ALLSEEN ALLIANCE

# Technical Steering Meeting

December 15, 2014

# Antitrust Compliance Notice

- AllSeen Alliance meetings involve participation by industry competitors, and it is the intention of AllSeen Alliance to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of and not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at AllSeen Alliance meetings and in connection with AllSeen Alliance activities are described in the AllSeen Alliance Antitrust Policy. If you have questions about these matters, please contact your company counsel, or if you are a member of AllSeen Alliance, feel free to contact Lee Gesmer or Andrew Updegrove, of the firm of Gesmer Updegrove LLP, which provides legal counsel to AllSeen Alliance.

# Reminder:

# This call is being recorded

# Agenda

1. Approve minutes from previous meeting
2. Core 14.12 Status
3. Developer engagement call for participation
4. Vote on Interface Review Board charter
5. Location Services project proposal
6. Cordova AllJoyn integration overview

# Core 14.12 Status

# Core 14.12 Status

- Test status
  - All system and regression testing complete
  - All JIRA tickets assigned to this release are either closed or in the process of being closed
- Release status
  - Core WG held a final review
    - All major contributors agreed to release 14.12 on Dec 17th
      - Microsoft
      - QCE
      - Technicolor

# Developer Engagement

**Phillip DesAutels**

# Developer Engagement
## Call for Participation

- The success of the AllSeen Alliance depends on ‘Developers’
  - Developers ***contributing*** to the AllJoyn code base
  - Developers ***leveraging*** the AllJoyn code base to build and deliver connected products
- We need to grow these developer communities i.e. increase developer engagement
- Engaging developers has technical and content components

  - Technical aspects of engagement are handled by the normal operations of the TSC

  - Content aspects of engagement need a plan…
    we have resources, but to get a plan developed, I need your help!

Email pdesautels@linuxfoundation.org

# Interface Review Board charter - Vote

# IRB Charter – Purpose & Scope

- **Purpose**
  - Over time, the various work groups within the AllSeen Alliance will produce a great amount of standardized Interface definitions that model different entities and services found within the Internet of Things. The purpose of the Interface Review Board is to act as an oversight committee for the definition and standardization of all these Interfaces, ensuring a minimal quality, stylistic homogeneity and well-defined taxonomy for the defined Interfaces.
- **Scope**
  - *Define the guidelines for standardized Interface design*.
  - *Provide practical guidance to Interface designers*.
  - *Review Interface definitions and advise the TSC on whether to approve them for standardization*
  - *Manage the overall Interface taxonomy.*

Detailed charter was submitted to the TSC mailing list

# IRB Volunteers (1/3)

**Ru Zhao**, *Haier*

- Standard R&D senior engineer
- Contributor to SmartHome WG & HAE Project
- Area of interest: interconnected smart appliances, AllSeen-oneM2M interworking

**Inhwan Choi**, *LG Electronics*

- Principal Research Engineer
- Chief contributor to HAE Project
- Area of interest: convergence of TV, mobile and home appliances

# IRB Volunteers (2/3)

**Josh Spain**, *Affinegy*

- Director of Engineering
- Committer of the Gateway Agent Working Group
- >10 years of experience as lead developer, software architect

**Virgil Cameron**, *Two Bulls*

- Principal Designer
- Area of interest: user experience & interaction design, user interface design

# IRB Volunteers (3/3)

**Marcello Lioy**, *Qualcomm Connected Experiences*

- AllJoyn Core Development Team Lead
- >4 years of involvement with AllJoyn, a.o. with proto-Base Services
- Area of interest: wireless networks, developer facing software and APIs

**Steve Kinneberg**, *Qualcomm Connected Experiences*

- AllJoyn Core Developer, maintainer of AllJoyn OpenWRT feed
- Involved with AllJoyn Standard Client from the very beginning
- Previously involved with Bluetooth SIG Car Working Group
- Very active in developer guidance and code reviews (GW Agent)

# Location Services Project Proposal

– Beechwoods, Politecnico di Milano

# Location Service Project Proposal

- Purpose and intent of the project
- The Location Service will provide a common interface that allows devices to provide various location related services to the proximal network. Four such services have been identified: presence, proximity, location and containment.
- These services will enhance scenarios proposed in the Living Scenarios project as well as other Allseen programs by providing events and triggers to deploy such scenarios.
- This proposal is appropriate for devices as motion sensors, beacons, cameras, mobile phones, wearables, etc…
- This proposal is technology neutral it attempts it is designed for all kinds of transports,  Bluetooth LE, Cellular, Zigbee, LTE, NFC, Wireless
- This project will support both the service/client (core) and the publish/subscribe (data driven) paradigm.

# Presence Service

- The Presence service will provide presence information about other known or unknown entities.
- The Presence service will allow a device to determine if another object is present on the proximal network.

# Proximity Service

- Provide distance information about other items
- Allow a device to know its distance with respect to a location

- The proximity service will allow the query for devices within the proximal network to be read.
- The proximity service will allow the distance between two devices to be read
- The proximity service will provide notifications of a new device entering or leaving the proximity fence.
- The proximity service will provide notifications of a change in distance of a device within the proximity fence.

# Location Service

- The Location Service will provide Location information (GPS, GIS, 4G location service) of a device on the network

# Containment Service

- The containment service will provide information on whether a device or object is within a proscribed area.

# Technology Transparency

- **Allow the developer to seamlessly integrate various kinds of location technologies into their application.**
  - Precision. Different technologies provide different degrees of precision. The spectrum goes from cellular (miles), to WiFi (yards), to BLE (feet), and finally to NFC(inches). Given the current nature of AllJoyn we can assume that the devices share the same WiFi, which means we might be initially focusing on BLE and NFC.
  - Interaction Modality. BLE-enabled devices listen for signals from wireless transmitters (BLE Beacons). Interaction is passive. NFC, on the other hand, requires that the user of the device actively initiate the engagement. While BLE supports a one-to-many interaction, NFC supports a one-to-one interaction. BLE requires that the device respond to what is similar to a notification, while NFC requires that the device control the timing and engagement.

# Technology Transparency

- Privacy and Security Issues. Given the different natures of BLE and NFC there are differences in this area. BLE is more intrusive since a device can be configured to continuously monitor a consumer's movement as he moves in and out of range of a beacon. NFC is less intrusive since movement can only be monitored based on the tags that have been engaged.

# Scenarios

- Geo-fencing: understanding whether the device has entered or exited a distance range. Geo-fencing might involve both proximity and containment, depending on the application's needs. Some times it might be enough to know if the device has entered the proximity range of a beacon (e.g., the proximity range of a store’s entrance). Some times we might want to know if the device has actually entered the store, meaning that it has actually entered a specific room.

- Hand-off like capabilities: allowing applications to pass their engagement with a user to another device. This will require proximity, but may also require location and/or containment. For example, the hand-off might only make sense within a specific context (e.g., I am at home, in the office, etc.).

# Scenarios

- Detailed Indoor Navigation: in this case we want an application to be able to track a device's movement within a specific indoor location with varying degrees of precision. This may require various kinds of joint use of proximity, location and containment features, depending on the specific needs. Indoor navigation might combine passive and active user interactions, since the could be tracked passively in a room, while going from one room to the next might require an active engagement (e.g. to open an NFC-based locked door).

- Intention Discovery: can we use location services (together with other things) to infer a user's intentions? This might be something to start researching for the future.

# Dependencies, Project Name, Working Group

- Dependencies
  - AllJoyn Core and Base services such as About interface
- Proposed Project Name
  - Proposed name for the project : “Location Service Framework
  - Proposed name for the git repository : “device_services/location_services”
- Proposed Working Group
  - This project should be under “Device Services” Working Group

# Committers and Contributors

- Maintainer – TBD
- Committers
  - Brad Kemp - Beechwoods Software,
  - Sam Guinea - Politecnico di Milano
- Contributors
  - Rodney Hess, Scott Knowlton, Carol Howard, Andrew Phillips -Beechwoods Software,
  - Valerio Panzica (MIT)
  - Various Students - Politecnico di Milano,
  - Open to any interested party

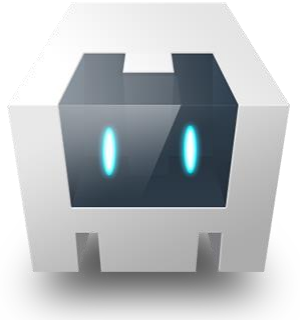
# Proposed Schedule

- Project Plan
  - High-level system description document : January 2015
  - AllJoyn interface specifications : February 2015
  - High-level design (HLD) documents for foundational components : April 2015
  - Foundational component implementations for Linux : July 2015
  - Certification test suite : September 2015
  - Reference controller applications for Android & iOS : September 2015
  - First official release : October 2015

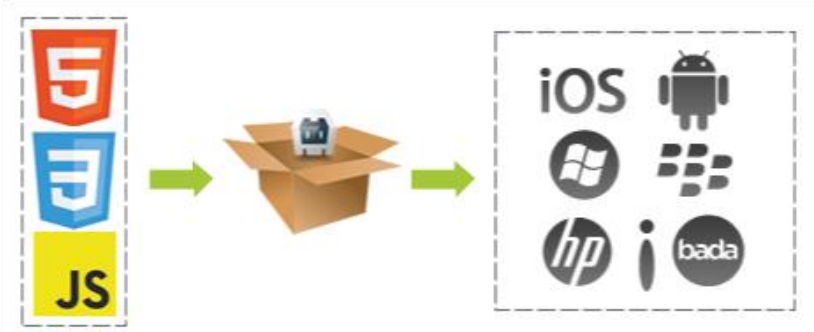
# Cordova AllJoyn overview

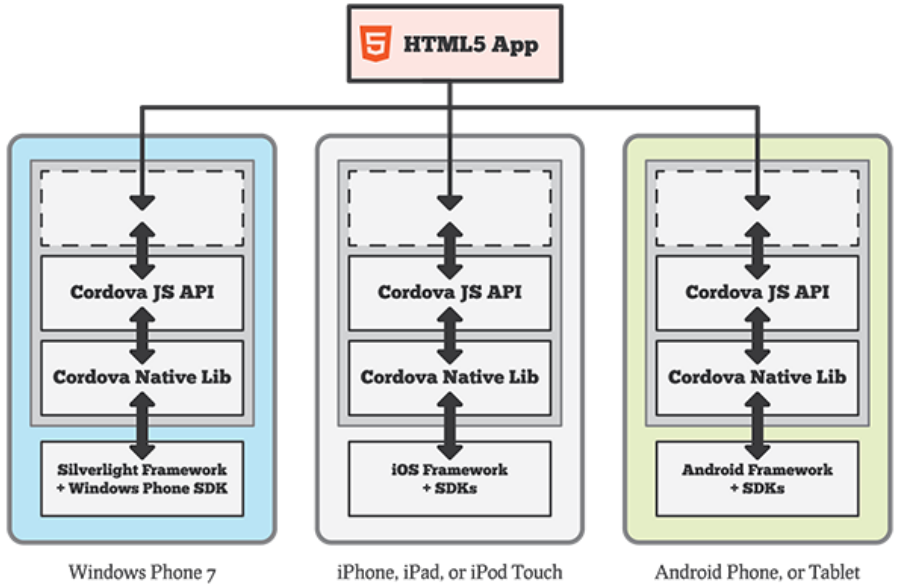
Ivan R. Judson, PhD
TED, Strategic Engagements
Microsoft

# Apache Cordova



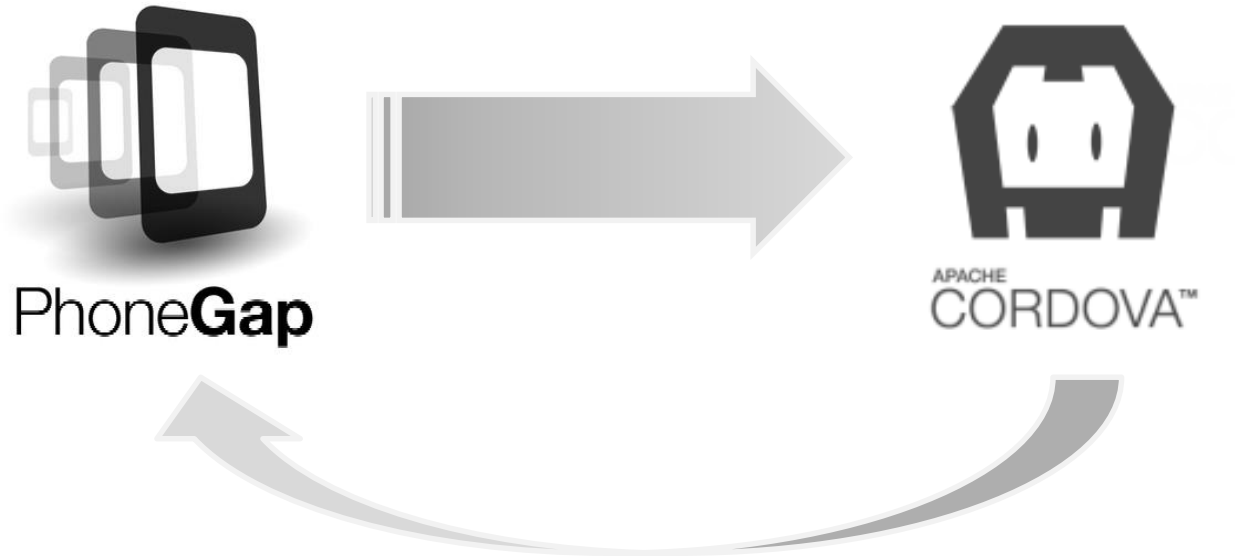
- **Apache Cordova** is a platform for building natively installed mobile applications using **HTML**, **CSS** and **JavaScript**



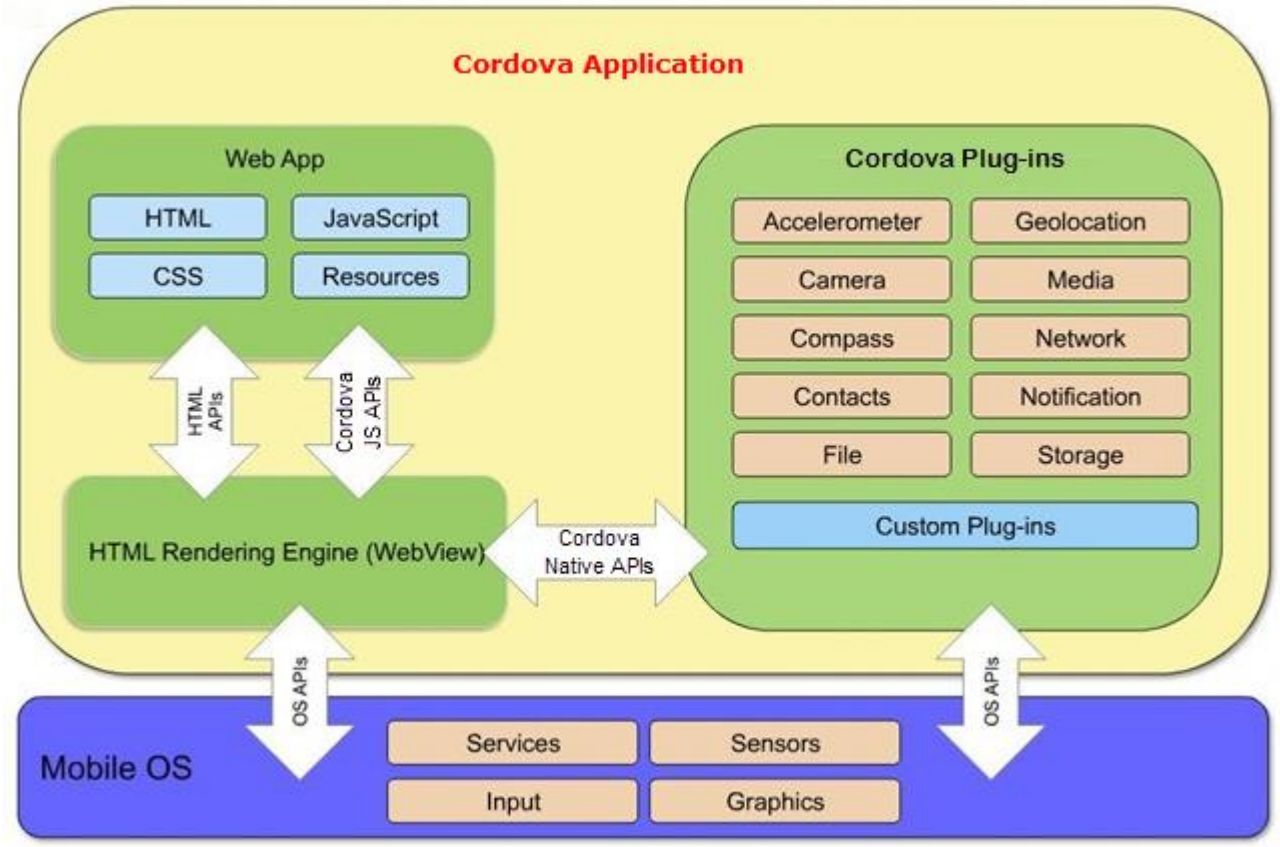



# History

- Apache Cordova was originally called **PhoneGap** build by Nitobi
- Open-source & free software from the beginning (MIT License), Apache License now
- Nitobi then acquired 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.

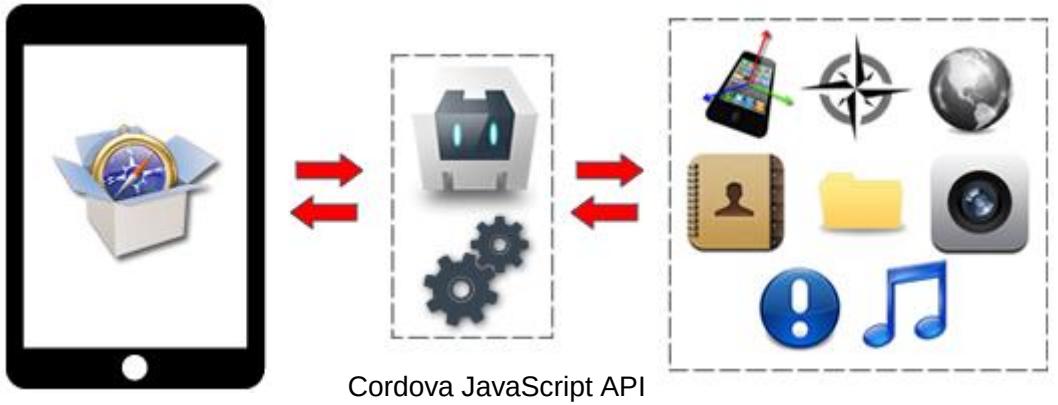


# Cordova Architecture



# Apache Cordova API

- Provides an application programming interface (API)
  - **access to native operating system functionality using JavaScript**.
  - APIs for Accelerometer, Camera, Compass, Media, Storage, etc.
  - Extendable using native plug-in
- http://cordova.apache.org/docs/en/4.0.0/

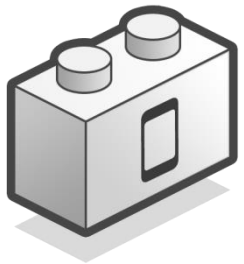


# Supported Platforms

| | amazon-fireos | android | blackberry10 | Firefox OS | ios | Ubuntu | wp8 (Windows Phone 8) | windows (8.0, 8.1, Phone 8.1) | tizen |
|---|---|---|---|---|---|---|---|---|---|
| cordova CLI | ✔ Mac, Windows, Linux | ✔ Mac, Windows, Linux | ✔ Mac, Windows | ✔ Mac, Windows, Linux | ✔ Mac | ✔ Ubuntu | ✔ Windows | ✔ | ✘ |
| Embedded WebView | ✔ (see details) | ✔ (see details) | ✘ | ✘ | ✔ (see details) | ✔ | ✘ | ✘ | ✘ |
| Plug-in Interface | ✔ (see details) | ✔ (see details) | ✔ (see details) | ✘ | ✔ (see details) | ✔ | ✔ (see details) | ✔ | ✘ |

| | amazon-fireos | android | blackberry10 | Firefox OS | ios | Ubuntu | wp8 (Windows Phone 8) | windows (8.0, 8.1, Phone 8.1) | tizen |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Platform APIs | | | | |
| Accelerometer* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| BatteryStatus* | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✘ | ✔ |
| Camera* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Capture* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✘ |
| Compass* | ✔ | ✔ | ✔ | ✘ | ✔ (3GS+) | ✔ | ✔ | ✔ | ✔ |
| Connection* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Contacts* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | partially | ✘ |
| Device* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Events | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| File* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✘ |
| File Transfer* | ✔ | ✔ | ✔ * Do not support onprogress nor abort | ✘ | ✔ | ✘ | ✔ * Do not support onprogress nor abort | ✔ * Do not support onprogress nor abort | ✘ |
| Geolocation* | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Globalization* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✘ | ✘ |
| InAppBrowser* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | uses iframe | ✘ |
| Media* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Notification* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Splashscreen* | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ | ✔ | ✘ |
| Storage | ✔ | ✔ | ✔ | ✘ | ✔ | ✔ | ✔ localStorage & indexedDB | ✔ localStorage & indexedDB | ✔ |
| Vibration* | ✔ | ✔ | ✔ | ✔ | ✔ | ✘ | ✔ | ✘ | ✘ |

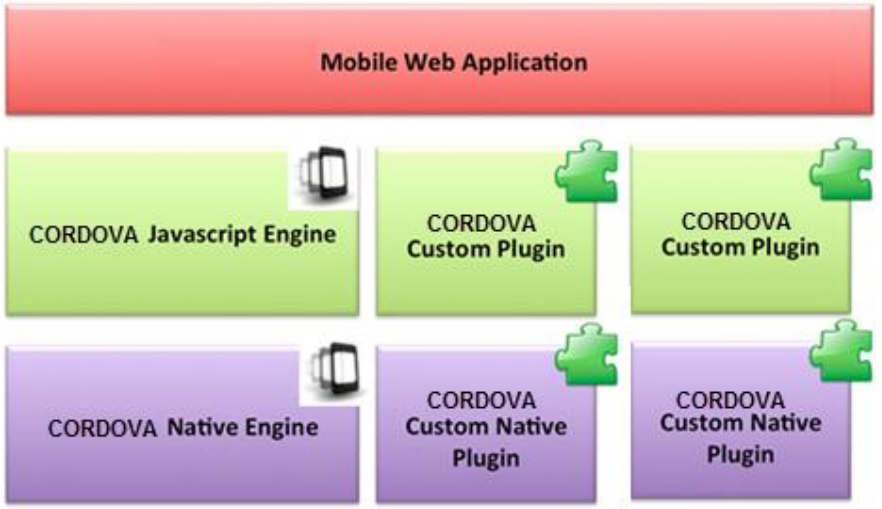
# Apache Cordova Native Plug-in



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

Cordova 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 `Cordova.exec()`
- Plug-in class mappings:
  - **Android**: res/xml/plugins.xml
  - **iOS**: www/Cordova.plist
  - **BlackBerry**: www/plugins.xml




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

# Plugin Example (Android Native Code)

```
package org.allseen.alljoyn;
//Imports excluded
public class AllJoyn extends CordovaPlugin {
        private static final String TAG = "AllJoyn";
        static {
                System.loadLibrary("alljoyn");
        }
        @Override
        public void initialize(final CordovaInterface cordova, CordovaWebView webView) {
                super.initialize(cordova, webView);
                alljoyn.AJ_Initialize();
        }
        @Override
        public boolean execute(String action, JSONArray data, CallbackContext callbackContext) throws JSONException {
                if (action.equals("discover")) {
                        Log.i(TAG, "Calling discover");
                } else if (action.equals("version")) {
                        Log.i(TAG, "Calling version");
                        String version = getVersion();
                        if (version == null) {
                                callbackContext.error("Error retrieving version.");
                                return false;
                        } else {
                                callbackContext.success(version);
                                return true;
                        }
                }
                return false;
        }
}
```

Extend the Cordova Plugin class

Load ajtl based shared library

Implement execute method

# Application-Plugin Example (HTML)

```
<html>
    <head>
        <meta charset="utf-8" />
        <meta name="format-detection" content="telephone=no" />
         <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-
scale=1, width=device-width, height=device-height" />
        <link rel="stylesheet" type="text/css" href="css/index.css" />
        <meta name="msapplication-tap-highlight" content="no" />
        <title>AllJoyn Example</title>
    </head>
    <body>
        <div class="app">
            <h1>AllJoyn Example</h1>
            <div id="deviceready" class="blink">
                <p class="event listening">Connecting to Device</p>
                <p class="event received">Device is Ready</p>
            </div>
            <br/>
            <div id="alljoyn">
                <button id="alljoynDiscoverStart">Start Discovery</button>
                <button id="alljoynDiscoverStop">Stop Discovery</button>
                <div id="allJoynDevices">AllJoyn Devices: None</div>
            </div>
        </div>
        <script type="text/javascript" src="cordova.js"></script>
        <script type="text/javascript" src="js/index.js"></script>
        <script type="text/javascript”> app.initialize(); </script>
    </body>
</html>
```

# Application-Plugin Example (JS)

```
var app = {
    // Application Constructor
    initialize: function() {
        this.bindEvents();
    },
    // Bind Event Listeners
    bindEvents: function() {
        document.addEventListener('deviceready', this.onDeviceReady, false);
        document.getElementById('alljoynDiscoverStart').addEventListener('click',
this.alljoynDiscoverStart, false);
        document.getElementById('alljoynDiscoverStop').addEventListener('click',
this.alljoynDiscoverStop, false);
    },
    // deviceready Event Handler
    onDeviceReady: function() {
        app.receivedEvent('deviceready');
        allseen.alljoyn.version(function(version) { // Success Callback },
function(error) { // Error Callback } );
    },
    alljoynDiscoverStart: function() {
        console.log('Starting discover...');
        allseen.alljoyn.discover(function(msg) { // Success Callback }, function(error)
{ // Error Callback } );
    }
    // Other code removed for clarity
};
```

# Plugin Example (JS Code)

```
var argscheck = require('cordova/argscheck'),
     utils = require('cordova/utils'),
     exec = require('cordova/exec'),
     cordova = require('cordova');
var alljoyn = {
     version: function(success, error) {
          argscheck.checkArgs('', 'allseen.alljoyn.version',
arguments);
          exec(success, error, "AllJoyn", "version", []);
     },
     discover: function(success, error) {
          argscheck.checkArgs('FF', 'allseen.alljoyn.discover',
arguments);
          exec(success, error, "AllJoyn", "discover", []);
     }
};
module.exports = alljoyn;
```

# Plan

- Microsoft's TED-SE Team is implementing:
  - AllJoyn Native Libraries for:
    iOS, Android and Windows 8+
  - Cordova JS API wrapping #1
  - Cordova Example Application
  - AllJoyn enabling Oxide, a mobile application for http://nitrogen.io/
  - Training materials for how to develop Cordova Applications that use AllJoyn

# Resources

- Apache Cordova Website
  http://cordova.apache.org/

- Apache Cordova Documentation
  http://cordova.apache.org/docs/en/4.0.0/

- 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)

- Original Deck, heavily modified for our use:
  http://www.slideshare.net/ejlp12/intro-to-apache-cordova

# Thank You

Follow Us On 

- For more information on AllSeen Alliance, visit us at: allseenalliance.org & allseenalliance.org/news/blogs