Bridge-Stage Framework for the Smartphone Application Development using HTML5

Mukesh Vijay[1] Prateek Diwan[2] Manoj Raman[3]
[1,2,3]Department of Computer Science
[1,2,3]Vivekananda Institute of Technology, Rajasthan, India

Abstract— Now a days, the Web has become an integral part of our everyday lives. The rapid growth of the smart phone market has brought the Web from our home desks to anywhere we are, and enabled us to access this vast source of information at any time. The mobile operating systems (OS) used by modern smart phones are too diverse such as Google's Android, Apple's iOS, Microsoft's Windows Phone, and so on. Smartphone application development is done using native platform such as iPhone using Objective-C, Android using Java, Windows Mobile using C# and so on. Therefore, a bridge stage framework which supports 'Write once and deploy everywhere' is required to support the development of Smartphone applications. This paper presents the HTML5-based bridge stage framework which uses Phone Gap and Web kit to support the development of Smartphone applications that are written as Web applications. A big problem with developing applications for mobile devices is platform fragmentation [6]. That means that there are many different mobile platforms that are further divided by the different versions available [5][2]. Users with older hardware are left without support and updates as newer devices are put out on the market [9]. This means that the developer has the choice between limiting the solutions and only aim for a minor part of the spectra or to develop for more platforms to reach as many users as possible. To maximize the amount of possible users, the developer has to create an application for each platform and make sure that they are backwards compatible so that users with older devices can use them.

Keywords: Smartphone Application, Cross-Platform, HTML5, PhoneGap, Web Application, Android, iOS, Windows Phone

I. INTRODUCTION

In parallel with the rapid growth of the Web, mobile phones have evolved from briefcase sized "portable" telephony devices into modern pocket sized computers. The mobile revolution has already changed the world and more people have access to the Web from a mobile device than from an Internet connected desktop computer. [1] The Web is not constrained into computers and mobile phones only, even Tablets, TVs, e-book readers, watches and house hold appliances are connecting to the Internet. For the first time in history, we have a truly ubiquitous digital medium. [1] Universal accessibility and openness are the keys to being the ubiquitous information platform of the digital age [4]. Now the Web is closer in accomplishing its original principles in equality and universality; anyone can access this vast source of open information from anywhere, with any device. All you need is a web browser that supports the open standards of the Web.

The goal of the Web is to serve humanity. Being the universal digital medium, mobile devices has some unique characteristics that other mass media lack. Mobile is personal, always-on, always- carried medium with a built-in payment channel. Mobile is in your pocket at the moment you have your creative impulse. Various mobile operating system is available in the market, in which apple iOS is widely used now a days. Fig 1 shows the mobile OS currently being used in market.

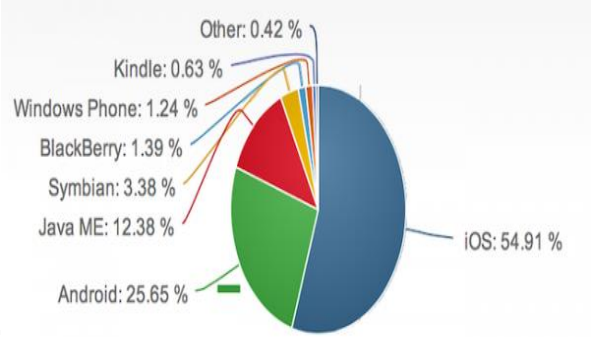

Fig. 1: Various Mobile OS

The viral spreading of mobile phones has raised the need for building scalable applications that can handle the whole spectrum of devices, screen sizes, and form factors that are used to access the internet. This is the need that HTML5 with all the related tools and APIs have promised to solve. Performance is the foundation of a great user experience [3]. Performance means the speed of downloading, initializing and using an application as perceived by the user as well as the responsiveness and smoothness of the user interface influencing the overall user experience. Native tools have been carefully optimized to provide the best possible performance and responsiveness, and web applications are often unfavorably compared to them. In the end, however, the received savings in development time, deployment, cost-efficiency, and cross-platform support can often outweigh the possible compromises. [3] Performance of HTML5 as a cross-platform application platform for different device form-factors. To maximize the amount of possible users, the developer has to create an application for each platform and make sure that they are backwards compatible so that users with older devices can use them. Because of the fragmentation some companies have turned to cross-platform solutions that work on several platforms with a minimum of adaptation. The various mobile phone OS and the programming language required to make the application for that OS is given below in the table.

Mobile OS Type	Skill Set Required
Apple iOS	C, Objective C
Google Android	Java (Harmony Flavored, Dalvik VM)

RIM BlackBerry	Java (J2ME FLavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows 8 Phone	.NET
HP Palm web OS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Table 1: Mobile OS and Language Required

II. HTML 5 FEATURES

A. Semantic Markup

Google did a study in 2005 of a sample of over a billion HTML documents about the popular class names, elements, attributes and related metadata. This analysis had a large impact on which elements and attributes were considered in the upcoming HTML5 standard. HTML5 defines several new elements and attributes. The objective is to make the markup more semantic for developers and for content processors such as search engines and screen readers. The specification aims for a more semantic structure of HTML by dropping many presentational features.

B. Extensibility

HTML5 defines the main constructs of a semantic and accessible document. However, some specific use cases require a more precise and context- dependent and fine grained semantics. Also, web browsers might introduce new features that must conform to the standards. This is why HTML5 is made extensible for adding more semantics or additional features on top of the existing standard. There are several ways to extend HTML5. The simplest approaches include using the defined general attributes with certain vocabularies.

C. Media

Multimedia support is crucial for modern applications. HTML5 defines elements and APIs for audio, video, subtitles, and embedded content. Previously to use these rich content types, developers have had to rely on third-party plugins and browser extensions. Not having to rely on plug-in and extensions has been one of the main goals of the HTML5 standard for improving the openness and accessibility of web content.

D. Canvas 2D Context

HTML5 defines the canvas element. It is a resolution-dependent bitmap canvas for dynamically rendering graphics. It can be used, for example, for graphs, games, or other visuals. [7] The Canvas 2D Context specification draft [4] defines a JavaScript API for programmatically drawing on the 2D canvas surface. The API defines functions for drawing shapes, paths, text, gradients, and images on the canvas and other functions for handling the bitmap data.

E. Form Enhancements

Forms are an essential construction in interactive HTML documents. How- ever, due to their relative simplicity in terms of expressiveness and the lack of proper accessibility features, developers have been forced to build lots of JavaScript solutions to enhance and fix some of these problems. HTML5 brings several enhancements to forms. A new input type for numbers, dates, email addresses, and etc. obsoletes the need of scripted widgets by using native platform controls. New form attributes like placeholder and autofocus bring easy-to-use accessibility and usability improvements and also reduce the need for scripting. [9]

F. Offline web applications

By design, web sites have always needed a working network connection. Applications, however, should be able to work offline or in unreliable and flaky networks. Especially mobile networks are unreliable [8], which has raised the need for offline support in HTML5.

III. USER EXPERIENCE

User experience is all about how users feel when they interact with an app in a specific context. It is about utility, ease of use and efficiency. While positive UI is subjective in nature, a user knows it when it is there and recognizes when it is not. To be fair, the HTML5 standard has delivered more native-like capabilities such as access to the GPS location or accelerometer for mobile web applications. However, these still fail to deliver the same user experience on different devices and perform slower when compared to a native implementation on the iPhone or Android. HTML5 capabilities only represent a small number of the new native features. For example, Apple introduced 1,500 new APIs in iOS 5 for developers to leverage, including access to iCloud Storage, Newsstand and Twitter. Both Apple and Google continue to deliver new releases each year with thousands of new APIs. HTML5 is actually falling farther behind native versus the popular belief that it's catching up.

IV. PERFORMANCE

A central part of user experience is performance of the application on the mobile device. Performance comes in two forms: rendering and loading. Slow rendering and loading can mean death to an application's usage and hence its success. With the increased computing power on today's mobile smart phones and tablets, the tolerance for delays from rendering and loading is at an all time low. Native applications have set the standard for what a user expects from a performance perspective and reflect the best possible approach to delivering seamless and immediate responses to user actions and requests. This is clear when comparing the performance of loading the application locally to serving it up through a browser where data connectivity can be weak or missing. Once the app is launched, performance can be measured against device actions and operating-system-specific gestures like scrolling, swiping, rotating, and clicking. The performance gap between HTML5 and native is so great that even non-technical users easily notice the difference. For example, it is unacceptable to use a calendar as part of a mobile app and have it take 4 seconds to load and then be slow to the eye and touch when navigating from one day to another.

V. MONETIZATION

By distributing new versions and updates to applications via the browser, HTML5 apps are not dependent on an app store. Apple's App Store represents the most vibrant ecosystem on the planet, with 400 million accounts with credit card information stored for each user. This is the

largest number of participants for a transactional site on the Internet. Consumers have downloaded 30 billion apps and Apple has written checks for over $5 billion to developers.

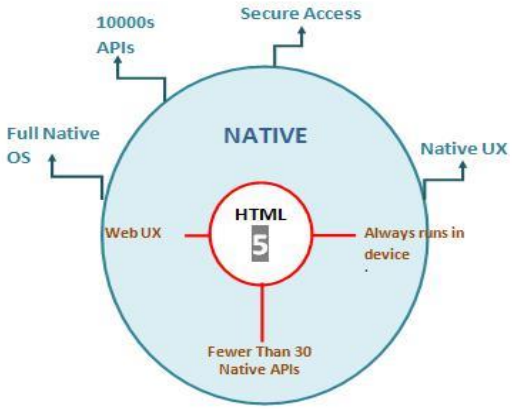

Fig. 2: The Native APIs

VI. Bridge Stage Deployment Cost

Driven by the desire for a cutting-edge app with beautiful user experience, many developer's and organizations first choice is to develop a native iPhone and iPad app. The next step for most businesses is to build the same app on the Android platform. But building out another team to create the same application for all versions of Android phones is problematic in most cases and can double or triple overall costs.

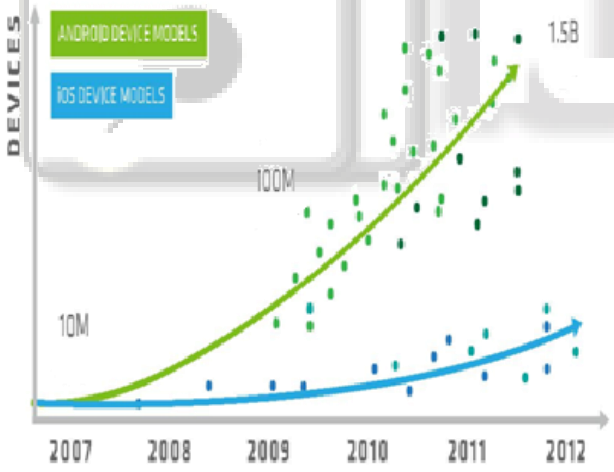

Fig. 3: Users of iOS and Android OS

The additional costs of each new development team are easy to quantify. In comparison, building a mobile web application that includes a rich experience based on HTML5 and that can run on all browsers would seem to significantly reduce development costs and time. Truly understanding the cost benefits and write-once-run anywhere claim of HTML5 contrasted against a native user experience requires understanding two types of fragmentation challenges. The first is driven by all the various Android variations. The second highlights the lesser-known but equally critical problem of mobile web browser fragmentation.

VII. Fragmentation

With multiple releases of the Android OS over the past few years and over 28 implementations by multiple manufacturers the pains associated with writing an application for Android deployment are well documented. Unfortunately, the promise of "write once – run anywhere" is not real today for HTML5 either. Web applications were touted to be the unifying factor for mobile devices. But this assumes that all mobile devices access web applications in the same way. There are not only many different browsers in existence (including Internet Explorer, Safari, Chrome and Firefox), but also many different versions of those browsers. The level of support offered for HTML5 varies widely across those different browsers. There are 200+ different types of available browser interpretations of HTML5 and the pace of browser updates is increasing. Over the past two years, Android has had 21 updates to their browser. The following figure shows the various web browsers available in the market and is widely used in different OS.

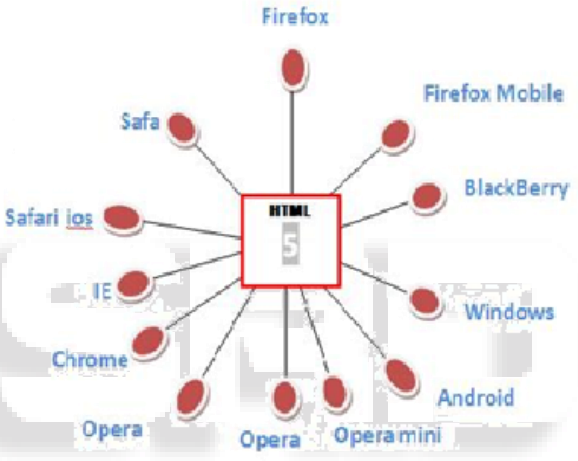

Fig. 4: Different Web Browser

VIII. Native V/S HTML5 Comparisons

Timeliness of new OS innovations Standards bodies are not known for speed or innovation. W3C is the main governing body of the HTML standard to date. While W3C works to deliver new OS support from the manufacturers with every release of the standard, the OS and handset developers like Apple and Android are in control of when and how fast they implement new capabilities in their operating systems for developers to play with. As mentioned above, HTML5 does not keep pace with the multiple new releases of Android per year or annual iOS releases. It has taken five years to ratify the standard in its current form. Slow-evolving standards force browser vendors to innovate ahead of the standard. Straying from the standard will further complicate the aforementioned browser fragmentation problem. Slowing the innovation in the HTML standard further is the recent news that the standards bodies for HTML5 have now split. The Web Hypertext Application Technology Working Group (WHATWG) and the World Wide Web Consortium (W3C), the two bodies working on HTML5, are parting ways, with WHATWG taking charge of an evolving, “living standard” and W3C working on a more static “snapshot.” It

is debatable how splitting the standards body driving HTML will speed the release of more support of mobile OS.

IX. A SMOOTHER PATH TO SUCCESSFUL APPLICATION DEVELOPMENT

The mobile revolution has spread the Web from our home desks to anywhere we are, to be used at any time of the day. The roots of the Web lie in openness and universal accessibility for everyone, and today more and more people can afford a device to access the vast information spread all over the Web. One crucial factor in the universality is the open standards used for defining the protocols and APIs of the Web. HTML5 tackles many of the growing pains of the Web by defining standards to handle all the devices capable of accessing the Internet. The set of new specifications drafts is very large and growing all the time. We introduced the latest specifications and drafts related to modern web application development, some of these specifications already have very good implementations in several browsers but some are just very early drafts. We also presented modern tools and libraries for developing mobile web applications. Performance is one of the main components of a successful and usable application. We took a practical focus on performance optimization of mobile web applications.

X. CONCLUSION AND FUTURE WORK

The quality of HTML5 based application developed for smart phone can be easily evaluated by using the method discussed in this research paper. Another goal of this research paper is to find the cost effectiveness of developing HTML5 based applications. HTML5 provides us capability to develop such application which are platform independent and also these applications are of good quality. Future work involves to develop more complex applications which can be evaluated on those complex scenario.

REFERENCES

[1] Burnett, D. C., and Narayanan, A. get user media: Getting access to local devices that can generate multimedia streams. W3C Editor's Draft, W3C, Dec 2011.
[2] Charland, A., and Leroux, B. Mobile Application Development: Web vs. Native. Communications of the ACM 54, 5 (2011), 49–53.
[3] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1995.
[4] Brubeck, M., Moon, S., and Schepers, D. Touch Events version 1. W3C Candidate Recommendation, W3C, Dec 2011. Available at: http://www.w3.org/TR/touch-events/. Accessed 7-Febuary-2012.
[5] Anderson, O., et al. Scalable Vector Graphics (SVG) Tiny 1.2 Specification. W3C Recommendation, W3C, Dec 2008. Available at: http://www.w3.org/TR/SVGTiny12/. Accessed 6-Febuary-2012.
[6] Fielding, R. T. Architectural Styles and the Design of Network-Based Software Architectures. PhD thesis, University of California, 2000.
[7] Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Kreutzer, K., Kubiatowicz, J., Morgan, N., Patterson, D., Sen., K., Wawrzynek, J., et al. A View of the Parallel Computing Landscape. Communications of the ACM 52, 10 (2009), 56–67.
[8] Carlisle, D., Ion, P., and R., M. Mathematical Markup Language (MathML) Version 3.0. W3C Recommendation.
[9] Berners-Lee, T. Long Live the Web. Scientific American 303, 6 (2010), 80–85.

