Native vs. HTML5 Mobile App Development

Which option is best?

appcelerator.com

appcelerator®

Introduction: The Native vs. HTML5 Debate

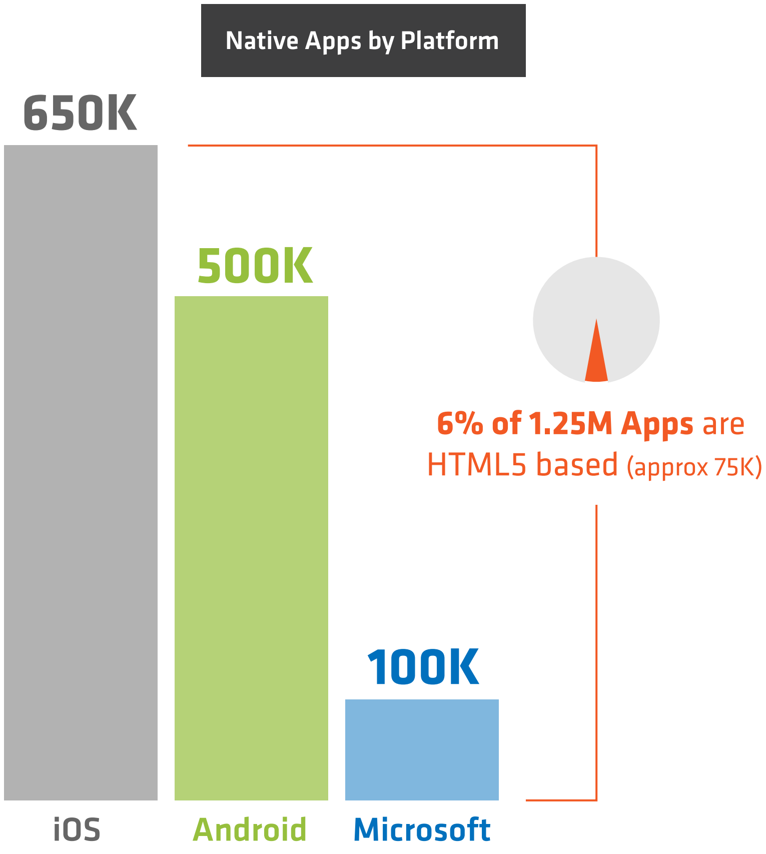
While there are far more native than HTML5 mobile applications in the market today, the debate rages on regarding what is the better approach: go native or go HTML5. This paper will explore the benefits of each. It will show that while HTML5 applications have a place for a segment of applications, the standards body governing HTML5 will never evolve fast enough to match the constantly rising bar of user experience delivered via native operating systems.

Exploring this debate requires discussing the importance of a rich, high performing native user experience versus the cost benefits associated with "write once – run anywhere." To underscore this point, let’s determine how many native versus HTML applications are in production today. This baseline will show us current user preference.

Native applications in production today
By most estimates, there are roughly 1.25 million native applications used in production today. This total is largely made up of iOS, Android and Microsoft applications deployed through their respective application stores.

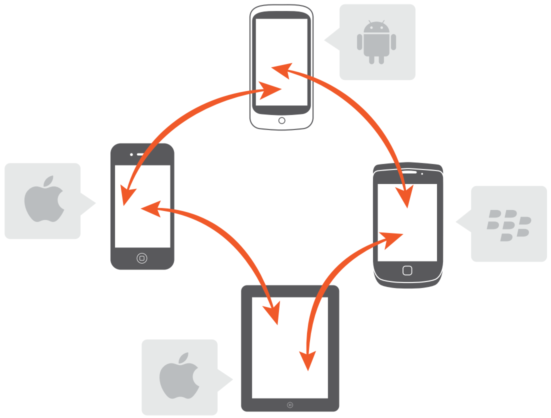
iOS – 650,000
Android – 500,000
Microsoft – 100,000

HTML apps in production today
It is difficult to determine exact numbers as there is no single way to track browser usage. Given the difficulty, this can be best estimated by identifying leading indicators. In the largest survey of its kind for mobile development, Appcelerator/IDC’s Q2 2012 survey of 3,600+ developers found that 6% of applications were targeted for development in HTML5. With this



statistically relevant sample of the developer community, we can estimate that of the 1.25 million native apps in production roughly 6% or 75,000 are mobile web or HTML5 based. Another data point to check is the number of HTML5 applications listed within the Facebook app store. One of the biggest proponents of HTML5, Facebook now claims to have about 20,000 applications in their app store. Given this data point, the 75,000 total HTML5 applications in the production estimate appear reasonable.

1.25 million native applications versus 75,000 HTML5 applications in production today in public app stores: these are telling statistics. They clearly underscore user preferences for a rich, high performing user experience. However, costs can be prohibitive when building native applications on every required OS. In the same developer survey referenced above, 80% of developers expressed the need to deploy their applications on more than one operating system. Development team costs need to be doubled or tripled to deliver this native experience on multiple OSes (if those developers can even be found and hired). If costs were the only consideration, one would expect HTML5's promise of "write once – run anywhere" to play a bigger role in the debate.



The promise of "write once - run anywhere"

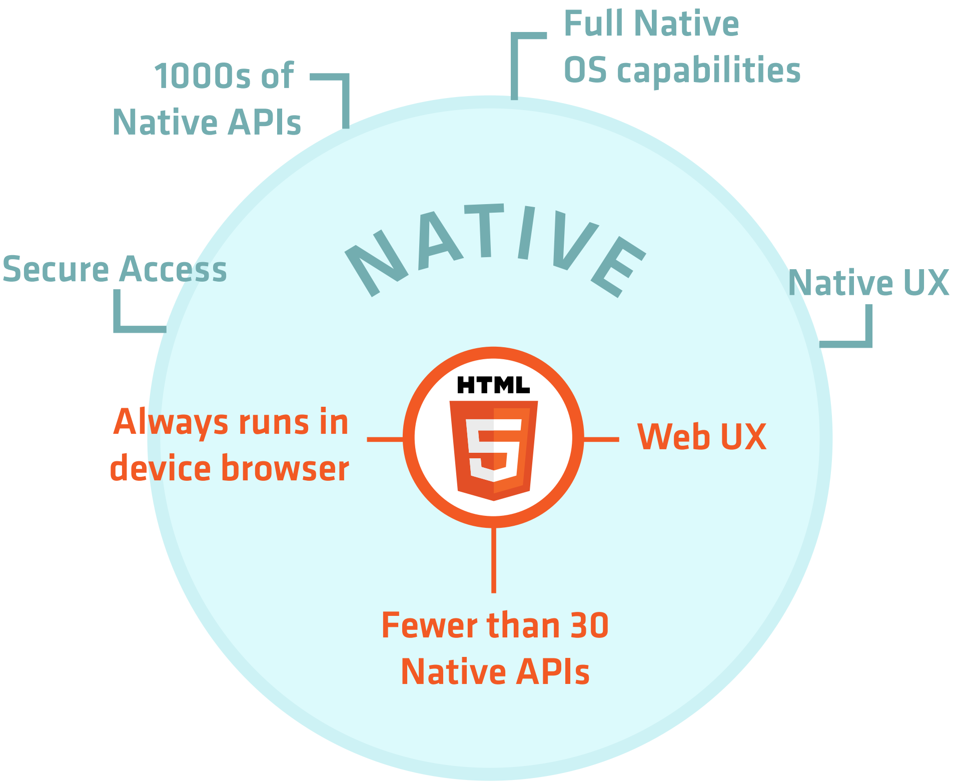
In addition to cost, a successful mobile application requires a beautiful and engaging user experience that leverages the rich features on the device. Natively built applications provide the best user experience. HTML5 applications promise to have parity with native applications while eliminating costs given the "write once – run anywhere" benefit. To better decide upon a mobile strategy, it is key to dive deeper into the pros and cons.

Native and HTML5 Applications Defined

A native mobile app is a device-specific program designed to run on a mobile device and its associated OS. Native apps can come preinstalled on smartphones or tablets, but can also be downloaded from a public and internal/enterprise app stores. Native apps are developed to leverage mobile device capabilities directly from the operating system, such as the camera, geo-location, animation and more. These device-specific functionalities add to the richness of the user experience and are a prime differentiator between native and HTML5 applications. HTML5 is the fifth version of the HTML language and provides the basic building blocks of web pages. It's the first HTML version to support multimedia without plugins. The HTML5 standard was created so web apps can be accessible and used on any device via a browser. HTML5 apps also have the ability for offline access and usage via the application cache, which means working without a network connection is now possible.

Creating a successful mobile app that realizes the promise of "write once – run anywhere" doesn't come from the specification itself but

Native and HTML5 Applications Defined


1000s of Native APIs
Full Native OS capabilities
NATIVE
Secure Access
Native UX
HTML
Always runs in device browser
Web UX
Fewer than 30 Native APIs

from the implementation of the standard by each browser vendor. It's questionable how much incentive Apple and Google have to deliver timely standards-based support in their browsers with the latest and greatest APIs. By its nature, the standard puts at risk traffic to these large commercial app stores and marginalizes the unique capabilities of their devices.

Strategic decision-makers for successful apps should weigh the importance of the following:

1. User experience
2. Performance
3. Monetization
4. Cross platform deployment costs
5. Fragmentation
6. Availability of programming expertise
7. Importance of immediate updates and distribution control
8. Timeliness of new OS innovations
9. Security

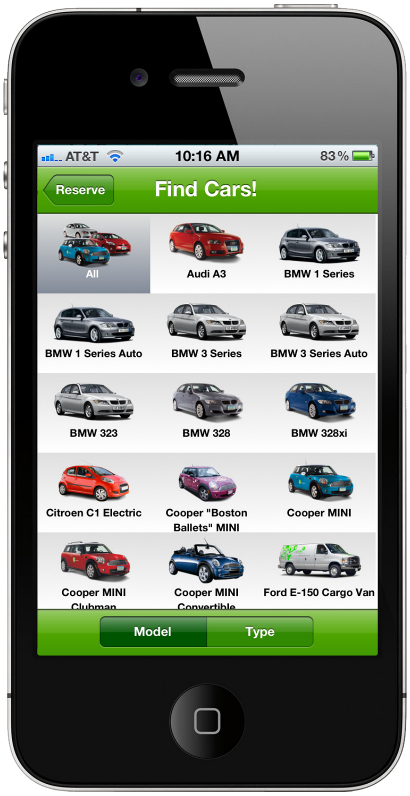
User Experience

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 and more as of late. 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. More importantly, 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.

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 smartphones and tablets, the tolerance for delays from rendering and loading is at an all time low.

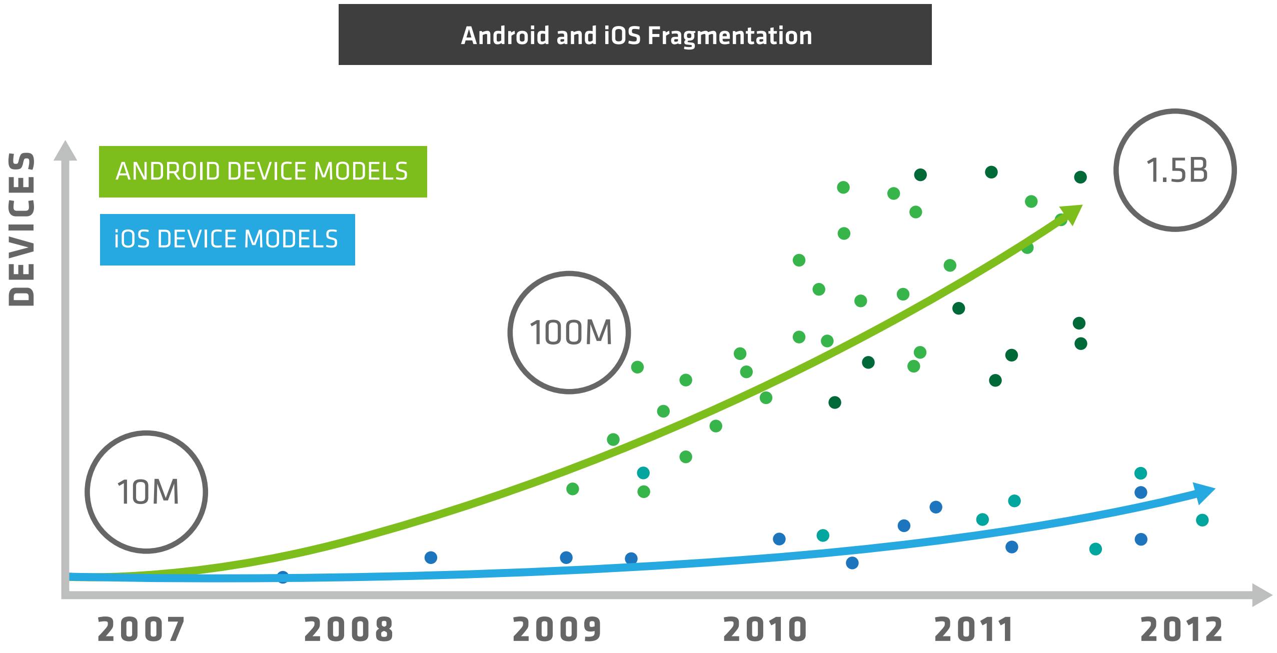
Native applications have set the standard for what a user expects from a performance perspective and reflect the best possible approach to delivering seamless and immediate responses to user actions and requests. This is clear when comparing the performance of loading the application locally to serving 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.



Performance can be slow with many graphics in HTML 5

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. The store is now available in over 155 countries around the world. HTML5 lacks any monetization model similar to Apple's or Google's app stores. More importantly, it has no ecosystem of developers. Although applications can be directly distributed by a developer organization via a browser in HTML5, promotion of those apps to a public or private enterprise is not available without an app store platform.



Cross platform deployment costs

Driven by the desire for a cutting-edge app with beautiful user experience, many developers' 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. 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.

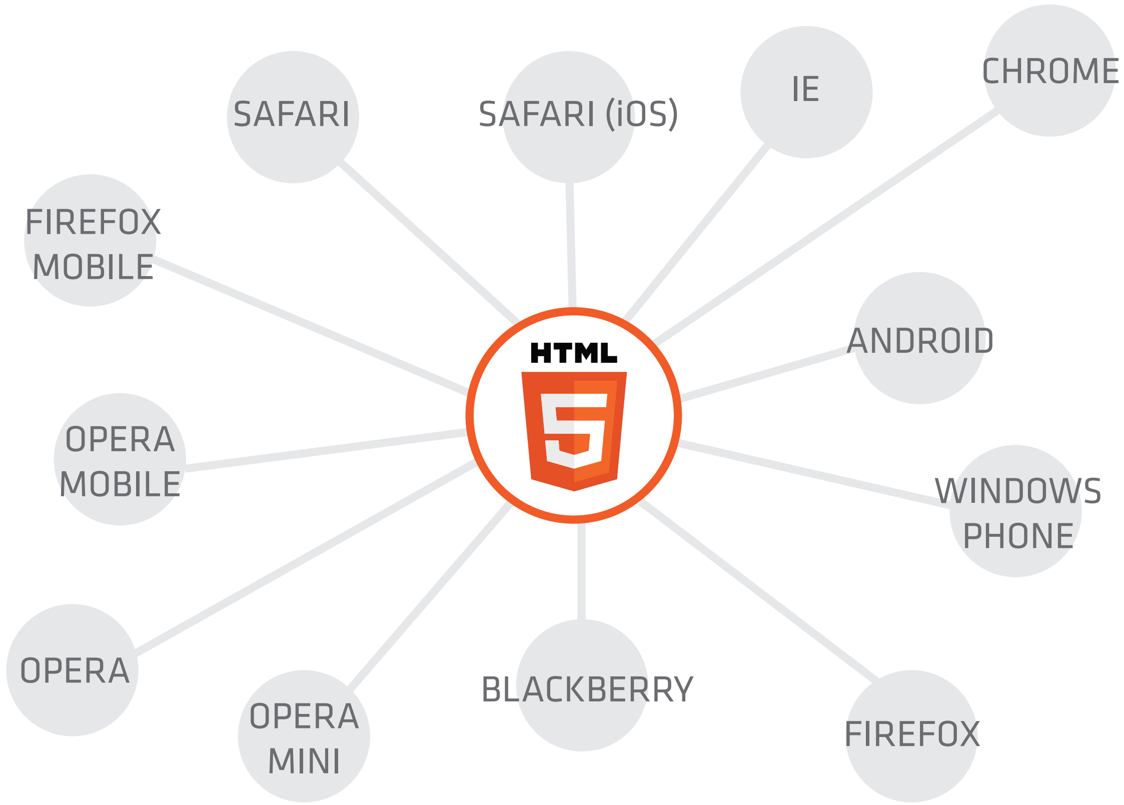
Truly understanding the cost benefits and write-once-run anywhere claim of HTML5 contrasted against a native user experience requires understanding two types of fragmentation challenges. The first is driven by all the various Android variations. The second highlights the lesser-known but equally critical problem of mobile web browser fragmentation.

Fragmentation

With multiple releases of the Android OS over the past few years and over 28 implementations by multiple manufacturers (e.g. HTC, Samsung, Motorola, Kindle, Google itself, Sony and more), 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, which adds to the challenge. Over the past two years, Android has had 21 updates to their browser.

Browser Fragmentation

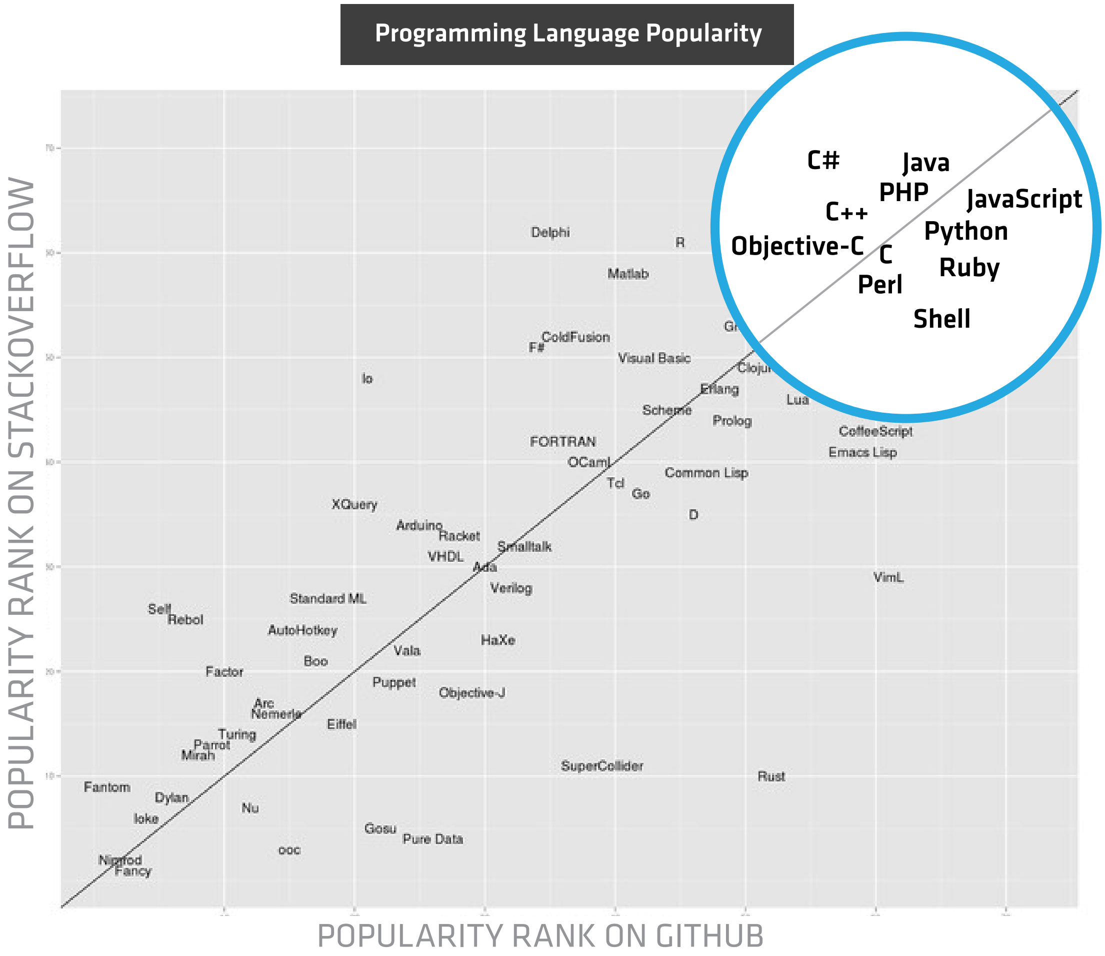


Browser fragmentation and the varied support of HTML5 motivated Facebook to create Ringmark as well. Ringmark was designed to be a testing suite for those mobile web browser capabilities. It helps developers know whether their app can or cannot run on any given mobile browser.

HTML5 as a standard has a long way to go. The fragmented support for HTML5 in different browsers drives the user experience to an even lower common denominator. The only way around this is to have development teams implement multiple web app versions, thus eliminating the cost-benefit argument for multi-platform deployment.

Mobile browser fragmentation rivals the challenge presented by Android device fragmentation. We have seen the browser fragmentation issue play out before in the desktop world. In the desktop era, the main issue was massive browser fragmentation, but with only one operating system. What makes the mobile browser issue even more challenging is the additional variable of multiple operating systems. Considering all these obstacles, "write once, run anywhere" quickly becomes "write once, optimize everywhere."

Availability of programming expertise
An important factor in any development project and especially a mobile app project is having access to developers with the right skills. It’s widely believed that there are far more HTML and JavaScript than Objective-C developers. Enterprises have spent the last 15 years building up internal developer expertise in web technologies. The ranking of projects in GitHub and the number of questions in StackOverflow support this belief. In a recent study, JavaScript and HTML5 held 2 of the top 4 rankings of 18



different programming and markup language frameworks. The web development community by most estimates represents one of the largest developer segments today. As the demand for traditional websites declines, any mobile developer project that requires HTML and JavaScript skills has a greater pool of developers from which to choose when compared to the pool of Objective-C programmers. Due to higher demand, Objective-C developer skills usually cost more as well.

Immediate updates and distribution control

One of the biggest benefits to IT organizations developing mobile applications in HTML5 is the ability to deploy those apps and updates directly to the user community via the browser. No third party or extra step is needed for distribution. Complete control is governed by the developing organization. This can shorten upgrade, change and release cycles.

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

Native vs. HTML5 Comparisons

	Native	HTML5
Rich user experience	●	
Performance	●	
App store monetization	●	
Cross platform deployment costs	HIGH	LOW
Fragmentation challenges	●	●
Availability of programming expertise		●
Immediate updates and distribution control		●
Timely access to new OS innovations	●	
Security	●	

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 OSes.

Security

HTML5 presents unique security risks when compared with native apps. In HTML5 based apps, more of the logic and data can now be executed and presented in the HTML format on the mobile client itself (this is also true of most hybrid “container” based apps). Hybrid container-based apps that run in a native shell also store and present their data in the very open HTML format. Whether the apps are HTML5 or hybrid in nature, there are four key security concepts to evaluate for your organization.

1. App source code
2. Data at rest
3. Data in transit
4. URL security issues

Application source code for HTML5 is freely available on the mobile device. Anyone using the application can follow a simple set of steps (e.g., routing the mobile URL to a desktop) to examine the source code. With the right tools, a developer could then insert malicious code into the application and it would execute along with the rest of the app or run in the background depending the nature of the injected code. With compiled native applications, it is much more difficult for anyone outside the application developer to access this source code.

HTML5 now offers the ability to cache data within the browser. Since both native and HTML5 applications manage data on the device, this data should be appropriately secured and encrypted while at rest. iOS and Android deliver data security for native apps in ways above and beyond HTML5 apps. First, Objective-C and Java API data encryption calls are readily available for native applications. There is no equivalent capability in the HTML5 world. Secondly, MDM and MAM vendors provide another level of security for native apps. None of these vendors can affect data on the browser's cache so therefore, they cannot secure or manage that data.

Secure Socket Layer (SSL) is the most common way that both native and HTML5 apps secure their data in transit. For additional security, many organizations also implement Virtual Private Networks (VPNs). While VPNs can support both native and HTML5 traffic, they typically are not implemented for HTML5 due to significant performance overhead. When it comes to data in transit, HTML5 forces a tradeoff between performance and greater security.

URL security vulnerabilities are unique to web apps. Techniques such as cross-site scripting (XSS), cookie manipulation, and SQL injection can be used to gather sensitive data from the back end servers as well as from the mobile device itself. These attacks are not possible in the native app world.

When does using HTML5 make sense?
Given the above advantages and disadvantages, when does it make sense to use HTML5? The short answer: When the information is always updating; you don't want Apple or Google as intermediaries, or if the user doesn't want to download an application.

Today's mobile users care deeply about having a rich and interactive mobile application experience. This native experience is driven by a personal connection with the device and how it has become a part of our everyday lives.

Today's smartphone and tablet have become the modern day encyclopedia, dictionary, navigator, camera, personal assistant and communicator for our workplace and beyond. We haven't yet begun to leverage such capabilities as geographic location, camera, voice-activated search, and Bluetooth to their full potential. Therefore, it stands to reason that the class of applications best suited for HTML5 would be those with more traditional forms-based requirements or those built for access to and consumption of information. An example of this would be the mobilizing of a website where rich and alphanumeric data can easily be input, requested and retrieved from a web and app server. Other examples of this that are heavily internet-reliant are information-based applications such as Wikipedia, Facebook, Twitter, Yelp, and OpenTable.

Creating forms-based applications or mobilizing websites leverages the device more as an extension of the desktop, much like in the traditional client-based computer model. This minimizes the need for using the smartphone or tablet's computing power and their frequently improving device capabilities whether they are online or offline.

A Smoother Path to Successful App Development

What if you could leverage the low cost of HTML5 development and deploy on multiple platforms? What if you could not only build rich, immersive, high performing native applications but also produce HTML5 apps from that same environment when needed?

Introducing Titanium

At Appcelerator, we build Titanium Studio and our Cloud services to deliver cross-platform (iOS, Android and mobile web) applications at a fraction of the cost without sacrificing a high performing, rich mobile experience. A single team with JavaScript experience can build an app for iOS, Android, Blackberry, Windows (coming soon), and HTML5, with up to 90% code reuse. The result is significantly lower cost and faster time to market. No Objective-C or Java skills are required. We do this by delivering a mobile platform that allows developers to create apps built in JavaScript with access to over 5,000 APIs and parity across platforms. Appcelerator's development environment also includes a broad set of cloud-based services for push, email, photo sharing and more. No knowledge of IaaS, infrastructure provisioning or server programming languages such as Ruby or PHP is required. If there are any native APIs we don't cover, our ecosystem of over 270+ partners in our marketplace will.

Many customers, such as NBC, Zipcar, PwC and Safeguard, have delivered rich, native applications with Appcelerator while lowering their cost of development and time to market. If you are interested in realizing the same benefits for your mobile strategy, please contact us.

Find out more at **appcelerator.com**

Appcelerator is the #1 Mobile Cloud Platform

Appcelerator is the leading enterprise-grade, cross-device mobile development solution on the market today, with over 350,000 mobile developers worldwide and 40,000+ cloud-connected applications used on 50 million devices every day.

The company's flagship offering, Appcelerator Titanium, is the only platform to enable fully native, cross-device development from a single codebase for iOS, Android, Blackberry, and HTML5 mobile web apps. Appcelerator's Cloud Services enable easy integration of compelling features such as push notifications, user logins, and photo uploads.

Appcelerator's customers include NBC, Zipcar, eBay, PayPal, and LEGOLAND. These companies develop their mobile applications on Appcelerator products so they can decrease time-to-market and development costs, increase customer adoption and revenues, and enjoy greater flexibility and control. **Download Titanium for free at www.appcelerator.com.**