HTML 5 in Mobile Devices – Drivers and Restraints

Antero Juntunen
Aalto University
antero.juntunen@aalto.fi

Eetu Jalonen
Aalto University
eetu.jalonen@aalto.fi

Sakari Luukkainen
Aalto University
sakari.luukkainen@aalto.fi

Abstract

Application stores have played a crucial role in the proliferation of applications for smartphones and other mobile devices. However, web-based mobile applications are challenging the application store model by allowing developers to directly reach the end users. These web-based applications are enhanced by the HTML5 standard, which provides additional capabilities for the use of developers and brings the performance of mobile web applications closer to that of native applications. In this paper, we analyze the potential of HTML5 and identify drivers and restraints that affect the future of the technology.

1. Introduction

Today's mobile phone landscape is increasingly dominated by smartphones with advanced computing capabilities and features. In the second quarter of 2011, smartphone shipments exceeded the shipments of feature phones (normal phones) for the first time in Western Europe [29]. In the United States, the number of smartphone users exceeded the number of feature phone users in May 2012, with two thirds of new users choosing smartphones [34]. This increase in the technical capabilities of mobile phones has coincided with a proliferation of applications available on these devices. In fact, the main selling point of mobile phones has changed from the hardware capabilities of the devices to the applications they can run. A significant reason for the abundance of mobile applications has been the emergence of application stores. Application stores have simplified the process of finding and installing the applications for the end users, increasing the demand for mobile applications. In addition, a more open policy by device manufacturers has allowed small developers and hobbyists to develop and publish their applications for mobile devices, thus increasing the supply of mobile applications.

The application store model works with native mobile applications, which are downloaded into the user's mobile phone and stored and executed locally. Native applications can fully use the capabilities of the mobile devices, require no Internet connectivity, and can be distributed through the application store, leading to increased visibility among the end users. However, there are certain problems with the native application model. First, the mobile space is fragmented and native applications are tied to a specific platform (such as Apple's iOS or Google's Android). As a result, a mobile developer targeting a larger user base has to create applications for many different operating systems (OS), significantly increasing the time and resources required in application development. Moreover, fragmentation is an issue even within a single OS, as new versions of an OS may not support old applications. Second, application developers are tied to the revenue sharing terms set by the application store provider. These terms (typically a 70/30 % split for the developer) may not be suitable for all developers and all applications. Third, while the performance of mobile devices has increased, native applications are still limited by the constraints of the devices such as limited computing resources and battery power.

These issues can be partially addressed by using web-based mobile applications instead of native applications. OS fragmentation can be addressed if the user can access the web-based application through a standard mobile browser, forgoing the need for the developer to tailor the application to each platform. Web-based applications can also bypass the revenue sharing constraints of application stores, with the developer establishing a direct billing relationship with the end user. In addition, the Mobile Cloud Computing (MCC) model can help address the hardware limitations of mobile devices by running a part of the computation in the cloud. MCC can be used with pure web applications [31] or with native applications, the processing of which is partly offloaded into the cloud [30]. MCC is also sometimes used to describe a mobile device cloud [38], in which mobile devices form a cloud by pooling their resources.

Web-based mobile applications are enhanced by the HTML5 standard, which provides some of the features

from traditional desktop-style software to the browser. HTML5 is currently being developed by two standards bodies, the Worldwide Web Consortium [46] and the Web Hypertext Application Technology Working Group (WHATWG) [28]. Both standards are still in a draft stage, with the WHATWG standard being the more fast-changing or fluid of the two. Support for HTML5 is predicted to grow from 336 million mobile phones with HTML5 browsers sold in 2011 to one billion sold devices in 2013 [41], while further estimates put the number of mobile phones with HTML5 browsers in 2016 to 2.1 billion devices [1].

In this paper, we used exploratory research to examine the HTML5 technology and evaluate its potential. Our research goal was to identify drivers and restraints that affect the technology evolution of HTML5.

We base our work on general literature of HTML5, which was chosen by identifying academic articles focusing on HTML5 and mobile applications from databases ScienceDirect, ACM Digital Library, ProQuest ABI/Inform Complete, IEEE Xplore Digital Library, JSTOR, EBSCO Business Source Complete, and Google Scholar. Due to the scarcity of academic articles addressing HTML5 and mobile applications, we supplemented this material with a more general web search. In order to avoid becoming overwhelmed with the available data, we used our own research framework as a research focus [16], basing the research framework on relevant existing business literature. We analyzed the literature using the research framework, which allowed us to classify the data within the different dimensions of the framework. These dimensions were then compared with the research target, which produced the drivers and restraints of HTML5 under each dimension.

The remainder of the paper is structured as follows: Section 2 describes the framework and its theoretical background and section 3 provides an overview of the HTML5 technology. We analyze HTML5 using the framework in section 4, summarize and discuss the results in section 5, and give our conclusions in section 6.

2. Theoretical background

2.1. Technology evolution

Industries evolve through a sequential development of technology cycles. These cycles are initiated by technological discontinuities [4] that emerge through scientific advance or through a unique convergence of existing complementary technologies, which eventually substitutes existing products [3]. At some point, diminishing returns begin to surface as the technologies start to reach their limits and new, substitute technologies emerge [4]. The threat of substitute products depends on several factors including relative price, new features and added value, performance, and switching costs [37].

The success of many new entrants has lead to coining a phenomenon called the "attackers' advantage". This term refers to those new entrants who are better than the incumbents in developing and commercializing emerging technologies because the new entrants are smaller in size, have limited path-dependent history, and are not commitment to the value networks of the previous technology [9,24]. New entrants can be successful despite the incumbents' greater resources and experience with the existing technology. However, industries have barriers to entry, which protect the profit levels of the incumbents and hinder the market entry of new entrants. Barriers to entry are unique to each industry and include factors such as cost advantage, economies of scale, brand identity, switching costs, capital requirements, learning curve, regulation, access to inputs or distribution, and proprietary products [37].

Christensen [10] states that the incumbents improve their technological performance on an existing trajectory and finally exceed even the most demanding customers' needs. Simultaneously, new, more cost-effective technologies are developed by new entrants, first for the needs of the customers of other industries. These new technologies start to increase their market share among less-demanding customer segments and will later enter the existing mainstream market. Christensen refers to these technologies and the related innovations as 'disruptive', which can be seen as an extension to the concept of technological discontinuity. Similar to technological discontinuities, disruptive innovations significantly change the current market structures, customer usage patterns, and value propositions. If the markets of disruptive technologies develop fast, new entrants gain advantages due to economies of scale. If the development is slower, the incumbents will have more time to react on the new entrants.

Rogers [39] considers the most important factor affecting innovation diffusion to be the relative advantage (price and performance) over competing technology substitutes. Among other factors highlighted by Rogers, trialability relates to how easily the product can be experimented with. Easy trialability for the early adopters enhances the diffusion of an innovation. This is also supported by Gaynor [27], who emphasizes the importance of experimentation, especially in times of great market uncertainty and

Thomke [45], who stresses the role of experimentation with new technologies.

The product platform is a concept that allows a company to build a series of related products around a set of common components [32]. An industry platform differs from a product platform in that these components are likely to come from different companies called complementors and that the industry platform has relatively little value to users without these complements [26]. Eisenmann, Parker, and Van Alstyne [17] define platforms as products or services that bring together two distinct groups of users in two-sided markets. They consider four different roles in platform-mediated networks: demand-side platform users (end users), supply-side platform users (complementors), platform providers (users' primary point of contact with the platform), and platform sponsors (who determine access to platform) [18]. Platform openness can differ for each role, leading to varying strategies for managing openness.

2.2. Research framework

Based on the above literature review on technology evolution, we created the following framework for the empirical part of this study. The most important factors affecting the technology evolution of HTML5 are summarized in Table 1. The 'Added value' category emphasizes the value of the HTML5 technology over existing solutions and focuses on the viewpoints of the main actors – end users and application developers. Relevant theoretical concepts in this dimension are added value [37] and relative advantage [39]. 'Ease of experimentation' concentrates on the ability of developers to adopt HTML5 and to use the technology to create new applications and services. Relevant theoretical concepts include trialability [39] and experimentation [27,45].

Table 1. Research framework

Dimension	Meaning
Added value	The relative advantage over existing technologies
Ease of experimentation	The threshold of end users or third parties (developers) to experiment with new services
Complementary technologies	The interdependence between complementary technologies
Incumbent role	The product strategy of existing players
Technological performance	The performance or capability of the technology

The category 'Complementary technologies' examines supporting technologies, which can be especially important in the emergence of technological discontinuities [3] and in the case of platforms [18,26]. 'Incumbent role' focuses on the roles of major incumbent actors, including device manufacturers, mobile OS providers and mobile network operators. This dimension can be especially relevant when considering the effect of new entrants [24] on the market, particularly in the case of disruptive innovations [10]. 'Technological performance' compares the performance of HTML5 to substitutes, which relates to the concept of a sufficient level of performance [37,40]. The chosen categories were considered especially useful for a developing technology and the categories arose from both the literature on technology evolution and HTML5.

3. HTML5 overview

HTML5 is both an evolution of the previous HTML version, but also a response to the change in the way that content is used and viewed on the web. Application developers providing multimedia-rich and interactive services have previously relied on solutions provided by third parties, primarily Adobe Flash, and to a lesser extent, Microsoft Silverlight. HTML5 standardizes some of the core aspects of the previously mentioned technologies, allowing the browser to directly provide those features without the need for additional drivers or plug-ins. These new capabilities also bring HTML5-based solutions closer to the traditional realm of desktop or native applications, thus lowering the barrier between the traditional and web-based solutions [43].

Although HTML5 is a standard itself, it is also used as a blanket term for other related technologies such as Cascading Style Sheets version 3 (CSS3) and JavaScript (JS). Roughly speaking, HTML5 is used for content, CSS3 for presentation and JS for defining the behavior of the other two.

Table 2 contains a selection of the most relevant features when considering using HTML5 on mobile devices.

Mobile applications built on HTML5 usually rely on different frameworks in cutting down development time and cost. In general, these frameworks can be roughly divided by the input they take and the end product they produce. Basic mobile HTML5 frameworks such as LungoJS, jqMobi, Sencha, Jo and others use HTML5, CSS3, and JavaScript, but also offer added library functions that help in the development of the application. The end product is a page, site, application, or any other target the developers were aiming for that is then usually run inside a web browser on the targeted platforms.

Table 2. HTML5 features [47]

Feature	Comment
Multimedia	*<video>* and *<audio>* tags, support for both media formats without 3rd party plug-ins.
Hardware integration	Access to mobile device features such as *GPS, accelerometer, microphone, camera,* etc.
Device adaptation	Modifying the page based on the device's *screen size, keyboard type,* etc.
User interactions	Support for *touch* and *speech* interaction, also *haptic feedback* (vibration).
Data storage	Data can be stored *offline* within the *browser* or on the underlying *filesystem*, though there is also a simple *key-value* based *database*.
Network	Cross-domain requests with *XMLHttpRequest. Server-Sent Events* or *Push Events* for sending data to HTML5 applications even when the page is not active on the browser. *WebSocket* [21] allows for more efficient data transfer, based on a TCP stream (two-way).
Widgets	HTML5 applications can be run off-line with the *ApplicationCache* feature, but also shared as archive files that can be unpacked and deployed in the same way as more traditional applications as per the *W3C Widgets* family of specifications.

PhoneGap is similar to the previously mentioned frameworks, but instead of running applications within the browser of the device, PhoneGap [36] outputs a stand-alone application for the selected and supported mobile platforms. The source is HTML5 and it can include parts of other HTML5 frameworks or JavaScript libraries. The end result is an application that runs inside browser view that in turn runs inside the aforementioned PhoneGap stand-alone application or container.

Titanium SDK [5] is another step towards native applications from PhoneGap, as its only input is JavaScript that is then cross-compiled to the selected mobile platforms. The output is platform-specific code and the end result is in a sense a true native application. The limitation of this approach is that the framework is restricted to the libraries provided by Appcelerator.

Figure 1 displays the inputs and outputs of the different frameworks, and Figure 2 shows how these frameworks fit into the scale between HTML5 applications and native applications.
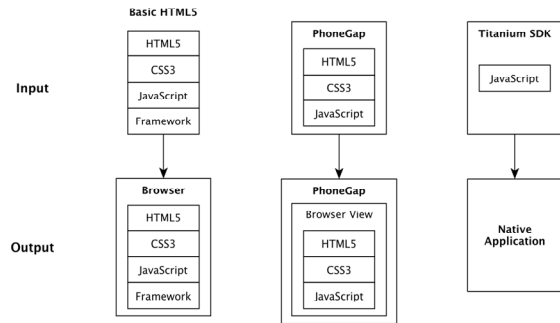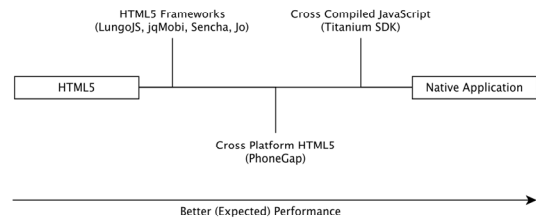

Figure 1. HTML5 frameworks input/output

Figure 2. HTML5-Native Application Scale

4. Analysis

In this section, we apply the research framework of Table 1 to HTML5. Section 4.1 examines the added value provided by HTML5 to both developers and end users, while section 4.2 evaluates the ease of experimentation with HTML5 from the point of view of third-party application developers. In section 4.3, we analyze the role of complementary technologies, and section 4.4 examines the role of incumbent actors such as application store providers. Finally, section 4.5 examines the technological performance of HTML5 applications.

4.1. Added value

End users can benefit from web-based applications and HTML5 in several ways. First, the users do not need to manually install or update their applications, as is the case with native applications. Because web applications use the mobile browser as the run-time environment, the user always has access to the newest version of the application without explicit installation

or update [42]. Second, users who have multiple devices such as mobile phones, tablets, and laptops on several platforms may have to use different applications on different devices. Web-based applications can offer a unified user experience regardless of the device or platform used.

Third, HTML5 provides both platform-specific and custom user interfaces depending on the device in question or the needs of the application. Applications developed directly for a certain phone model by leveraging its native programming interfaces and programming model can have a better usability than traditional web applications. Native applications can take a stronger advantage of the user interface controls such as certain gestures on touch screen devices and the placement of control buttons around a display in keypad-operated mobile phones. In addition, users may not be comfortable in using and installing applications the user interface of which greatly differs from the rest of the mobile device, thus creating confusion. However, HTML5 applications can offer better interactivity for the user and more closely mimic the behavior of native applications. For example, HTML5 applications can be designed not to look like web pages by disabling the traditional web page elements such as tabs, URLs, and back/forward buttons [44].

Fourth, web applications normally require a network connection to function properly, but HTML5 provides offline data caching, which enables applications to be developed to function at least partially even when the connection is unavailable. In addition, the application can also function completely off-line and to only exchange data with the host server when required.

In addition to end users, developers can benefit from HTML5 and web-based applications in multiple ways. First, web applications can help overcome the fragmentation of the mobile space. Application developers only need to develop one web application that will be used through the browser rather than provide applications for each platform they want to target. This cross-platform development not only significantly reduces application development costs, but precludes the need to have the programming expertise necessary to develop an application for each platform. One major driver for the development of web technologies has always been the interoperability of different operating systems and computer architectures, thus making it a natural choice for developing applications for the heterogeneous mobile environment. Increasing diversification and uncertainty about the future direction of the mobile operating system market may drive more developers to adapt web technologies in developing their applications.

Second, developers may find it financially lucrative to challenge the revenue sharing terms of the applications store monopolies. With the application store providers typically retaining 30% of the application revenues, developers may wish to bypass the application store and sell the application directly to the end users. This approach has been used by content providers such as Financial Times, who withdrew their application from Apple's AppStore and launched an HTML5 application [14], eventually resulting in increased revenue and more subscribers [40].

Third, for certain applications and developers, web applications can offer more visibility than native applications. Web applications are visible in Internet search results, which can produce a larger audience for the application in question. On the other hand, application stores are a good way to gain visibility for certain applications, in which cases not having access to an application store could be considered a downside. A hybrid or a compromise solution would be to use a framework such as PhoneGap, which allows the developers to wrap their web application in a native application that can be placed in the application store.

4.2. Ease of experimentation

The trialability or ease of experimentation of HTML5 depends on how easy it is for application developers to start using the technology and how HTML5 affects the software development process. First, use of HTML5 builds on existing knowledge with web technologies such as JavaScript, meaning that web developers should be comfortable moving to HTML5. In general, web development is considered more economical and faster than traditional software development as the technologies and tools are usually more user-friendly then their native counterparts. This all leads to a lower threshold for HTML5-based software development and increases the ease of experimentation of the technologies in question.

Second, some of the intrinsic advantages of running applications on the web include the ease of deployment as well as the speed and ease of updating the applications. Application stores are typically vendor-locked to their respective operating systems, which means that developers have to first develop, compile, and submit the application to the store before it can be downloaded by the users. On the other hand, a web application is usually distributed as source code that the browser interprets, resulting in a more dynamic process. Code can be updated on the fly, with users downloading the changes while they are using the application. Thus, the process of software deployment is greatly hastened. For more popular applications, this means having the necessary server hardware and

bandwidth to host your application, instead of relying on an application store's hosting service, a trade-off between the ease of deployment and the ease of upkeep [11].

4.3. Complementary technologies

As was already mentioned in section 3, HTML5 is a blanket term for several related web standards and technologies, and HTML5 together with CSS3 and JavaScript represent the complete package or idea that is HTML5. Underneath the surface there are several related APIs to provide the multitude of functionalities that are currently available on mobile devices, but ultimately it is up to the browser to implement these standards. A listing of how mobile browsers support different HTML5 features is available on the web [23] and no mobile browser offers complete as of June 2012. Adequate browser support is a prerequisite for HTML5 adoption and the dichotomy with the platform vendors is that of native applications versus web applications. The platform vendors are striving for a strong ecosystem around their respective platforms, but the ultimately correct path is still unclear. The level of support they want to provide for the two options, native applications and browsers/HTML5, is a balancing act.

4.4. Incumbent role

The current native application market is mainly controlled by the platform vendors, the largest two being Google and Apple [25]. Their respective native application stores are the primary way in which users on these platforms find, download, and update their applications. As native applications already have access to the hardware features of the devices through a multitude of APIs, platform vendors also provide help and documentation on using these features as well as the necessary software tools to develop the applications. The only limiting factor for developers with this model is the vendor lock-in caused by the vendors themselves.

The problem of how to cross-develop applications for multiple platforms has been left unanswered and this opening is something HTML5-based solutions are able to exploit. The wide variety of HTML5-based frameworks allow for solutions based on it to adapt to a fairly large number of situations. Even if a particular method of application deployment might prove in the long run to be unsuccessful, it is more likely that a large quantity of the development already done can be transferred to another application, minimizing the amount of waste in development.

One of the main benefits of the application store model comes from the simplicity of monetizing applications, but at the same time it ties down the application developer, both technically and legally. To be able to publish in an application store, the application developer has to follow the guidelines set by the store and also accept the 70/30 revenue split. With HTML5, the application itself can be hosted as a traditional website, but it can also be deployed as a more traditional application, via a mobile platform vendor's application store, or in some cases even by simply downloading it from a website. This wider set of options for deployment for the applications publisher might be a key factor in switching over to HTML5.

The areas where HTML5 is lagging behind its native competitors are performance, ease of use, and added value. The performance aspect is viewed in detail in section 4.5, but the ease of use and added value factors have to also be addressed. If the benefit proposal is clearly biased in the favor of the application developer, that is, the user experience of an HTML5 application compared to a native application would only be marginally better, the same or even worse, then the user has very little incentive to switch over from an existing native application unless forced. For new services with no existing native applications the situation might be simpler, but overall, if the HTML5 user experience cannot reach the level of native applications, it can be considered a serious hindrance for its wider adoption on mobile devices.

Although many Mobile Network Operators (MNOs) have launched their own application stores, these initiatives have gained little success compared with those of handset manufacturers or operating system providers. Moreover, because MNOs normally provide subsidized handsets for numerous platforms, creating and maintaining an application store for each platform can be costly. With web-based applications opening up the application store model, MNOs may be able to reach new relevance by mediating between content providers, advertisers, and end users. MNOs can bring value to these actors by helping users find relevant applications, and by providing application developers with more flexible billing models and revenue sharing than current application store providers. In addition, MNOs could utilize their access to anonymous user data like device type, location, and behavioral data to better target applications for users. Similarly, because HTML5 uses the client-server paradigm, HTML5 applications are well suited for utilizing open APIs offered by MNOs [12]. These APIs can provide developers with additional capabilities such as user location, billing, and SMS messaging [31]

while fitting seamlessly to the client-server paradigm of HTML5.

4.5. Technological performance

As previously mentioned, HTML5 is still a work in progress, even though some of the new features it defines have already been implemented in certain browsers. Problems with mobile use include the adaptation of the web applications view to the quirks of the particular platforms conventions, considering for example the extra buttons on an Android device compared to an iOS device. Browser compatibility is another issue, as not all browsers treat the same piece of code in the same way. Browser performance must also be taken into account, as the browser itself adds another layer of complexity between the application and the hardware. Even though there has been progress in the last few years on the execution speeds of JavaScript on browsers compared to native code, JavaScript still has to be downloaded, parsed, and only then can it be executed, adding a time penalty.

There are a few frameworks available to address some of the issues, such as the previously mentioned PhoneGap and Titanium SDK. Both frameworks allow access to most of the internal APIs of their supported mobile platforms, but provide them in a platform-independent way for cross-platform development and deployment. These frameworks, HTML5, and web-based applications in general exchange application execution smoothness and responsiveness for flexibility and a more universal deployment scheme when compared to native applications.

5. Summary of results and discussion

5.1. General results

In our analysis, we identified several factors that act as both drivers and restraints to the diffusion of HTML5. Table 3 displays these drivers and restraints and how they relate to the theoretical framework dimensions presented in Table 1.

The added value [37] or relative advantage of HTML5 over substitutes [39] is crucial for the success of the technology. In the case of HTML5, the most important driver in this dimension is cross-platform compatibility (D1), which allows developers of mobile web applications to more easily target various mobile platforms, thus minimizing the negative effects of mobile OS fragmentation. On the other hand, the end user benefits of HTML5 and web applications are somewhat limited and the user experience (R1) of

these applications can, in fact, be inferior compared to native applications.

Table 3. Drivers and restraints of HTML5

Dimension	Driver	Restraint
Added value	Cross-platform compatibility (D1)	User experience compared to native apps (R1)
Ease of experimentation	Cheaper, more flexible development and deployment (D2)	
Complementary technologies		Browser support (R2)
Incumbent role	No reliance on restrictive policies (D3) Flexible revenue models (D4)	Infrastructure and marketing expenses (R3)
Technological performance		Performance compared to native apps (R4)

Ease of experimentation [27,45] or trialability [39] in this context refers to the ability of mobile developers to adopt HTML5 and to develop applications using HTML5. An important driver is the cheaper and more convenient development of HTML5 applications (D2) compared to native applications. In addition, the deployment of these applications is much more flexible, and the developers can choose to use different HTML5 frameworks (see Figure 1), allowing them to tailor the application deployment to their needs (D2).

Complementary technologies can play an important role in the emergence of technological discontinuities [3] and in the case of platforms [18,26]. In the context of HTML5, the most important complementary technology is mobile browsers, and their support for HTML5, which is still incomplete (R2). The crucial issue is the incentives that mobile browser providers have in developing the browsers and the effectiveness of browser standardization.

The role and actions of incumbents is important in determining what effect new entrants can have on the market [24], especially in the case of disruptive innovations [10]. In the case of HTML5 and web applications, the most important incumbents are the current application store providers or platform vendors, such as Google and Apple. Their current restrictive policies on application approval (D3) and revenue sharing (D4) can function as an incentive for

developers to move to HTML5 and web applications. On the other hand, the platform vendors provide an infrastructure for the deployment of applications and a marketing venue for application developers, which would not be available for pure web applications (R3).

A sufficient level of performance [37,40] of HTML5 is a precondition for the success of the technology. Mobile web technologies, such as the execution of Javascript in browsers, have seen considerable progress, but mobile web applications still suffer from a performance gap compared to native applications (R4), which can in part lead to a worse user experience. However, these performance issues can partly be reduced by opting for a hybrid solution between native and web applications, using frameworks such as PhoneGap.

5.2. Practical examples

The adoption of new technologies can be accelerated by the example of successful products or services using these technologies. For technologies aimed at end users, so-called "killer applications" can be instrumental in driving end-user adoption. However, because HTML5 benefits mostly developers and not end users, examples of successful HTML5 applications are more likely to be relevant in demonstrating the benefits of the technology for developers.

The following real-life examples highlight some of the drivers mentioned in Table 2 and provide a quick overview of how HTML5 has and currently is impacting the mobile application market.

- [D3] Grooveshark offers a HTML5 client [19], as the native mobile application was pulled from both Google's and Apple's application stores due to ongoing legal issues [8] between Grooveshark and the music labels EMI, Sony, Universal, and Warner.
- [D1, D2] OpenAppMkt [35] is a marketplace for mobile HTML5 applications, and they have a client available for iOS and Android. Mozilla and AT&T have also brought or are bringing out application stores based on HTML5 applications, called Marketplace [33] and AppCenter [7], respectively.
- [D3, D4] Financial Times completely switched from the AppStore to an HTML5-based application [14].
- [D1, D2] According to PhoneGap [36] and Titanium SDK [5], several companies and other groups, such as the Wikimedia Foundation, eBay, and NBC, have released applications built on their frameworks.
- [D1] Facebook [20], Amazon (Kindle Cloud Reader) [2], and Dropbox [15] all offer HTML5-based

applications for services for which they also provide native mobile applications.

The examples presented in the list above do not all rely on a particular benefit that HTML5 provides, but accentuate the multitude of new opportunities it brings. Financial Times, for example, has taken the route of removing its application completely from Apple's AppStore, while Facebook has done the opposite and deepened its relation with Apple's mobile operating system, iOS [6]. Facebook's iOS application used to be a HTML5 solution wrapped as a native application, but they released a completely new native iOS application [13] written in Objective-C in August 2012, citing "that when it comes to platforms like iOS, people expect a fast, reliable experience and our iOS app was falling short." While a single application moving away from HTML5 might not be that significant, it highlights the fact that HTML5 still has issues to solve, mainly related to performance.

What Facebook and Financial Times have in common is that they do both provide a viable HTML5-based application for their service. In the case of Facebook, the HTML5-based mobile version of Facebook [20] is used as a tool to reach a wider audience, that is, users that are not using native mobile applications for one reason or another. From the perspective of Financial Times, HTML5 is used as a replacement for a native application because of the restrictive policies of a single platform provider [40], as Financial Times does provide an Android application [22].

The HTML5 mobile application stores by AT&T, Mozilla, and OpenAppMkt present an attempt to create a cross-compatible application store for all compatible mobile platforms. The number of potential customers is naturally higher than in any platform-specific store, but at the same time they suffer from a much larger pool of differing hardware and software combinations. Ensuring an adequate user experience for so many different devices will undoubtedly require more resources, negating a part of the easier development aspect of HTML5 but fully taking advantage of the technology's cross-platform capabilities.

PhoneGap and Titanium SDK both offer an enticing solution to the native vs. web-application question. Developers can, to a varying degree, use the same codebase for both versions and leave out native development altogether. In addition, the switching cost on the developers' side to either version in the future is minimal compared to fully porting a native application to HTML5 or vice versa.

6. Conclusions

Although HTML5 applications are not equal to the performance levels of native applications, the lower cost and cross-platform availability of a web application might prove crucial for vendors or organizations looking to support the largest number of customers possible, without writing platform-specific implementations of their application. Political or financial decisions to opt out of the mobile platform vendors' ecosystems can also motivate companies and individuals to directly adopt HTML5 and web-based applications. Which actor or actors would provide the largest push away from native applications and towards HTML5 remains to be seen.

Mobile browser support for HTML5 features is a key factor in the diffusion of the technology, and efforts to adopt and integrate HTML5 standards into a growing number of browsers [23] are ongoing. For certain types of applications, HTML5 will surely be a viable option, but at the same time it is unknown if platform vendors would simply halt the development of their own mobile application platforms and let web applications take over.

The final issue is that of end user preference, in which the largest obstacle comes from the fact that if HTML5 cannot offer a level of usability and added value that users currently receive from native applications, then there is no pull from their part to adopt the technology. A level of parity is needed, which requires support from the platform and hardware vendors to open up their systems for the browsers to utilize with HTML5. Nevertheless, the performance gap between native and HTML5 applications is closing, and the performance of HTML5 applications is already suitable for many end user needs.

In this paper, we provided an initial analysis on how HTML5 affects different actors in the mobile phone ecosystem. In the future, more research is needed to clarify the effect of HTML5 especially on the role of mobile network operators.

7. Acknowledgement

8. References

[1] ABI Research, "2.1 Billion HTML5 Browsers on Mobile Devices by 2016 says ABI Research", 2011. Retrieved June 10, 2012 from http://www.abiresearch.com/press/3730-2.1+Billion+HTML5+Browsers+on+Mobile+Devices+by+2016+says+ABI+Research.

[2] Amazon, "Kindle Cloud Reader". Retrieved June 10, 2012 from https://read.amazon.com/about.

[3] P. Anderson and M.L. Tushman, "Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change", Administrative Science Quarterly 35 (4), 1990, pp. 604–633.

[4] P. Anderson and M.L. Tushman, "Technology Cycles, Innovation Streams and Ambidextrous Organizations", In Managing Strategic Innovation and Change, Oxford University Press, New York, 1997.

[5] Appcelerator, "Titanium SDK -- App showcase". Retrieved June 10, 2012 from http://www.appcelerator.com/thinkmobile/showcase.

[6] Apple Inc, "Apple Previews iOS 6 With All New Maps, Siri Features, Facebook Integration, Shared Photo Streams & New Passbook App", 2012. Retrieved June 10, 2012 from https://www.apple.com/pr/library/2012/06/11Apple-Previews-iOS-6-With-All-New-Maps-Siri-Features-Facebook-Integration-Shared-Photo-Streams-New-Passbook-App.html.

[7] AT&T, "AT&T AppCenter". Retrieved June 10, 2012 from https://appcenter.wireless.att.com/.

[8] BBC News, "EMI terminates Grooveshark deal", 2012. Retrieved June 10, 2012 from http://www.bbc.co.uk/news/technology-17610811.

[9] C.M. Christensen and R.S. Rosenbloom, "Explaining the attacker's advantage: Technological paradigms, organizational dynamics, and the value network", Research Policy 24 (2), 1995, pp. 233–257.

[10] C.M. Christensen, The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail, Harvard Business School Press, Boston, 1997.

[11] L. Corral, A. Sillitti, G. Succi, A. Garibbo, and P. Ramella, "Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm", 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software, 2011, pp. 181–183.

[12] M. Crossey, "Telco 2.0: HTML5 - The Catalyst for Network as a Service?", 2012. Retrieved June 10, 2012 from http://www.telco2.net/blog/2012/07/html5_-_the_catalyst_for_netwo.html.

[13] J. Dann, "Under the hood: Rebuilding Facebook for iOS", 2012. Retrieved June 10, 2012 from https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920.

[14] K. Doctor, "FT Declares Independence (from Apple) Day", 2011. Retrieved June 10, 2012 from http://newsonomics.com/ft-declares-independence-from-apple-day/.

[15] Dropbox, "Dropbox Mobile HTML5 Client". Retrieved June 10, 2012 from https://m.dropbox.com.

[16] K.M. Eisenhardt, "Building Theories from Case Study Research", Academy of Management Review, 14 (4), 1989, pp. 532–550.

[17] T. Eisenmann, G. Parker, and M.W. Van Alstyne, "Strategies for Two-Sided Markets", Harvard Business Review, 84 (10), 2006, pp. 92–101.

[18] T. Eisenmann, G. Parker, and M.W. Van Alstyne, "Opening Platforms: How, When and Why?", SSRN eLibrary, 2008.

[19] Escape Media Group, "Grooveshark Mobile". Retrieved June 10, 2012 from http://html5.grooveshark.com/.

[20] Facebook, "Facebook Mobile HTML5 Client". Retrieved June 10, 2012 from https://m.facebook.com/.

[21] I. Fette, "The WebSocket Protocol", 2011. Retrieved June 10, 2012 from https://tools.ietf.org/html/draft-ietf-hybi-thewebsocketprotocol-05.

[22] Financial Times, "The Financial Times app for Android", 2011. Retrieved June 10, 2012 from http://apps.ft.com/android/.

[23] M. Firtman, "Mobile HTML5 - compatibility on iPhone, Android, Windows Phone, BlackBerry, Symbian, and other mobile and tablet devices". Retrieved September 10, 2012 from http://mobilehtml5.org/.

[24] R.N. Foster, Innovation: The Attacker's Advantage, Summit Books, New York, 1986.

[25] Gartner Inc, "Worldwide Smartphone Sales to End Users by Operating System in 1Q12", 2012. Retrieved June 10, 2012 from https://www.gartner.com/it/page.jsp?id=2017015.

[26] A. Gawer and M.A. Cusumano, Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation, Harvard Business Review Press, 2002.

[27] M. Gaynor, Network services investment guide: maximizing ROI in uncertain times, Wiley Publishing, Indianapolis, Indiana, 2003.

[28] I. Hickson (Ed.), "HTML Living Standard", 2012. Retrieved June 10, 2012 from http://www.whatwg.org/specs/web-apps/current-work/multipage/.

[29] IDC, "Smartphones Outstrip Feature Phones for First Time in Western Europe as Android Sees Strong Growth in 2Q11, Says IDC", 2011. Retrieved June 10, 2012 from http://www.idc.com/getdoc.jsp?containerId=prUK23024911.

[30] A. Juntunen, M. Kemppainen, and S. Luukkainen, "Mobile Computation Offloading - Factors Affecting Technology Evolution", ICMB 2012 - International Conference on Mobile Business, 2012.

[31] A. Juntunen, V. Suikkola, Y. Raivio, and S. Luukkainen, "Innovation in Mobile Clouds - Analysis of an Open Telco Application", CLOSER 2011 - International Conference on Cloud Computing and Services Science, 2011.

[32] M.H. Meyer and J.M. Utterback, "The Product Family and the Dynamics of Core Capability", Sloan Management Review, 34 (3), 1993, pp. 29–47.

[33] Mozilla, "Mozilla Mobile Marketplace". Retrieved June 10, 2012 from https://marketplace.mozilla.org/.

[34] Nielsen, "Smartphones Account for Half of all Mobile Phones, Dominate New Phone Purchases in the US" , 2012. Retrieved June 10, 2012 from http://blog.nielsen.com/nielsenwire/online_mobile/smartphones-account-for-half-of-all-mobile-phones-dominate-new-phone-purchases-in-the-us/.

[35] OpenAppMkt, "OpenAppMkt -- Marketplace for Mobile HTML5 Apps". Retrieved June 10, 2012 from http://openappmkt.com/.

[36] PhoneGap, "PhoneGap | About", 2012. Retrieved June 10, 2012 from http://phonegap.com/about.

[37] M.E. Porter, Competitive Advantage, Free Press, New York, 1985.

[38] M. Raatikainen, V. Myllärniemi, S. Ghosh, et al., "Towards Mobile Device Cloud", Communications of the Cloud Software, 1 (1), 2011.

[39] E.M. Rogers, Diffusion of Innovations, Free Press, New York, 2003.

[40] J. Saba, "FT Web-based app more popular than app sold in Apple store", 2011. Retrieved June 10, 2012 from http://www.reuters.com/article/2011/09/22/us-ft-idUSTRE78L49Q20110922.

[41] Strategy Analytics, "One Billion HTML5 Phones to be Sold Worldwide in 2013", 2011. Retrieved June 10, 2012 from http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5145.

[42] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen, "The Death of Binary Software: End User Software Moves to the Web", Ninth International Conference on Creating, Connecting and Collaborating through Computing (C5), 2011, pp. 17–23.

[43] A. Taivalsaari and T. Mikkonen, "The Web as an Application Platform: The Saga Continues", 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), 2011, pp. 170–174.

[44] Telco 2.0 Research, "HTML5: market impact and telco strategies". Retrieved June 10, 2012 from http://www.telco2research.com/articles/EB_HTML5-what-does-it-mean-for-telcos_Summary.

[45] S. Thomke, Experimentation Matters, Harvard Business School Press, Boston, 2003.

[46] W3C, "HTML5, W3C Working Draft 29 March 2012", 2012. Retrieved June 10, 2012 from http://www.w3.org/TR/html5/.

[47] W3C, "Standards for Web Applications on Mobile: current state and roadmap", 2012. Retrieved June 10, 2012 from http://www.w3.org/2012/05/mobile-web-app-state/.