



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

PLATFORM INDEPENDENT DESIGN CONCEPT FOR SMARTPHONE APPS DEVELOPMENT

Mendapara Vijay R.¹, Ghodasara Yogesh R.², Kamani Krunal C³

¹IT Consultant, iOS Softwares Solutions, Rajkot, Gujarat, India

²Associate Professor, College of AIT, Anand Agricultural University, Anand, Gujarat, India

³Assistant Professor, S M C College of Dairy Science, Anand Agricultural University, Anand, Gujarat, India

ABSTRACT: Currently many platforms such as iOS, Android, Symbian OS, BlackBerry OS, Windows Phone, MeeGo, HP webOS, or bada with different programming languages and paradigms each currently struggle for majority of the smartphone market. Not even looking at the vast amount of different devices, the total number of application stores indicates a growing diversity of today's market. When producing mobile applications, the key problem arises in the conflict between market perceptions on the one hand and cost of development on the other. If you want to address the majority of today's mobile users you could either develop your applications for each of these platforms independently, or try to follow a platform independent strategy in order to cut the finance. In this paper, we propose a platform-independent framework for smartphone application development. The core technology of our framework is based on building interpreter with common syntax for different platform. Because we are targeting on OS independent platform. To this end, we propose the framework architecture and evaluate the system.

Keywords: platform-independent; framework; architecture; smartphone.

I. INTRODUCTION

Platform independent means the execution of the program is not restricted by the type of OS environment provided. Smartphones and tablets are becoming universal and with them mobile applications. But what does the creation of different devices mean for developers entering the market? Do they target a single platform or build their applications twice for iOS and Android or other platform [8].

There were smartphones before the iPhone and there were mobile apps as well. Windows CE was first released in 1996, but it was Apple's iPhone (launched in 2007) that really signaled the start of the smartphone era. Originally, the iPhone didn't allow third-party native-code apps to be installed; developers wanting to add their own content had to do so using HTML web apps running in the phone's browser. There were some extensions to make such apps look like Apple's apps rather than web pages, but access to the hardware was limited. With 2008 came the App Store, and with it the opportunity to develop in native apps in Objective C. This wasn't without limitations: development required a Mac, Xcode, and only documented parts of the API were available. The resulting app then had to be vetted before being published. However, the iPhone's dominance at this stage meant that mobile developers really only had to target one platform, one device, and a very small set of OS versions.

Android founded in 2003, acquired by Google in 2005, and with the first handset launched in 2008, the Android platform has since seen an explosion of devices since the original G1. Android devices now collectively account for over 50% of worldwide smartphone sales (Gartner, Q1 2012), but, unlike the iPhone, there's huge variation in just about everything: screen size, resolution, processor speed, and memory, features, and OS version.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

Suddenly the idea of developing a single app that runs everywhere or nearly everywhere became a goal that was much harder to achieve – but one that remains as attractive as ever.

Many traditional software development cases, developers love platform-independent approaches because you write one piece of code, and deploy it on different places. That shoots down the development time and heavily cuts down the software maintenance costs. And for the same reason, there has been lots of buzz about platform-independent application development tools in current mobile development community. And yes, there has been a good progress, with such idea coming into the scene.

But still, why these emerging tools haven't been able to become the part of mainstream strategy in mobile app development? The reasons are many. This paper focused on writing-once-running-everywhere comes from the cost reduction in having only one code to write and maintain, and the time reduction being able to write one code and target multiple devices and platforms, making researching cross-platform mobile applications development worth one's effort and findings. In this regards we have proposed design of platform independent architecture for smartphone. Our design is building framework which will consist of interpreter on specific platform for the common syntax across platform. [1].

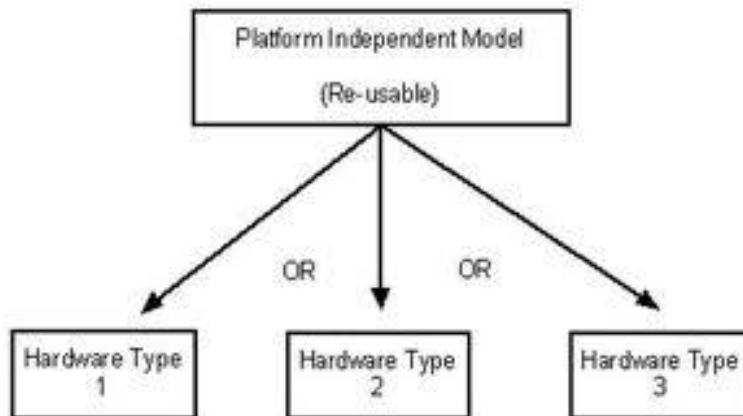


Figure 1 Platform Independent Architecture

II. INTRODUCTION TO SMART PHONES

A smartphone is a new breed of mobile computing device with advanced computing resources and equipped with technologies that facilitate access to the internet, run web applications, email sync, run and install applications.

A classic smartphone will have a touchscreen, access to the internet and an operating system to run apps on. A smartphone packs a whole raft of features such as a still and video camera, a calculator, an alarm, a watch, an mp3 player, and a gaming facility. Features that are driving to extinction specialized devices that used to run these features alone.

Multicore processing has now been the dominant pattern of processing architecture for quite a number of years on stationary computers, but in the first quarter of 2011, a number of smartphone manufactures brought to the market an extremely miniaturized smartphone that boast dual-core processor on board, a 1GB memory, a spacious storage capacity up

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

to 32GB, and the ability to play and capture HD Video, connecting to HD monitors, which could make these devices easily pass as a veritable replacement for stationary system units for the majority of consumers the majority assumption being text processing, browsing, gaming people.



Figure 2 Windows, Android and iPhone Smart Phone

III. RESEARCH CHALLENGES

Reaching large number of users in diverse and continually evolving Mobile OS (MOS) landscape. It is not sufficient to build an application or architecture that supports a single application. Difficulties of developing for so many platforms like, Time consuming, Requires detailed knowledge of each platform, Develop familiarity with different development environments and increased development costs [2].

Following are the research challenges in order to design platform architecture design for smart phones:

A. Mobile Device Portability

If an application is to hit it with end users on a mobile device, the mobile device has to be small, and convenient to handle not one which will be worn around the waist, on arms and head or carried in the bag. Small and convenient devices come with low power due to smaller battery size, constricted display and user interface necessitating alternative input mechanisms such as speech recognition, and smaller storage capacity which might render the device useless when it comes to business and corporate data intensive applications or multimedia content. Another key problem to consider is risk to data due to loss, theft or damage to mobile devices [3].

B. Mobility challenges

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

As mobile devices achieve full mobility, achieving smooth connectivity becomes a major problem. Mobile devices have to search for a connection spot and establish connection as they move dynamically an operation which could drain the battery fast and a stressful feature for traditional connection model. Address migration, location dependent configuration such as IP address, and service locality starts to get in the way of smooth operations of a connected mobile application.

C. Wireless communications

Greater bandwidth variation from high to low network bandwidth, susceptibility to security problems, signal corruption, signal blocking buildings, and natural topographies together bring about an extra overhead in wireless communications. Wireless networks range from Wi-Fi to GPRS to 3G hotspots. A robust mobile applications development process must take into account the above challenges when developing mobile applications [4, 6].

Building the interpreter facilitates the creation of platform-independent software, but also allows you to create software that is platform-specific. When you write a program in common syntax, platform independence is an option.

IV. PROPOSED PLATFORM INDEPENDENT ARCHITECTURE DESIGN

The core logic to develop platform architecture design is explained in figure 3.

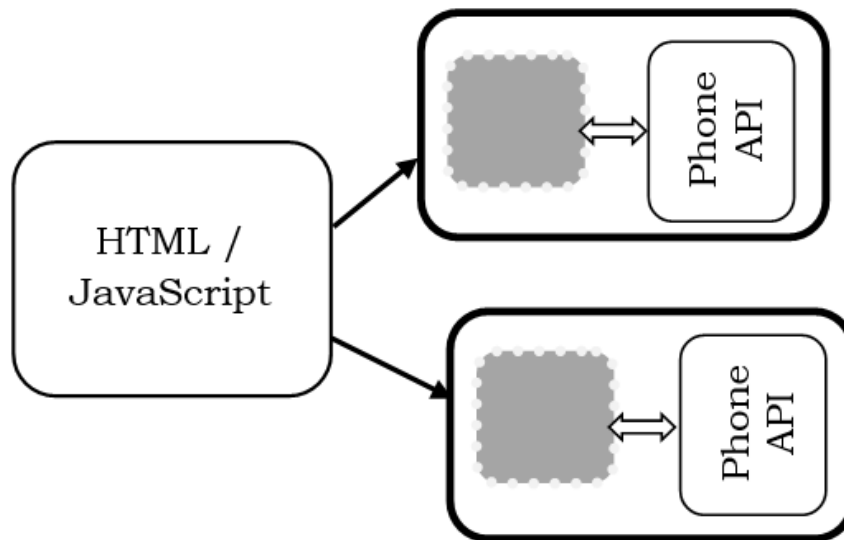


Figure 3 Nutshell of Proposed Architecture

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

The proposed framework is divided into two logical parts, which are server framework and the client framework. They are connected by a communication protocol called proposed Net, based on HTTP. Each of these parts consists of several modules that can be utilized by the application developers [5, 7].

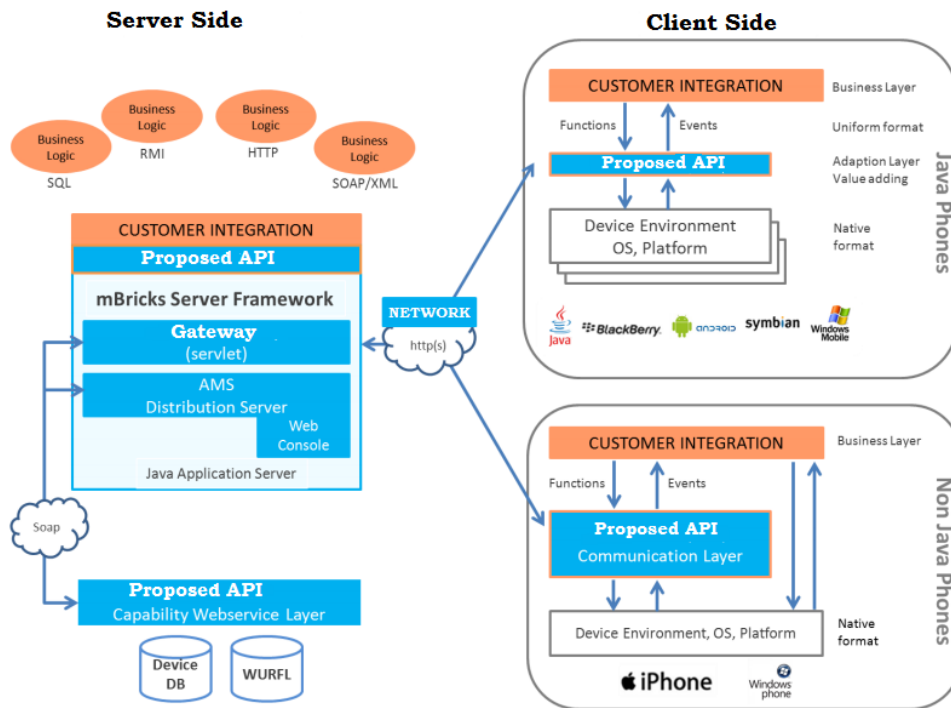


Figure 4 Proposed Platform Independent Architecture

The elements in orange color are our code and business logic, while the elements in the blue are elements to develop. The blue boxes with the orange frames are our proposed API's that your code interfaces with.

- **Client Side:**

The client side is divided into two parts. The top part applies to Java phones:

- The orange box is your customized code where you develop your business logic. In this layer you may take advantage of the many widgets, which can be customized for the user interaction (UI).
- Instead of communicating directly to the specific phones API, you communicate to the API layer in the proposed framework. This is an adaption layer which takes care of the communication to the phones native format. It puts a lot of value adding to this layer, for you to relate to only one API, while the adaption layer uses the native format for file system, communication, accessing device functions (GPS, cameras etc.). Your code simply calls on functions in the API, and receives events triggered from the device.

The lower right part applies to non-Java phones, like iPhone and the new Windows Phone

Operating System:



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

- Because of the nature of iPhone which is based on Objective C and not Java, one cannot take advantage of all the Java based widgets, meaning that one have to communicate from our business logic directly to the device using the native iPhone format. However – when it comes to communication and all server-side integration, one may take advantage of the communication layer, communicating to the server.
- **Server Side:**
- The orange rings show our existing business logic, which may run on any environment, as long as it supports communication based on either Web Services (soap/xml), HTTP, RMI (Remote Method Invocation) or SQL functions.
- The developer creates their own customized integration on a Java Application Server (i.e. TomCat) which communicates to the Java API.
- Inside the Server Framework, the Gateway takes care of the communication to the mobile devices.
- The AMS (Application Management System) is included in the server framework. The AMS Distribution server is your “local application store” where the end-users connect using HTTP from their mobiles, to download your applications. The AMS also has a web-based console for managing the applications. The Distribution server also communicates to a Web Service for collecting the correct capabilities of a mobile device.

For providing communication between server and client. The communication layer is based on pure HTTP, but you’ll automatic take advantage of the optimized and secure protocol to communicate in an efficient way to a mobile client. It is also possible use industry standard HTTPS at the expense of excluding some older phones.

V. CONCLUSION

Developing mobile applications for multiple platforms involves many challenges. Device fragmentation, lack of standardization, and the on-going race for being the dominant OS, mean that mobile application development may be difficult to handle without a proper application development tool.

This paper has explained the different aspects of the fragmented mobile world. It’s complicated and costly to write applications supporting all the platforms/operating systems such as Symbian, Nokia OS, RIM OS, Apple OS, Palm OS, Mobile Linux, Android, Windows Mobile and more. There is also implementation diversity due to different interpretations of the standards.

The fragmentation continues with device diversity such as screen size and orientation, input types like keyboard, soft-keys, hard-keys and touch screens. Continuing with diversity of memory size, processors, camera and GPS support in addition to connectivity options like Bluetooth, GPRS, 3G, Edge, HSDPA etc.

We have presented the idea behind supporting Cross Platform Mobile Applications. When developing using our proposed Framework, the key benefits are:

- Reduced time-to-marked – reuse the many widgets and the complete library for creating your applications
- One (or as few as possible) code base(s) for all platforms and devices
- End-to-end solution – server framework, communication layer and client framework



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

- Network, content, device, and OS agnostic
- Secure communication protocol
- Easy operation and distribution of your mobile application

The main objective behind proposed architecture is to enable cross platform support for mobile application developers. By doing this, one can focus on your main business idea, while the framework takes care of the rest.

So you can stick to what you do best – developing your business logic and content, and let us do what we do best – providing you a great framework for mobile applications that will reduce your cost, frustration and time spending.

REFERENCES

1. Redda, Y.A. (2012). “Cross Platform Mobile Applications Development”, Norwegian University of Science and Technology.
2. Cross Platform Mobile Apps (2010). Paper presented by mBricks [http:// https://mbricks.atlassian.net](http://https://mbricks.atlassian.net)
3. Cross Platform Mobile Development, Unique Computer System. <http://www..ucssolutions.com>
4. Eberhardt, C. (2013), “Cross Platform Mobile Development - Looking beyond the HTML5 Hype”. <http://www.slideshare.net/colineberhardt/cross-platform-mobile-development-looking-beyond-the-html5-hype>
5. http://www.academia.edu/3009209/Comparing_cross-platform_development_approaches_for_mobile_applications
6. www.coronalabs.com
7. www.mosync.com
8. <http://en.wikipedia.org/>

BIOGRAPHY

Vijay R. Mendapara (M.Sc. IT) is an IT consultant at iOS Software, Rajkot, Gujarat, India. His publication includes 1 national conference/seminar.



He is working on Mobile application development for iOS as well as Android platforms for various business domains with different frameworks, APIs.

He also contributes to online forums in related technologies.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013



Yogesh R. Ghodasara (Ph.D. (Computer Science)) is currently working as an Associate Professor at College of Agricultural Information Technology and University Librarian at Anand Agricultural University, Anand, Gujarat, India. His publication includes 1 book, 5 international research papers, and 5 national/state level conferences / seminars on various topics of IT. He is also involved in syllabus designing in different UG and PG courses in Gujarat – India. He is recognized Ph.D. guide in various universities. He is involved in various academic, research and IT administration activities at Anand Agricultural University, Anand, Gujarat – India.



Krunal C. Kamani (M.Phil. (Computer Science), MCA, M.Sc.IT) is currently working as an Assistant Professor (Computer Science) at College of Sheth M. C. College of Dairy Science, Anand Agricultural University, Anand, Gujarat, India. His publication includes 8 research papers in international journals, 2 book chapters, 2 papers in national journals and 13 papers in national conferences/seminars. He received 3rd rank for the best research paper at the national seminar at Shri M & N Virani Science College – Rajkot, Gujarat, India in the year 2006.