

Comparative Study on Techniques for Cross-Platform Mobile Application Development

¹Abhinav Swami, ²Dr. Rajiv Singh

¹M.Tech Student, ²Head of Department

¹Computer Science,

¹Monad University, Hapur, U.P, India

Abstract— every person use smart phone today. Many mobile applications are used and launched as per user need. In order to do work on Mobile Device, many desktop applications are getting converted to mobile version by developers. From developer point of view for application to reach to most of the end users it need to run on max platforms, this needs redevelopment of application, we can solve this problem to some extent by developing cross platform application without additional investment. Several techniques are available which will help to make it happen. Survey of these techniques could help application developers to make a proper choice [1]. But here we are pick up most traditional approach to develop mobile application.

Cross-platform mobile application development tools contribute in solving this problem largely. This paper presents a pragmatic comparison among six very popular cross platform tools, which are Xamarin, Adobe Air, Rhodes, PhoneGap, DragonRad and MoSync.

Index Terms— Cross Platform Application, Cross Compiled Approach, Cross Platform Application Development Tools.

I. INTRODUCTION

One of the problems in mobile development is that of end user isolation. The choice of platform, required features and operating system backwards compatibility segments users into groups which often limit the growth of the application user base. To combat this there are a limited number of options available. The first solution was to rewrite the entire mobile application independently for each platform. This of course leads to high development and Maintenance costs for these applications. [2]

The solution of this problem is to go to cross platform application development.

Cross platform applications means developing single application or writing single source code and get it executed on different platforms. While development of mobile application developer checks for technique which will be best to save money, avoid learning new languages and make application fit for maximum devices. [1]

Cross platform development tools gives a solution. There are number of tools available for cross platform development for example, Rhodes, Titanium, PhoneGap, Xamarin, MoSync, IBM Worklight, Corona, Marmalade, Adobe Air, DragonRad [1].

II. XAMARIN

Xamarin is a relatively new firm. Indeed, the engineers of Mono, and MonoTouch have created it in 2011 [3]. This software intends to improve the developers experience in writing mobile applications for the three giants iOS, Android and Windows Phone. Although, Xamarin is recent more than 330,000 developers have adopted it. Several well-known companies have selected Xamarin to build their mobile application such as the famous Rdio app [4]. Xamarin Starter package is free for developers. This package comprises Xamarin Studio software and documentation for further technical support. It also offers a market place to quickly add components or frameworks usually free otherwise for a certain amount of money [5].

Xamarin provides a set of different services and software for fresh developers as well as confirmed developers or organizations.

Xamarin Studio

Xamarin studio is a software allowing developers to build, test, and publish rich mobile application for iOS, Android, and WP. It includes Xamarin iOS, Xamarin mobile, and Xamarin Android. This software comply C# apps, through the .NET runtime, into native ARM executable package for iOS or Android platform.

Developers can achieve the following tasks with Xamarin Studio:

- Create complex design through iOS and Android designer tools included
- Create structured apps depending on the platform

Sr. No.	Tools	Programming Language Used	License	Device capability Support	Integrated Development Environment support and build environment	Cross platform development on
1.	Adobe AIR	ActionScript, HTML, CSS, JavaScript	Flash Builder, Flash Professional, IntelliJ IDEA - Commercial licenses available Adobe AIR SDK (command line tool) – Free	-	Flash Builder, Flash Professional, IntelliJ IDEA	iOS, Android, BlackBerry
2.	Xamarin	C#	Open Source licence- LGPLv2 license	-	Xamarin Studio or Visual Studio	iOS, Android and Windows
3.	Rhodes	Ruby with HTML interface features compiled through an interpreter into native applications.	Rhodes is free and open source under the MIT License, RhoSync is under GPL or commercial, Commercial support available. Subscription for RhoHub	Barcode, Contact, Calendar, GPS, NFC (near field Communication), Screen rotation, Camera, Signature	Xcode or Eclipse, on-demand RhoHub version includes full IDE	supports iOS (incl. 3.0) on iPhone and iPad, Windows Mobile 6.1 Professional, Mobile Windows 6.0 Standard, BlackBerry 4.6, 4.7, 5.0, 6.0 (BlackBerry 4.2 and 4.5 supported but database access is very slow on these devices), Symbian and Android 1.6 and higher
4.	PhoneGap	HTML, CSS, JavaScript	Apache 2	Geolocation, Files, Network, Accelerometer, Notifications, Camera, Contacts, Compass, Storage and media	Any SDK can be used just PhoneGap plugins need to be installed.	iPhone, Android, Tizen, Windows Phone, BlackBerry, Symbian, Palm, Bada
5.	DragonRaid	D and D	GPL2	Maps, Location based service, Calendar, Camera, Native Device keys, Contacts and payments	Own IDE	iPhone, Windows Mobile BlackBerry, Android, Linux, Mac and Windows
6.	MoSync	C, C++, Lua, HTML5, CSS, JavaScript	Free, GPL 2.0, Free Indie Subscription; commercial subscription available	-	Eclipse, Visual Studio 2005 and later, MoBuild w/ text editors	Android, Java ME, Moblin, iOS (iPhone), Smartphone 2003, Symbian, Windows Mobile (Pocket PC), Blackberry (experimental)

- Code revision via a module/github
- Test apps through devices or emulators
- Deploy apps into apps stores

III. ADOBE AIR

Adobe AIR (formerly **Adobe Integrated Runtime**) is a cross-platform runtime system developed by Adobe Systems for building desktop applications and mobile applications, programmed using Adobe Flash, [ActionScript](#) and optionally Apache Flex. The runtime supports installable applications on Windows, OS X and mobile operating systems like Android, [iOS](#) and BlackBerry Tablet OS. It also originally ran on Linux, but support was discontinued as of version 2.6 in 2011[6].

Adobe AIR is a runtime environment that allows Adobe Flash content and ActionScript 3.0 code to construct applications and video games that run outside a web browser, and behave as a native application on supported platforms. An application developed for Flash Player or HTML5 and deployed in a browser does not require installation, while AIR applications requires installation from an installer file (Windows and OS X) or the appropriate App Store (iOS and Android). AIR applications have unrestricted access to local storage and file systems, while browser-based applications only have access to individual files selected by users [7].

Adobe AIR framework can be called as multi-screen, multi-operating system runtime, can be used by developer with good web development skills for building a rich internet Application(RIA) for desktop as well as mobile devices. Adobe AIR uses HTML, JavaScript and Ajax for application building. You can make use of any of the technology among Flash, Flex, HTML, JavaScript, and Ajax as per your experience for example, application can be build using single or combination of Flash / Flex / ActionScript or HTML / JavaScript / CSS / Ajax. User interaction to Adobe AIR is exactly same as that with native applications, application runtime is installed at one time and it can be used as native application. Extension of web-based applications to the desktop is possible without need to learn traditional desktop development technologies or the complexity of native code [8].

Adobe AIR can build application for Windows and Mac OS desktops as well as iPhone, iPad, Kindle Fire, Nook Tablet, and other Android™ devices [9].

IV. RHODES

Rhodes 3.3.3 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 [14].

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 [16]. 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), Screen Rotation, etc. [15].

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 connection between these elements [15]. 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 [17]. Rhodes provides mainly three possibilities to add extensibility 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. 1 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 [15].

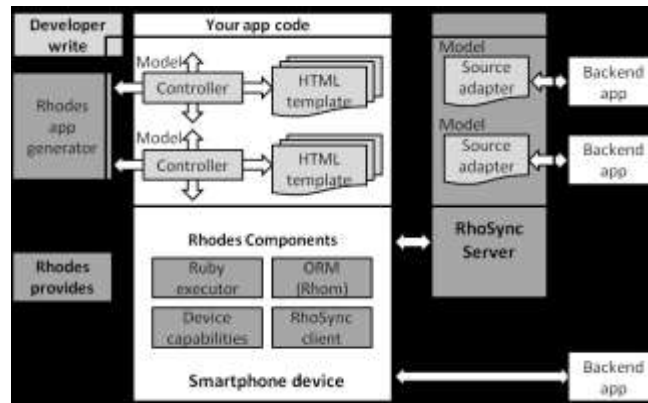


Fig. 1. Rhodes interfacing architecture between the Smartphone and Rhodes components [16].

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. Applications developed with Rhodes are compiled into Java byte code to be executed on BlackBerry OS, or compiled into Ruby 1.9 byte code to be executed on all other OS's [14].

V. PHONEGAP

PhoneGap 1.9.0 is an open-source mobile development tool developed by Adobe System Inc. under Apache 2.0 license. PhoneGap allows developers and companies to build free, commercial and open-source applications, and give them also the possibility to use any licenses combination [18]. The development environment is cross-platform and permits the creation of applications for Android, Bada, BlackBerry, iOS, Symbian, web OS and Windows Phone OS's.

PhoneGap is a useful solution for building mobile applications using modern web programming languages, such as HTML, HTML5, CSS, CSS3 and JavaScript, and the functionality of SDK are instead to use less-known languages such as Objective-C or other languages [9]. It has the benefit to bring many advantages to skilled developers and specially to attract web developers [21].

Essentially, PhoneGap is a "wrapper" that allows developers to enclose applications written in known programming languages into native applications. Moreover, as each valid open-source software it is composed by many components and extensions. PhoneGap applications are hybrid, which means that they are not purely native or web-based. The meaning of "not purely native" comes from the layout rendering that is done via web-view instead of the native language of the OS, whereas "not purely web-based" comes from the lack support of HTML in some functions [14]. Besides, PhoneGap also offers the possibility to extend the tool by developing own plug-ins.

Adopting a cross-platform approach the applications building and maintenance can be enhanced because developers have to write a single source code for any mobile OS supported by the tool. PhoneGap does not provide an IDE to develop applications, but developers have to write the source code with an IDE and port the it on other IDE's (e.g. Eclipse for Android, XCode for iOS, etc.). This approach does not allow developers to have a centralized development environment, so the effort required to compile the source code and produce the executable application (final product) is high. Thanks to the use of different IDE's for the development, PhoneGap can be performed on different PC OS's such as Mac, Linux and Microsoft Windows. Unfortunately, sometimes there are some exceptions because not all IDE's are compatible with all PC OS's.

PhoneGap architecture: The PhoneGap's architecture is composed mainly of 3 layers: Web Application, PhoneGap, and OS and native API's.

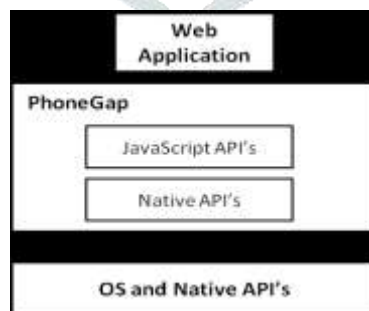


Fig. 2. Interfacing layers of the PhoneGap architecture [15].

In Fig. 2 the top layer represents the application source code. The central layer is composed by JavaScript and native API's. Mainly, this layer is responsible for the interfacing between web application and PhoneGap layers. Furthermore, it also takes care of the interfacing between JavaScript API's that are used by the application with native API's that are used by mobile OS's. The functionality of this layer is to maintain the relationship between JavaScript API's and native API's of each mobile OS. PhoneGap provides JavaScript API's to developers that allow the access to advanced device functionality, such as Accelerometer, Barcode, Bluetooth, Calendar, Camera, Compass, Connection, Contacts, File, GPS, Menu, NFC, etc. [18].

In Fig. 3 is shown a more detailed architecture schema provided by IBM. It represents all components about the web application, HTML rendering engine, PhoneGap API's and OS layers. Moreover, some different interfaces are shown in detail, such as the interfacing between PhoneGap API's and native API's layers.

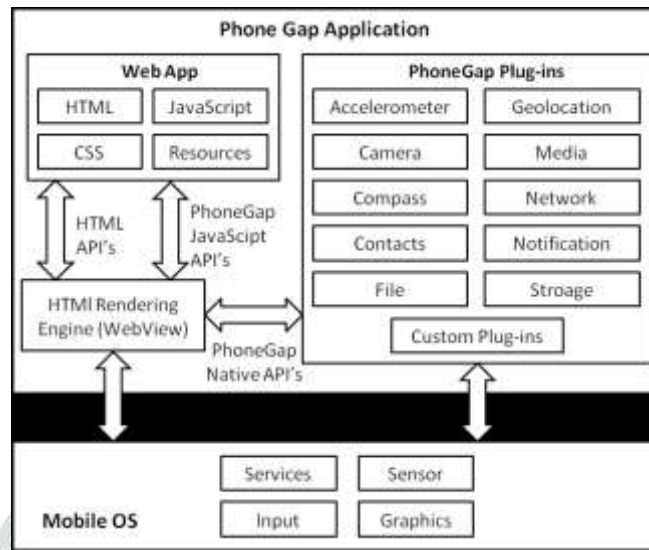


Fig. 3. Complete schema of PhoneGap architecture and interfacing among components [13].

VI. DRAGANRAD

DragonRad 5.0 is a cross-platform mobile application development platform by Seregon Solutions Inc. and distributed

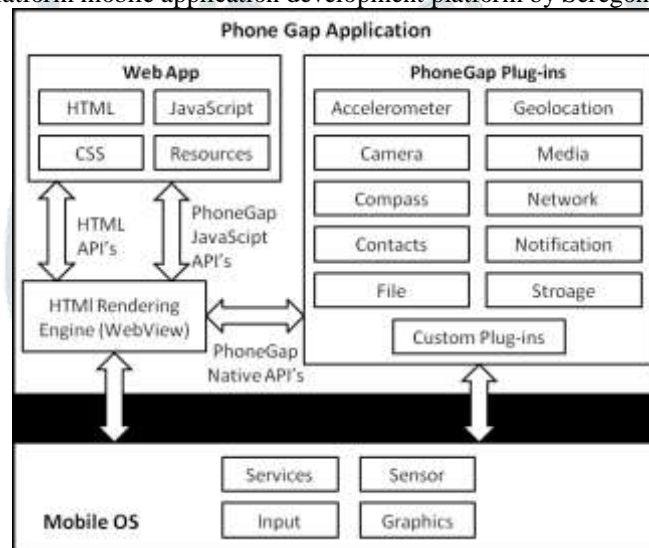


Fig. 3. Complete schema of PhoneGap architecture and interfacing among Components [13].

under a commercial license. It allows developers to design, manage and deploy mobile applications once and use it across iOS, Android, BlackBerry and Windows Mobile [10]. The tool focuses on database driven mobile enterprise applications with easy and wide range of databases support. It provides the D&D environment which help developers to save programming time and to create logics. DragonRad provides their own built IDE, that can be configured for different simulators like iOS, Android, BlackBerry, Windows Mobile etc. As DragonRad has host-client architecture, it is required to setup server and database based on the needs of developers but it also comes in complete package with all prerequisites of server and database like Tomcat, MySQL etc. DragonRad is commercial tool with the support to its own language D&D, the possibilities of extension in terms of adding plugins and other support to the framework are quite limited. DragonRad facilitates the integration and synchronization of database system with native functions of above defined mobile OS's, such as Contacts, Calendar, Geolocation, Menu and Storage. The Architecture of DragonRad mainly composed of three major components [12]:

A. DragonRad Designer

It is a D&D visual environment or GUI for developers to design, develop and install mobile applications. Features of D&D are not only helping developers to design applications, ,but also reduce the efforts for maintenance and coding [12].

B. DragonRad Host

DragonRad host component could be run on either Linux or windows server which fill the gap between database of enterprise and mobile applications. It helps to maintain the communication with mobile device, which also includes query of transaction during

network unavailability. It also plays the role to establish problem free connection with database access and updates with synchronization. The following list summarizes the most relevant features of DragonRad host [12]:

- Taking data query from the device;
- Executing data query on specific target;
- Sending data back to device based on request;
- Handling data updates from the devices and updating databases; and
- Compression and data checking of data packets.

C. DragonRad Client

This component behaves like a native application on device which helps to run and interpret code of the created application by designer. DragonRad has the emulator to run and debug the application. This component also has the feature to customize application like change icon, application name, DragonRad project and link for installation of DragonRad host. By changing the link to DragonRad host, the application automatically connects to the given host when it started. It would also help in updating the project when required. One advanced function provided by the DragonRad company is to compile the application online [10].

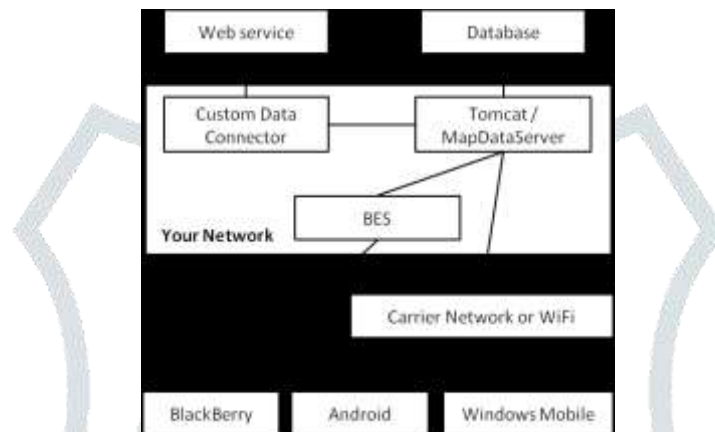


Fig. 4. DragonRad architecture with the connectivity of different components in it [8].

In Fig. 4, the area containing your network provides all backend support such as a custom data connector that could fit for any web service available, while Tomcat/ MapDataServer is to support for the database. This full network is connected to mobile phone with different OS's with the help of the Wi- Fi. The other sub-part BlackBerry Enterprise Server (BES) is specifically for BlackBerry products, such as PlayBook. This tool allows creating application in very easy manner with having only three step, which are connect, build and deploy[12].

In DragonRad, it needs one connector that could be used to connect data source to DragonRad host web application. Therefore, it provides custom data connector to meet features. To handle the data transmission between back-end and device this connector is useful. As shown in Fig. 5 custom data connector is responsible to make connection of DragonRad host web application with database. DragonRad designer and device runtime are for respective work. There are few other features of this custom data connector [12].

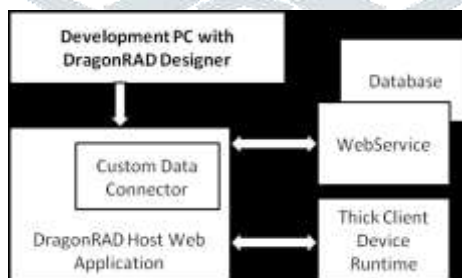


Fig. 5. DragonRad custom data connectors architecture [12].

- Responsible for transferring data request and updates form device to database with keeping synchronization;
- Receiving data queries from device;
- Processing received data queries;
- Re-query data or resend stored data based on the query;
- Receiving notification form device; and
- Sending data to device.

VII. MoSYNC

MoSync 4.0 is an open source solution developed by a Swedish company targeted to mobile market. MoSync has fully fledged SDK which helps developers to build and package all type of mobile applications, such as simple, advanced and complex application that share the same code base. MoSync SDK is proving to be very powerful tool with many components tightly coupled together like Libraries, Runtimes, Device Profile Database and Compilers, and so on. It provides the full fledge Eclipse-based IDE and the use of standard C/C++. It also added the support with web-based language like HTML, HTML5, CSS and JavaScript. It provides well documented API's both in C/C++ and web-based. The idea involved to support multiple mobile OS's is different from other tools and also in very isolated way from other mobile operating code. Applications in MoSync are built to target a device profile by using GNU Compiler Collection (GCC) and pipetool. After writing the application, pipe-tool is used to compile the resources present in the application. Then GCC backend is called and path to target device profile passed to it. GCC uses it to produce MoSync intermediate language, which then fed in to pipe-tool. Then, pipe-tool behaves as the bridge between MoSync applications to target device profile.

The profile database helps the application in ensuring that it

has adapted correctly to the device. Runtimes are libraries

which are bound to provide support related to all like regarding graphics, audio, communications, input, uniform interface to low level system API's and other device features [1]. MoSync is completely open source and based on the Eclipse for IDE, so it provides to add extensibility in the same way as Eclipse does, i.e plugins or adding external library.

MoSync Architecture

MoSync has mainly two architectures specific to each C/C++ and web based languages. The idea and most of the components are same except few. Architecture mainly consists of eight components as shown in Fig. 6.

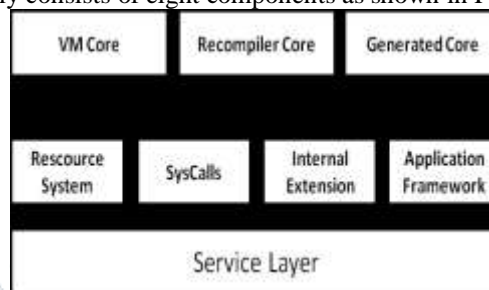


Fig. 6. MoSync Runtime Architecture, Showing the layered structured concept use in it [11].

- 1) Service layer: supports many functions like file I/O, threading, networking, memory management and many more functions [11].
- 2) Application framework: is responsible for runtime entry point. It is primarily to manage platform specific events like event management, initialization and destruction. The size and level of responsibilities varies across multiple platforms [11].
- 3) Syscalls: provides support to all features required by OS's, such as camera, contact, images, audio and networking etc which is specific to multiple platform. It is also responsible to interprets MoSync resource files and supports some like event management, initialization and destructions [11].
- 4) Resource System: manages resource objects such as images, sound and data blobs with the support of dynamic creation and destruction of resources [11].
- 5) Internal Extensions: specifies the design and configuration of each OS. Not all platforms contain the same type of features except few mandatory one, so missing features are implemented through single Syscall and known as numbered functions. When function is not found by Syscall it throws one error related to feature, which makes developers life little easy to determine a non-universal API is accessible in runtime [10].
- 6) Core: is responsible for the execution of MoSync programs by interoperating with Syscall and resource system, MoSync offers mainly three different types of core, which all share same common interface [11].
- 7) Virtual Machine Core: is the component that provides the support to load, interpret and run MoSync bytecode directly. The execution is taken care by single; small function that allows efficient Just in time (JIT) optimization. The whole structure is very similar to the core of Java Mobile Edition (Java ME) [11].
- 8) Recompiler Core: is the component that loads MoSync Byte code and recompiles it on the specific platform or typically Advanced RISC Machine (ARM) machine. After this recompilation, the generated code is executed. This core has many similarities with windows mobile and symbian [11].
- 9) Generated core: The core is responsible for the exhibiting interface with the generated native code. At this level it does not have any connection with the bytecode. The type of core is like iOS core. So the reason for having three different types of core in MoSync has versatile advantages. For example VM core is best for debugging and its dynamic approach makes it possible to load new code at runtime. This property proved to be very useful for many applications. Recompiler core is more efficient but less debugging support and its dynamic behavior also help in fast recompilation of some code. At last generated core have zero overhead for low end devices which are not able to load code at run time [11].

Conclusion

Thus, various tools and approaches used for cross platform development. The choice of approach and tool for application development totally depends on requirements of user. Here, we can co relate approaches and tools. From complete paper we find Xamarin can be used when hybrid application is to be developed. Xamarin application can be developed using any IDE as it supports special plug-in for porting application. Adobe Air application development requires just web programming knowledge thus both of these tools saves developer's efforts. Adobe Air builds its own web server for receiving and serving request, thus we can say that Adobe Air is uses a web approach.

Xamarin presents itself to be a good option for those who meet a few basic criteria. First, you should be targeting multiple platforms. If not, you may be better off going with a native solution. The next is that native user experience should be a priority. If native UI doesn't matter then it may be a case for a hybrid solution. When the previous criteria are met and you are also looking for great performance, Xamarin is probably the solution for you.

REFERENCES

- [1] Vandana S. Jagtap, Mamta S. Bhamare, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 10, October 2014*
- [2] Xamarin Mobile Development, Jared Dickson, *Grand Valley State University 2013*
- [3] B. Andrew. (2011, June 11). *NET Alternative In Transition* [Online]. Available at: <http://www.informationweek.com/architecture/net-alternative-in-transition/d/did/1098050?>
- [4] Xamarin. (2013, June 15). *Xamarin - Build cross-platform iOS, Android, Mac and Windows apps with C# and .NET* [Online]. Available at: <http://xamarin.com/>
- [5] Xamarin. (2013, June 15). *Components | Xamarin* [Online]. Available at: <http://components.xamarin.com/>
- [6] Adobe Air Wikipedia Available at: https://en.wikipedia.org/wiki/Adobe_AIR
- [7] "Adobe AIR: Browser vs. Desktop". Adobe Systems. Retrieved March 15, 2011.
- [8] Building ADOBE® AIR® Applications handbook by Adobe
- [9] <http://www.adobe.com/in/products/air.html>
- [10] C. Best. Seregon solutions announces support for blackberry playbook. <http://www.prweb.com/releases/DragonRAD/PlayBook/prweb4960674.htm>, January 2011.
- [11] P. Broman. The runtime architecture. <http://www.mosync.com/documentation/manualpages/runtimearchitecture>, June 2010.
- [12] DragonRad. <http://dragonrad.com>, 2012.
- [13] B. Curtis. PhoneGap and the Enterprise. <http://www.slideshare.net/drback/phonegap-day-ibm-phonegap-andthe-enterprise>, July 2011.
- [14] V. G. Sarah Allen and L. Lundrigan. *ProSmartphone Cross- Platform Development*. Apress, 2010.
- [15] MokaByte. Applicazioni mobili negli scenari Enterprise. <http://www2.mokabyte.it/cms/article.run?articleId=O5R-R6L-HN8-8R8-7f000001-18359738-2ff5bd55>, March 2010.
- [16] Leckylao. Rhodes framework: Agile mobile web development. <http://leckylao.com/2010/06/12/rhodes-framework-agile-mobileweb-development>, June 2010.
- [17] RhoMobile. What is so special about the rhodes smartphone app framework? <http://rhomobile.com/blog/whats-so-special-about-therhodes-smartphone-app-framework>, March 2010.
- [18] PhoneGap. <http://phonegap.com>, 2012.
- [19] Motorola Solutions. <http://www.motorola.com/Business/USEN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite>, 2012.
- [20] J. Feroso. Seeks to Bridge the Gap Between Mobile App Platforms. <http://gigaom.com/2009/04/05/phonegap-seeks-to-bridge-the-gap-between-mobile-app-platforms>, April 2009.
- [21] T. Myer. Beginning PhoneGap. Wrox, November 2011.