IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 5 (May. 2019), ||S (VI) || PP 01-05

Review Paper on Intelligent Way to Develop Application to Target All Platforms.

Shubhojeet Chatterjee[1], Laxmi Dodke[2], Pawan Thakurpelne[3], Shekhar Salam[4]
[1](ETC, Jhulelal Insitute of Technology / RTMNU, India)
[2](ETC, Jhulelal Insitute of Technology / RTMNU, India)
[3](ETC, Jhulelal Insitute of Technology / RTMNU, India)
[4](ETC, Jhulelal Insitute of Technology / RTMNU, India)

Abstract: *The recent varieties of mobile platforms makes it extremely difficult for developers to find the most appropriate platform or target multiple ones. A review of key concepts on Android, iPhone and Qt points out the bridge that will help to better connect multiple platform. Study the use of Ministro to create support for cross platform development to remove the dependencies of a developer regarding platform.*
Keywords: *Qt, Dalvic virtual Machiene, Ministro, Java*

I. Introduction

The last couple of years have seen smartphone sales take off. While this hype mainly started out with the introduction of the iPhone in 2007, many more mobile platforms have been introduced since then. At the same time, with the advent of more and more platforms, competition is intensifying. Due to thisfragmentation, targeting an application for as many markets, devices and/oroperators has become very difficult, As cross-platform development toolsare rare yet, developers have to crucially select a platforms for their native application. Thus a common approach followed by the developers is developing for a single platform first, and later porting to various other. This practice is highly liable to Market share, ease of development, popularity, an active developer community, usability and target group and many other factors. Developers on the other hand would prefer to implement an application once, and use it for various other platform with minimal effort. This is where comes the use of cross platform like Qt.

II. Methodology

The comparison basically focuses on three platforms: the first one is which started the hype (IOS) the second one is gaining more and more momentum and becomingincreasingly popular (Android) third which dominates in the PC department (Windows).

2.1. Platform Introduction

Android:It is an open-source software stack created for mobile phones and otherdevices that includes an OS, middleware and key applications [1]. It is based on

Linux kernel version 2.6 to provide important system services such as security, memorymanagement, process management, network stack, and driver model. The Dalvikvirtual machine is an excellent example of the the approach by the developers, it was optimized for minimal memory footprint, executes Android applications written in Java that were first converted into DalvikExecutable format. From a developer's perspective, it is the Android SDK thatprovides the tools and APIs required for developing applications.

iOS: It is the mobile OS of Apple's iPhone series. It is based on the sametechnologies used by Mac OS X, namely the Mach kernel and BSD interfaces,thus making it a UNIX-based system [2]. IOS applications are developedusing Objective-C, which is a superset of ANSI C that has been extended withcertain syntactical and semantic features to supportobject-oriented programming.

2.2. Platform Comparison

Concepts like memory management, user interface realization or communication, etc are provided by all platforms in one way or the other. This section will cover these basic concepts and find differences between the platforms.

2.3. Diff erences:

1. Granularity level

- Intents are used to pass information between broadcast receiver, activities or services. They operate on a whole screen service.
- Notifications and Signals can be used to pass information between all kind of objects, even if it is fine-granular to notify somebody when a single GUI element changes, or coarse-granular

2. Inter-process Communication (IPC)

- Intents can be used for inter-application communication, as well as intra-app communication.
- Notifications are limited to intra application communication on iOS.

2.4. Commonalities:

1. All three platforms have a asynchronous communication system in place where events can be bound to operations. The internal implementation is substantially diff erent though.
2. The mechanisms is used to register for any notifications that report system changes to one's application.

Intents/Notifications/Signals can all be utilized to trigger lifecycle state changes.

2.5. OS/Platform-speci □ c

Memory Management is essential for all computer systems, but due to the memory constrained nature of smartphones it is important on mobile platforms. Thus it is indispensable to delocate objects that are no longer needed which could either be taken care ofby the system, or be an important necessity to be carried out by the developer.

2.5.1. Android

Android applications depend on automatic memory management controlled by Dalvik's garbage collector (GC), whichGoogle states, can sometimescause performance issues if you are not careful with memory allocations [1].Each process uses a different GC instance thus, the collectors does not interfere with instances of other applications. The type used by Dalvik is known as tracing GC and utilizes a mark-and-sweep approach [4].

2.5.2. iPhone

Objective-C supports two environments for memory management: automatic garbage collection and reference counting [2]. Garbage collection was introduced in Objective-C 2.0, an automaticmemory management mechanism isunavailable on iOS (i.e., it is onlysupported by Mac OS X and above). It would pass responsibility for determining the lifetime of objects to an automatic collector.

Thus such differences causes incompatibility of same application type on different OS.

This arises the need for use of a cross platformtechnology through which developers are more focused on business logic rather than application compatibility.

III. Cross platform development

Cross-platform mobile development has long been a viable alternative to fully native engineering. Following the classic native approach, you produce two different applications: one written in Java for Android and the other in Swift or Objective-C for iOS. This path is pretty time-consuming and expensive because the same code must be written twice. On the other hand, it ensures the highest code consistency with the original app look and feel.

Cross-platform development, in contrast, allows engineers to write the code once and apply some parts of it across all platforms. Usually, this happens at the cost of performance and application behavior. Cross-platform engineering communities strive to mitigate these disadvantages by continuously introducing new approaches and tools.

3.1. Perks of cross-platform development

1. Greater Market Potential

With cross-platform technology, you can reach a maximum number of your target audience by being available on more platforms. The availability of your application on competing platforms like iOS and Android benefits your business with greater market potential.

2. Faster Development

Developing an application in different platforms separately may be a lengthy process. Developers can leverage one codebase in cross-platform and customize on multiple platforms rather than creating a new codebase for each platform. Hence, developing apps on cross-platform is faster and also helps to release your app quickly.

3. Cost-effective Solution

Cross-platform is a cost-effective solution as it requires single development skillset. The maintenance charges and updates will sync across all platforms and devices, which saves your time and money.

4. Lower Technical Barriers

Working on cross-platform technology tools like ActionScript & Flex or HTML & JavaScript is easier for developing an application. It helps a developer in lowering the technical barriers and allows them to focus in creating the application.

5. Uniform Look and Feel

When an application is running on a single codebase, the uniformity in look and feel of the app can be maintained easily. This kind of uniform app may portray your brand image.

Though cross-platform assists your business with various advantages, some parts of this technology have to be improved. It has some limitations with user experience, platform integration, and flexibility. Hence, it is necessary to evaluate all the possibilities and absurd when deciding whether to choose native or cross-platform. The ideal choice you make depends on your business goals.

3.2 QT

Qt is a free and open-source widget toolkit for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed.[3]. This product can solve the main problems faced by developers that is removal of dependencies of a developer regarding platform.

For use of Qt application in in OS like android and iOS an application called ministro needs to be downloaded this application basically facilitates the use of QT apps.

3.3. MINISTRO

Ministro is a central repository for Qt shared libraries. It acts as a bridge between apps and Qtlibraries.The connection API protocol used by applications to connect and query the service is released under BSD license, it mean that any application can connect and query its services.

Minitro can also be used to reduce the size of application. It enables several applications to share the libraries. To use this deployment method, one must set up a repository for the libraries that one want to distribute

IV. Market Evaluation

4.1. Mobile

In mobile OS market Android makes up for the majority of shares, the second most dominant player in this market is iOS. Development in these sector is generally for these two platforms.

As seen above the gap between these two platform can be reduced by use of cross platform technology. This can reduce the might and capital that is wasted on developing the same product for different platform. Use of such technology will also promote use different OS like Samsung and Microsoft which hold minority stakes due minimum app support.

The below mentioned pie chart shows the mobile OS market share according to stats counter.

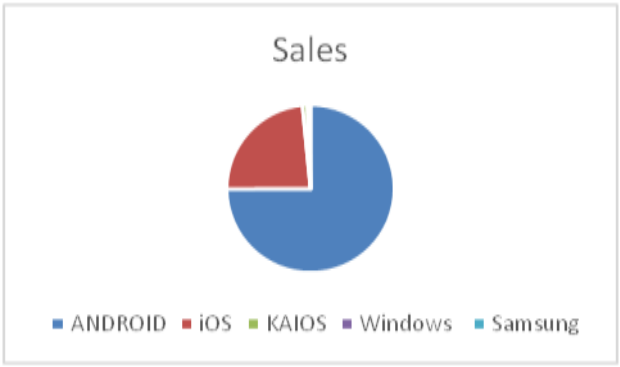

Figure 1: Mobile operating system market share worldwide

4.2. PC

Similarly just like mobile OS market PC OS have their fair share of competition the leader in this competition is windows and then comes OSX with second highest market share. Thus using cross platform technology we can reduce the gap between mobile computing and desktop application as a crossplatform implemented application supports multiple OS.

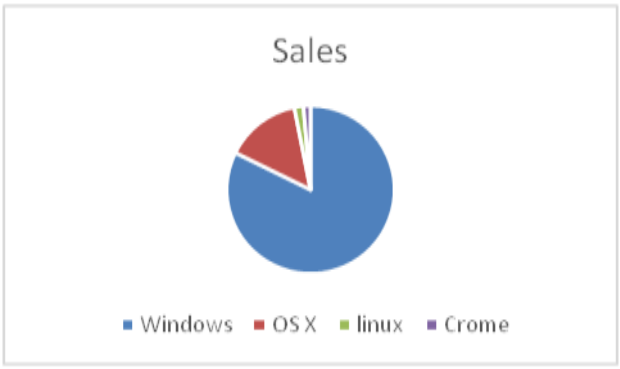

Table2: Desktop Operating System Market Share Worldwide

Market for app development

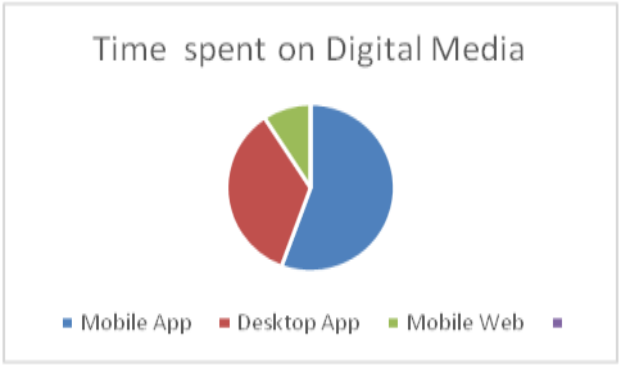

Table 3: Market potential for development

According to info-graphic of`dot com infoway' an average user spends 2.5 hr. /day on mobile apps .almost six million apps are present in play store itself and quite a similar stats for iOS as well. Thus this market being the latest boom in technology has a lot of untapped potential. But this market suffers un-organization and misdirection ie. Same application needs to be developed in different platform. This leads to market being cumbersome, daunting and costly affairwhere as the same work-energy and capital can be spent on further R and D or business logic development. This problems needs to be sorted as according to IDC the market till 2020 will see revenue growth of 47 billion dollar. To counter this problem use of cross platform technology is necessary.

V. Conclusion

Finally after concluding all review which were related to topic of Cross Platform Application Development, the concepts related to operating systems and their differences. So concluding, we can build an integrated development environment with cross platform technologies like Qt which will help us in achieving our aim. Basically in this IDE the developer will be given a choice to develop his application either in Java or Objective C. After he designs the GUI and works on the coding part and after that debugs and ensures his application is complete he would be given a choice to deploy his application in multiple OS. Basically the IDE will perform all the code conversions, semantically changes and logical changes required to deploy the application to a different platform. This IDE will also feature a sophisticated benchmarking or analyzing tool which will help the developer to debug his program, check for memory leaks and do other stuff related to the testing part. This IDE will also be quite platform independent that is, there would be variants for Linux based systems as well as Macintosh based systems too.

References

[1]. Android Developers Guide,http://developer.android.com
[2]. iOSDev Center, http://developer.apple.com/devcenter/ios
[3]. Qt Developer Network, http://developer.qt.nokia.com/
[4]. Dalvik Android Source Code, http://android.git.kernel.org/?p=platform/ dalvik.git;a=blob_plain;f=vm/alloc/MarkSweep.c