ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 2, February 2015*

# State of art approaches using PhoneGap for Portability of Mobile Apps

**Prof. Bhushan S. Thakare[1], Naghma Parween[2], Shama Parween[3], Dhanashree K. Shirodkar[4]**

Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, India[1,2,3,4]

**Abstract:** The demand for cross platform mobile development is constantly growing in the market. The basic need for programmers on any platform is compatibility. There are various cross-browser tools that will take any developer a long way in creating flexible apps that stand the compatibility test such as Whoop, Mosync, Phonegap, Appcelator etc. Phonegap is one of the most developer friendly tools present in the market which provide a good range of native APIs and platforms. This paper deals with the available approaches of building a cross-platform application using phonegap.

**Keywords:** PhoneGap, PhoneGap APIs, native APIs.

## 1. PHONEGAP

PhoneGap applications are developed using HTML, CSS, and JavaScript, however the final product of a PhoneGap application is a binary application archive that can be distributed through standard application ecosystems[3]. PhoneGap was previously called as Apache CallBack but now it is commonly known as Apache Cordova. PhoneGap was first built by Nitobi and was then purchased by Adobe systems. For iOS applications the output is an IPA file (iOS Application Archive), for Android applications the output is an APK file (Android Package), for Window Phone the output is a XAP file (Application Package), etc. These are the same application packaging formats used by "native" applications, and can be distributed through the appropriate ecosystems (iTunes Store, Google Play, Amazon Market, BlackBerry App World, Windows Phone Marketplace, etc.)

PhoneGap is extensible with a "native plugin" model that enables you to write your own native logic to access via JavaScript[5]. A PhoneGap plugin bridges a bit of functionality between the WebView powering a PhoneGap application and the native platform the PhoneGap application is running on[2]. Plugins are composed of a single JavaScript interface used across all platforms, and native implementations following platform specific Plugin interfaces that the JavaScript will call into.

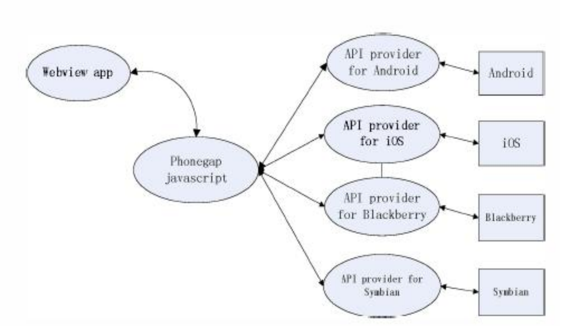
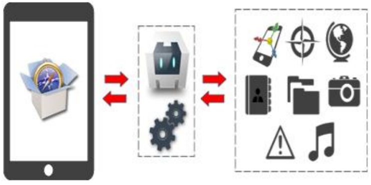


Figure 1. Communication between Phonegap and native API

Different frames used for developing cross platform apps are PhoneGap, Titanium and Rhodes. PhoneGap is considered the best amongst them as it supports all the major platforms like Android, iOS, Windows etc. PhoneGap supports nearly seven operating systems. PhoneGap applications are developed using HTML5, CSS and JavaScript which are supported by majority of the platforms.

## 2. PHONEGAP API

PhoneGap provides an application programming interface (API) that enables you to access native operating system functionality using JavaScript. We can build our application logic using JavaScript, and the PhoneGap API handles communication with the native operating system.



### 2.1 Contact API

PhoneGap provides contact APIs in order to create, store, copy, search and delete contact from the contact list. This API works with the device's database.

- **Org.apache.cordova.contact**

This plugin is used to navigate through the device's contact book in order to create ,store or delete any contacts.

### 2.2 Camera API

The camera API provides the functionality of capturing images through the device's camera. It also provides an option of choosing images from the system's image library.

- **Org.apache.cordova.camera**

This plugin provides an API for taking pictures and for choosing images from the system's image library.

### 2.3 Geolocation API

Geolocation API can acquire the real time positioning of the device along with latitude and longitude. The location can be tracked or monitored in order to acquire the geographical location.

- **Org.apache.cordova.geolocation**

This plugin provides information about the device's location, such as latitude and longitude. Common sources of location information include Global Positioning System (GPS) and location inferred from network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and

DOI 10.17148/IJARCCE.2015.4217

GSM/CDMA cell IDs. There is no guarantee that the API returns the device's actual location.

This API is based on the W3C Geolocation API Specification, and only executes on devices that don't already provide an implementation.

### 2.4 Accelerometer API

The movement of the device can be tracked by the accelerometer API. The accelerometer API acquires the device’s movement by tracing the X, Y, Z co-ordinates and the speed at which the device is moving.

- **Org.apache.cordova.device-motion**

This plugin provides access to the device's accelerometer. The accelerometer is a motion sensor that detects the change (*delta*) in movement relative to the current device orientation, in three dimensions along the *x*, *y*, and *z* axis.

### 2.5 File API

The file system is present on all the devices regardless of the operating system present on it. This file system is accessed by PhoneGap in order to perform various operations such as data access, copy data, transfer data etc.

- **Org.apache.cordova.file**

This plugin implements a File API allowing read/write access to files residing on the device.

### 2.6 Compass API

The functionality of the compass API is similar to Geolocation API and Accelerometer API. It provides the direction in which the device is pointing or heading towards. Most of the smartphones have compasses embedded in them.

- **Org.apache.cordova.device-orientation**

This plugin provides access to the device's compass. The compass is a sensor that detects the direction or heading that the device is pointed, typically from the top of the device. It measures the heading in degrees from 0 to 359.99, where 0 is north.

### 2.7 Media API

This API provides the functionality of recording and playing the audio files present on the system. It can play, pause and record any audio file.

- **Org.apache.cordova.media**

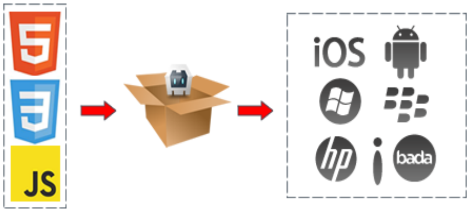
This plugin provides the ability to record and play back audio files on a device.

### 2.8 Storage API

PhoneGap supports WebSQL as well as local database to create client side database. The database can then be accessed for storage purpose.

## 3. PHONEGAP APPLICATION PACKAGING AND DISTRIBUTION

PhoneGap applications are developed using HTML, CSS, and JavaScript, however the final product of a PhoneGap application is a binary application archive that can be distributed through standard application ecosystems.



For iOS applications the output is an IPA file) (iOS Application Archive), for Android applications the output is an APK file) (Android Package), for Window Phone the output is a XAP file (Application Package), etc. These are the same application packaging formats used by "native" applications, and can be distributed through the appropriate ecosystems (iTunes Store, Android Market, Amazon Market, BlackBerry App World, Windows Phone Marketplace, etc.)

## 4. ADVANTAGES OF PHONEGAP

Developing Android apps or iOS applications requires code to be written in their native languages like Java and Objective C respectively. In order to develop an application that is compatible on the operating system a cross platform app can be developed using PhoneGap. Cross platform apps are developed using HTML5, CSS and JavaScript. When one needs to develop an app in its native OS, one needs to have the basic understanding about that language. Developing apps using PhoneGap does not require any extra knowledge about the language. PhoneGap coding includes standard HTML5, CSS and JavaScript. The coding is implemented once and then it can be run on multiple platforms. PhoneGap supports seven different mobile platforms, including iOS, Android, BlackBerry, webOS, Symbian, Samsung Bada and Windows Phone. The applications developed using PhoneGap have a cleaner interface as it does not include the browser window.

## 5. CONCLUSION

Phonegap is a frame that realizes cross-platform application using web technology (HTML+CSS+ JAVASCRIPT), and it provides rich interfaces used to visit phone's native API by Javacript. Phonegap builds a bridge between web development standards and smartphones to make it easy for web developers to develop mobile application. But, to develop cross-platform application, it still needs different development IDE, not the same environment configuration, and adjusted program. Although that, the compatibility of Phonegap still relieves many work, which has a better prospect. In this paper, we have discussed about the architecture of PhoneGap, the APIs provided by the phonegap and the architecture of PhoneGap. Also we came across the actual working ie calling of native APIs through Javascript in two major OS environment respectively using PhoneGap. PhoneGap shows a lot of promises in the field of cross-platform application development.

## 6. REFERENCES

[1] “A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach”, Rahul Raj C.P and Seshu Babu Tolety

[2] “Cross Platform Application Development”,http://www.ndot.in/cross-platform-mobile-developmen t.html

[3] “Cross-Platform Multi-Network Mobile Application Architecture”, Wen-Hann Wang, CV Vick, Christos Georgiopoulos, Jon Bork

[4] http://www.phonegap.com/ “The Discussion of Cross-Platform MobileApplication Based on Phonegap” LiT ian, HuaichangDu,LongTang, YeXu Communication University of China