

**Mobile Application Development**  
The Tools of the Trade

**Abstract**

The landscape of mobile application development has had its milestones in history and is full of various platforms that have progressed in industry to where it is today. What some might find hard to believe, the mobile application industry has actually its roots back as far as 1996, and rather up until recently, it had a monolith of companies directing the development of this new and exciting technology. (Dediu) The mobile world as we now know it consists of the heavy-hitters, Android and iOS, with a few other notable camps like Windows Phone and BlackBerry. The increasing breadth of these camps has left mobile developers in a fragmented state, where developing native applications has become quite cumbersome for industry to handle. Developers must now trade off developing for one OS or the other, or have the money, time, and skills necessary to port their applications to the other OSs as well. However, there is a variety of options that these application designers can take part in to maximize the reach of their code, while maintaining high efficiency when developing mobile applications.

**1. Introduction**

Mobile application development has been a hot issue in today's consumer industry, and has driven many enterprises and producers to begin to develop for the mobile market that now takes up a considerable share of all devices on the market, and also a considerable amount of one's interaction with technology. As such, and as with many industries and products in the modern

consumer world, there are various products that are available on the market. The current mobile OS marketplace is a staggered between the various camps that mainly consists of Apple's iOS, Google's Android, and BlackBerry.

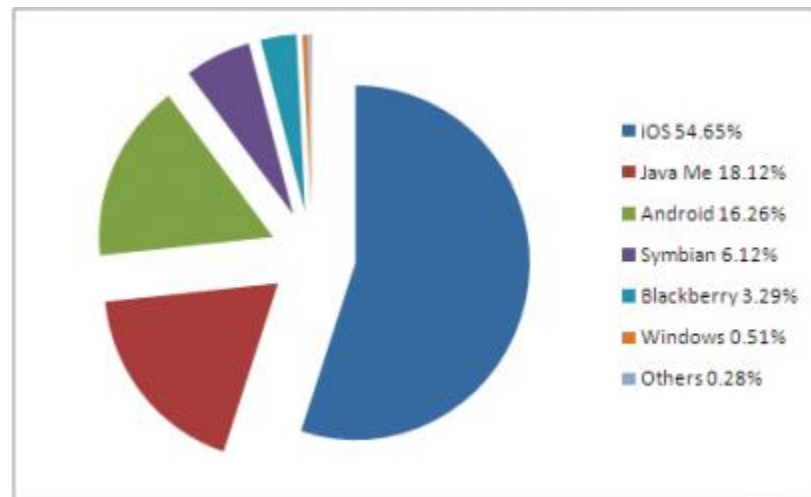


Figure 1: The market share of all mobile operating systems as of 2012 (Asthana)

Little do many people tend to realize, that the mobile phone market actually extends itself back more than a dozen years, and has seen a lot more than just these handful of companies that dominate it today.

### 1.1 History of Mobile Applications

There has been a large range of mobile operating systems that have lived in the mobile ecosystem since 1996. PalmOS was one of the earliest founders within this system, followed soon after by Windows Mobile in 1997 and Java Micro Edition (Java ME) in 1998. (Dediu) In the early 2000s emerged Java 2 Micro Edition (J2ME), which as Luis Corral and Alberto Sillitti state in their article, “Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm,” started “opening ways to develop software for different target devices, allowing the creation of a software system in any mobile device capable to

execute the Java framework.” It wasn’t soon after that, that a large range of competitors joined in the mobile arms race, and that “the use of J2ME decreased while major platform-specific software development kits (SDK) grew.” (Corral)

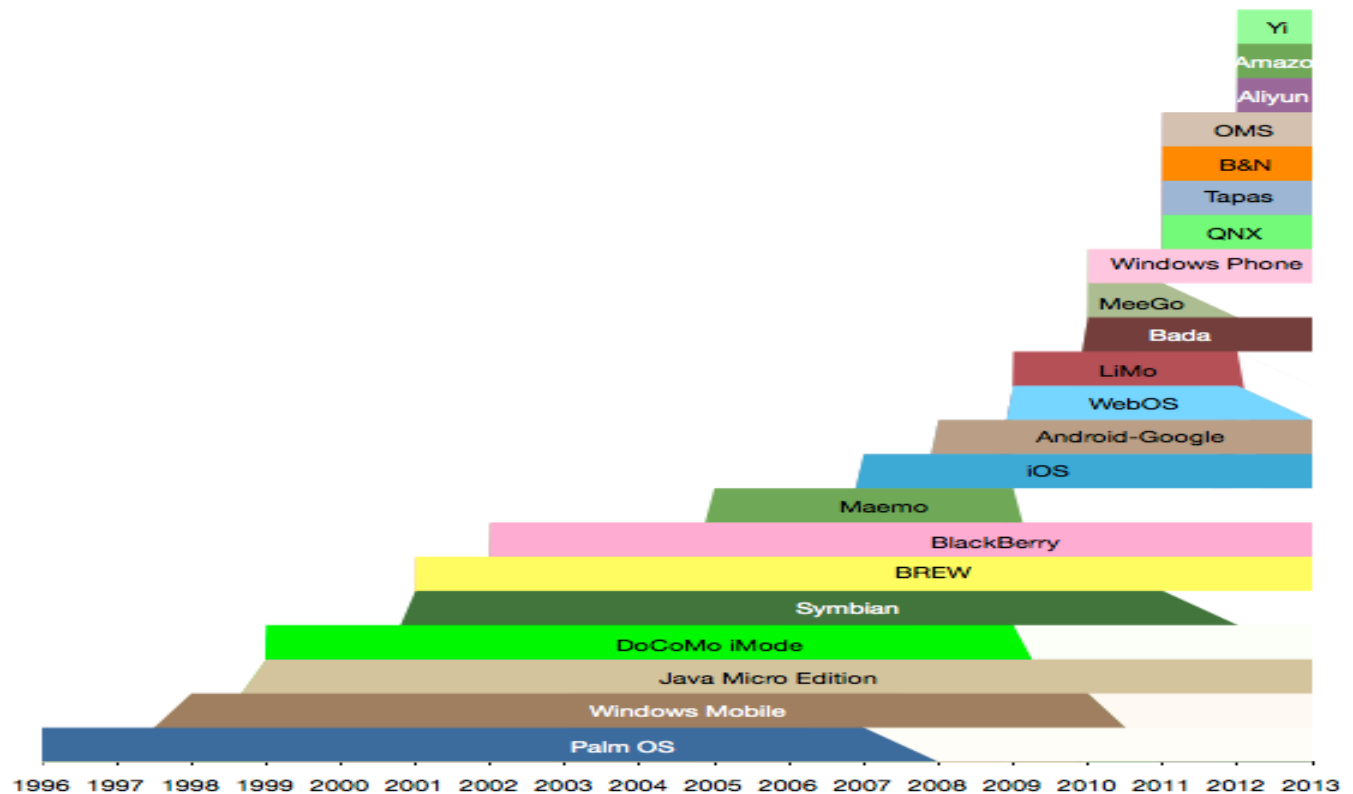


Figure 2: The history of mobile operating systems, by platform, as per public announcements of the beginning and end of platform. (Dediu)

Again, the three notable OSs that we most see today are BlackBerry, iOS, and Android. To begin to understand where the development world sits in relation to these three groups, it is worthwhile in looking into each operating system’s recent past.

BlackBerry had its humble beginnings since 2001, when the first BlackBerry devices began to hit the market. BlackBerry’s first device, in 2001, was based entirely on J2ME. In 2002, they introduced their first development platform where programmers could develop for BlackBerry devices with J2ME. (RIM) Since then, BlackBerry has expanded its tools to allow

for a larger base of developers to program for the platform, allowing for developers to develop with C/C++/Qt, or a HTML/CSS/JavaScript platform, or Java. (BlackBerry Developer)

Although Apple has been one of the longest standing companies in building computers and creating operation systems, it wasn't until 2007 that released their first iPhone, which original had little to no applications for it, other than the one Apple would develop itself. The original intent of the iPhone was to allow people to develop applications on the web, and the phone's mobile browser would be the link to those applications. Effectively, no third-party applications would be allowed on the iPhone. (Markoff) It was in 2008 that Apple introduced its first SDK, and also the App Store, where developers could now sell their mobile apps to the general public. Developers could begin to create custom applications using Apple's IDE, Xcode, and write their applications using the Cocoa Touch framework and the Objective-C programming language (Rogers).

Finally, Google released the Android OS in 2007 as an open source mobile platform. (Hansen) The platform is mainly developed in Java, on platforms such as Eclipse or Sublime, but lower level C/C++ code can also be executed on the device. Due to Java's popularity as a programming language, it allows many developers to access the variety of libraries that Google has created and included in the Android APIs. (Hansen)

## **1.2 The Mobile Development Problem**

With so many factions within the mobile landscape, and the growing popularity of these groups to produce their own proprietary SDKs, one can begin to realize the variety of problems and questions are produced from this. Erick Vargas, a software architect at Avantica Technologies highlights these, as he explains how slow “development speed and [large] cost,” developing

“requires knowledge” to use each environment, variations in “user experience,” the lack of mature “tooling” to build apps, and a great lack of “cross-platform support” all plague the land of fragmented mobile app development. (Vargus) Developing native applications becomes difficult and time consuming, and can cause a great deal of additional problems.

Native development does, however, have many advantages to it, which include access to the deep levels of the phones hardware through API access, features to make a flawless user experiences, and access to a grand archive of tutorials and support for each platform. In addition to the advantages that native development offers, Ron Vetter, an editor for IEEE’s *Computing Now*, highlights in his article how native applications have “the ability to reach hundreds of millions of customers simply by uploading your app to a store.” (Vetter) Another big area that native applications usually far surpass their mobile web counterparts, is in the area of user experience. Although a developer can develop a very similar design that will be used cross platform, the properties of these designs, such as those used in CSS, don’t perfectly port from platform to platform. Andrew Charland and Brian Leroux, both writers for Communications of the ACM, point out in their article that

The Web platform is consistent, for the most part, but the number of built-in or SDK-included controls is limited. You have to roll your own. Sometimes the differences among browsers can cause pain, but – at least in the modern smart-phone world – most devices sport the very capable WebKit rendering engine, and only small differences prevail... Unfortunately for the Web [though], those small differences are becoming a big deal. For example, on iOS, the CSS position property does not properly support a value of “fixed.”

This, as Charland and Leroux exclaim has caused some to jump ship, and develop native code, which is why we still to this day see more 2 million applications between the various app stores (Victor H.), and why in 2012 there were roughly 36 million native apps downloaded. (Quilligan) These are just some of the reasons why native applications remain on top of their counterparts.

In the end, however, when one chooses to develop a native application, they run into problems when or if they need to copy the application to another platform. The cost of learning to develop in a new environment if its needed, in addition to the time to develop, test, and push the application into a brand new app store causes considerable problems. Additionally, maintaining several different codes bases means support for an app will inevitably suffer, and poor end product is made. The state that mobile application development finds itself in is highly segmented, profoundly different in various aspects of each OS group, and has no guarantee of fixing any of these problems in the near future. The cost of native applications across various platforms has become almost too great, and thus, it is time to begin to discuss new frameworks that will stretch across the mobile **landscape**.

### **1.3 A Possible Solution**

Currently, there are two popular development approaches when it comes to developing mobile applications, a mobile web-based approach, and developing native applications for each platform. The web-based approach will be briefly looked at, as it currently provides a platform that developers can access a broad spectrum of devices. As Vetter proclaims though, native apps will still remain popular for some time. Although native and mobile web applications are the dominant forces in the mobile development world right now, Vetter points out that many developers will begin to solely move towards a more cross-platform friendly development approach, one other than a mobile web solution. (Vetter) That approach, and quite possibly the best solution towards developing across the highly fragmented mobile market, is known currently as the “hybrid” approach, which Vetter, along with many others say will see a “natural

migration” towards for “developing cross-platform code that can run in a device-independent” applications. (Vetter)

There are various reasons that this ‘hybrid’ approach towards developing applications might be seen as the up-and-coming trend in which mobile apps are developed. The largest of these reasons, is that it can leverage native APIs for each language, which allows developers access to more powerful features of each device. In his article, Vargas examines this approach, stating that it “essentially means that they take advantage of the cross-compatibility of Web technologies such as HTML5, CSS, and JavaScript and use the native component to leverage device-specific features such as hardware sensor or the camera.” (Vargas) No longer will developers be restricted to developing multiple code bases to achieve the desirable end product for the multiple platforms, but rather, build one code base to access them all. Developing these hybrid applications, although not a popular approach in mobile development circles right now, might be the solution that many stakeholders in the mobile industry have been looking for.

## Works Cited

- BlackBerry Developer. "Developer Platform Choice." BlackBerry.com. Oct 2013.  
[http://developer.blackberry.com/develop/platform\\_choice/index.html](http://developer.blackberry.com/develop/platform_choice/index.html)
- Dediu, Horace. "The proliferation of mobile platforms." Asymco. Sept. 4 2011. Sept. 2013.  
[www.asymco.com/2011/09/04/the-proliferation-of-mobile-platforms-continues/](http://www.asymco.com/2011/09/04/the-proliferation-of-mobile-platforms-continues/)
- Corral, Luis. Sillitti, Albert. "Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm." ACM. Oct. 2011. Aug. 2013
- Asthana, Ankit. "IOS 5, Android 4.0 and Windows 8 – A Review." IEEE Beacon. Mar. 2012. Sept. 2013. <http://ieeabeacon.com/>
- Vargas, Erick. "Hybrid or Native Mobile App Development: Six Key Considerations." Sand Hill. Jun. 26 2013. Aug 2013. <http://sandhil.com/article/hybrid-or-native-mobile-app-development-six-key-considerations/>
- Vetter, Ron. "Challenges and Opportunities in Mobile Web and App Development." Computing Now. Nov. 2011. Aug. 2013  
[www.computer.org.ezproxy.csbsju.edu/portal/web/computingnow/archive/november2011](http://www.computer.org.ezproxy.csbsju.edu/portal/web/computingnow/archive/november2011)
- Quilligan, Aiden. "HTML5 vs. Native Mobile Apps: Myths and Misconceptions." Forbes. Jan. 1 2013. Aug. 2013. [www.forbes.com/sites/ciocentral/2013/01/23/html5-vs-native-apps-myths-and-misconceptions/print/](http://www.forbes.com/sites/ciocentral/2013/01/23/html5-vs-native-apps-myths-and-misconceptions/print/)
- Charland, Andre. Leroux, Brian. "Mobile Application Development: Web vs. Native." Communications of the ACM. May. 2011. Aug 2013.  
<http://cacm.acm.org/magazines/2011/5/107700-mobile-application-development/fulltext>



RIM. "Rim History." BlackBerry.com. Date Unknown Oct. 2013

[http://www.blackberry.com/select/get\\_the\\_facts/pdfs/rim/rim\\_history.pdf](http://www.blackberry.com/select/get_the_facts/pdfs/rim/rim_history.pdf)

Rogers, Michael P. "There and Back Again: Leveraging iOS Development on Mac OS X."

Consortium for Computing Sciences in Colleges. 2011. Aug. 2013.

<http://dl.acm.org/citation.cfm?id=1961613>

H., Victor. "Android's Google Play beats App Store with over 1 million apps, now officially largest." PhoneArena. Jul. 24 2013. Oct. 2013.

[http://www.phonearena.com/news/Androids-Google-Play-beats-App-Store-with-over-1-million-apps-now-officially-largest\\_id45680](http://www.phonearena.com/news/Androids-Google-Play-beats-App-Store-with-over-1-million-apps-now-officially-largest_id45680)

Markoff, John. "Phone Shows Apple's Impact on Consumer Products." NYTimes. Jan. 11 2007.

Oct. 1 2013. [http://www.nytimes.com/2007/01/11/technology/11cnd-apple.html?\\_r=2&oref=slogin&](http://www.nytimes.com/2007/01/11/technology/11cnd-apple.html?_r=2&oref=slogin&)