

Taxonomy of Cross-Platform Mobile Applications Development Tools

Amarbir Singh*

*(Department of Computer Science, Guru Nanak Dev University, Amritsar)

Abstract:

Most of the developers use the cross-platform mobile development solutions to develop the mobile application once and run it on many platforms. Many of these cross-platform solutions are still under research and development. Also, these solutions are based on different approaches such as Cross-Compilation approach, Virtual Machine approach, and Web-Based approach. There are many survey papers about the cross-platform mobile development solutions but they do not include the most recent tools, including Component-Based approach, Cloud-Based approach, and Merged approach. The main aim of this paper is helping the researchers to know the most recent approaches and the open research issues. This paper surveys the existing cross platform mobile development tools, introduces a comprehensive categorization to the cross-platform tools, defines the pros and cons of each approach and compares the cross-platform mobile development solutions.

Keywords —Android, PhoneGap, Rhodes, Appcelerator, Xamarin

I. INTRODUCTION

One of the joys and pains of being a software developer is choosing the right tools for the job. Technology moves so fast and everyday there are new frameworks and libraries being shared with the world in order to do the job more efficiently. Most people think this would always be a good thing, but choices can be hard. The best one can do is make the best choice with the information you have, and that information is always changing. We are going to take a look at some of the leading frameworks that might be able to help you create your mobile applications a little more efficiently. Before we look at some of the frameworks, we have to set the stage for why there are options. When the latest mobile movement emerged during the release of the first iPhone there were limited choices on how you could produce your applications. Your only real option was

Objective C; that was what Apple was pushing and all their software development kit documentation and examples revolved around that.

Then Google got involved in the game and after that everything their team produced and documented for doing development was with Java. Of course some people were still dabbling with their Palm and Blackberry at this time, which would have their own set of tools and learning curve. As time progressed, developers started getting a little tired of having to learn a new language and a new way of doing everything on each mobile platform. Frameworks started emerging claiming to be able to develop once and distribute to multiple platforms, allowing developers to focus on getting good at one thing and leveraging that knowledge across mobile devices [1]. The promise was that it was going to allow developers to focus on learning one thing well.

II. GOAL AND BENEFITS OF USING CROSS-PLATFORM DEVELOPMENT TOOLS

There are number of reasons due to which need for cross platform development tools arisen. The primary goal was to create code once and then distribute it to all the leading mobile devices having

different operating systems. Another reason is that, if we use cross platform development tool then we can focus on one or our favourite programming to develop applications otherwise there is a need of expertise in the language supported by the mobile device. By using different Cross platform development tools quality code can be created faster which helps in getting the products to market faster and such code is also supported by current as well as future devices [3]. Various benefits of using cross platform development tools are listed below.

A. Code Is Reusable

Rather than having to write the specific action or sequence for each platform, a developer can just write the code once and then reuse those bits in later projects or on other platforms.

B. Plugins

Major frameworks, including Appcelerator and PhoneGap offer easy access to plugins and modules that can easily plug into other services or tools.

C. Easy for Web Developers

Because most frameworks are dynamic or scripting languages, they are easy for web developers to jump in and use. Moreover, most frameworks have the facility to manage and support HTML5 and CSS3 alongside the calls to more native functions.

D. Reduced Development Costs

This is perhaps the biggest advantage because it allows companies or brands to get an app onto other platforms *without* having to invest in a team or developer specific to that ecosystem.

E. Support for Enterprise and Cloud Services

In addition to plugins and modules for specific functions, most frameworks also have the option to directly integrate with cloud services, including Salesforce.com, AWS, Box.net and others.

F. Easy Deployment

Deploying and compiling apps is much faster in a cross-platform scenario. This is especially true with many of the new cloud-based build

tools that various frameworks are starting to push out.

III. VARIOUS CROSS-PLATFORM APPLICATIONS DEVELOPMENT TOOLS

A. PhoneGap

In PhoneGap all of the front-end is actually delivered as HTML and JavaScript, but the experience is packaged and surrounded with the Cordova magic. What this gives developers the ability to do is use their past web experience and create apps like they would create websites and then use specialized markup to access specific device features. This would include gaining access to the camera, triggering a vibration and getting input back from the accelerometer. You can accomplish a lot, very easily by going this direction with your mobile development. It's easy because you can leverage your teams experience with web development and put that to work for you with mobile applications. However, I feel this route is the most limiting to the developer and the future of the mobile product. You can really get boxed in by the types of features you can ultimately deliver since you are so abstracted from the core device APIs [4]. This route is going to be good if you have a very sound understanding of the application you are creating and all of it's current and expected future features. If you can with 100% confidence determine everything you need is do-able with this route, then go for it. I, however, often have a fear that 6, 12, 24 months down the road I'll get a feature request from a client that will seem like a common mobile feature or no big deal and get completely stone-walled by this approach to mobile development. For that reason I'm pretty hesitant to do anything terribly complex with these technologies. However, they are awesome for mimicking a set of website features on mobile or enabling a website to work offline via a mobile app.

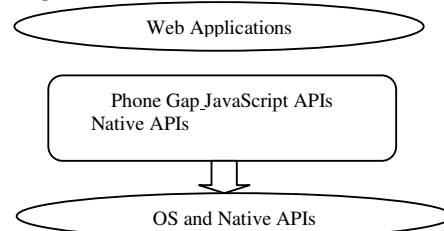


Fig 1 Interfacing layers of PhoneGap architecture [5].

PhoneGap holds the top slot in developer mindshare. Cordova/PhoneGap developers write their mobile applications using HTML, JavaScript and CSS. These assets run in a “WebView” inside a native application container on the target platform as shown in Fig 1. The various advantages and disadvantages for PhoneGap are discussed below.

Pros

1) Regardless of server side platform & language experience, a significant number of developers have experience with HTML, JavaScript and CSS. Apache Cordova or PhoneGap allows developers to immediately leverage these existing skills. The value of this can't be overstated – as it reduces training and can enable a quick-to-market stance in companies ready to adopt it.

2) PhoneGap apps install just like a native application, and are able to leverage app store discoverability.

3) PhoneGap follows a plugin architecture, which means that access to native device APIs can be extended in a modular way. There are a lot of PhoneGap plugins to choose from – enabling developers to focus on the web-based skills they already have. Cordova is open source and free, so there are no licensing costs

Cons

1) The strength of being open source and leveraging the talents of a wide array of contributors is both a blessing and curse. If you need to extend your app with a custom Cordova/PhoneGap plugin, odds are you will find one. Yet it may be out of date and not support the target platforms you need.

2) The performance of PhoneGap apps has often been criticized. Native UI will always outperform a hybrid solution, but improvements in device hardware and WebView implementations have narrowed the gap. Your web developers will need to pay close attention to performance, which means their knowledge of profiling tools as well as which web UI frameworks are mobile-friendly is essential.

B.Rhodes

Rhodes is a cross-platform mobile application tool developed by Motorola Solutions Inc. under Massachusetts Institute of Technology (MIT). It is developed to rapidly build native applications for

all major mobile OS's (iOS, Android, BlackBerry, Windows Mobile/Phone and Symbian). The main goal of Rhodes is to provide a high level of productivity and portability in programming. It is an open source Ruby-based mobile development environment. Thanks to this environment, developers can create and maintain enterprise applications and data based on single source code across different mobile OS's [6].

RhoMobile suite provides an IDE called RhoStudio which is an innovative solution dedicated to users that want to develop applications through a hosted IDE. This solution can be used across Linux, Mac, and Microsoft Windows OS's. Alternatively, RhoMobile offers the possibility to write applications with any other editor or IDE which supports HTML, HTML5, CSS, JavaScript and Ruby. The most popular editors are Eclipse, Visual Studio, Netbeans, IntelliJ and TextMate [7]. Rhodes provides native device applications to improve the end-user experience, which work with synchronized local data and take advantage of device capabilities, such as Barcode, Bluetooth, Calendar, Camera, Contacts, GPS, Menu, Near Field Communication (NFC) and Screen Rotation. Rhodes is the only framework that uses Model View Controller (MVC) pattern to develop mobile applications. The MVC pattern creates applications that separate data definitions (models) from business logic and (controllers) from interfaces (views), providing at the same time a point of

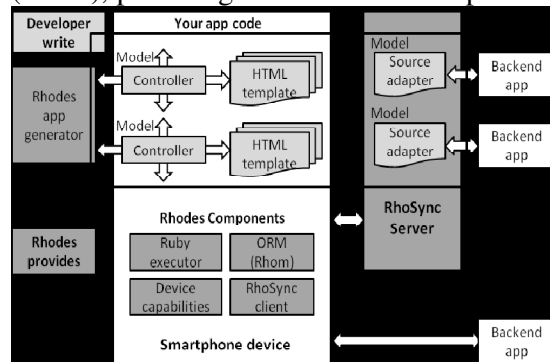


Fig. 2. Rhodes interfacing architecture between the Smartphone and Rhodes components [9].

connection between these elements. Languages used in the view element are HTML, CSS and JavaScript to make mobile applications, whereas in

the controller element is Ruby to make the backend support. Moreover, with MVC approach is also possible to write applications that use only the view element. Obviously, it is realizable for applications or sites that require a low level of complexity [8]. Rhodes provides mainly three possibilities to add extendibility in its framework, first can be done by adding external Ruby library to Rhodes, second by creating native extensions for specific Software Development Kit (SDK) of each OS and last by extending the already existing views available in Rhodes. In Fig. 2 Rhodes architecture is shown. Controller, HTML templates and source adapter components are the parts which developers have to implement for the creation of applications, whereas other components are provided by Rhodes such as Rhodes App Generator which is an IDE than can be RhoStudio or another editor, Ruby Executor is the executor of the Ruby code, Device Capabilities are the API's, Rhom is a mini database ORM (object relational mapper) which provides a high level interface to make it quickly and easily (i.e. the database is SQLite for all mobile OS's except BlackBerry that is HSQLDB), RhoSync Client is a library to add sync data capability to your applications, and RhoSync simplifies the development of connectivity to enterprise backend applications. Moreover, performing the backend application integration between RhoSync Client and RhoSync Server is reduced by 50-80% the development effort [7]. Various advantages and disadvantages of using Rhodes are given below.

Pros

- 1) Rhodes development files are compiled into native applications that can be executed on real or a virtual devices, indeed, this tool offers a desktop simulator where is possible to run applications.
- 2) Provides high level of productivity and portability in programming.

Cons

- 1) The plugin architecture works well if you can find the plugins you need or if your web developers are capable of changing gears to write their own custom plugin(s) as needed.

C. DragonRad

DragonRad 5.0 is a cross-platform mobile ADT developed by Seregon Solutions Inc. DragonRad

has Drag and Drop (D&D) features, which requires reduced programming talent to develop applications. It allows developers to design, manage and deploy mobile applications with ease. These deployed mobile applications can be used across iOS, Android, BlackBerry and Windows Mobile. This tool concentrates on database driven mobile enterprise applications with easy and wide range of databases support. DragonRad facilitates integration and synchronization of database system with native functions such as Contacts, Calendar, Geolocation, Menu and Storage. Fig 3 shows the detailed architecture of DragonRad for Android. Various advantages and disadvantages of using DragonRad are given below.

Pros

- 1) This tool is pretty easy to use and has seamless integration to MySQL database.
- 2) It is very intuitive and application development is extremely fast.

Cons

- 1) DragonRad is a Proprietary platform having a small user community.
- 2) Documentation is not really good and examples are tough to understand.
- 3) Dragonrad is not suited for building games, it is useful for building data-driven enterprise applications only.

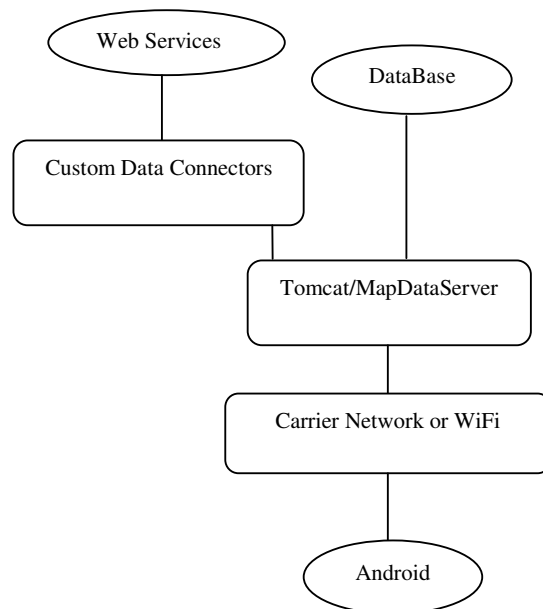


Fig 3 Architecture of DragonRAD for Android [5].

D. Appcelerator

Appcelerator has been in the cross-platform space since 2006. They originally had an application that was targeted towards making it easy to write desktop apps for both Windows and Apple. They then changed their focus to work on building a tool that made it easier to program cross-platform mobile applications.

Titanium is their open source IDE/compiler. It allows you to write your mobile application using JavaScript syntax and tap into the necessary core libraries to create native mobile experiences. With Titanium we have found that it's not really a one-and-done workflow. There are still differences that need to be accounted for when developing cross-platform applications. For example, an iPhone really only has a home button. All app interaction must be programmed within the application [10]. However, with Android there is a physical back button and other buttons on the keyboard that might interact with your application. Specific situations still need to be accounted for when developing for both of these platforms.

Appcelerator also claims that their framework is supporting Windows and Blackberry, but from what I've read, it's been mixed results and certainly isn't a one click process to get your application running on those platforms. Appcelerator has paid subscription options, but they still have a free, open-source version that has majority of what you need to succeed. The biggest downside to Titanium is that you can't access everything in the core frameworks. Titanium tries to model and make methods available to you for the most popular device features, but unlike other options, everything is not available out of the box. The good news is that you can always create a custom Titanium module that calls a core feature you need and include that in your project.

Appcelerator's Titanium provides a unified (across devices) JavaScript API, coupled with native-platform-specific features. Developers write JavaScript and utilize a UI abstraction (the Alloy MVC framework) that results in the use of native UI components, greatly aiding UI performance compared to other hybrid options [11]. The various advantages and disadvantages for Appcelerator are discussed below.

Pros

- 1) The use of native UI components is a performance win, and the Alloy framework attempts to normalize UI across platforms.
- 2) The use of JavaScript to normalize code across platforms enables you to leverage existing skills on multiple target platforms.
- 3) Appcelerator provides value-adds such as a Backend-as-a-Service (BaaS), app analytics and a marketplace for 3rd party components.

Cons

- 1) Developers are required to manage target platform SDKs locally. It's highly recommended for your team to establish a controlled build environment/CI process if you choose to manage SDKs locally, especially if you target multiple platforms. SDK version & build-related issues can be a horrific time sink, when you really need your team delivering features.

- 2) Normalizing the UI across platforms, while arguably a "pro", is also a "con" in that your team will need to train on a proprietary technology to gain skills that are not directly transferrable outside Titanium.

E. Xamarin

"Xamarin" has only been around for a couple years, however the team and technology behind it has been in the cross-platform industry for many years. Before developers were concerned about cross-platform mobile apps they were concerned about cross-platform desktop applications [12]. Members of the Xamarin team were behind the very popular Mono project and later the MonoTouch project, which was acquired, spun off, and things happened... then poof: Xamarin.

If you are a Microsoft .Net developer, then you'll know that Mono was the rogue project that worked tirelessly to keep the C# language running and available on Linux. They know a thing or two about working with cross-platform technologies, and they are huge C# fans. Therefore, the programming language preferred for Xamarin is C#. If you are a .NET developer this will probably be your favorite. Xamarin has been excelling at building really cool cloud-testing tools as well. This makes it easier for developers to test their

applications across a variety of different devices [13].

Pros

1) Compiles into a truenative APP or APK, uses native UI tools and has cross platform benefits as well as stability.

2) Favourite tool for the .NET developers.

Cons

1) Does not provide build once experience, so little more work has to be done if code has to be reused.

| | | | |
|--------------|-------------------------|--------------------|-------------|
| | HTML5, CSS, Java Script | RhoHub | |
| DragonRad | D&D | DragonRad Designer | NA |
| Appcelerator | HTML, Java Script | Titanium Studio | Java Script |
| Xamarin | .Net, HTML | Xamarin Studio | NA |

Table 2 Development Features

IV. COMPARISON

The Comparison of the five popular cross platform applications development tools is given in Table 1 and Table 2. Table 1 contains the comparison different general features of tools and Table 2 contains the comparison of various development features of tools.

| Name | Platfoms | OS Support | Open Source | MVC |
|--------------|--|---------------------|-------------|-----|
| PhoneGap | IOS, Android, Windows, Blackberry, Symbian | Linux, Mac, Windows | Yes | No |
| Rhodes | IOS, Android, Windows, Blackberry, Symbian | Linux, Mac, Windows | Yes | Yes |
| DragonRad | IOS, Android, Windows, Blackberry, | Linux, Mac, Windows | No | No |
| Appcelerator | IOS, Android, Windows, Blackberry | Linux, Mac, Windows | Yes | Yes |
| Xamarin | IOS, Android, Windows | Linux, Mac, Windows | Yes | No |

Table 1 General Features

| Name | Language | IDE | Acessibility To Native API |
|----------|--------------------------------|----------------|----------------------------|
| PhoneGap | HTML, HTML5, CSS3, Java Script | Eclipse, XCode | Java Script |
| Rhodes | HTML, | RhoStudio, | Java Script |

V. CONCLUSION

There is never a winner when it comes to programming languages and frameworks. The winner is the option you think will be best for you and your product now and into the future. It very well might be that the best option is to write Objective C or Swift for iOS and Java for Android, just because you're awesome at those. Then you can learn C# if needed for Windows later. The choice is yours, and that's the beauty of it. Our paper is an attempt to make this task of choosing an appropriate tool easier.

REFERENCES

- [1] Ma, L., Gu, L. and Wang, J., "Research and Development of Mobile Application for Android Platform", International Journal of Multimedia and Ubiquitous Engineering, Vol. 9, No. 4, pp. 187-198,2014.
- [2] Kim, J.H., Karunaratne, S., Regenbrecht, H., and Wunsche, C.B.W.I., "Evaluation of Cross-Platform Development Tools for Patient Self-Reporting on Mobile Devices", Proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management, Sydney, Australia, pp. 27 – 30. 2015.
- [3] Pawar, P.A., Jagtap, S.V. and Bhamare, S.M., "Survey on Techniques for Cross Platform Mobile Application Development", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 3, Issue 10, pp. 3552-3558, 2014.
- [4] Pandey, G. and Dani, D., "Android Mobile Application Build on Eclipse", International Journal of Scientific and Research Publications, Vol. 4, Issue 2, pp. 1-5, 2014
- [5] Palmieri, M., Singh, I., Cicchetti, A., "Comparison of Cross-Platform Mobile Development Tools", 16th International Conference on Intelligence in Next Generation Networks, Berlin, Germany, pp. 179-186, 2012.
- [6] G. Sarah Allen and L. Lundrigan. Smartphone Cross-PlatformDevelopment. Apress, 2010.
- [7] Motorola Solutions. <http://www.motorola.com/Business/USEN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite,2012>.
- [8] RhoMobile, <http://rhomobile.com/blog/whats-so-special-about-therhodes-smartphone-app-framework>, March 2010.

- [9] LeckylaoRhodesframework,
<http://leckylao.com/2010/06/12/rhodes-framework-agile-mobileweb-development>, June 2010.
- [10] Ertaul, L. and Shakoor, S., "Implementation of Pinkas Partial Matching (PM)-Semi Honest Protocol Using Mixed Multiplicative Homomorphic Encryption (MMH) For Location Based Services (LBS)", *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 15, No. 3, pp. 1-10, 2015.
- [11] Farto, D.G., Endo, T.A., "Mechanisms to support automated testing of mobile applications", *Thesis and Dissertation Workshop on Software Quality*, Blumenau, Santa Catarina, Brazil, pp. 1-6, 2014.
- [12] Kumar, A.L., 2012. "Mobile Application for News and Interactive Services", *ARNP Journal of Science and Technology*, Vol. 2, No. 1, pp. 1-16, 2012.
- [13] Nepali, J., Kadariya, L. and Kandel, Pd.N., "Restaurant Guide GPS based Android App", *Minor Project Report*, Tribhuvan University, Nepal, pp. 1-30, 2012.