

## AN APPROACH TO HYBRID TECHNOLOGY FOR MOBILE APPLICATION DEVELOPMENT

H T Palliyaguru<sup>a</sup>, L I E P Weerathunga<sup>b</sup>, E A L R Ekanayake<sup>c</sup>, S P Wijesuriya<sup>d</sup>, M P A W Gamage<sup>e</sup>  
<sup>abcde</sup> Sri Lanka Institute of Information Technology, Colombo, Sri Lanka  
*Corresponding email: tship111@gmail.com*

---

### Abstract

Today, as the development of mobile hardware is getting better, the performance index is much higher than the actual requirements of the software configuration. Mobile Phone's features depend more on software. As the Android operating system is getting more widespread in the population of smart devices, the application based on Android SDK attracts much more attention and becomes more useful. Since the differences among programming languages between these devices, a lot of developers spend much time studying the programming language for development. In the case of conducting the research on Fine Payment and Penalty Point System to Sri-Lankan government problems related to the development of native android applications arise. Also, the most significant point is that all the available applications that are developed are not good at cross-platform, which causes some inconvenience to the users as well. In this paper, we propose a method to build an application for smart devices using the Cordova framework, which uses building apps in HTML and JavaScript. It is also good for web developer dives in mobile development and a simple way of making applications for cross-platform within a short time

**Keywords:** M-government, Hybrid Mobile Application, Cordova

---

### 1. Introduction

In recent years, the emergence of smart phones has changed the definition of mobile phones. A phone is no longer just a communication tool, but also an essential part of the people's communication and daily life. It is certain that the future of the network will be the mobile terminal.

Now, the Android system in the electronics market is becoming more and more popular, especially in the smartphone market. Because of the open source, some of the development tools are free, so there are plenty of generated applications. This greatly inspired the people to use the Android system. In addition, it provides a very convenient hardware platform for developers so that they can spend less effort in realizing their ideas. This develops Android further (Shabtai et al., 2010; Butler, 2011; Proffitt, 2011; Tracy, 2012).

As the smart phones and Android system are getting popular, the operations like bio-metric identifications, map and location detection, access and update documents and some others can be moved from the computer to a phone now.

Meanwhile, because of the unfair competition of IT, many applications build illegal programs to steal user information and cause some damage to user's personal privacy. Sometimes, users will pay more attention to the user experience of software. Therefore, the development of the application can not only be limited to the function, but more attention should be paid to the user's experience.

After studying some previous Android applications and access to large amounts of materials, this novel application is build on utilizing Hybrid Technology. This system has an attractive interface and smooth operations.

As Sri Lanka is a developing country, when introducing an application to a government organization/ department, developers must consider the approach of e-governance to m-government (2015). In developing countries where internet penetration is low due to lack of infrastructure and mobile connectivity, m-government is a better option.

When considering some of the cases in Sri Lanka, all Government Departments, Agencies shall develop and deploy mobile applications for providing all their public services through mobile devices to the extent feasible on the mobile platform.

They shall also specify the service levels for such services. A good government to citizen interaction can be done through tools like SMS. Also, open standards shall be adopted for mobile applications for ensuring the interoperability of applications across various operating systems are some of the cases in emerging countries.

In Mobile Government, services can be defined "as a strategy and its implementation involving the utilization of all kinds of wireless and mobile technology, services, applications and devices for improving benefits to the parties involved in e-government including citizens, businesses and all government units" (Kushchu & Kuscu, 2003).

The following benefits can be enriched and analysed under three categories which are benefits to the government, citizens and the industry.

- Cost reduction
- Efficiency
- Transformation/modernization of public sector organizations
- Added convenience and flexibility
- Better services to the citizens
- Ability to reach a larger number of people through mobile devices than would be possible using wired internet only

As a final point, Cordova framework is highly influenced to develop the application of Fine Payment and Penalty Point System under m-Government concept.

## 2. Research Methodology

In the current generation, usage of smart devices like iPhone, iPad, Galaxy-s and Galaxy Tablet PC have become a widespread technological trend. The most competitive advantage of each device is the applications. Therefore, development of the application for these devices is becoming a hot issue. Building apps across different mobile platforms is hard and time-consuming for programmers because each platform has a complex, proprietary language.

The most common and traditional approach is based on programming language of Object-c, a language used mainly to write for the Apple iPhone, Java for RIM Blackberry OS and Google Android and .Net or unmanaged C++ for Windows Mobile and C++, Java, Python, Flash, WRT for Nokia Symbian etc. There's no doubt about that it is difficult to learn all the languages. The most important objective is that the applications developed in these languages of each mobile OS, are not compatible with other platforms. But the larger pool of developers in the industry, who know how to write for standard languages like JavaScript, CSS, and HTML (Ajax), now stand ready to blow this system wide open. They'd do it with the help of Cordova (formerly PhoneGap), an open-source mobile application development framework.

Cordova uses the standard web languages to allow programmers to write one code that can work on many different phones. By expanding the reach of apps to more phones, Cordova could give developers both a larger customer audience and faster channel on which to test

the commercial viability of those apps. The huge need for a multiple-platform service is only heightened by the recent growth in app store solutions. Cordova currently allows developers to work on the iPhone, BlackBerry and Android OS, and others' systems as well.

Before committing to a platform strategy, it's important to evaluate the technical and non-technical assets of hybrid versus alternative like native, especially as it relates to the mobile application's requirements. This table will serve as a comparison between the two mobile app types.

Table 1: Comparison between Native and Hybrid Mobile App Types

	<b>Native</b>	<b>Hybrid</b>
Development language	Native only	Native / Web
Coding language	Java, Objective-c	JavaScript (native portion) + HTML5 (WebView) and CSS
Development time	High	Medium
Cost	High	Low
Portability	One code for one platform only.	One code for major mobile platforms.
Device Access	Platform SDK enables access to all device API's.	Many device API's closed to web apps can access.
UI consistency	Comes with familiar original UI components.	UI frameworks can achieve a fairly native look.
Distribution	Neutral	Neutral
Performance	Has a direct access to platform functionality.	Sub-optimal performance.

## I. Development of the Hybrid project

Using Cordova may be the fastest way for many web developers to start building a mobile application. With Cordova, developers do not need to learn a new set of programming languages. Hybrid mobile application can be simply built by using the most commonly used languages such as HTML, CSS, and JavaScript.

*jQuery*: jQuery is a robust mobile development framework to build **cross-mobile-platform** JavaScript library designed to simplify the client-side scripting of HTML. jQuery support a wide range of different platforms, from a regular desktop, smart phone, tablet or an e-reader device like Nook or Kindle. And it is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. Introduction to mobile application development, jQuery is designed to change the way of writing JavaScript (2015). It is a quite simple way to develop and reduce codes and time efficiency. But some libraries for jQuery are too heavy to run in mobile devices, and will work not as well as in PC. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library.

*HTML, CSS and JavaScript*: For many web developers, which may only be familiar with HTML, CSS and JavaScript, developing a native mobile app could be unfamiliar territory. Technically speaking, mobile apps in Android, iOS, and Windows Phone are built using completely different programming languages; an Android app uses Java, an iOS app uses Objective-C, while a Windows Phone app uses .NET.

But now, anyone with a decent knowledge of HTML, CSS, and JavaScript can build a mobile application. One key advantage of using web technology to build your app is Portability. Using a packager/compiler, like Cordova, you will be able to port and install your app on many different platforms.

*Ajax (Asynchronous JavaScript and XML) AJAX* is not a new programming language, but a new way to use existing standards. AJAX is the art of exchanging data with a server, and updating parts of a web page, without reloading the whole page (2015). This technology plays a significant role in runtime of the mobile application.

However, with the capabilities of above mentioned programming languages, user can compile the codes into numerous mobile platforms including iOS, Android, Windows Phone and BlackBerry using the Cordova build tool. Therefore, it is essential to have a deep understanding about the procedure of developing a new Cordova project.

Cordova comes with Command-line Interface (CLI) that includes the command to create a project and build the app. The Cordova CLI is available as a Node.js Package. Therefore, users have to install Node.js to be able to install it.

Node.js is an open-source, cross-platform runtime environment for developing server-side web applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, Linux, FreeBSD, Nonstop, IBM AIX, IBM System z and IBM I (2015).

Assuming that user have installed in the computer, can type this command line in Terminal or Command Prompt to install Cordova CLI.

```
npm install -g cordova
```

In order to proceed with the development of Cordova project, follow the Apache Cordova Documentation (2015).

## II. Cordova Plug-ins

Cordova plugin bridges functionality between the Web View powering a Cordova application and the native platform that Cordova application is running on. Plugins are composed of a

single JavaScript interface used across all platforms, and native implementations following platform-specific Plugin interfaces that the JavaScript calls into.

The entry point for any plugin is JavaScript. The main reason developer's use Cordova is ability to use JavaScript, without using Objective-C, Java and C#. All of the core Cordova APIs are implemented using this architecture.

Cordova is a platform that provides a consistent set of JavaScript APIs to access device capabilities through plug-ins, which are built with native code. These plug-ins include APIs for accessing the device's accelerometer, contacts, camera, and more. There is also a number of plug-ins that are built and maintained by the developer community at-large. These can be found in the Apache Cordova Plug-ins Registry (2015).

### **3. Research findings/Results and evidence**

Hybrid mobile applications provide a way for developers to re-use their existing skills in web and mobile application development. Developers do not like the prospect of getting limited into copyrighted platforms. This includes the programming languages and SDKs provided by platform vendors.

Hybrid mobile application development looks tempting to an organization's bottom line. There's no reason to hire a developer for each platform where one developer can target all of them through HTML, CSS, and JavaScript. It is true that hybrid mobile application development enables developers to target more than one platform.

However, each platform comes with a set of cautions when it comes to its web runtime or Web View. This is especially true with Android, which is inconsistent between OS versions. Moreover, there might be unique capabilities of platforms to which a developer may wish to target. In those cases, a combination of plugins and platform-specific code must be utilized in order to take advantages of those capabilities.

### **4. Comparison between Hybrid Vs. Native**

There are benefits and drawbacks to using either technology. In this section, some of the special attributes of using hybrid app development (HTML5, CSS and JavaScript) vs. native app development technologies are discussed.

#### **Design of Interfaces**

Most mobile interfaces can be implemented using either native app or hybrid app development technologies; however, there is more flexibility in designing these interfaces using hybrid / Cordova app development technology. Hybrid app development uses HTML5 and CSS3 that have advanced capabilities when designing interfaces with many form factors and highly dynamic content.

However, there are few drawbacks and interface design issues that may come up when using hybrid app development.

Sometimes, it is tedious and time consuming to implement certain standard native user experience designs and complicated interfaces using hybrid app development technology.

Cost:

Development of mobile apps using hybrid / Cordova app development technology is highly cost effective. Hybrid / Cordova apps leverage high quality and diverse sets of libraries, as well as providing the tools required to moderate the development time.

Also, with hybrid app development, it can build once and submit it to all of the platforms (iPhone, Android, Windows Phone) using Cordova technology. So, it would save money by not having to build the app using native programming language of each platform.

Below table shows the average cost of mobile app development using hybrid vs. native mobile app development.

Table 2: Cost: Hybrid Vs Native

Description	Small to Medium Size Projects Average Cost of Native App Development	Small to Medium Size Projects Average Cost of Hybrid App Development
Back-end programming, APIs, admin and cloud deployment	\$10,000- \$20,000	\$10,000- \$20,000
iOS native development	\$15,000- \$30,000	\$0
Android native development	\$10,000- \$20,000	\$0
Hybrid app development using Cordova technology	\$0	\$10,000- \$20,000
Total	\$35,000- \$70,000	\$20,000- \$40,000

The above cost is just a ball park average cost. The cost can be lower or higher based on mobile app project size and specifications.

There are times when the developer uses native coding within the hybrid environment in order to implement certain features that are only available through native development. For the purpose of providing the above cost, assume that the hybrid project does not require any native app development (2015).

#### Development Timeline

Hybrid apps are easier and faster to develop and deploy. However, if there are a lot of native and complicated features that are beyond the capability of the hybrid app framework and environment, then it will be more time consuming to build for hybrid apps.

#### Security

Depending on the security requirements of the project, most of the security issues are created based on the poorly written codes and developer's lack of experience in security and server-side security issues.

#### Code Reusability/Portability

The biggest weakness of native apps is their lack of portability to other platforms (iPhone, Android, Windows Phone and Blackberry). The appeal of web apps is that you can have one Code base and run it on any major mobile platform. The appeal of hybrid apps is similar, because of high ability to reuse a large amount of codes for each platform.

### Maintainability

Hybrid / Cordova apps are more maintainable as long as the developer chooses a right framework such as KendoUI, jQuery mobile, Ionic, etc. This is because it is easier to maintain the Web app technology (HTML, CSS and JavaScript) than a native app technology.

Also, it is important to note that if the right framework is not used for hybrid app and the code is not written properly, the app will not be maintainable.

## II. Cordova for Enterprise Mobile Application development

While evaluating the potential solutions for creating a mobile app, developers always prefer to use Cordova, due to its superior features and advantages.

Firstly, the platforms that the application should support were considered and mostly the goal was to launch for Android, as well as iOS devices. Later, plans were made to expand the support to BlackBerry and Windows devices. Creating applications for each platform using the native programming platforms like Objective-C on iOS and Java on android required maintaining at least four different set of codes in the long term.

Advantages of using Apache Cordova for Enterprise Mobile App development:

- Various companies that appreciate the standard based approach to multi-platform mobile development and HTML5 becomes the preferred choice as it balances compatibility of platform along with user experience.
- The APIs of Apache Cordova for accessing device features are powerful applications that access many mobile device capabilities like address book, accelerometer, camera and so forth.
- Cordova's extensive architecture also allows the enterprises to utilize the full potential of the mobile device.
- Since a lot of organizations already have in house pool of HTML/ JavaScript expertise, it is easy for them to use their skills in most cost effective manner to develop applications that can run on various mobile devices.

As mentioned in above, Cordova is exclusive for any vibrant projects and it is supported by a great community for a great framework that offers substantial benefits, in terms of enterprise application development.

When considering about developer's perspective of Hybrid technology,

- Free & Open source.
- Use of standard web development - any text editor, firebug (no investment on special tools)
- Most important & most used plugins are free in-app purchase.
- No need to know native APIs and native app development is almost completely abstract.
- Development of the tool is at a much faster compared to other tools in this space.
- Giant development community to acquire support.

When considering about customer's perspective of Hybrid technology,

- Support for all major platforms (iOS, Android, BlackBerry, Windows mobile, and Web) with one code base.
- User friendly interfaces.

All of the above mentioned features are highly influenced to the rapid increase in market trends of Hybrid Technology.

### III. The Technology Life Cycle

Development of Hybrid applications has gone to peak level with its ups and downs throughout past few years. But, despite prominent rejections hybrid development continues to be used by a significant number of developers. Gartner hype cycle (shown below) is to be an elegant way to visualize the life cycle of most technologies (2015).



Figure 1: Gartner Hype cycle [13]



The hype cycle breaks a technology's life into five phases as Technology Trigger, Peak of Inflated Expectations, Trough of Disillusionment, Slope of Enlightenment, and Plateau of Productivity.

*Technology Trigger:* The technology trigger for hybrid was the rapid rise of multiple mobile platforms. Developers were and still are overcome by the need to support a growing number of platforms. Hybrid mobile development is a single codebase approach, particularly used by web developers who attempted at the opportunity to develop native iOS and Android apps with the technologies they already knew.

*Peak of Inflated Expectations:* Cordova provides an elegant way to create hybrid apps without having conflicts with device SDKs. PhoneGap's success, and its acquisition by Adobe in 2011, is considered to be hybrid's *Peak of Inflated Expectations*.

*Trough of Disillusionment :* Hybrid's popularity were increase rapidly, but after sometime, because of lack of tooling and performance a number of companies publicly abandoned hybrid. A few months later, LinkedIn joined Facebook by switching from HTML to native for its mobile apps.

These public moves represent hybrid's *Trough of Disillusionment*.

*Slope of Enlightenment :* Over the last year 2014, mobile developers have been overcome with tools and services. Due to this, the current state of hybrid, or the *Slope of Enlightenment* from the hype cycle, as the urgency to provide the tooling needed for hybrid to be a viable mobile app development option. Remote debugging, cloud-based builds, backend are some tooling improvements.

#### IV. The Future of Hybrid Technology

Google trends shows that the popularity of Hybrid technology is still slowly growing because of the improvement of performance and tooling. According to analysis, it shows more than half of mobile apps deployed by enterprise by 2016 will be hybrid.



Figure 2

## V. Testing of Cordova Project

Fine payment and Penalty Point System is mainly developed with the objective of “Identifying the issues faced by motorists, in traffic fine payment process and provide an automated and computerized fine payment system to increase the level of satisfaction towards Sri Lanka Police Department”, under the m-government concept. According to the requirements of the system, Hybrid is selected as development technology after conducting long term research on development technologies.

The facts which motivate to select the Hybrid Technology and their performance are graphically represented as follows;

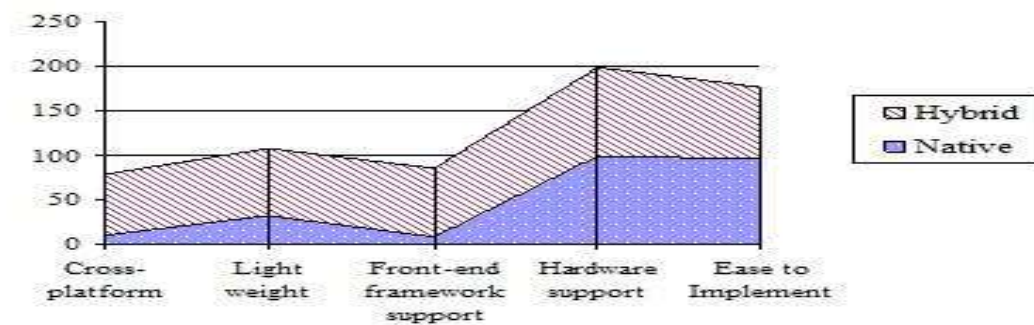


Figure 3

## Conclusions & Future works

The applications based on Hybrid framework have the following advantages against traditional applications of smart devices:

- 1) Using all technologies are already well known worldwide, thus making it easy to get started.
- 2) With the simple way of developing, the total time of development is becoming shorter.
- 3) Cross-platform is the most attractive point of this approach; which can easily make the

Application running any other devices if it has any platform.

However, choosing the correct development model (Hybrid or native) may not always be easy. Therefore, users should keep in mind all of the specific requirements and critical parameters like project budget, performance, time to market and resources and so on.

## References

- i. AJAX Material required. 2015. Available at: <http://www.w3schools.com/ajax/default.asp> (Accessed: July 21, 2015)
- ii. Apache Cordova Documentation. 2015. Available at: [http://cordova.apache.org/docs/en/3.1.0/guide\\_Overview\\_indexmd.html#overview](http://cordova.apache.org/docs/en/3.1.0/guide_Overview_indexmd.html#overview) (Accessed: July 07, 2015)
- iii. Butler, M., 2011. Android: Changing the Mobile Landscape. *Pervasive Computing*, pp. 4-7.
- iv. Cordova Plugins. 2015. Available at: <http://plugins.cordova.io/npm/index.html> (Accessed: August 11, 2015)
- v. jQuery Google Feed Plugin. 2015. Available at: <http://jquery.com/> (Accessed: August 16, 2015)
- vi. Kushchu, I. & Kuscü, M. H., 2003. *From e-Government to m-Government: Facing the Inevitable*. The 3rd European Conference on e-Government, pp. 253–260.
- vii. M-Government. 2015. Available at: <https://en.wikipedia.org/wiki/M-government> (Accessed: August 29, 2015)
- viii. Native vs. Hybrid / PhoneGap App Development Comparison. 2015. Available at: <http://www.comentum.com/phonegap-vs-native-app-development.html> (Accessed: September 24, 2015)
- ix. Node.js. 2015. Available at: <https://en.wikipedia.org/wiki/Node.js> (Accessed: September 10, 2015)
- x. Proffitt, B., 2011. Open Android-For Better and For Worse. *Spectrum*, pp. 22– 24.
- xi. Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S. & Glezer, C., 2010. Google Android: A Comprehensive Security Assessment. *Security & Privacy*, pp. 35–44.
- xii. The State of Hybrid Mobile Development. 2015. Available at: <http://developer.telerik.com/featured/the-state-of-hybrid-mobile-development/> (Accessed: September 17, 2015)
- xiii. Tracy, K. W., 2012. Mobile Application Development Experiences on Apple's iOS and Android OS. *Potentials*, pp. 30 – 34.