

A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach

Rahul Raj C.P

Siemens Technology and Services Pvt Ltd
Bangalore, India
c.raj@siemens.com

Seshu Babu Tolety

Siemens Technology and Services Pvt Ltd
Bangalore, India
seshu.tolety@siemens.com

Abstract— Mobile phones have become an inevitable part of day-to-day life and so do the mobile phone applications. User expectations about the applications are remarkably high. It is a real challenge for the application vendors to provide versatile applications in this competitive market in a short time. The challenge is even more if the application is targeted for multiple platforms. Cross platform mobile application tools are of great help in this scenario. These tools support to develop the applications for multiple platforms in less time. However cross platform application development is also not seamless, it has its own fallacies and pitfalls. The application vendors have to face lots of challenges in cross platform application development. The article targets at bringing awareness on such issues and provides information to make a suitable choice.

Keywords—cross platform; mobile web; hybrid applications; interpreted applications; cross compiled applications.

I. INTRODUCTION

With growing popularity of smart phones, smart phone application development has found a remarkable identity in software industry. There are over thousands of applications which are being used in day-to-day life. Also lots of desktop applications are finding their mobile versions with increased computing power in smart phones. Android, iOS and Windows share the majority of the smart phone market. RIM, Symbian and WebOS share rest of the market space.

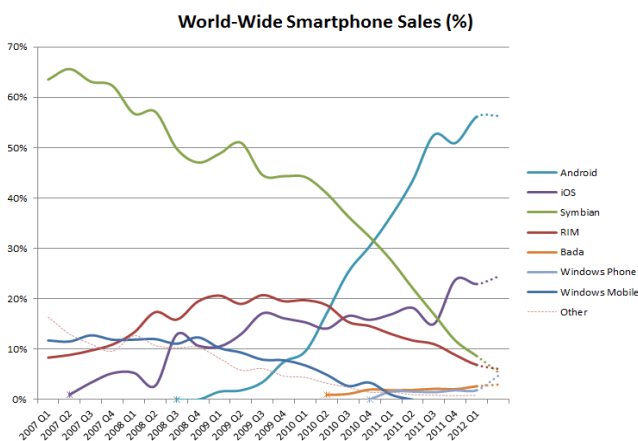


Figure 1. Smartphone sales rate [2]

All the referred platforms have their own native application development infrastructure. It allows accessing the platform and hardware features of the mobile device. Each platform has its own architecture and application development support. The main challenge for an application vendor is to provide their solution across all these platforms. Providing a native specific solution might not be preferred always due to the cost, resources and time associated with the development activity for each platform. Going by native methodology the vendor has to create and maintain separate solutions for each platform. Maintaining separate applications and corresponding code is definitely a pain point for the application vendors.

An ideal solution to this issue is to create and maintain a single application for all the platforms [4]. Cross platform mobile development finds its relevance in this scenario. Cross platform development targets on creating a single application which can be used across multiple platforms. This helps the application vendor to maintain the same code base for multiple platforms. Maintenance and release overheads for multiple platforms can be reduced by cross platform application development.

Though cross platform application development seems to be simple and ideal, it has its own fallacies and pitfalls. As a first step, the application vendor has to identify the real potential for applying cross platform application development. Next to that one has to identify appropriate tool for achieving platform independency, knowing the short comings.

II. CROSS PLATFORM MOBILE DEVELOPMENT APPROACHES

An application can be made cross platform in different methodologies [3]. Some methodologies concentrate on the application construction phase to achieve this, while some concentrate on the application execution phase. Each methodology has its own purpose and context [1]. The application vendor has to choose an appropriate methodology based upon the application requirements. Primarily cross platform development approaches can be classified into,

- Web Approach
- Hybrid Approach
- Interpreted Approach
- Cross Compiled Approach

III. WEB APPROACH

A mobile web application is a web application which is designed to execute in the web browser of the mobile device. Mobile web applications are developed using HTML, CSS and JavaScript. In this approach the mobile device will not have any application specific components installed. The applications will be browser based and the application data is server driven. Since the application is browser based, the application will be platform independent.

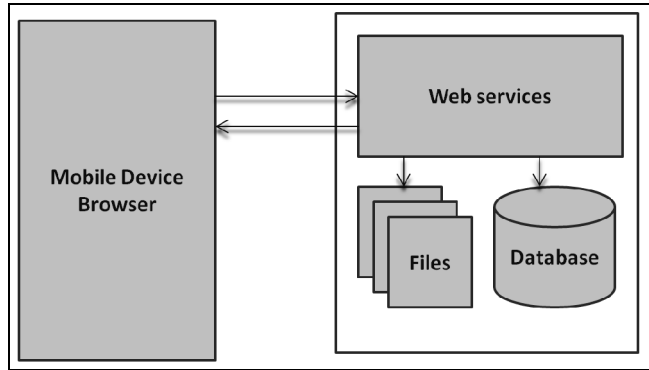


Figure 2. Web Application

In this approach some operations of the application are performed on the client side, while the critical operations are server driven. The client hosts the application user interface and user data validation logic, while the server implements the business logic.

A. Advantages

- A web application does not require any installation in to the mobile device. The application can be accessed using a URL through the mobile web browser.
- Since the data and application is hosted on the server, any update to application is maintenance-free on device. No application update in the mobile device is required.
- Since the mobile web browsers are fairly standardized the Web User Interface can be reused across different platforms.

B. Challenges

- Mobile applications cannot be distributed through mobile application stores; they have to be accessed using the URL. Nowadays users searches application store for desired application. The absence of an application in the application store might have a negative impact on the application popularity.
- A web application might suffer from inferior performance due to connection and network delays.
- Due to application sandboxing web applications cannot access the mobile device hardware and software. Web application cannot access the platform hardware such as camera, GPS sensors etc.
- Unlike desktop applications, a mobile web application has to support different screen resolutions. This is one

of the important factors which need to be considered while developing mobile applications.

- Also the manual testing required to baseline the look and feel of the application across different screen resolutions consumes considerable amount of time.
- The application developer has less control on how different browsers render the content.
- A web application is limited to leverage the gestures offered by the platforms.
- Monetizing a web application is not straightforward as a native application.

IV. HYBRID APPROACH

Hybrid approach stands in between web and native methodology. Hybrid application is developed using web technologies and gets executed inside native container on the mobile device [6]. Hybrid approach uses the browser engine of the device which renders and displays the HTML content in full screen Web view control. The device capabilities are exposed to the hybrid application through an abstraction layer. The abstraction layer exposes the device capabilities as JavaScript APIs (Application Programming Interface). Hybrid approach can take the advantage of both browser engine and device capabilities. Hybrid approach can be used for both server backed and standalone applications. Unlike web applications Hybrid applications needs to be downloaded and installed on the mobile device.

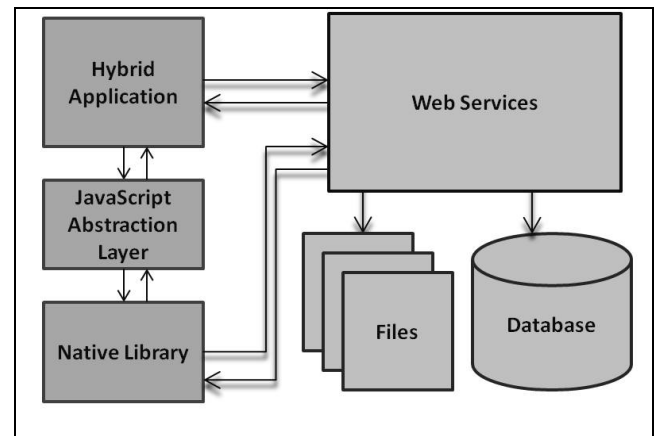


Figure 3. Hybrid Application

A. Advantages

- A hybrid application is distributable through the application store.
- The main advantage of hybrid approach is that the user interface can be reused across different platforms utilizing native platform features.
- Since native platform features are made available through the Hardware abstraction layer, application can make use of device features.
- In this case the web application is powered by device computing capabilities.

B. Challenges

- Hybrid applications are inferior in performance compared to the native applications since the execution happens in the browser engine.
- Since a hybrid application uses JavaScript Hardware abstraction layers, it is subjected to cross space communication vulnerabilities. Hybrid applications also suffer from platform specific behavior of JavaScript and threading model incompatibilities with JavaScript.
- Even though the user interface can be reused across different platforms the user interface will lack the look and feel of native application. To achieve the native look and feel the platform specific styling might be required.

V. INTERPRETED APPLICATION

In the case of interpreted applications the application code is deployed to the mobile device and gets interpreted thereafter. There is an interpreter which executes the code at runtime. The native features are made available through an abstraction layer. The interpreter interprets the source code on runtime across different platforms and thus supports cross platform application development. The interpreted application interacts with the abstraction layer to access the native APIs (Application Programming Interface). Interpreted applications make use of platform-specific native user interface elements for user interaction. The application logic is captured in a platform-independent way. This can be a set of commands in XML or another description language.

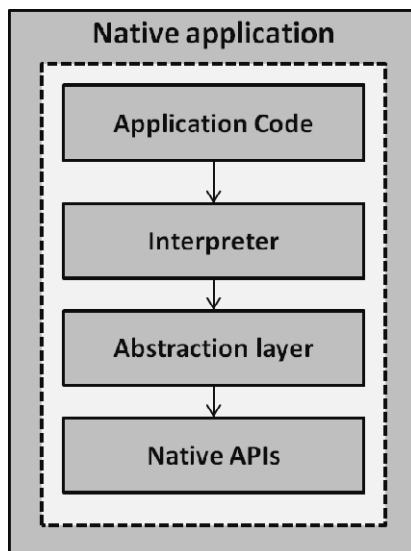


Figure 4. Interpreted Application

A. Advantages

- An Interpreted application provides the look and feel of the native application. The business logic can be reused across different platforms.
- Interpreted application is distributable through the application store.

- The device hardware and platform features are wrapped with specific framework API (Application Programming Interface).

B. Challenges

- Reusing the user interface depends upon the framework level abstraction.
- The main disadvantage of interpreted application is that the development is dependent on the feature set provided by the selected framework.
- The performance of the application might degrade a little due to the run time interpretation of the code.

VI. CROSS COMPILED APPLICATION

In the case of cross compiled applications the cross compiler converts the source code to native binaries. The cross compiler is responsible for generating the executable code for a particular platform [5]. The developers can write the source code in a common programming language and the cross compiler compiles the source code into particular native code. This whole approach is dependent upon the efficiency and reliability cross compiler.

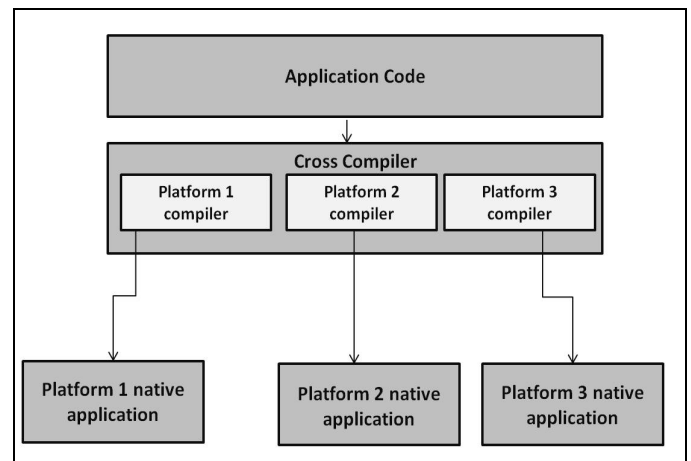


Figure 5. Cross compiled Application

A. Advantages

- Cross compiled application provides all the features that the native application provides.
- Device hardware and software can be accessed here. All the native user interface components can be used.
- Performance is the main highlight of these applications.

B. Challenges

- The main disadvantage of cross compiled application is that the user interface cannot be reused. Also the platform specific features such as camera access, location services, local notifications, etc., cannot be reused. These features are platform specific and the way to access these features varies from platform to platform.

- This approach would be appropriate for simple applications but for sophisticated applications cross compilation will be outweighed by native approach.
- Identifying and rectifying the cross compilation phase issues will be difficult for the application developer. The cross platform solutions currently available in the market are not mature enough.

VII. SELECTION OF APPROPRIATE METHODOLOGY

Selection of the cross platform methodology depends primarily upon the application requirement. It also depends upon the targeted platforms, type of application, etc. The tool and framework support also is a factor to consider while selecting the methodology.

Table 1. Cross platform methodologies and corresponding frameworks which adopt them [6]

Methodology	Frameworks Adopting the Methodology	Language Used	Platforms Supported
Web	Web Technologies	HTML,CSS, JavaScript etc.	Android, iOS, Windows, BlackBerry
Hybrid	Phone Gap	JavaScript	Android, iOS, Windows, BlackBerry
Interpreted	Titanium, JMango	JavaScript	Android, iOS, BlackBerry
Cross Compiled[5]	Mono, applause	C#	Android, iOS, Windows

Each mobile application has its own characteristics. An appropriate cross platform methodology needs to be applied for each of the mobile application. Primarily mobile applications are classified into Server data driven, Sensor/IO based standalone application and client-server applications. The methodology needs to be decided wisely for these types of applications.

Native execution refers to the executions of Interpreted and Cross compiled applications since they both execute in native space. Interpreted application will be inferior in performance compared to cross compiled application due to the dynamic interpretation.

A. Server driven applications

Server data driven applications are thin client applications where the business logic resides on the server. The mobile client allows information display and user interaction. Web approach would be appropriate for this kind of application. By following web approach the entire application user interface can be reused. Since the application business logic resides on the server, mobile client do not have to bother about any application updates in the mobile device. If the application wants to perform any platform level actions, for example

schedule a local notification upon data retrieval, the hybrid approach would be more appropriate.

B. Sensor/IO based applications

Sensor based applications primarily uses the device hardware. Some applications process the sensor data locally while some depends on the server for processing. Approach needs be selected wisely based upon the processing area for these kinds of applications. Cross compiled applications would be of better choice if the data is processed on the mobile device. Because cross compiled applications have native hardware accessibility and are performance rich, interpreted approach could be of second choice provided, corresponding framework offers desired feature set. However the interpreted application will be inferior in performance compared to the cross compiled ones.

If the application is processing-intensive and requires a server to process the data, then hybrid approach could be of more significance.

C. Standalone applications

Standalone application here refers to the applications in which the data is produced and processed on the mobile device itself. It also includes applications where data is from the server and processing happens in client. The mobile client locally processes and displays the information display and allows user interaction. For standalone applications, cross platform or interpreted approach would be a preferred choice since they can leverage the native platform features.

D. Client-Server applications

In the case of client-server applications both the server and client are involved in application data processing. Hybrid application methodology will be appropriate here as the web application can be enriched by utilizing the device capabilities.

It is very important to decide upon the cross platform methodology to be followed for a particular application type. Table 3 provides rating to each of the methodology for different application types. The rating shall be interpreted as follows.

- 1 – Not preferred
- 2 – Preferred, but not the perfect methodology
- 3 – Perfect methodology

Table 2. Application type and preferred approaches

Application Code	Web	Hybrid	Native Execution (Interpreted/ Cross compiled)
Server data driven applications	3	2	1
Standalone applications	1	2	3
Sensor/IO based application(Data processing in device)	1	2	3

Sensor/IO based application(Data processing in server)	1	3	2
Client Server applications	1	3	2

VIII. CONCLUSION

Cross platform solutions are preferred when the application is targeted for multiple platforms with time to market and cost being the critical factors. The cross-platform software vendors could offer their solutions in different working models. Each approach has its own purpose with pertinent advantages and disadvantages. Application vendor has to wisely choose a right approach for his context based on the application types and its relevance.

REFERENCES

- [1] Inderjeet Singh and Manuel Palmieri, "Comparison of cross platform mobile development tools," [online]. Available: http://www.idt.mdh.se/kurser/ct3340/ht12/MINICONFERENCE/FinalPapers/ircse11_submission_16.pdf
- [2] "Mobile operating system", [online], Available: http://en.wikipedia.org/wiki/Mobile_operating_system
- [3] "Native, hybrid and Mobile web", [Online], Available: <http://www.appcelerator.com/platform/titanium-sdk>
- [4] "Cross Platform Application Development", [online], Available: <http://www.ndot.in/cross-platform-mobile-development.html>
- [5] "Building cross platform mobile applications using C#", [online], Available: <http://xamarin.com/>
- [6] "How PhoneGap Works", [Online], Available: <http://phonegap.com/about/>