State of Art Approaches to Build Cross Platform Mobile Application

Bhushan S. Thakare
Assistant Professor
Department of Computer Engineering,
Sinhgad Academy of Engineering,
Pune, India

Dhanashree Shirodkar
Department of Computer Engineering,
Sinhgad Academy of Engineering,
Pune, India

Naghma Parween
Department of Computer Engineering,
Sinhgad Academy of Engineering,
Pune, India

Shama Parween
Department of Computer Engineering,
Sinhgad Academy of Engineering,
Pune, India

ABSTRACT

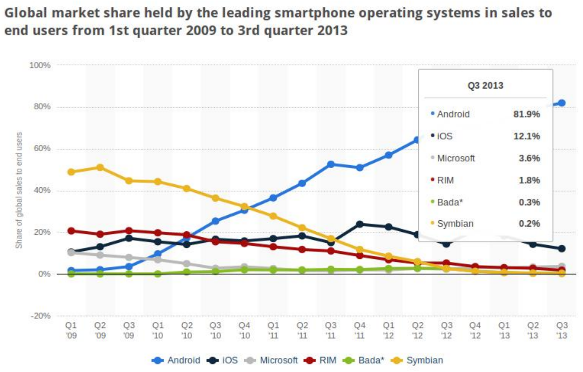
Mobile application market has grown exponentially. The market is becoming increasingly fragmented with at least five important mobile platforms making native development of mobile applications a challenging and costly endeavor. Cross-platform development might alleviate this situation. Cross platform tools support to develop the applications for multiple platforms in less time also the developer need not have the prior knowledge of native language. This paper deals with available frameworks, cross-platform solutions and evaluating the features, performance, and development experience of these tools. Also, this paper targets on bringing out the fallacies and pitfall in different cross-platform approaches so as to raise awareness on such issues and help in selecting suitable approach.

Keywords

Cross platform; smart phones; web applications; hybrid applications; interpreted applications; generated or cross compiled applications.

1. INTRODUCTION

Now-a-days mobiles are more a necessity then need. Except making calls, there are many features which are gaining popularity like Camera, Music, Global precipitation system (GPS), Accelerometer etc. The smart phone market has shown a promising growth. There are over thousands of applications developed and used every day. Major available mobile Operating Systems such are Android, iPhone, Blackberry and Windows phone. The OS provides space for developers to built application for their users using the built-in features provided by them. Since each Operating system vendor uses different Software Development Toolkit, the basic architecture and support of programming language of OS is very different from each other. One built application is not compatible for all other OS, which forces the developers to build the same application again for other OS.



Cross-platform development approaches emerged to address this challenge by allowing developers to implement their apps in one step for a range of platforms, avoiding repetition and increasing productivity [3]. On the one hand, these approaches need to offer suitable generality in order to allow provision of apps for several platforms. On the other hand, they still have to enable developers to capitalize on the specific advantages and possibilities of smart phones. Even though cross platform approach looks simple and ideal it has its own drawbacks. In the following, we present existing types of cross-platform frameworks their advantages and their challenges.

2. APPROACHES FOR CROSS PLATFORM APPLICATION

Platforms are a combination of hardware (system plus any possibly built-in sensors or actuators), operating system, vendor-provided software SDKs and standard libraries [BH06], which together offer the base for building software for that platform.

A cross-platform framework allows reusing parts of the application source code for multiple platforms and may additionally offer building blocks such as an architecture style (e.g. MVC2), user interface API or other functionality that is not specific to a single platform. Such frameworks can also expose APIs for platform-specific features.

An application can be made cross platform by using different methods. Some methods can achieve this in application construction phase while other may achieve it in application execution phase. The developer has to select which method is best suited for his/her requirement as every approach has its own uniqueness [1].

The approaches can be classified primarily into following categories:

- Web approach
- Hybrid approach
- Interpreted approach
- Cross compiled or generated approach

2.1 Web Approach

Web apps are browser-based applications in which the software is downloaded from the web. Web apps are based on widespread Internet technologies such as HTML, CSS and JavaScript. Any application specific components are not installed in the device [2]. The applications will be browser based and the application data is server driven. Since in this approach, the application is browser based, the application will be platform independent. Most of the critical operations such as business logics are run on the server side where as the client side deals with the user interface and data validation logic.

The major advantage of Web apps is that they do not require installation and subsequent upgrades; the application can be accessed using URL in the web browser. The data and the application is hosted on the server so the application is maintenance free on the device.

The main disadvantage of web apps is the limited access to the underlying device's hardware and data. Another problem is the extra time needed to render the web pages and the extra cost needed to download the web page from Internet. Also, since they can't be installed physically on a device there are situations such as when the device is in Airplane mode where web apps are inaccessible for the end user.

2.2 Hybrid Approach

Hybrid apps try to combine the advantages of web and native apps. Hybrid apps are primarily built using HTML5 and JavaScript, and a detailed knowledge of the target platform is not required. Hybrid application is developed using web technologies and gets executed inside native container on the mobile device. Hybrid apps are installed on the device and access to the underline device hardware and data is feasible through specialized APIs. Unlike web apps, they need to be downloaded and installed first. An example of the most popular container for creating hybrid mobile apps is PhoneGap [2]. 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.

The main advantage of hybrid approach is the reusability of user interface across different platforms utilizing native platform features. Also, these apps are distributed thorough app stores. The application can make use of device's features as native platform features are made available using APIs.

On the other hand, the disadvantage of these apps is that hybrid applications also suffer from platform specific behaviour of JavaScript and threading model incompatibilities with JavaScript. These apps cannot completely achieve native look and feel. Also they require more processing time than native apps. Thus they are inferior compared to native apps.

2.3 Interpreted Approach

In interpreted apps native code is automatically generated to implement the user interface. An interpreter is present that interprets the source code on runtime across different platforms and thus supports cross platform application development. The end users interact with platform-specific native user interface components, while the application logic is implemented independently using several technologies and languages, such as Java, Ruby and XML etc [1].

The main advantage of this approach is efficiency because of the native user interfaces; they provide the look and feel of native application. Just like hybrid apps, these are also distributed on the app store.

The downside of this approach is the complete dependence on the software development environment. More specifically, new platform-specific features (e.g. new user interface features of a new android version) can be available to apps only when and if supported by the development environment. The performance gets affected due to the run time interpretation of the code.

2.4 Cross Compiled or Generated Approach

In cross compiled approach, cross compiler generated apps are compiled just like a native app and a platform-specific version of the application is created for each target platform. The developer uses cross compilers to convert programs written in any common programming language to convert it into the required native code [1]. A popular example of software development environment for creating generated apps is Applause. This approach is highly dependent on the efficiency of the cross compiler.

The advantage of this approach is that the generated apps achieve high overall performance because of the generated native code and provides all the features of native app including its native interface components.

The major drawback of this approach is that the user interface cannot be reuses. Also, many of the features cannot be reused as the method to access those features for example camera access; geolocation etc is different for each platform.

This method cannot take the load of big and sophisticated applications. It is best suited for small applications.

3. CONCLUSION

Mobile app development is an evolving field that has attracted great interest. The most common type of mobile apps was, until recently, native mobile apps. Native mobile apps are developed for a specific platform and offer without doubt the best user experience. However, the increasing number of mobile platforms has constituted the development of native mobile apps inefficient both in terms of development time and development/maintenance cost. Cross platform solutions are preferred when the application is targeted for multiple platforms with time to market and cost being the critical factors. The aim of this paper was to investigate the cross-platform approaches that have been proposed and their contribution towards achieving the aforementioned goal. Our review showed that the current trends in developing cross-platform mobile applications focus on four primary app types, which are web, hybrid, interpreted and generated/cross compiled apps. 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.

4. REFERENCES

[1] Xinogalos and Spyros Xanthopoulos, "A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications "

[2] Rahul Raj C.P and Seshu Babu Tolety, "A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach"

[3] "Cross Platform Application Development", http://www.ndot.in/cross-platform-mobile-development.html

[4] Henning Heitk¨otter, Sebastian Hanschke, and Tim A. Majchrzak," Evaluating Cross-Platform Development Approaches for Mobile Applications"

[5] About PhoneGap, "How PhoneGapWorks", http://phonegap.com/about