



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Cross Platform Mobile Application Development Tools: An Analysis

Wellington S. Manjoro^{#1}, John Mangirazi^{#2}, Trevor Makausi^{#3}, Wellington Makondo^{#4}, Chiedza Hwata^{#5}

^{#1} Lecturer, Software Engineering Department, Harare Institute of Technology 277 BE Belvedere Harare, Zimbabwe

^{#2} Lecturer, Information Security and Assurance Department, Harare Institute of Technology 277 BE Belvedere Harare, Zimbabwe

^{#3} Lecturer, Computer Science Department, Harare Institute of Technology

277 BE Belvedere Harare, Zimbabwe

^{#4} Lecturer, Software Engineering Department, Harare Institute of Technology

277 BE Belvedere Harare, Zimbabwe

^{#5} Lecturer, Information Technology Department, Harare Institute of Technology

277 BE Belvedere Harare, Zimbabwe

Abstract-Mobile application development of mobile applications is a fast growing as well as challenging field because of many diverse platforms and their software development kits. In a bid to ease the cost of software development and reach out to many users across several platforms, developers are preferring to use cross platform application development tools. This paper, looks at several write-once-run-anywhere tools like HTML 5, Codename One, Xamarian and Apache Cordova. It looks at their requirements, strengths and weaknesses. A comparative analysis is done on the available cross platform mobile application tools on the market together with a discussion on the general architecture for cross platform application development. This paper also looks at the possible future of mobile application development.

Keywords-Android, Xamarian, Codename One, HTML 5, Cordova, Phone gap, Mobile development, cross platforms

I. INTRODUCTION

The phrase “cross platform tools” refers to development tools that allow the development of applications for multiple platforms from a single codebase. Some of the advantages of using cross platform tools include reduction of entry and exit barriers, reduction of fragmentation, provides easy access to new platforms and device screens. Cross platform tools also allow efficient use of developer resources. This paper provides a description of the types of cross platform that exist and examine cross platform development tools frameworks against

three native Software Development Kits (SDKs) namely Android, IOS and Windows.

II. TYPES OF CROSS PLATFORM TOOLS

A. Web based Frameworks

This framework relies on HTML for the user interface technology and can be combined with pure web development frameworks. For example, JQuery has support for mobile application called JQuery Mobile which provides support for mobile application development. JQuery provides user interface controls and components.

B. Compiled or Interpreted Code Frameworks in this category do not use web technologies but they incorporate a different user interface Application Programming Interface (API) and use single programming language which is then used across different supported platforms.

III. GENERAL ARCHITECTURE

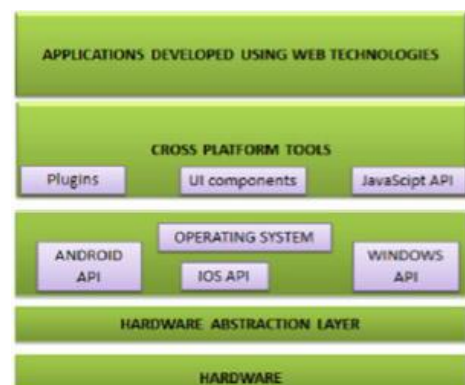


Fig 1. General Cross Platform Architecture

The application developer implements the functionality and business logic using web technologies. The cross platform framework provides a facility to implement user interface and to access device storage facility and other features such as sensors, camera, contacts e.t.c which interacts with a JavaScript API. The API will in turn interact with the native API of a mobile platform.

The application is then built separately to generate the executable files for different platforms. The APIs for the mobile platforms allows for generation of the respective application. The generated application can then be run in corresponding mobile device.

IV. CROSS PLATFORM TOOLS EVALUATION CRITERIA

The following outlines some of the major aspects considered for the evaluation of each cross platform tool.

A. Licence

This criteria examine whether the framework in question is distributed as a free software or open source. The licence of publishing. If a developer is free to create a commercial licence.

B. Platforms supported

Considers which platform is supported by the tool.

C. Access to specific devices features Includes access to device hardware specific features like camera, GPS, contacts and notifications. Compared according to application programming interfaces and web services

D. Look and feel

This criteria focuses on the support of native components and native-like styling capabilities

E. User Interface Design

This criteria looks at the development of the GUI. The availability of a UI editor.

F. Ease of development

Quality of documentation and the learning curve

G. Distributions and Deployments

Evaluate how easy it is to distribute to the consumers. Use of application stores. Availability of updates

CROSS PLATFORM TOOLS

This section consists of an analysis of the various cross platform tools on the market against three native SDKs namely Android, IOS and Windows.

A. Appcelerator Titanium

Titanium [8] is a framework by Appcelerator Inc. Titanium supporting Android, IOS, Blackberry, Windows phone, Tizen and HTML5 with the latest version 3.0 at the time of survey. It uses JavaScript, HTML and CSS for application development. The architecture comprises of two main parts: an application is combined with a standalone JavaScript interpreter to execute the application code, and the library that supports APIs for device functionalities such as file system access, sensors, native user interface, Titanium components and others, abstracting the differences between the platforms. Titanium library can be used from anywhere in the source code by an application. Calls to methods or properties of the global object are passed to the native implementation for the respective platform. This bridging facility also allows own native extensions (platform specific native code) to be added and used in a project from JavaScript code.

B. Apache Cordova

Apache Cordova [9] is a framework that does not provide any user interface generation functionality, but only the capability for displaying web technology based content. Initially Apache Cordova was called Phone Gap. Phone Gap was initially developed by the company Nitobi which was acquired by Adobe Systems in 2011. As of the evaluated version 4.0, Cordova supports the following list of mobile platforms: Android, iOS, BlackBerry, Web OS, Windows Phone 7, Window Phone 8, Firefox OS, Tizen, Amazon Fire OS and Bada. It enables development of mobile applications through the use of web technologies by providing web view interface component and tools that create platform-specific project files and also initial source code that shows the web view. In addition, Cordova provides APIs (Application Programming Interfaces) for device and platform functionality through a JavaScript-to-native bridge and the native code can also call back JavaScript functions inside the web view. The JavaScript-to-native bridge implementation is done differently on each platform. Developers can extend Cordova with native plugins by implementing a simple interface which also

differs across the platforms. Features that are not natively supported on a platform are mimicked by Cordova's own implementation.

C. MoSync

MoSync [2,3] is an open source tool that is used to develop all types of applications ranging from simple to complex sharing the same codebase. Application development is done using C++ and Java. MoSync applications are native applications, where a service layer supports a number of functions such as file I/O (Input and Output), memory management, networking, threading and other functions. The supported platforms are IOS, Windows Phone, Android, J2ME, Blackberry and Symbian.

D. RhoMobile Rhodes

RhoMobile Rhodes [10] is an open source framework for cross-platform Smartphone applications, developed by RhoMobile. It aims to manage enterprise application and data, and to provide a high level productivity and web programming portability. The RhoMobile framework is composed of a number of components Rhodes (Develop), RhoConnect (Integrate), RhoHub (Deploy) and RhoGallery (Manage). It can be used across Linux, Macintosh and Windows. RhoMobile Current application data remains on users' device by a standalone server (RhoSync) which is provided by RhoMobile in addition to a hosted development environment (RhoHub). The application can be written using HTML and Ruby. The supported platforms are Android, BlackBerry, IOS, Windows Phone and Symbian.

E. Xamarian

Xamarian [4,5] allows developing cross-platform applications written in C#. It produces a native application. It provides an Integrated Development Environment (IDE) called MonoDevelop IDE and Xamarian plug-in for Visual Studio. On each platform the application user interface uses native controls, taking advantage of native user interface toolkits. With Xamarian context-sensitive template, developers can edit and restructure their code. The supported platforms are iOS, Android, and Windows phone

F. Codename One

Codename One [6] is a cross device platform allowing one to write code once in Java and it works

on most devices more especially iOS, Android, Blackberry, Windows Phone, & J2ME. The biggest goals for the Codename one are ease of use/RAD (Rapid Application Development), deep integration with the native platform & native speed. Codename One extended the Lightweight User Interface Toolkit (LWUIT) platform abstraction by adding a simulator and a set of cloud-based servers that build the actual native applications from the Java byte code.

G. PhoneGap

PhoneGap [11, 12] is a cross platform mobile application development framework that supports development across android, iOS, webOS, Windows Phone, Symbian, Blackberry and Bada. It is an open source framework developed by Nitobi Software under the MIT licence. Phone Gap mainly uses HTML5, CSS3 and Javascript for developing the applications.

H. Dragon Rad

Dragon Rad [13] is mobile application development framework that supports android, window and blackberry platforms. It was developed in D programming language and offers its own development IDE. It is relatively easier to use due to the use of its drag and drop Graphic User Interface (GUI) environment and mainly focuses on mobile applications that are database driven and supports databases such as SQL Server, Oracle and MySQL.

I. Sencha Touch

Sencha Touch [14] is an open source mobile application development platform that supports Android, iOS, BlackBerry, kindle, and Bada. Sencha Touch makes use of HTML5, CSS3 and Javascript and requires a webserver to run applications locally.

J. jQuery Mobile

jQuery Mobile [15] is an open source mobile application development framework that supports Android, iOS, BlackBerry, Windows Phone, Bada, webOS, Symbian and MeeGo. The applications built using this framework are based on HTML5 user interfaces thereby making it relatively easy to use. Native applications cannot be built using this framework.

K. Unity 3D

Unity 3D [16] is a 3Dimension (3D) cross platform game engine used to develop and model 3D

mobile applications usually games. Unity makes it easier for developers by providing drag and drop features for some code to reduce the amount of code needed to be written by the developer. The code used to build the application can be JavaScript, c# or Boo. Unity 3D provides developers with access to a number of motion sensors for the mobile devices.

L. Corona SDK

Corona SDK [17] is a cross platform mobile application development framework that supports Android, iOS, Kindle, NOOK and Fire. Corona uses web views that are compliant with HTML5 as well as OpenGL functionalities for the graphics, hence making it relatively easy to use. Corona provides a free trial version to develop applications for an unlimited duration but the developer will need to pay a certain subscription amount in order to be licenced to publish the application in an app store.

VI. DISCUSSION AND STATISTICS

Most tools allow developers willing to compromise user experience and put more priority to launching off the application in several platforms to reach to maximum users. Most of the tools have a favourable licences, developed using technologies like HTML, CSS and JavaScript. These technologies promote rapid application developments as a majority of developer are familiar with them. Thus the cost of developing and time to market the application is reduced

Fig 2 and 3 shows the statistics as of July 2014 and March 2017 respectively, according to statistics [7] the total number of applications in the android market (Google Play) was at 2.8 million from 1.3 million in July 2014. Apple remains the second biggest with 2.2 million apps in the market (Apple App Store) from 1.2 in July 2014. Windows is the third largest with approximately 669 thousand applications in the market (Windows Store) from 300 thousand applications in 2014. Amazon comes fourth with 600 thousand applications in its Amazon Appstore as at March 2017. Blackberry comes fifth with approximately 234 500 applications as at March 2017 in its Blackberry world from 130 000 in June 2014.



Fig2. Application Store statistics as of July 2014



Fig3. Application Store statistics as of March 2017

VII. TABLE I. COMPARATIVE ANALYSIS

Tool	Licence	Platform	Look and Feel	User interface design	Ease of Development	Distribution and Deployment
Titanium	Open Source	Android, IOS, Windows, Blackberry	Titanium interpret JavaScript to create for creating the native UI elements	Cumbersome and Time consuming to produce the UI. No WYIWYG editor	Good documentation. Full example of code available. Accustomisation to framework is relatively slow.	Titanium app can be distributed via different app stores without difficulty
Apache Cordova	Open Source	Android, IOS, Bada, Windows, Firefox OS, Blackberry	Does not use native UI elements. It uses CSS and relegated the process of UI development difficult	As a web app. UI element may be developed using tools like Dreamweaver	Good documentation. Full example of code available.	Cordova apps can be distributed via different app stores without difficulty
Mosync	Open source	Android, IOS, Windows, Blackberry	Use native UI elements in C++. The process of UI development is slow not developer prefer to use HTML/CSS instead	Native UI library available.	Good documentation. Full example of code available.	MoSync apps can be distributed via different app stores without difficulty
Rhodes	Open source and Subscription	Android, IOS, Windows, Blackberry	Use native UI elements is restricted. Rhodes developers prefer to use HTML/CSS instead	As a web app. UI element may be developed using tools like Adobe Dreamweaver	Good documentation. Full example of code available. Apps can be developed using Rhodes studio and plugin to support Visual studio	Rhodes apps can be distributed via different app stores without difficulty. Enterprise specific deployment can be via RhoMobile Gallery
Xamarian	Commercial	Android, IOS, Windows, Blackberry	Use native UI elements	Native UI library available. GUI builder is available	Good documentation. Full example of code available. Apps can be developed using Xamarian Studio and Visual studio	Xamarian apps can be distributed via different app stores without difficulty.
Codename One	Open sources	Android, IOS, Windows, Blackberry	Use native UI elements	Native UI library available. GUI builder is available	Fairly good documentation. Full example of code available. Apps can be developed using Netbeans IDE and Eclipse	Codename One apps can be distributed via different app stores without difficulty.
Phone Gap	Open source	Android, iOS, webOS, Bada, BlackBerry, Symbian, Windows	Use native IDE e.g. Eclipse for Android, XCode for iOS	Can be Web app or native applications	Relatively easy due to use of HTML and CSS	Applications can be compiled in the cloud
Dragon Rad	Commercial	Android, Windows, Phone, BlackBerry	Uses its own IDE	Native applications	Drag and Drop environments helps developers	Can be distributed easily
Sencha Touch	Open Source	Android, iOS, Bada, BlackBerry, Windows, Phone	Use HTML, CSS and JavaScript	Can be native or web based applications	Relatively easy to use but needs a web server to run apps locally	Distributed easily but developer pays to publish the app in an app store
jQuery Mobile	Commercial	Android, iOS, webOS, Bada, BlackBerry, Symbian, Meego, Windows, Phone	HTML based user interface	Does not create native applications	Works on a write-less do more ideology using jQuery UI.	Distribution is not a challenge
Unity 3D	Commercial	Android, iOS, Windows, Phone	Developers can drag codes from one object to another	Creates native apps	Provides access to different motion sensors for the mobile device.	Can be distributed easily
Corona SDK	Commercial	Android, iOS, Kindle, Fire, NOOK.	Use HTML5 and can add other features like Facebook using Corona APIs	Can create native apps	Integration of OpenGL enables easy graphics handling	Easily distributed upon payment of a fee

VIII. CONCLUSION

This paper performs a critical analysis of the mobile application development tools focusing on cross platform development tools since they build apps for several platforms and the development cost and time to market are less. Although the user experience is not as good as native applications, but the apps can be released in several platforms at once to reach out to most of the potential users. The paper discussed the general architecture for cross platform application development. Also a detail survey is presented that covers several aspects of the tools allowing developers to gain insight about the tools.

REFERENCES

- [1] <http://www.cordova.apache.org>
- [2] Singh I, Palmieri M., *Comparison Of Cross-Platform Mobile Development TOOLS*. Mälardalen University. 2011
- [3] <http://www.mosync.com>
- [4] https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_1_-_understanding_the_xamarin_mobile_platform
- [5] "Xamarin - Create iOS, Android and Mac apps in C#", <http://xamarin.com>.
- [6] <https://www.codenameone.com>
- [7] <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores>
- [8] <https://www.appcelerator.com/Titanium>
- [9] <https://cordova.apache.org>
- [10] <http://rms.rhobile.com>
- [11] <http://phonegap.com>
- [12] Singh I, Palmieri M., "Comparison of cross-platform mobile development tools", Mälardalen University, 2011.
- [13] <http://dragonrad.com>.
- [14] <https://www.sencha.com>
- [15] <https://jquerymobile.com>
- [16] <https://unity3d.com>
- [17] <https://coronalabs.com>

