Suyati
dream. dare. do

YOUR GUIDE TO PLATFORM INDEPENDENT MOBILE PROGRAMMING

With smartphones and tablets becoming all pervasive, mobile programming, especially app development has come to the center stage of IT developments. However, programming for the mobile is not as simple and straightforward as programming for desktop devices.

Suyati℠
dream. dare. do

Suyati
dream. dare. do

01 Introduction

In the era of desktops, all computing devices ran on just a handful of operating systems, with Microsoft Windows dominating the scene, and Linux and Mac offering the alternatives. Adopting the code a few times over would ensure that it works across all desktops. In contrast, the mobile space is now highly fragmented. In the beginning, there was only iOS, but soon Windows and Android started crowding the market.

A 2012 Gartner study estimates that Android devices collectively account for over 50% of worldwide smartphones.

Initially developers rolled out iOS, Windows, and Android version of mobile apps. However, the proliferation of mobile devices with various flavors and versions of operating systems, and different device configurations makes rolling out a separate development for each environment an impossible task.

Android, for instance, has branched out to many flavors and compounding the mix is the varying screen size, resolution, hardware, memory, processor speed, and device configurations of Android devices.

Suyati
dream. dare. do

02
Dealing with diversity

The challenge before mobile app developers is to reach out to users in such a diverse and continuously evolving landscape.

The options are:

1. Roll out different versions of the apps for different operating systems, as before. This worked well when the number of OS variants and devices were limited. But the constraints of time, the huge costs involved, difficulty in acquiring in-depth knowledge of each platform, and familiarity with each environment makes this a non-starter.

2. Use emulation software to generate code for different OS. While such software exists, it often fails the test of practical applicability, especially when the structure of the program is complex.

3. Opt for platform independent development. Even when the focus is to target just a few major operating systems that most mobile phones in the developers crosshairs may be using, it makes far more sense now to develop in a cross-platform framework, or platform independent framework rather than build the app multiple times for iOS, Android and other OS.

There are mainly two approaches to platform independent programming:

1. Mobile web apps, mainly in HTML5
2. Cross platform scripting (XMS)

Suyati
dream. dare. do

03
Mobile Web Apps

The mobile web app approach to programming is hosting all or some parts of the software, including the user interface, on the cloud or remote server, with users accessing the same through the web browser. The page is loaded every time the user wants to access the app, in a similar way websites operate.

Mobile web apps use a common code base, such as HTML5 or Javascript / CSS3, along with some extra libraries, and come wrapped in a relatively thin native-code wrapper.

HTML5 works across all devices and operating systems, thereby doing away with the need for duplicating the code and using different language for different OS, such as Objective-C for iOS and Java for Android. Moreover, it has the capability to provide a rich client-side experience to the code running on the web browser.

It also has the capability to access much of the common features requested by most apps, such as local storage, geo location, audio and video, accelerated graphics, offline working, accelerated graphics and more, to deliver on a great user experience. Although such apps require internet connectivity, users may still download content for later use. The superior cache features enable users to access repeated content without a web connection.

Hosted build services circumvent the need for software development kits (SDKs) to compile apps into their native wrappers. Adobe has integrated the PhoneGap Build service into their new Creative Cloud offering. Red Gate's Nomad offers another option.

A good example of mobile web apps at work is Amazon's Kindle web app on iOS. Changes in App Store policies prompted Amazon to go in for a pure HTML5 app, and this app is a standing testimony to the fact that when done right, the distinction between a native app and a website is very blurred. Twitter's web app is another good example of web app at work.

Advantages

Apart from the obvious convenience to developers of not having to write different codes to different operating systems or mobile environments, mobile web apps offer several advantages to users as well.

- It remains future-proof. Changes in device configuration would not render the program obsolete and users would not have to download updates frequently.
- All users remain in the same latest version, doing away with compatibility and version conflict issues.
- It spares users from downloading software, thereby saving space, which comes at a premium in most devices.

Limitations

However, for all the advantages and convenience, mobile web apps come with significant shortcomings as well.

- Both native apps and mobile apps require internet connectivity to fetch data. However, mobile apps are more dependent on internet connection. Mobile web apps have to fetch the user interface and content from the remote server on every request. This slows things down considerably. Poor internet connectivity makes it much worse. Offline caching mitigates the issue to some extent, but it does not do away with the problem. In native web apps, the user interface is already downloaded, and as such the app works regardless of the state of the internet connection.

- Mobile web apps can access many of the device's capabilities, but technology has not yet developed to enable it to access all the device features that native apps access. This restricts the capability of mobile apps to some extent. Also, web apps face some limits on the APIs available, and the amount of data that can be stored locally.
- Most people are now used to apps listed in app stores. Such listing makes it possible for users to search and install their desired apps easily. The seamless nature of app discovery, installation, complete with ratings and review contributes significantly to the popularity of native apps. Web-based apps do not offer such listing or easy search facility. From the developer's perspective, an app store makes it very easy to monetize an app. For web apps, the options of monetization are limited to clumsier paywall or subscription set ups.

The verdict

So what is the verdict on mobile web apps? A study by Global Intelligence Alliance reveals that 23% of developers give greater focus on web apps compared to native apps. 30% of developers now give equal focus to web apps and native apps, and 24% of developers continue to give greater focus on native apps over web apps. The other 23% of developers are on a wait and watch mode.

Suyati
dream. dare. do

04 Cross Platform Mobile Development Tools (XMT)

Cross platform mobile development tools (XMT) offer tools to develop native apps that may be deployed on different platforms. Such development requires minimal change of code for customization to each specific platform.

XMTs come in a variety of guises. At the basic level is C compilers that link against the various platform SDKs. This allows development in a common language, but with vastly different APIs.

Cross platform development tools however have evolved significantly in recent times to offer much high end capabilities compared to the basic C compiler. Such tools compile JavaScript or the common language in which coding takes place into native code, and links it against their platform-neutral SDKs. The bridge between Java Script and native code allows communication between HTML and native platform.

The three popular XMTs in vogue today are:

1. **Apache Cordova (formerly PhoneGap)**
2. **Appcelerator's Titanium**
3. **RhoMobile, developed by Motorola Solutions**

All three support native code for further extensibility, but each comes with its own distinct flavors.

The open-source Apache Cordova project, initially known as the PhoneGap framework before it was donated to the Apache Software Foundation makes it possible to code in HTML and Javascript, with the framework compiling the JavaScript code into native app automatically.

Apache Cordova facilitates development of mobile apps using a common development language with a common set of APIs for all supported platform. Every Apache Cordova application has UI layer in HTML5 / CSS Javascript that runs across all devices and provides wrappers for Android, iOS, BlackBerry and Windows Phone that is platform-specific. In order to code against, it has also exposed a platform-independent API in JavaScript.

Compared to pure HTML5 web app development, it offers enhanced access to the device. The enhanced access makes available local storage over the standard 5MB limit, ability to upload photos from the local device, running background services and more, all not normally possible in a pure HTML5 development.

Appcelerator Titanium offers a wide variety of choice of coding in HTML, Javascript, Ruby, PHP and Python. RhoMobile offers coding in Ruby.

Advantages

Cross platform XMTs reduce deployment cost and spares developers the hassle of generating different programs for different environments. Unlike web apps where such advantages come with prospects of slowing up things considerably, cross platform XMTs speed up things considerably. Everything is downloaded to the device, just as in native apps.

Limitations

However, compilation is still slow compared to native code, as the software has to churn out code compatible with the platform where it is run, every time compilation takes place. Moreover, there is a significant downside of cross platform XMTs churning out inefficient and bloated code, as the code optimization features available in native code is not available for cross platform XMT codes.

Similar to the limitation of web apps, cross platform XMTs too have some restrictions on accessing all device features and capabilities.

05 Conclusion

All things considered, it makes sense to opt for platform independent development in about 90% of all instances rather than waste time setting up multiple development environments and re-developing the app in several different and unfamiliar languages. This is more so considering that the major stumbling block for platform independent development – the non-availability of desired parts of platform SDK or peripherals in the cross-platform framework is fast becoming a thing of the past, thanks to the advancements in computing.

The other 10% too is going away. This belongs to functions where development in a platform independent environment is not viable at present. For instance, while it is still not possible to develop high end games in a platform independent environment, the enhanced graphics performance made possible through canvas has brought this task within the realm of possibility and something that would soon take place.

Suyati
dream. dare. do

Suyati Technologies

Suyati is a young, upwardly mobile company focused on delivering niche IT services to support myriad Digital Engagement strategies. Our expertise also includes integration and delivery of CRM, CMS and e-commerce solutions.

www.suyati.com **services@suyati.com**

Reference:

- *http://www.codeproject.com/Articles/388811/An-introduction-to-cross-platform-mobile-developme*
- *http://www.twago.com/blog/developing-mobile-apps-compatible-and-platform-independent/*
- *http://stackoverflow.com/questions/11391226/platform-independent-mobile-development-sdk-for-a-small-task*
- *http://www.slideshare.net/whatanidea1/a-quick-guide-to-cross-platform-mobile-development*
- *http://www.gartner.com/newsroom/id/1622614*
- *http://www.visionmobile.com/blog/2013/07/web-sites-vs-web-apps-what-the-experts-think/*