# Comparison of Cross-Platform/Native Mobile development tools

M.Yunus R.Tonse
Tonse Technologies
Manipla,India – 576114.

Ashika S. Kalmady
NMAMIT, Nitte
Karkala, India –574110.

Monisha Kunder
NMAMIT, Nitte
Karkala, India –574110.

*Abstract* — **New mobile applications are being released everyday as per the increasing needs of the user. When the majority of target audiences are using different platforms, the choice for platform becomes difficult. Catering to huge audience in the ranks of iPhone, Android, Windows Phone, BlackBerry or others, there's a need to design for multiple platforms. As different mobile operating systems use different programming languages and development tools, the developer has to code application for each of the platform adding to complexity, more of time and money. The much researched solution is with the help of cross-platform mobile applications wherein a unified code base defines the application and the nativity and UI/UX are brought to optimum.**

*Keywords:* Application, Cross-platform, Android, Native Mobile Development, Xcode, Phonegap, Eclipse, Visual Studio, AppStudio, Comparison, tools, Tonse Technologies, Mobileware Technologies.

## I. INTRODUCTION

Mobile phones are now widely used by lot of people all over the world. Mobile technologies are opening up new channels of communication between people and governments, potentially offering greater access to public information and basic services to all. Over the past two decades, we have witnessed significant technology advances in mobile devices. This evolution also means that solution providers must constantly update their technology to stay relevant in this extremely competitive market.

However, there are numbers of smart phones with heterogeneous platform in current mobile market, which are android, iOS, Symbian, Window Phone, BlackBerry and others. Competition among the diverse operating system is the main factor that triggers the developers to add in new features to the operating system and feasibility to develop [1][12]. The basic architecture of each OS is very different from each other's, forcing re-development to be a must. Native applications can access all API's made available by OS vendor. SDK's are platform-specific. Each mobile OS comes with its own unique tools and GUI toolkit. Designer is required to be familiar with different UI components of each OS.

Efforts have been made over the last years to address this problem, mainly with the emergence of mobile cross-platform tools or frameworks. Essentially, these tools consist in software that allow applications to be created and distributed to multiple platforms, reducing the incremental cost Per platform and maximizing the code reuse [1].

In this paper we briefly compare cross-platform and native mobile development tools. Cross-platform

tools considered in this paper are Phonegap and App Studio.

## II. MOBILE OPERATING SYSTEM

A. *Android:*
Android is software platform and operating system for mobile devices. Its current versions are based on Linux Kernel version 3.04. Android phone has an inbuilt 3rd party application store known as Google play store. It supports apps of extension .APK format. Application of Android was developed in java language using Android software development kit (SDK).

B. *IOS:*
IOS is Apple's mobile operating system. The latest mobile OS of Apple is IOS 7.1.2. This operating system is used in Apple's product such as iPhone, iPad and iPod Touch.

C. *Blackberry:*
Blackberry OS is Property Mobile Operating system, developed by Research in Motion. The operating system provides multitasking and supports special input devices. The track ball, track wheel and most recently the track pad and touch screen is one of its features. Blackberry is a close source OS and is not available for any other manufacturer.

D. *Windows phone*:

Windows mobile is a mobile OS developed by Microsoft. It is easily compatible with many Windows program such as Microsoft Office. Windows OS is programmed in C++ language. It is run on Pocket PCs, Smartphones and Portable media centers. [13]

## III. LIMITATION IN MOBILE OPERATING SYSTEMS

Mobile OS have many advantages as discussed in Section 2, beside all these it has certain limitations. The cost of building and maintain native app is usually higher because apps have to be developed and maintained separately for each mobile OS on which application is made available. The mobile apps have to be developed individually for each operating system using SDK or software Development kit specific to each platform. It is also time consuming to develop apps in each of different platforms for each OS.

## IV. ECLIPSE FOR ANDROID

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications in Java. The initial codebase originated from IBM. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.[3]

The Eclipse platform is primarily designed for building integrated development environments. It is a highly extensible platform, rather than a custom tool for a specific set of tasks. The Eclipse platform defines the mechanisms and the rules, and allows tools to be built on the top of them by providing a set of well-defined APIs.

Plug-ins is the smallest unit of the Eclipse platform. They are structured bundles of code that contribute a set of functionality to the platform.
To develop Android app in eclipse we need to:

- Download the Android SDK.
- Install ADT plug-ins for Eclipse.
- Download latest SDK tools and platforms using SDK Manager.[15]

Android-specific platform APIs and application packaging tools are required in order to develop android applications using Eclipse. ADT is a set of plug-ins for Android application development on the Eclipse platform. The Android SDK is a comprehensive set of development tools, including android platform's Java libraries, an application packager, a debugger, an emulator, and extensive documentation. In order to do anything useful with ADT, the Android SDK needs to be installed on the machine.[2]

## V. XCODE FOR IOS

IOS is an advanced mobile operating system, continually redefining what people can do with a mobile device. Together, the iOS SDK and Xcode IDE make it easy for developers to create revolutionary mobile apps.

The Xcode IDE is at the center of the Apple development experience. Tightly integrated with the Cocoa and Cocoa Touch frameworks, Xcode is an incredibly productive environment for building amazing apps for Mac, iPhone, and iPad. Because everything is so well integrated, workflows feel natural. As you compose a new interface, the Assistant editor intuitively presents the related source code in a split window pane. Simply drag the mouse to connect UI controls to the implementation code. Apple LLVM compiler technologies parse your code, keeping every symbol you see in the LLDB debugger consistent with the editor and compiler. As you type, that same engine is constantly at work, finding mistakes and offering Fix-it's for your code.[4]

Xcode even communicates with the Apple developer website, so you can enable services such as Game Center or Passbook in your app with a single click. In conjunction with OS X Server, Xcode can setup a remote bot to continuously build, analyze, test, and even package your app. When your app is ready, Xcode will bundle and submit your app to the App Store.

Xcode and the iOS SDK are available together as a free download on the Mac App Store. With tools like the Xcode IDE, Instruments, and iOS Simulator, you can have your first app running within minutes. And when you're ready to test on devices and distribute your apps, get the resources you need by joining the iOS Developer Program.[6]

## VI. VISUAL STUDIO FOR WINDOWS

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GU applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle(like the Team Foundation Server client: Team Explorer).[16]

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Its rich collection of tools for targeting Windows platforms equips you to create immersive applications that get the most of every Windows device using existing skills in languages such as C#, VB, C++, and JavaScript.

## VII. PHONEGAP

**Phonegap** is a mobile development framework which enables software programmers to build applications for mobile devices using JavaScript, HTML5, and CSS3, instead of device-specific languages such as Objective-C. Supported platforms are Android, iOS, Windows phone, Windows 8, BlackBerry 10, Firefox OS, Ubuntu, and Amazon Fire OS. The core of Phonegap applications uses HTML5 and CSS3 for their rendering and JavaScript for their logic. Although HTML5 now provides access to underlying hardware such as the accelerometer, camera and GPS, browser support for HTML5-based device access is not consistent across mobile browsers, particularly older versions of Android. To overcome these limitations, the Phonegap framework embeds HTML5 code inside a native Web View on the device, using

a foreign function interface to access the native resources of the device.

Phonegap provides an application programming interface (API) that enables you to access native operating system functionality using JavaScript. You build your application logic using JavaScript, and the Phonegap API handles communication with the native operating system.[20]

Phonegap applications are developed using HTML, CSS, and JavaScript, however the final product of a Phonegap application is a binary application archive that can be distributed through standard application ecosystems. For iOS applications the output is an IPA file) (iOS Application Archive), for Android applications the output is an APK file) (Android Package), for Window Phone the output is a XAP file (Application Package), etc.[20]

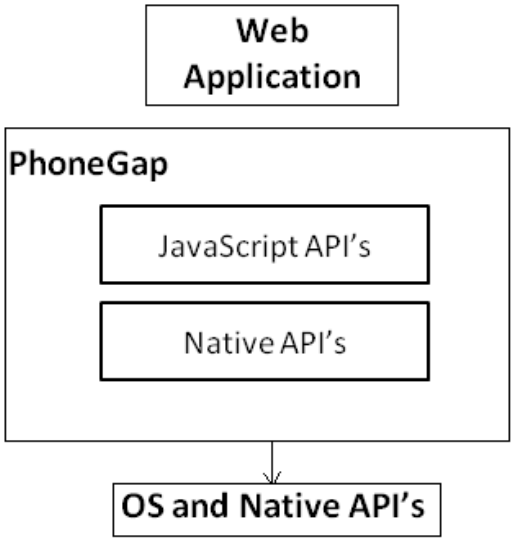
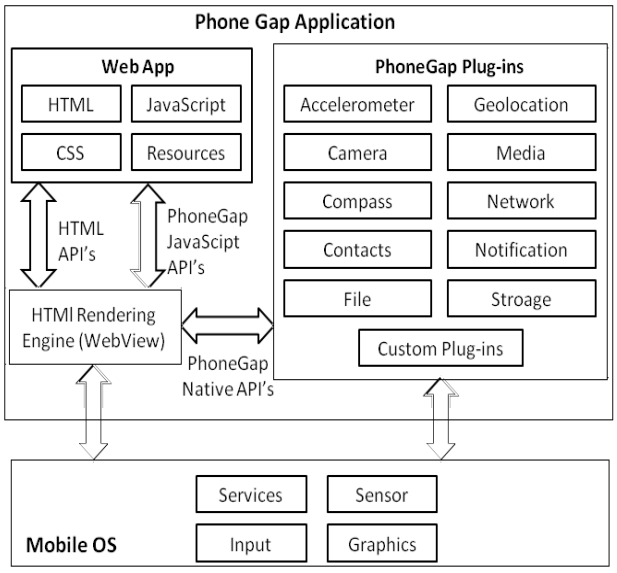



FIGURE 2: INTERFACING LAYERS OF THE PHONEGAP ARCHITECTURE

The Phonegap architecture is composed mainly of three layers: Web Application, Phonegap, and OS and native API's. In Figure 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. [17]




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.[17]

## VII. APP STUDIO

A proprietary cloud based mobile app generation platform developed to enable enterprise customers to create mobile apps across multiple platforms. App Studio allows non-technical users with basic technology understanding to develop mobile applications using App Studio as effectively as native app development tools.

App Studio has a cloud based web portal to which users can securely connect using their unique accounts and start creating.

The only infrastructure - hardware you need to get started on App Studio are:

- A web browser
- A mobile device

Architecture of AppStudio is as shown in figure 4 below,

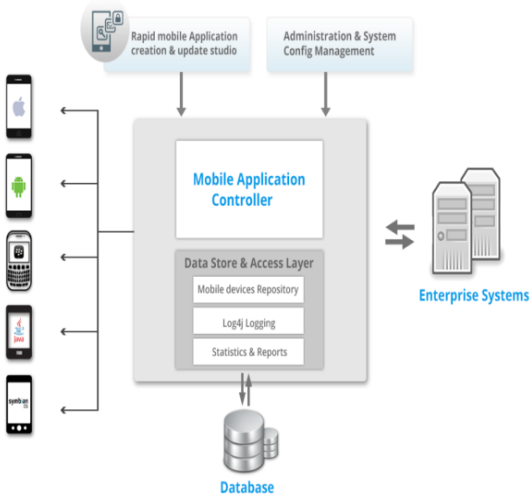



FIGURE 4: ARCHITECTURE OF APP STUDIO

Supported operating system is J2ME, Android, and Blackberry.

Differentiator:

- Development costs/ app is at a fraction of native development costs/ app
- Reduced development team size
- Non-mobility resources with basic technology understanding can develop apps.

Key Features:

- It is a Cloud based Rapid Mobile Thick Client Development system.
- Once built can be distributed across all platforms.
- Easy to customization the apps without any extra add on costs.
- Lightweight Middleware to integrate thick Client Applications with Enterprise Systems.

Cross platform development simplifies application development. Developers are free to code using any familiar web technology or a development tool that generates java script as well as native SDK's for android, iOS, J2ME and Blackberry. Easy compatible integration with SAP, ERP systems, databases, web services, SOAP/WSDL/XML.

## VIII. COMPARISON OF TOOLS

This section presents the comparison made among the tools.

| Tool Name | Mobile OS support | OS Support |
|---|---|---|
| Eclipse for Android | Android | Linux, Windows, Apple Mac OS X |
| Xcode for iOS | iOS | Mac OS |
| Visual Studio for Windows | Windows | Windows |
| App Studio | IOS, Windows 8 | |
| Phonegap | Android, BlackBerry, IOS, Symbian, Webs, Windows Phone | Linux, Windows, Mac |

TABLE 1. COMPARISON ON MOBILE PLATFORMS COMPATIBILITY AND DEVELOPMENT ENVIRONMENTS OS'S SUPPORT

From Table 1, it is easy to understand that PhoneGap Offers more compatibility for development on different mobile OS's, which is optimal to gain the maximum profit for both developers and business model owners [18]

| Name | Eclipse for Android | Xcode for iOS | Visual Studio for Windows Phone | App Studio | Phonegap |
|---|---|---|---|---|---|
| Language | Java | Objective-C | C# | | HTML5, CSS3, JS |
| Native APIs | Yes | Yes | Yes | Yes | Yes |
| IDE | Eclipse | XCode | Visual Studio | IDE native of the mobile OS | IDE native of mobile OS |
| Output | Native | Native | Native | Web | Web |

TABLE 2. COMPARISON ON DEVELOPMENT FEATURES

Table 2 describes several characteristics and development features, such as supported programming languages, Accessibility to native API's, IDE.[18]

| | | | | | |
|---|---|---|---|---|---|
| | | | | | Phonegap |
| | | | | App Studio | |
| | | | Visual Studio for Windows Phone | | |
| | | | Xcode for iOS | | |
| | | Eclipse for Android | | | |
| Name | | | | | |
| License | Eclipse Public License | | | | MIT |
| Open Source | yes | | | yes | yes |
| Architecture | local | local | local | | Local, Web |
| MVC | | | yes | | No |

Table 3. COMPARISON OF GENERAL FEATURES

| Javascript APIS | Eclipse | Xcode | Visual Studio | App Studio | Phonegap |
|---|---|---|---|---|---|
| Accelerometer | | | | | ✓ |
| Barcode | | | | | ✓ |
| Calendar | | | ✓ | | ✓ |
| Camera | | | ✓ | | ✓ |
| Compass | | | | | ✓ |
| Contacts | | | | | ✓ |
| File System | | ✓ | ✓ | | ✓ |
| GPS | | | ✓ | | ✓ |
| Menu | | | | | |
| Media | | | ✓ | | ✓ |
| NFC | | | | | ✓ |
| Network | | | ✓ | | ✓ |
| Notification | | ✓ | ✓ | | ✓ |
| Screen | | | ✓ | | ✓ |
| Rotation | | | | | |
| Storage | | | ✓ | | ✓ |

TABLE 4. COMPARISON ON MAIN SUPPORTED API'S

## REFERENCES


[1] André Ribeiro, Alberto Rodrigues da Silva "Survey on Cross-Platforms and Languages for Mobile Apps", 2012 Eighth International Conference on the Quality of Information and Communications Technology.

[2] "Android Apps with Eclipse", www.it-ebooks.info.

[3] "Eclipse (software)", http://en.wikipedia.org/wiki/Eclipse_(software)

[4] "Xcode", http://en.wikipedia.org/wiki/Xcode

[5] https://developer.apple.com/xcode/ide/

[6] "Develop apps for iOS-The world's most advanced mobile operating system" , https://developer.apple.com/technologies/ios/

[7] "Application Development - Use existing skills to create applications across multiple platforms" , http://www.visualstudio.com/en-us/explore/application-development-vs.aspx

[8] Avinash Shrivas,Anandkumar Pardeshi ,International Journal of Emerging Technology and Advanced Engineering, 9, September 2013," To Study and Design a Cross-Platform Mobile Application for Student Information System using PhoneGap Framework"

[9] API guide for android, https://developer.android.com/guide/index.html

[10] A. Charland and B. LeRoux, "Mobile Application Development: Web vs. Native", Commun. ACM, vol. 54, no. 5, pp. 49-53, New York, May 2011

[11] Windows API reference, http://msdn.microsoft.com/en-in/library/windows/apps/br211377.aspx

[12] G. Hartmann, G. Stead and A. DeGani, "Cross-platform mobile development", Mobile Learning Environment, Cambridge, March 2011

[13]Mobile-OS-Overview, www.shoutmeloud.com/top-mobile-os-overview.html

[14]Native Vs HTML Vs Mobile Apps by Asahitechnologies, May 13, 2014.

[15]"Building Your First App**",** https://developer.android.com/training/basics/firstapp/index.html?hl=p
[16] Microsoft Visual studio, http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
[17] "To Study and Design a Cross-Platform Mobile Application for Student Information System using Phonegap Framework", www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 9, September 2013).
[18] Manuel Palmieri, Inderjeet Sing h, Antonio Cicchetti , "Comparison of Cross-Platform Mobile Development Tools", 2012 16th International Conference on Intelligence in Next Generation Networks.
[19] Phonegap, http://en.wikipedia.org/wiki/PhoneGap
[20]"PhoneGap Explained Visually" http://phonegap.com/2012/05/02/phonegap-explained-visually, May 2, 2012.