

An comparative analysis of Cordova Mobile Applications V/S Native Mobile Application

Nitishkumar Singh

Master of Computer Applications (MCA)
Thakur Institute of management Studies Career
Development and Research Center Mumbai, India.
nitishkumarsingh71@live.com

Abstract:- With the increase in use of mobile devices, mobile technology is also developing day by day. In mobile devices also use of smartphones has been increased drastically. Peoples are using smartphones not only for voice calling or text messaging but also to perform their day to day work like shopping, paying bills etc. A lot of mobile applications has been introduced which allows for free voice/video calling and messaging. All you need to do to avail these services is to buy mobile data services. Increase in use of smartphones has also caused increase in use of Mobile data. Increase in use of smartphone has been caused by various mobile applications. The approach to develop these mobile applications has also been changed i.e. there are basically two types of approach for mobile application development. These are Native and Hybrid, we will discuss about both these approaches and their feasibility for business applications. In this paper we will cover their advantages as well as their disadvantage. We will also cover the topic why cordova application are the future of mobile application development.

Keywords—cordova, natuve, android, ios, windows mobile, buiness

I. INTRODUCTION

There is a rapid increase in the no of users for smartphones. One quarter of the global population is already using smartphones in year 2015. There will be around 2 billion consumers for smartphones by the year 2016. [1] Myntra.com one of the famous online shopping platform has shut down their website. They are only opearting on mobile application, because there 90% revenue was from mobile application. Flipkart.com india's biggest online shopping market is also planning to shut down their website within few months. These examples also show the popularity of mobile applications. There are several smartphones available in the market which runs on different mobile platforms such as Android, IOS, Windows, Firefox, Ubuntu etc. The cause of increased use of smartphones is the mobile applications. Mobile applications allow user to perform several useful activities from their smartphones only. Some of the examples of activities are Banking, Social Networking.

Previously mobile application for each platform is need to be designed separately. Even for each platform that particular mobile application performs the same task. Each mobile platform has its own native language. Using their native language application for the respective platform can be developed. Since, Android uses java, IOS uses Objective-C and windows uses C# and other mobile platforms use other different languages.

The idea behind Hybrid application is to arrive at cross platform solution along with native features. If you are targeting for one or three mobile platform at once, then hybrid mobile applications will yield better results compared to native mobile application. Hybrid mobile applications are developed

using HTML5, JavaScript and CSS. All these technologies are bundled together using the native framework of the respective mobile platform and deployed as mobile application.

HTML5 is the markup language used to presenting and structuring data over World Wide Web (WWW). Several new features has been included in the HTML5, some of them are web SQL & storage, geo-location, touch events, media capture, file api, battery status, device orientation and indexed database. All of these features are widely used in the development of mobile application. JavaScript is a dynamic programming language. It helps to create interactive web applications. JavaScript also supports asynchronous operations which help to perform individual operations separately with an ease. JavaScript asynchronous functionality comes very handy in the development of business/ real-time mobile applications. Since, there are several operations in business applications need to perform at once which are independent of each other. CSS stands for Cascading Style Sheet. CSS is one responsible for look and structure of a Hybrid Mobile application. Simple idea behind CSS is to separate content from presentation. This separation of content from presentation improves ease in managing presentation of an html page.

There are several big names in market of e-commerce and social-media who have preferred hybrid mobile apps over native mobile app. Amazon and LinkedIn are one of the famous examples for hybrid mobile applications. [7] Even Microsoft has included javascript into windows phone coding language for their future mobile operating system version starting from windows phone 8.1. This gives a clear idea about the increasing popularity of the Hybrid Mobile applications. Since the introduction of hybrid mobile applications, several

cloud development tool providers has been came into existence. These tool providers provide easy development process for hybrid mobile applications with the help of their online products and services. They also help to create build and publish them on various mobile platforms. Some of the examples are Phonegap, Telerik etc. Advantage of such cloud platforms are, you don't need to set a separate development environment for each mobile platform and charges are also nominal for these cloud platforms. Due to their good services and features these cloud platforms are getting more popular.

Mobile Operating Systems

Here we will only cover mobile operating system which holds major stake in market. They are Android, IOS and Windows.

Android: It is an operating system based on Linux kernel. Initially android was designed for touch based smartphones and tablets. Now it is also used in Desktop, Laptops, TVs, watches etc. Application code for android is done in Java. Android provides a full mobile operating system which provides an easy way to do application coding and then building the APK. Android provides a full integrated development environment (IDE) bundled with android SDK package. All developer has to do is just install it to their system. Android allows user application to access the facility of camera, Bluetooth, Photo Album, Contacts etc. All these services help user application to provide a better user experience. Android allows creating separate component, they can be invoked individually thus providing several entry points to application. Since, Android is open source technology build on Linux kernel. Hence changes can be incorporated by different distributors as per their requirement. Even it can be used to create a new mobile operating system too which is built on the core of android mobile operating system for ex. Cyanogen. Since in android user application have equal priority as build in application. This allows android users to have a great variety of android applications.

According to market survey, market share hold by android till first quarter of 2015 was 78.00% which makes android the dominating mobile operating system.[3]

iPhone OS: iPhone OS (IOS) is a mobile operating system designed by Apple Inc. and distributed only for Apple designed hardware's. IOS works on an iPhone, iPad and iPod touch. IOS devices are well known for their security features. Even an mobile app also need to go through a very strict review before launching it on app store. IOS applications are built using Objective-c programming language. Recently Apple Inc. has launched a new open source programming language known as swift for creating ios application. IOS was derived from Mac OS X. Hence, ios is based on Unix OS architecture. IOS does not allow third party softwares without jail-breaking in ios; so, reducing a greater extent of malicious

code to be executed on ios devices. IOS also supports multi-tasking from version 4, although it was long supported in android because of java. IOS provides xcode which can be used for the developing, testing and installing on ios devices. In ios for development too we need to specify ios device id on which we need test ios application. IOS has been designed in such way which attracts users to spend more time over ios devices. Design of ios is clean and simple. IOS look and feel provides unified feeling.

According to market survey, market share hold by ios till first quarter of 2015 was 18.30% which makes ios the second dominating mobile operating system. [3]

Windows Mobile OS: There have been a lot of changes in windows mobile os. The latest stable version for windows mobile os is windows 8. Windows 8 was a failure over desktop pc's but was quite successful for mobile. "Tiles" Design has been used to display information in windows 8 mobile. Unlike IOS windows allow different devices to install windows mobile os. Windows 8 mobile has provided true multitasking. It allows mobile developers to run an mobile app in background and also allows to resume it. Windows mobile uses Extensible Application Markup Language (XAML) for developing windows application. XAML is used to design user interface of mobile application and also allows binding events and codes with each component. Using XAML windows allow developers to use Model-View-ViewModel (MVVM) approach. We can use either C# or C and C++ programming language with XAML. Windows provide all development tools for free and also allows registering a windows phone for debugging. Windows is the first mobile platform to provide full development support for cordova apps. In windows 8.1 we can use javascript for development too. From Visual studio 2013 version 4, visual studio uses XAMARIN which can be used for the cordova development as well as can be tested on different platforms.

According to market survey, market share hold by windows mobile os till first quarter of 2015 was 2.7% which makes windows mobile os the third dominating mobile operating system. [3]

II. WHAT IS CORDOVA

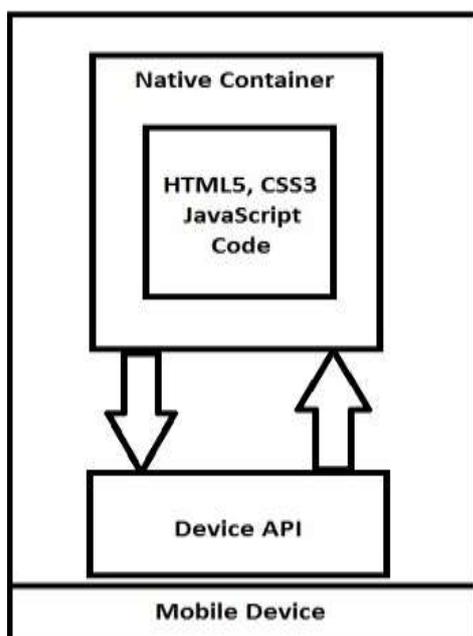
Apache Cordova is open source project within Apache Software Foundation (ASF). The sole purpose of cordova is to develop cross-platform mobile application. Cordova uses HTML5, JavaScript and CSS3 for the mobile application development. Here HTML5, CSS3 and JavaScript files are wrapped within the native container. At the time of execution they are interpreted by the device web engine. Cordova is best option if you want to launch your application over more than one mobile platform. Cordova also supports accessing device level apis. Cordova allows creating mobile website which can

be packed and distributed over several platforms. There are two development approaches for cordova:

Cross-platform: We can use this approach if we want to launch an mobile application over multiple platforms. The key focus of this approach is over web-view rather than native Software Development Kit(SDK) of mobile platforms

Platform-centered: In this approach we take the advantage of web based development, but more focus is on native functionalities. Here development is done using mobile platform specific tools. Hence, most of the time development is more specific to a single mobile platform.

III. HOW CORDOVA WORK WITH HYBRID APPLICATION



Above diagram shows the architecture of cordova mobile applications. It clearly shows that all web code is wrapped into a platform specific native container. Native container is the one which communicate with all device apis whenever web code demands. Native Container sends request to device api and receives the response from device api. Also native container parses data of both web code and device apis; so both the parties can understand each other's data. Since, there is native container which acts as middle layer between web code and device api. This makes cordova applications a little bit slow compared to native mobile applications. Since, hardware of smartphones is improving, which makes this difference also negligible.

Web code is executed by the web engine. Web engine is a kind of interpreter which interprets all the javascript, html and css code. Most famous device web engines are webkit, moz and ie. Webkit is used by android and ios devices; whereas moz is used by firefox os and ie is used by windows

mobile. While designing presentation web code for cordova mobile applications, all these web engines need to be taken care. Since, same code won't work for each web engine. Hence, it's a good practice to keep code for all web engines. These web engines automatically fetch the code appropriate as per them and will reject the remaining code.

Webcode can be provided in two ways to cordova mobile application. First approach is wrapping the web code into application itself and creates build. Another approach is keeping the web code over server and load the web code each time application itself. Each approach has its own advantages and disadvantages. In first approach it will take less time to load application. In second approach each time data is loaded from the web server, so it will take time. Even though caching is enabled, but caching also has limit. So for larger applications first approach is better and for smaller one second approach. In case of first approach updates need to be installed by user, but in case of second approach control for updates is in the hand of developer itself. For quick updates second approach is good. In case of first approach user can perform some task even web server is not available i.e. there will be an offline support. In second approach entire page is at web server only, so user won't be able to perform any task. For offline support first approach is good.

Since each approach has its own advantages and disadvantages. We should use mix of both approaches in a mobile application. First approach can be used for those features and functionality which don't change frequently and keep second approach for those which changes frequently. In this way we will be able to take advantage of both the approaches and will able to counter disadvantages of both.

IV. CORDOVA MOBILE APPLICATION V/S NATIVE MOBILE APPLICATION

Before writing this research paper we worked on multi-platform enterprise level mobile applications for two different domains; one for governance and another for info-media. Based on those mobile applications experience we have considered several points which are essential from mobile application development. Below we have mentioned description for all of them:

Effort: In case of native mobile applications we need to separately develop mobile application for each platform. Developers need to code for the same functionality for each platform. But for Cordova mobile applications efforts are reduced so much. Since, here we have to code once and use it everywhere.

Time: Native mobile applications separate time is allocated for the development of different platforms, but for cordova application we need to focus at one mobile application for entire time.

Development Team: Native application for each platform is developed on different programming languages. Hence for each platform you will require separate team of developers. But for cordova mobile applications web developer are required. With little extra efforts they can develop and build applications for each platform. So, here all we need is one development team for all platforms.

Maintenance & Upgrades: Maintenance and upgrades is hectic work with respect to native mobile application. As we have to work on the same issue for each platform, but for cordova based application write it once, and run everywhere policy is applicable.

Access to native features: native applications can by default utilize all the functionalities and features of a mobile application. Even though cordova apps can also use device and os functionalities. All basic and major functionalities are already built-in cordova platform, but sometimes they are not available in default package then we need to do native code for them.

Budget: Cost for native applications will be more than cordova mobile applications. Since, for each mobile platform we need to develop separate mobile application, but for cordova app we create one application. Hence, budget will be less for cordova mobile apps.

Mobile Web Engine: Native applications don't require web engine unless they want to show a webpage. Cordova applications are totally dependent on mobile web engines for their executions and user interface. For older smartphones web engines were not that powerful. Hence sometimes cordova apps don't execute in expected manner in older smartphones. But this is not a big disadvantage as older smartphones are moving out of market.

HTML5, CSS3 Standard Implementation: Although cordova applications work properly with old standard of html and css, but to take full advantage of cordova mobile applications full implementation of html5 and css3 is required. Different mobile platforms differ in their implementation of html5 and css3; also they have not implemented all the features of html5 and css3. Sometimes it can cause a little bit problem in app implementation.

Debugging: all mobile platforms provide very debugging support for the native applications. Even for cordova apps too debugging support is good but not as good as native apps. Many times you need to manually debug a cordova application.

Testing: Since cordova apps are platform independent, hence they need to be manually tested at several occasions. Native app has full testing support from their respective mobile platforms.

Simulator: There are several features for html5 and css3 which are not fully supported in mobile platform specific simulators. But this disadvantage is negligible, because even after executing on simulator we need to test an mobile application over several physical devices.

Error Handling: Native applications have a very good error handling mechanism. Since cordova apps use javascript, which have not that good error handling mechanism. Here developer need to take care of all possibilities which can cause error while execution of code.

Above we discussed key factors related to development. Let's now discuss important criteria for multi-platform from user's perspective:

Execution Time: Cordova applications do not lag much in terms of execution when compared with native mobile applications, provided that physical device hardware should be good. For low physical hardware too cordova mobile apps perform pretty well.

Performance: here native applications have a little upper hand over cordova mobile applications. Difference between cordova apps and native apps is almost negligible.

User experience: User experience for both cordova apps and native apps are same. Native Applications have full support of their mobile platform api's; where as cordova apps have support of html5, css3 and javascript.

Response time: Native mobile applications will have shorter response time compared to cordova mobile applications. Since for native apps execution is done directly, but in case of cordova apps web engine executes cordova mobile apps.

Usability: Cordova apps use HTML5 and CSS3 for user interface. HTML5 and CSS are capable of representing data in various ways which is not possible every time for native applications. Cordova apps have a big advantage over native apps in terms of usability.

In the below given chart we have shown the rating (Good, Medium, Poor) of different criteria for native mobile applications and Cordova Mobile applications for multi-platform support. This will help others who want to decide which approach to be chosen

Criteria	Native Mobile App	Cordova Mobile App
Effort	medium	good
Time	low	good
Performance	good	good
Development Team	low	good
Access to native features	good	good
Maintenance	low	good
Upgrades	low	good
Execution Time	good	good
Budget	medium	good
Reusability	medium	good
Testing	good	low
Simulator	good	low
Debugging	good	medium
Offline Support	good	good
Response Time	good	medium
User Interface	good	good
Usability	good	good
Security	good	good
Error Handling	good	low

V. WHY CORDOVA APPLICATIONS CAN SURPASS NATIVE MOBILE APPLICATIONS

A lot of improvements are happening in smartphones. Smartphones are becoming more powerful day by day, as powerful as computers. Since, gap between hardware quality of mobile phones and computers is minimized. This gives a chance for cordova applications to surpass native mobile applications. Let's discuss reasons behind it:

Plugin Development: This is one of the most powerful features of cordova platform. Whenever we want to include a native feature of a mobile platform then cordova allows incorporating that functionality/feature as plugin. Cordova provides architecture to include those features. All the basic services in cordova which are provided by default such as camera, geo-location etc. are nothing but plugins. Cordova also allows referring same functionality of different mobile platforms in the same way. Thus, removing ambiguity. Whenever developers want to use any mobile functionality which is not present in cordova platform then they can develop a plugin for that and include that feature or functionality into their project. Hence cordova can access any native feature of any mobile platform.

Use of HTML5, CSS3 and JavaScript: There are several frameworks available in market which uses HTML5, CSS3 and JavaScript. All these platforms are specifically designed

for cordova mobile applications. Initially there was issue with HTML5, CSS3, JavaScript framework, as they were not designed for mobiles. Now they fully support mobile events such as touch, gestures etc. Use of HTML5, CSS3 and JavaScript allows creating such look and feel which is not even possible for native mobile applications. Examples of such powerful framework are Sencha Touch, Kendo Ui, Ionic, Titanium, JQuery Mobile (JQM), Bootstrap Mobile etc. These platforms have large community support as well as commercial support too. They provide a good user interface as well as allows developers to manipulate as per there ease.

Hardware improvement in mobiles: Smartphones are getting more powerful. Mobile web engines which are the key player in cordova mobile applications are getting more powerful. Mobile web engines are now supporting all features which are supported by desktop browsers. Also there execution time has reduced and performance has improved. Nowadays smartphones are as powerful as desktop computers.

Platform Awareness: There are situations when we need to execute different operations on the basis of mobile platforms. Cordova allows detecting the current platform on which application is executing. Thus making it easy for developer's to execute different operations for different platforms. Each mobile platform has different implementation. So some of them support some functionality and some don't. Platform awareness helps to enable or disable such functionality in cordova mobile application.

VI. CONCLUSION

Demand of mobile applications over smartphones is increasing. As the demand is increasing discussion over the approaches to develop mobile application is also taking speed. Since, Cordova is an open source project and allows developing mobile application for several platforms. It provides a good way for business applications where demands are changing frequently, development and updates need to be done quickly. It allows them to focus on one development cycle, instead of several. This way it helps them to save cost, time along with improving productivity and user satisfaction. When compared to native apps cordova applications have very few disadvantages. Most of these disadvantages are negligible and the remaining disadvantages will perish soon. Dynamic updates are easy in cordova apps, since they contain business logic and presentation code in web code. In this paper we have try to cover all the aspects related to developer as well as user for mobile application. We have also tried to cover about cordova applications and their working. It's clear from all above points that cordova applications are the future of business mobile applications.

REFERENCES

- [1] "Smartphone Usage" <http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>
- [2] "Global Mobile Traffic" (http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html)
- [3] "Mobile Operating System Market Share" <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [4] HTML 5 (<http://www.w3.org/TR/html5/>)
- [5] JavaScript (<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)
- [6] Cascading Style Sheet (<https://developer.mozilla.org/en-US/docs/Web/CSS>)
- [7] <http://www.slideshare.net/iivanoo/these-mobile-apps-will-let-you-totally-rethink-hybrid-app-development>
- [8] Learning's from Early Native Apps Improve HTML5* and Hybrid Apps (<http://www.intel.com.br/content/dam/www/public/us/en/documents/white-papers/learnings-from-early-native-apps-improve-html5-and-hybrid-apps.pdf>)
- [9] Application Security in Android-OS VS IOS (International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014)
- [10] Development of Hybrid Applications with HTML (http://www.mindteck.com/pdf/Development_of_Hybrid_Applications_with_HTML5.pdf)
- [11] HTML5, Hybrid or Native Mobile App Development (ftp://public.dhe.ibm.com/software/fr/pdf/whitepapers/Worklight-HTML5_Hybrid_Native_Mobile_App_Development.pdf)
- [12] Cordova Platform (<http://cordova.apache.org/docs/en/5.0.0/>)
- [13] Stelios X., Spyros X. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications
- [14] Holzer. A., & Ondrus J. Trends in Mobile Application Development
- [15] Cao, Y., et al. (2009): Virtual Campfire - Cross-Platform Services for Mobile Social Software. In: Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, MDM 2009, pp. 363–364
- [16] Palmieri, M., Singh, I., Cicchetti, A.(2012): Comparison of Cross-Platform Mobile Development Tools
- [17] Ramadath, S., Collins, M. (2012): Mobile Application Development: Challenges and Best Practices
- [18] Heitkötter, H., Hanschke, S., Majchrzak, T.A. (2012): Comparing cross-platform development approaches for mobile applications. In: 8th International Conference on Web Information Systems and Technologies, WEBIST 2012, pp. 299–311
- [19] Yao Liu, Fei Li, Lei Guo, Bo Shen, and Songqing Chen. A Comparative Study of Android and iOS for Accessing Internet Streaming Services
- [20] Kamboj, Gupta, (2012) —Mobile Operating Systems, International Journal of Engineering Innovation & Research, Volume 1, Issue 2, ISSN: 2277 – 5668, Pp 115-120
- [21] Concepts and architecture for Windows Phone 8 ([https://msdn.microsoft.com/en-us/library/windows/apps/ff967549\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/ff967549(v=vs.105).aspx))