Abstract-- In academic institutions, corporations, and
government entities Smartphone’s laptops are joined as
genuine network endpoints. IT must be conscious of the
differences, as well as similarities, in the supervision of
security for the mobile devices between laptop and
Smartphone platforms. With these dissimilarities in mind,
best practices can be applied by IT to guarantee the
confidentiality and security of corporate communications
from both inside and outside of the network boundary. To
handle the security issues in smart phones is a major problem
these days. Different smartphone like Apple’s iPhone,
Google’s Android and Palm Pre using WebOS as graphical
user interface. Porting different applications to different
smart phones incurs elevated overhead because of its special
programming representation and environment. The progress
of it can be smoothen by generating an XML- on the basis
demonstration of byte code we entitle XMLVM which cross
compiles an Android app to the iPhone and the Palm Pre, and
it considerably decrease the Porting endeavor. The foundation
of cross compiler is Java byte code directives that are
interpreted to high-level programming languages which aim
Smartphone supports. The API between smart phones are
mapped using compatibility libraries.

Index terms- AR, API, AOSP, Cocoa Touch, Android, Palm
Pre, XMLVM.

I. INTRODUCTION

As the Adoption of the smart phones is increasing day by
day, these devices have become more prone to attackers
who try to infect them with Malicious Software (Malware).
These are very much convenient than a laptop, and
suggesting all the functions, smart phones have develop
into most accepted in current times. Typically smart phones
consists of an highly developed operating system also
called as the OS. As a result price of the gadget increases,
the OS is used to contribute the almost PC-like experience
of the gadget. The main purpose of a adapted OS on a
device is to formulate the operation of the cell phone quick
to respond and straightforward to utilize even for first time
buyers.

Evidently, to maintain an Operating System, the
processor cooperate, and phones that are high-end
processors are capable to support an OS without
obstructing the usability of the appliance as recurrent
crashes, deficient power to entire tasks, etc. Consequently,
all smart phones move towards their own Operating System
which are loaded on the device. [3].

The most common mobile OS used by modern smart
phones include Apple's iOS, Google's Android, Palm
Pre, Nokia's Symbian, RIM's BlackBerry OS Such
operating systems can be installed on many different phone
models, and typically each device can receive multiple
Operating System software updates over its lifetime [4].

This paper is organized as follows: Section 2 gives an
introduction to Android, the iPhone, and the Palm Pre. We
present versions for all three devices to highlight the
differences in their programming models. Section 3
presents our byte code level cross-compilation that can
cross-compile Android applications to the iPhone and the
Palm Pre. In Section 4, we discuss advantages of API’s in
consolidations. Finally, Section 5 provides a conclusion and
an outlook to future work.

II. OVERVIEW

Here we will compare Android-based HTC G1, the
iPhone 3GS, and the Palm Pre’s working environment. All
three smart phones are relatively similar with respect to
their hard-ware capabilities; they differ greatly in their
native application development models.

A. Android

Android is a platform designed for calculating some
smart phones and other gadgets. This expertise, the owner
of which is Google, Inc., consist of an OS, software, and
apps. The OS is foundation of Linux, that makes available
advanced computer processing. This skill is sustained and
constantly developed by the Android Open Source Project
(AOSP).
The purpose of the AOSP is to generate a unbeaten real-world creation that progresses the mobile skill for end users. AOSP also retains the Android Compatibility Program, classifying an "Android compatible" device that can be compiled on any appliance written by third-party developers via the Android SDK to avert incompatible Android completion. The well-suited program is also discretionary and with no charge, with the Compatibility Test Suite furthermore open-source and free.[4]

a Technology

Google introduced the HTC Dream as the initial marketed cellular phone to utilize Android technology in 2008. Since that moment, the utilization of this platform has extended to various smart phones. Java software applications are foundation of Android technology. This expertise involves the utilization of a unique software development kit (SDK) to generate apps for an Android device. The SDK is complimentary available for download via Internet. For this motive, and because it will exertion on multiple operating systems, many software developers favor Android technology over that used in other smart phones. Android expertise endow with detailed application programming interface (API) modules to developers that acquires benefit from it. Android technology is openly available to everyone who wants to enlarge applications as it sponsors the user to add new innovations by using the programming code available by them. The flexibility of Android technology makes it more convenient to the OS to have this as a base for smart phones. The only requirement is software development kit availability to bring any change in it.

There are two general editions of Android, one is cupcake in which sliding physical keyboard is there in the phone and another is HTC EVO which is thoroughly activated by touch screen. Customers can utilize it by physically touch the screen and execute various functions. Android avoids multitasking still yet interface is very good and comprehensible.

b. Version History

Android has been bring up to dated a number of times since its novel release, to fix bugs and adding new features with every release. The names of versions are in alphabetical order, following a dessert.

Recent releases

- **2.3 Gingerbread** has incidented the user interface, developed the flexible keyboard and copy and paste characteristics, improved gaming appearance, added VoIP calls, and supplementary support for Near Field Communication (NFC).
- **3.0 Honeycomb** was a tablet-oriented release, supports larger screen devices and introduces many new user interface features, support for multi-core processors, hardware acceleration for graphics and full system encryption. The first device featuring this version was the Motorola Xoom tablet, went on sale in February 2011.

1. **3.1 Honeycomb**, which was released in May 2011, extra support for extra input devices, USB host mode for transferring information directly from cameras and other devices, and the Google Movies and Books apps.
2. **3.2 Honeycomb**, announced in July 2011, further added optimization for a broader range of screen sizes, new "zoom-to-fill" screen compatibility mode, loading media files directly from SD card, and an extended screen hold API. Huawei MediaPad is the first 7 inch tablet to use this version.
3. **4.0 Ice Cream Sandwich**, was announced on October 19, 2011, passed Honeycomb features to smart phones and added new supplementary features that includes facial recognition unlock, network data usage monitoring and control, unified social networking contacts, photography enhancements, offline email searching, app folders, and information sharing using NFC. Android 4.0.3 Ice Cream Sandwich is the latest Android version that is available to phones. On November 14, 2011 Android 4.0.1 source code was make public.

Design
Android consists of a kernel that is based on the Linux kernel, with middleware, libraries, and APIs written in language C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android also uses the Dalvik virtual machine with just-in-time compilation to run Dalvik dex-code (Dalvik Executable), which is frequently translated from Java byte code. The ARM architecture is main hardware platform for Android. The support for x86 is from the Android project and Google TV uses a special x86 version of Android.

B. iPhone

A sequence of smart phones which are designed and marketed by Apple Inc. are iPhone. The first iPhone was revealed by Steve Jobs. An operating system known as iOS (previously iPhone OS) is run on iPhone. iPhone is an amendment of the Darwin OS core that is instigate in Mac OS X. It also comprises the "Core Animation" software component from Mac OS X v10.5 Leopard. Moreover the PowerVR hardware (and on the iPhone 3GS, OpenGL ES 2.0), which is accountable for the interface's motion graphics. The operating system takes less than half a gigabyte. The capability to sustain collection and prospect applications from Apple, and from third-party developers are included. Software applications cannot be copied directly from Mac OS X but must be written and compiled specifically for iOS.[2]

a. Cocoa Touch

Cocoa Touch is a UI framework for building software programs to run on the iPhone, iPod Touch, and iPad from Apple Inc. Cocoa Touch provides an abstraction layer of iOS, the operating system for the iPhone, iPod Touch, and iPad. Cocoa Touch is based on the Mac OS X Cocoa API toolset and, like it, is primarily written in the Objective-C language. Cocoa Touch allows the use of hardware and features that are not found in Mac OS X computers and are thus unique to the iOS range of devices. Just like Cocoa, Cocoa Touch follows a Model-View-Controller (MVC) software architecture.

Tools for developing applications based on Cocoa Touch are included in the iOS SDK [6].

b. Version History

- **iPhone** having CPU core 620 MHz (under clocked to 412 MHz) ARM 1176JZ(F)
- **iPhone 3G** having CPU core 620 MHz (under clocked to 412 MHz) ARM 1176JZ(F)
- **iPhone 3GS** having CPU core 833 MHz (under clocked to 600 MHz)
- **iPhone 4** having CPU core 1 GHz (under clocked to 800 MHz)
- **iPhone 4S** having CPU core 1 GHz (under clocked to 800 MHz)

C. Palm Pre

A multimedia Smart Phone framed and promoted by Palm with a multi-touch screen and a sliding keyboard is named as Palm Pre. This was the first Smartphone to use Palm's Linux based mobile operating system, i.e webOS. The Palm Pre uses JavaScript, HTML and CSS for the development of native applications. Palm debuted the Pre at the 2009 Consumer Electronics Show, which was the first mobile phone that used the Texas Instruments OMAP 3430 processor, and also its primary US exclusive carrier agreement with Sprint, that operates a CDMA network.

a. Web OS

The first Palm device to use webOS was Pre, a Linux-based platform that have replaced Palm's earlier Palm OS. The interface of webOS is based on a system of cards which are used to manage multitasking. Multi-touch gestures are supported by webOS, which enables nearly all navigational input to be made using the touch screen. The virtual keyboard is not included in Pre, just a portrait-oriented slide-out keyboard. An onscreen, virtual keyboard is embedded in the code and it can be made available via a third party patch. There are hundreds of third party patches that allow users to customize the usability and interface of the Pre.

b. Version History

- **Pre Plus** The Smartphone’s new edition, the Palm Pre Plus, was proclaimed at CES 2010. The variation included elimination of the center button, 16 GB storage memory (8 GB on the original Pre), RAM expanded from 256 MB to 512MB
- **Pre 2** The official announcement, The III generation of Palm Pre, the Palm Pre 2 by HP, that uses HP webOS 2.0. This device Included 1 GHz processor and 512MB RAM.
• HP Pre The Palm Pre's inventive CPU is a 600 MHz Texas Instruments OMAP 3430 (ARM Cortex A8 + PowerVR SGX) under clocked to 500 MHz

D. Augmented Reality

Most smart phones are now built with the technological necessities for AR—counting a camera, accelerometer, compass and GPS. The expertise begin to explain up in apps in 2009, and currently hundreds of them use AR.[?] Viewdle's Social Camera make use of AR for social networking as it recognizes people in photos by comparing their images to tagged photos of friends in Facebook. Google Goggles uses image recognition to provide information about real-life objects including books, artwork, wine and menus, which can be translated into a language you can read.

III. FRAMEWORK FOR CROSS COMPIlATION

Developers targeting smart phones ideally want their applications to be available on as many platforms as possible to increase the potential dissemination. Given the differences in the way applications are written for smart phones, this incurs significant effort in porting the same application to various platforms. In this paper, we introduce a cross-compilation approach, whereby an Android application can be cross-compiled to both the iPhone and the Palm Pre. The solution we propose not only cross-compiles on a language level, but also maps API between the different platforms. Although Android, the iPhone, and the Palm Pre differ in their user interface style guidelines, we believe that games are an ideal candidate for cross-compilation. Games typically take over the complete screen and use few special purpose widgets. Since all platforms support almost identical animation and graphic capabilities, games can readily be cross-compiled. The benefit of our approach is that only skill set for the Android platform is required and only one code base needs to be maintained for all three devices.[1]

As shown in the previous section, different smart phones differ greatly in their programming environment. Not only do they offer different APIs but they also require different programming languages. In this section we introduce XMLVM, a flexible, byte code level cross-compiler, that allows translating an Android application to the iPhone and Palm Pre. The main benefit is that only one code base needs to be maintained.

A. XMLVM

The foundation of our byte code consolidation framework is an XML-based programming language. Since the semantics of this language is modeled after the JVM and the CLR, we call it XMLVM. XMLVM basically allows representation of the contents of a Java class file as well as the contents of a CLR executable through XML. Another way to look at XMLVM is that it defines an assembly language for those virtual machines using XML as the syntax [3].

As we know that an Android is a canonical platform. The knowledge of Android system is required by developer, and then an Android application is cross compiled to other smart phones. There are various factors for selecting Android are that, we have a huge skill set for the Java programming language and various powerful tools are available to expand Java. A lot of improvement can be seen in JavaScript and Objective-C.

There are various advantages of Android design. The device’s capabilities can be explored by Android’s API to provide the application the chance to acclimatize accordingly. An Android developer is likely to deal with different device capabilities. This is not true for both the iPhone and the Palm Pre as they treat their devices as a homogeneous platform. Because Android applications are easily adaptable to different devices, it prepare them as ideal candidates to be cross-compiled to different platforms.

Another technical advantage of Android is its less of a lock-in which make it preferable. Android is maintained by a consortia and a reference implementation is available under a permissive Open Source license. The Android SDK is available for different platforms and the API is well documented.

Starting from a Java program we first use a regular Java compiler to generate a class file (1). The class file is used as input to our tool chain and as a first step we generate an XML representation of the contents of the class file (2). On the basis of the intermittent XML file we generate code in the target language, which in our case is either Objective-C or JavaScript (3). The following two sections explain in detail the cross-compilation process as well as the API mapping via a compatibility library.
B. Steps in Byte Code Level Cross-Compilation

A unique property of our tool chain is that we cross-compile from byte codes to high-level programming languages. We make use of the byte code instructions introduced by the Java Virtual Machine.[5] Using byte codes has several advantages. For one, byte codes are much easier to parse than Java source code. Several language features such as generics are already reduced to low-level byte code instructions. The Java compiler also does extensive optimizations to produce efficient byte codes.

1. The source code is first compiled to a Java class file via a regular Java compiler. The binary class file is then fed into our XMLVM tool. Internally, XMLVM generates the XML document. The reason our tool is called XMLVM is because the structure of the class file as well as the byte code instructions generated by the Java compiler are represented via appropriate XML tags.

2. When the XML representation of a byte code program is generated, it is possible to cross-compile the byte code instructions to arbitrary high-level languages, by simply mimicking the stack machine in the target language.

C. API Mapping

We need to make an Android application that uses the Android API, use the Cocoa Touch API instead. There are two basic ways to accomplish this.

1. Modify the application in such a way, that it uses the Cocoa Touch API instead of the Android API.

2. Achieve API mapping is via a compatibility library. Here the original Android application is unchanged, however, there is a compatibility layer that only uses Cocoa Touch API and offers the Android API.

D. Implementation Details

A prototype implementation based in the ideas described in this paper exists. The implementation of the Java to Objective-C/JavaScript cross compilation is almost complete owing to the simplicity of Java byte code instructions and the way they are mapped to high-level languages. Considering that both the Android and the Cocoa Touch API there are thousands of methods, around 5% of the API’s can be mapped by XMLVM. Though, at present the API that already support are allowed for complex applications. Other Open Source developers can also use XMLVM for the cross-compilation of Android applications to the iPhone that have been published on the Android Market as well as Apple’s App-Store [4].

IV. CONCLUSION AND OUTLOOK

The amalgamation of Smartphone makes them more popular and attractive these days. But different programming languages and different APIs lead to crucial overhead when porting applications to various smart phones. In this paper, we have demonstrated that a cross-compilation framework is feasible, thereby significantly reducing the porting effort. Only one code base needs to be maintained.

The byte code level cross-compilation introduced in this paper mimics the stack-based machine in the target language. This kind of code generation is not efficient and only suitable for applications that are not computationally intensive. We plan to introduce a stack- to register-based conversion within XMLVM to make the generated code more runtime efficient. The register-based machine will be described itself via appropriate XML-tags so that all code generating backends benefit from this optimization.

V. FUTURE WORK

In future, the work can be done to make the Smart phones more compact, strong and secure. Consolidation of Smart phones can be done to save them from threats by enhancing the Security features in it.
REFERENCES

AUTHOR’S PROFILE
Narender Madhan received M.Tech degree in CSE from Lingaya’s University, Faridabad in 2013 and B.E hon`s. degree Computer Science & Engineering from VTU, Belgaon, Karnataka in 2002. Presently, he is working as a Lecturer in Govt. Polytechnic for women, Faridabad. His area of interest are Wireless Sensor network and Compiler Design.

Garima Bhardwaj received M.Tech degree with distinction in CSE from Lingaya’s University, Faridabad in 2013 and B.Tech hon`s. degree Computer Science & Engineering from Uttar Pradesh Technical University in 2009. Presently, she is working as an Assistant Professor in Computer Science & Engineering department at Lord Krishna College Of Engineering, Ghaziabad. Her area of interest is Operating System and Database Management System.

Priyanka Sharma received M.Tech degree with distinction in CSE from Lingaya’s University, Faridabad in 2013 and B.Tech hon`s. degree Computer Science & Engineering from Uttar Pradesh Technical University in 2009. Presently, she is working as an Assistant Professor in Computer Science & Engineering department at Lord Krishna College Of Engineering, Ghaziabad. Her area of interest are Analysis and Design of Algorithms and Compiler Design.