The web as a platform

Pablo F. Iglesias
contacto@pabloyglesias.com
July, 2014 (edited November, 2014)

Abstract

In a highly competitive as mobile development environment, analyze the current situation of the web as a platform, and their proposals in the form of operating systems in the market.

Also we will review trends in a few years, new facilities and their competitive advantages over native development of conventional operating systems.

Keywords

HTML5, web, internet, webapps, mobile, operating system, SO, CSS3, Tizen OS, Firefox OS, Web OS, Android, iOS, Blackberry 10, Windows Phone 8, Sailfish OS, Ubuntu.

Índex

1. Introduction

Computers, and therefore, consumer electronics, has undergone a significant evolution over the past two decades with the democratization of mobile operating systems. For such a feat has been carried out, it was necessary to wait until the industry was able to miniaturize the classic components of an operating system, increasing battery life and its optimization with the rest of the existing hardware. The arrival of a new generation of processors based on ARM technology has meant, at least until recent years a standard for low-level communication, emerging platforms based on this paradigm.

The figure of the smartphone and tablet begins to blur, and in this study and discuss not only the two big winners of the new consumer electronics, but their specialization and permanently connected devices (wearables), and its implementation in traditional object unintelligent (IoT).

A booming sector, which has led to large technology companies in the twentieth century to evolve, turning the business model (as in the case of IBM) or directly disappearing before new figures born or adapted to the Internet age (Google, Samsung, Mozilla).

A journey through the bowels of current mobile operating systems, the estimated market for its disintegration and the importance of web languages are taking in and out of this ecosystem.

2. Architecture of a mobile operating system

An operating system is nothing more than a set of commands and programs able to abstract the logic to communicate with the hardware, so that other programs can operate above and provide access to information needed for user interaction with the machine .

Mobile operating systems inherit this functionality usually less powerful devices, sacrificing some performance in exchange for greater autonomy or more specific type of use.

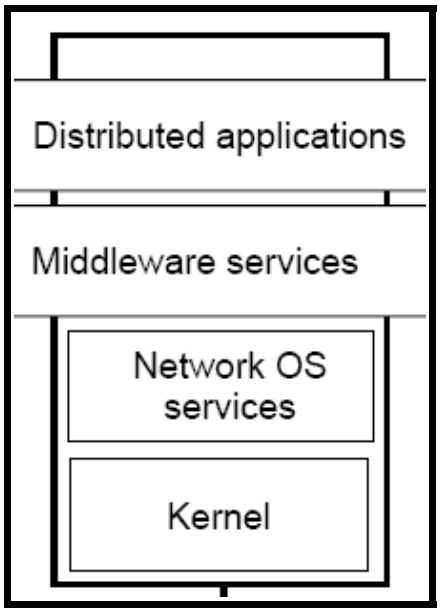
However, a strong hierarchy of permissions and the usual dependence on a service platform have allowed computers to mobile devices evolve differently from that it has on the desktop. Current mobile operating systems are strongly geldings in the face of user interaction. Application markets act as intermediaries, abstracting the respective technical implications for data security and possible vulnerabilities in them. The user thus becomes a consumer tool, freeing themselves from responsibility for their governance, and losing the way of the control previously held.

This has enabled a real breakthrough in the field. On the one hand, the user is consuming third-party applications, which after all, are the ones that offer real value to the operating system. And secondly, the community of developers, which was formerly only tied exclusively to open projects, has favored the growth of a limited market for each operating system, managed by the company behind it, and ultimately ends up being the monetary pillar of the entire ecosystem.

The mobile operating systems currently depend on the developer community on its platform, with its contributions, attract a greater or lesser number of users to it.

Access to processes of the system becomes so climb through permissions, so this apparent democratization of development (anyone can make an app and upload it to the market) is held, and on the other, narrowly manages interaction with the system, which becomes an operational base where the user launches the different services.

According to its modular architecture, we find similarities with most desktop operating systems.



2.1. Kernel

It is the software layer responsible for direct communication with the hardware. We speak of the lower layer, and therefore, its mission is to communicate the requests of the upper layers to the physical components, restoring access to one or other controller, memory management, file system and so on.

Its existence is not only linked to the mobile operating systems, but ultimately to any operating system today.

Thus, we find that many of the mobile OS on the market use a Linux kernel (Android, iOS (UNIX), ...) that have been removed components for the use to which it will provide.

Due to the great success of Android, there have been quite a few fork of the kernel, resulting in different versions of Android (Fire OS Amazon, for example) or genuine new operating systems (Firefox OS Mozilla or Tizen OS Samsung).

Within the mobile ecosystem kernels, we also find proprietary versions, like Windows Phone and Blackberry.

2.2. Middleware

Above the kernel usually have the middleware, a layer which is responsible for communicating with processes in the high level. Thus, this layer manages the communication engine and messaging services with counting device, multimedia codecs and so on.

It is also responsible usually manage permissions and discriminate between requests, following the restrictive modular architecture of mobile operating systems. Thus, each application will have access to only those permissions you order in advance, validation policies can be set and preliminary testing different systems (face to allow installation from an official market).

2.3. Application Execution Environment

We speak of a layer in charge of providing the necessary components to run third-party developments (applications). It is very dependent, therefore, the native language of the system, and the freedom that is given to the community (through APIs).

It also has an application manager, a fundamental tool for the coexistence of different services and use by the user.

2.4. UI

All operating systems must have a user interface. This is the layer with which the client interacts.

Served for this gesture and graphical elements (buttons, forms, menus, ...) that allow users to use services developed without resorting to console commands.

It also has a large group of services and applications that come installed by default, whether own operating system (diary, photo gallery, calendar, ...), developed by a company as a value proposition (or Google apps iApps, to give two examples) or installed on the device as agreements with operators or other stakeholders of the market (Dropbox on Samsung, Vodafone Cloud terminals purchased from the operator).

The user experience is different depending on the operating system being used, and evolves over time. The first-generation operating systems had barely gestures, having to use tools such as a physical keyboard for interaction. With the massification of the touch screens, we have experienced a migration from physical to digital, to the point that most interaction in operating systems is done by second generation digital buttons. Since a few years, the operating systems are increasingly gaining gestures for communication, which points to a near future in which much of its use is focused on this paradigm and not scanning button.

3. Market

Denial is absurd. Mobile operating systems, together with the rapid rise of smartphones (and to a lesser extent on the tablets) have displaced the historical importance of the desktop.

Most first world people choose to invest their money in renewing their mobile devices before the desktop. Even according to what uses, the tablet has started to offer more value than a laptop, with the default choice of a sector whose use of information through the media consumption and web browsing / social networking.

So we live a true technological revolution. Consumer electronics AD suits our needs as it has never done before. It is becoming more intuitive, less technical (at least for the user) and more practice. Internet has freed information, accessible by anyone where you are, what has certainly changed the habits of society towards technology. Use of an immersive, scheduled, I spent a continuous use, carefree. The useful information comes to us through notifications, informative small pills that warn us that something interesting may help us, so that systematic consultation is relegated to those times of need, or those timeouts among other actions.

All this is reflected, as expected, the substantial increase in the mobile market, and last year surpassed the desktop, and today is still owner and master of the sector.

So much so that giants like Samsung, whose niche was quite out of the computer user (appliances, construction and investment, mainly) have passed in just five years to dominate the sector. And others, strongly attached to the figure of the desktop, move your chips have managed to become the most profitable companies in history (Apple).

However, not all the playing field becomes golden, and example, the unfortunate situation that some years deals Blackberry (formerly RIM), or the loss of a Microsoft domain before anyone else arrived sector and did not know until recently a move to position in it.

In this section, therefore, we will address some of the latest market research industry. Different studies with different biases applied, that together allow us to observe some trends, emphasizing, as expected by the title of this research, the importance of the web in this ecosystem.

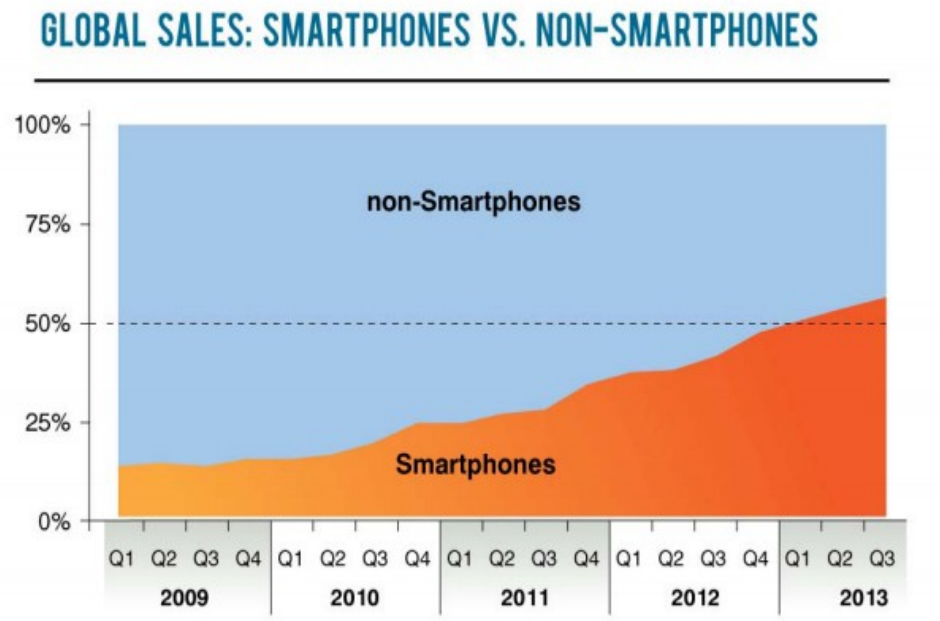
Since this article was published in July 2014, the majority of studies reviewed is to represent a picture of the Q1 of this year (mostly published between February and May) market.

3.1. Developers

The first study analyzed comes from the VisionMobile company, as every year, recently released the sixth edition of Economic Developers, a tour of the state of the industry from the interest shown by developers.

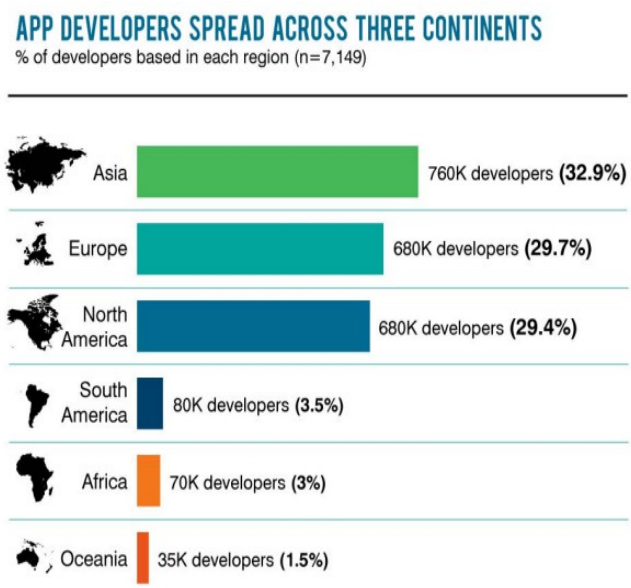
To generate the necessary knowledge, conducts surveys more than 7,000 employees in more than 127 countries, so we can consider it a pretty accurate scale. It is also sponsored by companies like Mozilla, Intel, Canonical or Blackberry (among many others), so a little bias rather accentuated expected.

The economic outlook of the sector in 2013 was 68 billion dollars, expected to increase to 143 billion in 2016. Therefore speak of a market with a strong upward trend, which continues to expand, and is still in infancy compared to what is coming to us.



In the upper graph we can see the evolution that has taken the smartphone sector compared to first-generation mobile devices. As early as 2013 the barrier was exceeded 50%, finishing at 55% and reaching 60% in the first Q1 of this year.

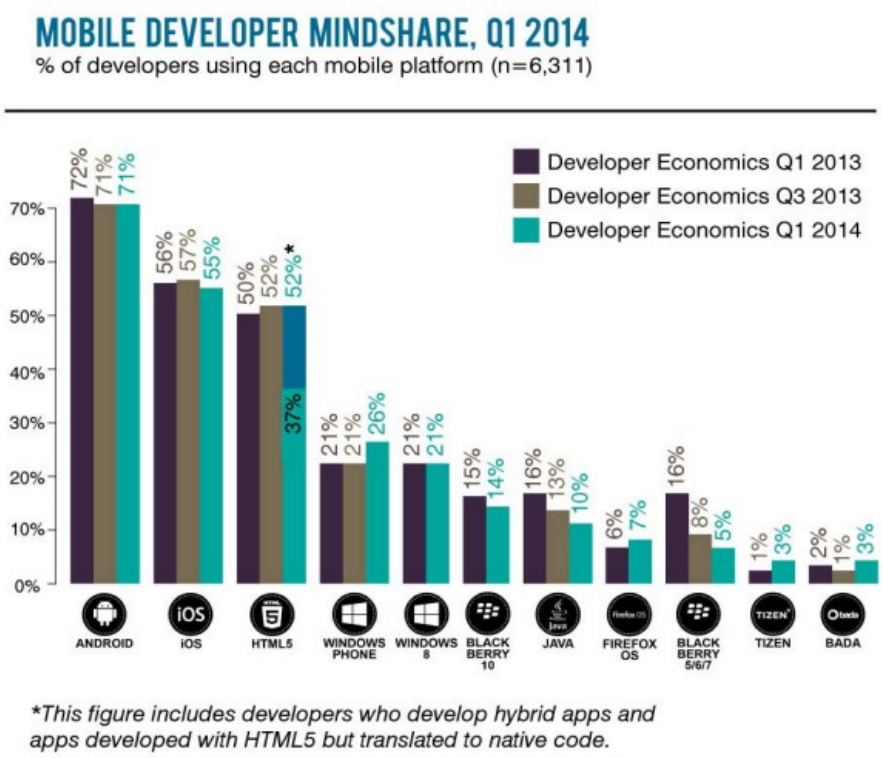
The tendency to the disappearance of non-smartphones, whose main presence is held by Bada and older versions of Blackberry, mainly in South America, Africa and Oceania, and where therefore confirms low cost operating systems such as Android (Samsung ended the year with 31% of the sector) or Firefox OS are accelerating the transition.



Considering the evolution of community development, Asia encompasses 32.9% of the market, overtaking Europe and North America (both with 29, X%) in presence, which highlights and enhances the figure of the Asian giant as the largest exporter and consumer of technology, shifting historical powers the industry, and advancing the loss of economic power and not only in Europe but in the U.S. against China.

Both South America, Africa and Oceania have a minority presence, mainly because many of its countries are developing. However, I find it interesting to note that while Africa and Oceania have a long way to go, market trends, and technological movements of recent years revolve around a

South increasingly advanced technologically, and the departure yoke of the first Blackberry, and start today to make the jump to the smartphone world, supported as they are in the market for low / medium range.

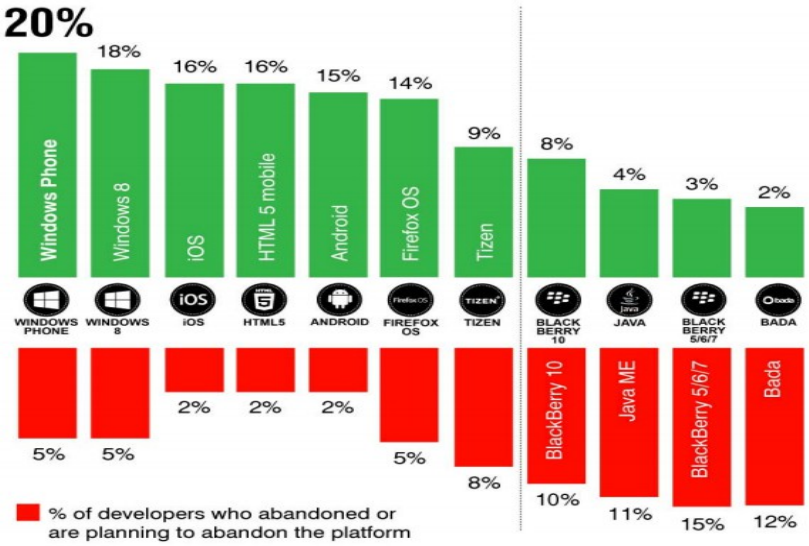


This figure includes developers who develop hybrid apps and apps developed with HTML5 but translated to native code.

Regarding the evolution of mobile operating systems by interest from developers, we see three clear winners: Android, iOS and HTML5.

Android remains the main engine of the sector, losing only 1% presence in favor of the new proposals, which it shares with iOS situation. On the other hand, is HTML5, which has it increased by 2% from Q1 last year, considering that this percentage is calculated not only in respect of applications developed for platforms of HTML5, but also for systems that allow hybrid applications (therefore, the percentage of fat plus other platforms).

Blackberry loses 1% of presence with its new version (BB10) and 9% with old versions, allowing Windows Phone up around 5% in the ARM version, Firefox OS 1% (considering that even has not taken off in most markets present in the roadmap), 2% for Tizen (supported by the engine representing Samsung in the sector) and a curious 1% Bada OS that refuses to go away.



In this graph we can compare the interest shown by developers enter into a platform (in green) versus percent giving it (red).

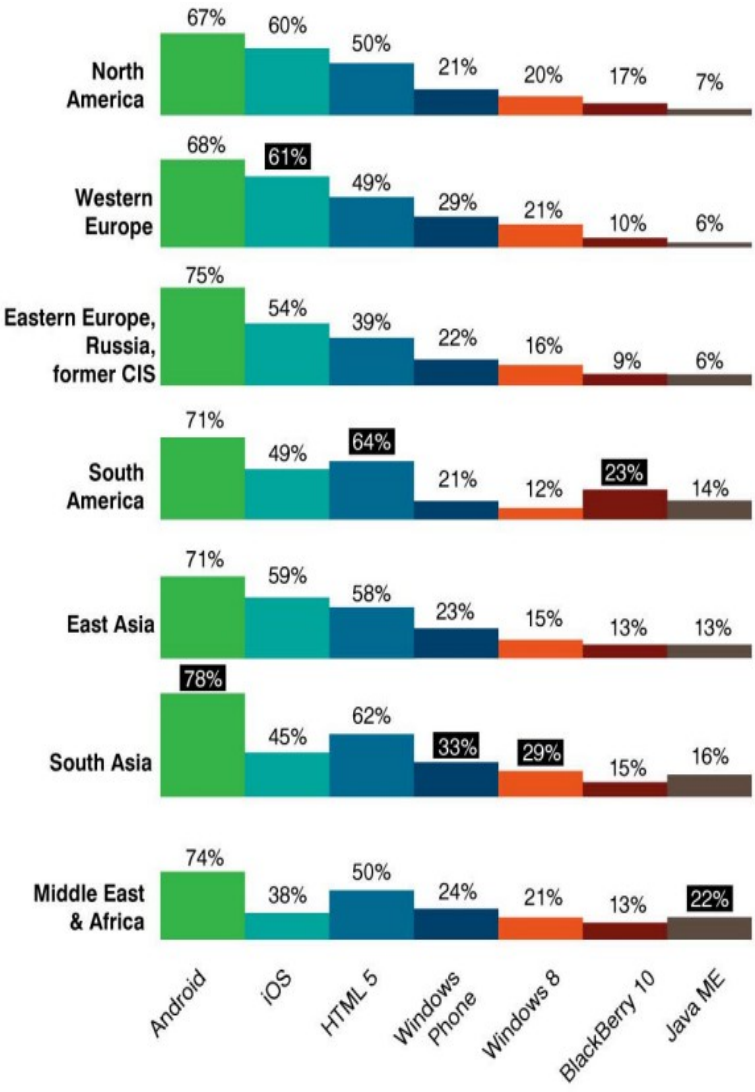
It is therefore clear that this year Microsoft with the new proposal (increasingly convergent) Windows Phone and Windows 8 raises the interest in the community, with 20 and 18% respectively, and only 5% drop.

What are iOS, Android and HTML5, with 16%, 16% and 15% and 2% dropout rate, Firefox OS (increased to 14% interest, compared to 5% drop), Tizen (9% interest versus 8% drop) and operating systems disappearing (Blackberry, Bada and Java applications).

Two platforms were left out of the study (at least) that a server would have liked to see represented: Blackberry OS and Sailfish Jolla.

The first, after about 18 months of development, has not yet gotten off the ground, which suggests that even with a truly disruptive proposal (a mobile terminal connected to a screen and a keyboard could function as desktop system) still it's too soon.

The second, based on Meego, offers a twist to the interaction via gestures, and although it had quite an impact at the end of 2013, has not been heard more from him.



We continue with the presence of SO different by region.

To be considered, Android wins with more or less success in all regions, with South Asia where more penetration rate is (to be expected given that both Samsung and HTC, Sony, XIAOMI and other Chinese manufacturers dominate the sector).

IOS is your best ally in Western Europe, while in South America makes HTML5, Windows Phone in South Asia.

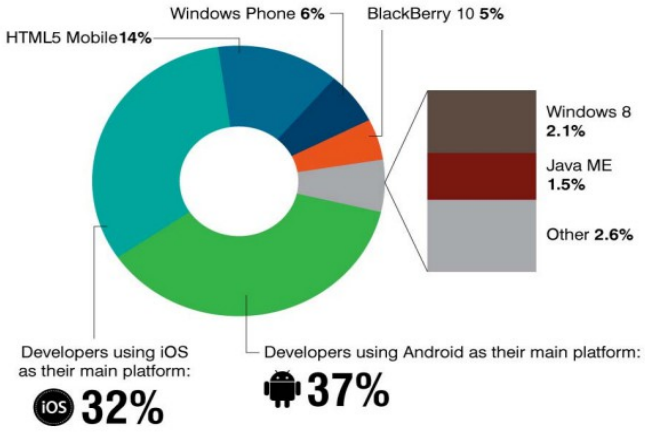
Finally, and as usual, Blackberry remains strong in South America and in Africa Java.

If we focus on the distribution of SO for each country markets, Kantar released earlier this year a similar study, whose graph (next page) shows some peculiarities in this first can not be observed.

Thus we see that if Spain does not serve as an example of the situation of the European market, with Android basically monopolized the industry, choking the other proposals.

The iOS brutal decline in Italy in just one year (10.3% absorbed by Windows Phone) and Android, a situation that has also lived in South America with the falling prices of Android devices and Blackberry receding.

iOS gets hurt in all markets surveyed by Kantar, indeed maintaining a strong presence in the U.S. (43% market share). It is also interesting to note that the portfolio of iOS devices only closes the portfolio of a company (Apple), while other operating systems have different manufacturers (especially Android references), so that despite its reduced paulativa market, it is worth considering.

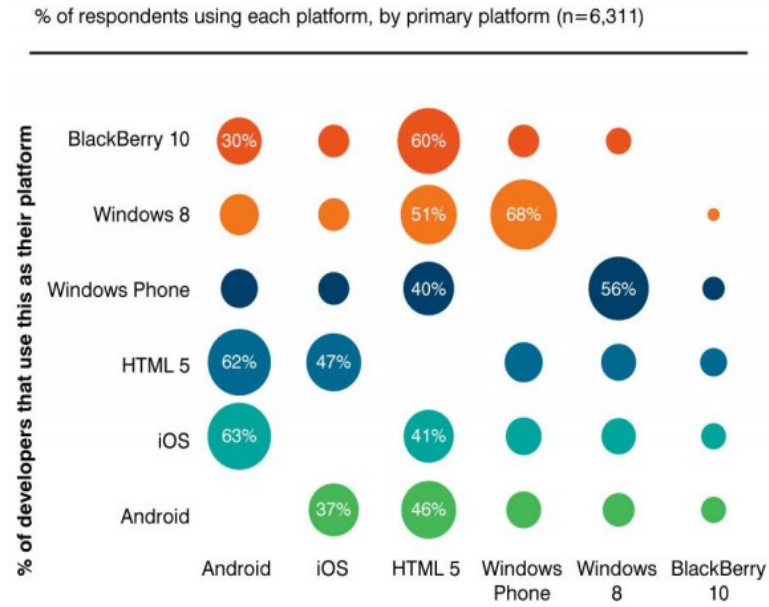


Germany	3 m/e Dec 2012	3 m/e Dec 2013	% pt. Change
Android	69.0	75.4	6.4
BlackBerry	1.1	0.5	-0.6
iOS	21.7	17.3	-4.4
Windows	3.4	5.9	2.5
Other	4.8	0.9	-3.9
GB	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	54.4	54.9	0.5
BlackBerry	6.4	3.2	-3.2
iOS	32.4	29.9	-2.5
Windows	5.9	11.3	5.4
Other	0.9	0.6	-0.3
France	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	61.0	65.9	4.9
BlackBerry	5.1	1.6	-3.5
iOS	23.7	20.3	-3.4
Windows	5.0	11.4	6.4
Other	5.1	0.8	-4.3
Italy	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	54.2	66.2	12.0
BlackBerry	2.6	1.8	-0.8
iOS	23.1	12.8	-10.3
Windows	12.7	17.1	4.4
Other	7.4	2.1	-5.3
Spain	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	85.9	86.2	0.3
BlackBerry	2.4	0.2	-2.2
iOS	7.3	6.7	-0.6
Windows	1.2	5.6	4.4
Other	3.2	1.3	-1.9

USA	3 m/e Dec 2012	3 m/e Dec 2013	% pt. Change
Android	46.2	50.6	4.4
BlackBerry	0.9	0.4	-0.5
iOS	49.7	43.9	-5.8
Windows	2.4	4.3	1.9
Other	0.8	0.8	0.0
China	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	73.7	78.6	4.9
BlackBerry	0.0	0.1	0.1
iOS	21.2	19.0	-2.2
Windows	0.9	1.1	0.2
Other	4.2	1.3	-2.9
Australia	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	56.0	57.2	1.2
BlackBerry	1.0	0.8	-0.2
iOS	38.5	35.2	-3.3
Windows	3.0	5.2	2.2
Other	1.5	1.7	0.2
LatAm 3 (BR, BX, AR)	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	61.6	83.5	21.9
BlackBerry	10.3	2.8	-7.5
iOS	4.4	4.3	-0.1
Windows	6.8	4.9	-1.8
Other	17.0	4.5	-12.5
EU5	**3 m/e Dec 2012**	**3 m/e Dec 2013**	**% pt. Change**
Android	62.9	68.6	5.7
BlackBerry	3.7	1.5	-2.2
iOS	23.7	18.5	-5.2
Windows	5.6	10.3	4.6
Other	4.0	1.1	-3.0

Regard to the interests of the community for a platform, 37% is decided by Android, 32% for iOS and 14% for HTML5.

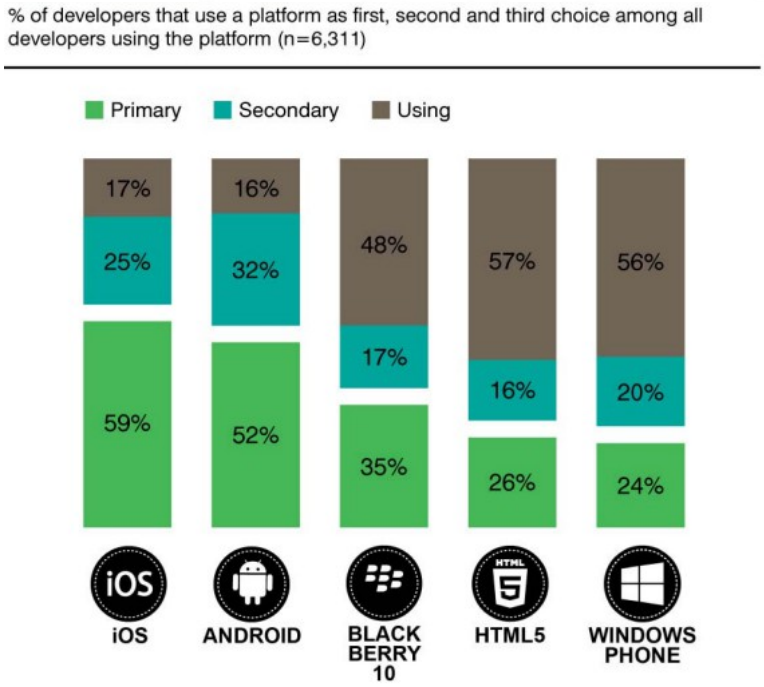
If we see these data with those in which two or more platforms are collated, HTML5 is the clear winner, despite having 14% of priority as the main platform, steals almost half interest in the multi-system development.



And it is here where we advance the first point to consider the web as a platform: **its ability to integrate with other operating systems**.

So we see 60% of the developments to take into account the Blackberry development in HTML5. A Windows 8 to 51%, 40% for Windows Phone, iOS 41% and 46% for Android.

Of course, it should be noted that 63% of iOS developers interested in Android (37% on the opposite side) and a similar situation occurs with the development of Windows 8 and Windows Phone (68% for the first, 56% for the latter).



Following development priority, iOS rises with interest as primary platform with 59%, Android and 32% secondary and tertiary as HTML5 as 57%.

And it's normal considering that today there are only two mobile operating systems and one desktop that have chosen HTML5 as the main platform, facing the wide range of operating systems with an SDK specifically designed for the web.

Curious that Windows Phone is the third choice of 56% of the community even though it has the advantage of being more and more compatible with the desktop operating system of the company, and a manufacturer in Europe as much weight as Nokia.

To conclude with the market overview, associate a graphical summary of the interest of both the developer community as the market for each operating system, and it brings out the objective target.

Thus we see how iOS is better positioned economically focused profiles, allowing developers to obtain higher profit rate per user.

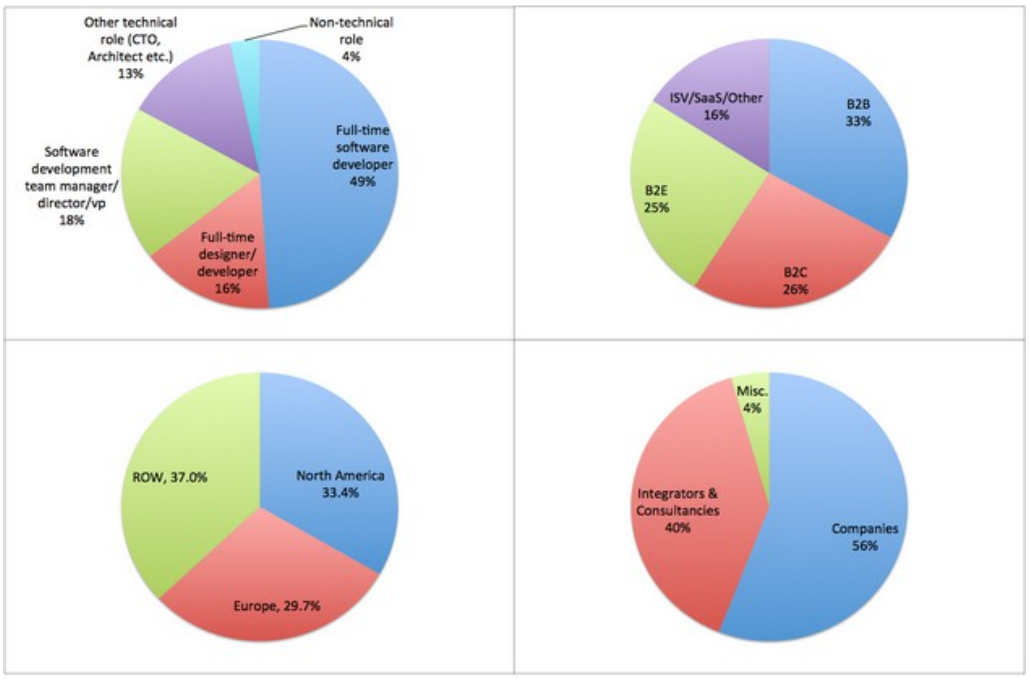
Meanwhile, Android has a much wider target audience, to cover both low-end and medium and high, and a market that tends to increase.

HTML5 is gaining ground, either as a hybrid development platform for other systems, or as a value proposition based on the universal market.

	ANDROID	iOS	HTML5 MOBILE	WINDOWS PHONE	BLACK BERRY 10
Sales market share (smartphones, Q3 2013)	81%	13%	-	4%	2%
Mindshare	71%	55%	52%	26%	14%
Priority	37%	32%	14%	6%	5%
Loyalty	52%	59%	26%	24%	35%
Most popular in	Asia	North America	South America	Asia	South America
Median revenues	$150	$750	$150	$25	$75
Differentiating selection criterion	Open Source	Revenue potential	Ease of porting	Choice of development environment	Documentation/ Access to hardware APIs
3rd party tools index	2,8	3,1	2,5	2,5	2,3
Top revenue model	Advertising	Contract development	Contract development	Advertising	Pay per download
Segments with a strong preference to the platform	Hobbyists, Gold Seekers	Digital Media Publishers, Hunters, Guns for Hire	Product Extenders, Enterprice IT	Hobbyists, Explorers	-

3.2. The web as a platform

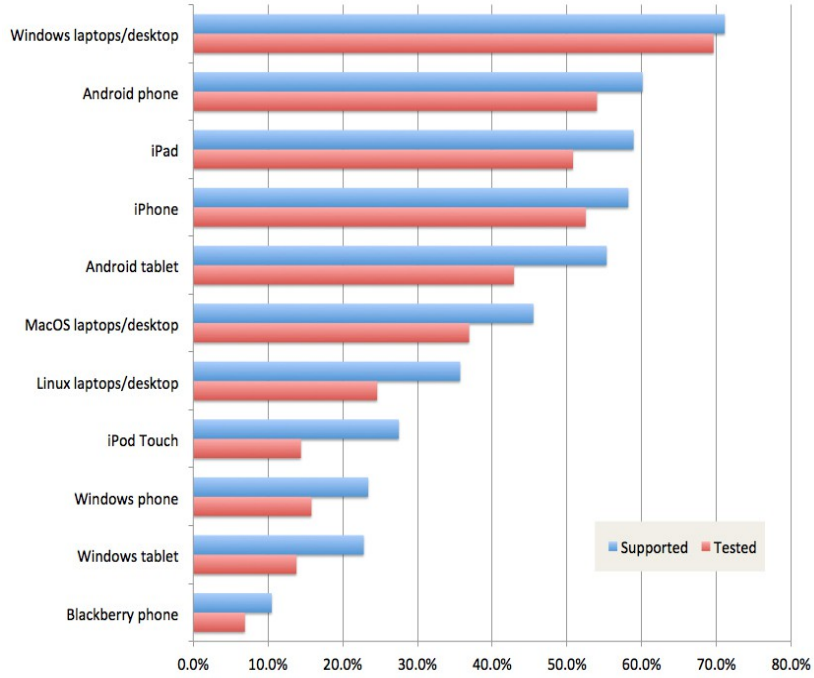
Throughout this study we have observed as in most situations, we discuss three major platforms, one of which HTML5.



The Sencha consultant presented a report on the state of HTML5 as a mobile platform in February, we can draw a lot of interesting facts:

Of the four graphs shown, it is interesting to note the interest of this platform for North America (33.4%) compared to Europe (29.7%) and the remainder (37%). Percentages that align with the market in each area (a Europe dominated by Android / iOS a North American divide between the interests of the major technological, and South America, Asia and Africa combined, with a profile of midrange, which increasingly bet HTML5).

Also interesting profiles about web development, with a clear dominance by the developer full time (49%).

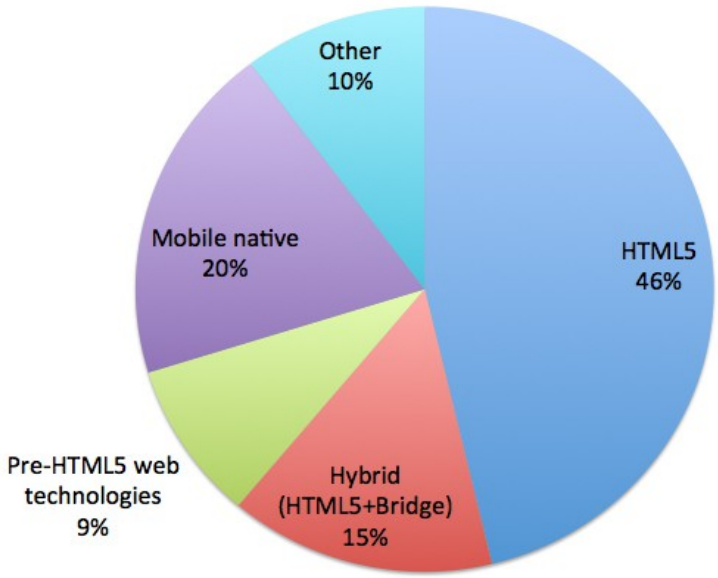


Considering the multisystem nature of HTML5, the lateral extent of the support graph shown by each device and testing level.

Understanding HTML5 as an evolving platform, it is normal that we see relatively low percentages, especially in those devices whose development environment is closed.

To avoid this problem, most operating systems have a new generation development environment ready for HTML5, as we shall see later in the proposal Windows 8/WP8, BB10, Tizen OS and so on.

The presence of HTML5 as a platform reaffirms developing countries where iOS loses steam in favor of low / medium range, led by Android and HTML5. And this is another major advantage of the website. A sleeping giant raising the interest of the developer community, and is now beginning to take shape and not only as a hybrid development environment, but relied on native platforms.



Nearly 70% of developers using or will use web technologies in your projects does not mean that HTML5 as a platform to iOS or Android is level. In fact, only 20% of these developments used as mobile web platform, leaving the rest for hybrid development, pre-HTML5 or for particular uses (documentation, contact form, access to the website, ...).

Has potential for it, and the market trend going on that course, but still years that the proposal for a web as a platform have enough to dethrone the duopoly implications of this generation. A duopoly which is based on the advantages of web technology (no choice) but rightly maintains its own closed ecosystem (separated markets such developments).

3.2.1. Advantages of the web as a platform

The history of HTML5 as a platform is the story of the struggle between social interest and self interest of the company.

HTML5 gives itself the freedom to migrate developments either platform, transferring information and data from one to another transparently to the user, rather than objectively not care if you own a model based on developments of third system which you participate making intermediary. The application markets have become the goose that lays the golden eggs from any operating system. A manager of cloud applications that ensures a steady flow of money to the company behind it, and that in cases such as Apple, comes to stay with 30% of all transactions made on it.

For the user, having to buy the same app twice or more by having to use different operating systems is a major drawback. Moreover, as in the case of most games (add a topology requires an application usage history), our progress is lost from one to another.

Moreover, the very business model leading companies like Google (web advertising agency), together with its repertoire of software products (Chrome OS based desktop HTML5, Android Java based) point to the convergence of developments web-based, inheriting characteristics of each ecosystem.

HTML5 is a proposal that is revolutionizing the paradigm of consumer electronics. Having a single platform for any type of device is a feature that a few years ago did not seem important (after all, had operating systems or desktop, or mobile), and expands as the Internet of Things starts be a necessity.

It is inconceivable to think of a future where each device has a different operating system. Different operating systems unable to communicate. Web languages, communication protocols and Internet thus define the way forward.

The specialization of the platform, with the creation of numerous APIs and services for managing advanced communications systems such as dialer, bluetooth, SIM or ambient location are already developed in HTML5, pending standardization.

Your **multisystem nature**, coupled with the ease of implementation on architectures that follow different patterns (hybrid applications), is now a point in favor of the web. Facing each face a development platform with HTML5 a company can develop your project and easily port it

to other way. In the course has a web version of the service, plus other versions for each operating system you want, reducing costs (both creation and maintenance) greatly.

Another feature to consider the web as a platform is that **you can make updates to the application directly from the server**. Dividing resources between client and server, you can update parts of the application without requiring the client update. New features that all users receive instantly without having downloaded an update, which unfortunately does not allow traditional applications.

Also, do not forget that the main purpose of the web as a platform is to **democratize access to information**. Applications should be available wherever we wanted to use them without having to pay for them and keeping the same information we had on the other device. The market applications Firefox OS points to this convergence: Applications available in both cloud and on premises, which can be installed on a device with operating system based on HTML5, or through the browser, on any device, be it desktop, mobile, wearable or IoT.

Of course, being web-based, it would not be necessary to download an application from an official market, but directly from the sweb service. Or even try without installing the application from the browser.

The requirements necessary to run a web-based operating system are lower in any case the requirements of a high level language like Java, which needs its own virtual machine. For Firefox OS, the system itself runs on a nerfed version of the browser, leaving as we shall see, the architecture of the system in only three layers. This makes it optimal for environments where the technical characteristics are strongly bounded (smartphones and tablets low-cost, smart devices, wearables, ...).

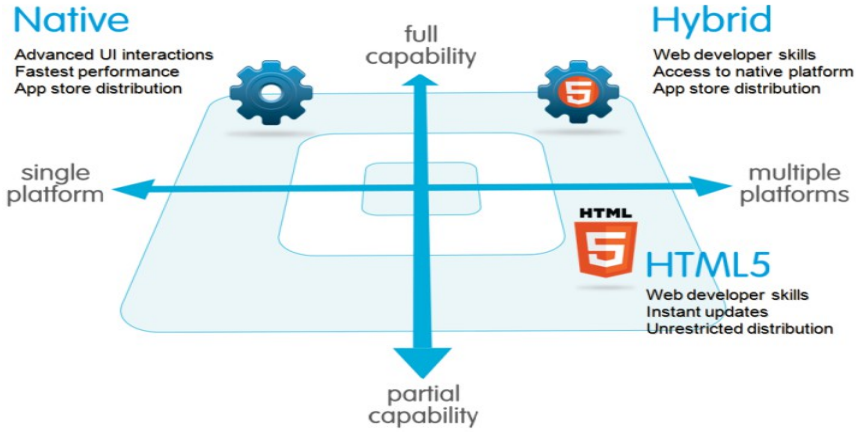
3.2.2. Drawbacks and historical myths

Under this section we collect the problems that the web as a platform had at least in the early stages.

The first one, and it has certainly caused an initial rejection of the proposal is that historically, HTML was not a programming language**.** HTML pure can not do nothing but logic presentation of information.

Do not understand HTML5 as a language, but as a platform composed of several languages, which is still HTML with a few new features, CSS3 capable of managing different conditional (ideal for proper viewing depending on the device where we are) and especially an evolution of JavaScript, which really is who gives full functionality and dynamic platform.

The second conflict is found in the idea that every web application must have connection to work. Again another mistake of not knowing the platform. **HTML5 can run both online and offline**. Both indexDB, as the very system cache, or using local database so testify. So the problem could come from own use that give developers the service, which occurs both as a web application in a Android or iOS. Facebook Offline only you can show that you have cached the last update, but in the case of first Angry Birds, you could play without problem.



Communication with the hardware is another obscure points of the platform. Indeed, the problem does not come from the same, but the components that give support to HTML5. Until about two years ago, there were no HTML5 APIs to manage voice communications or push notifications. This situation is now resolved. Some of these libraries have already been standardized, but still depend on other companies like Google, Microsoft or Mozilla, waiting for the community to decide what becomes the standard. In addition, platforms like PhoneGap already have specific interpreters for developing hybrid apps equal privileges in respect of each native operating system.

Performance of hybrid versus native apps: Here without position return a response in any of the two sides is quite complicated. For 99% of applications available on the market, HTML5 is as or more powerful than any other programming language. Assailed by doubt when we are dealing with applications that require rendering really high level. For these cases, developers may choose to use own apks as processor type on which the roll (in the case of Android and Intel) or Apple tools for graphics engines. When you develop a hybrid application (ie an application in HTML5 designed to run on a non-native operating system), make use of tools that act as intermediaries between the HTML5 media and the native language, so it is expected that this layer extra consume more resources than it was. That does not mean that a well-developed and optimized application to work as well or better than another even hybrid and native alike, something that was demonstrated in the answer he received a few years ago Facebook to abandon development of its hybrid Android application by a native.

For web-based operating systems, such difference does not exist (the system itself is optimized for the language).

4. Operating Systems and web languages

In this section we study the value proposition that HTML5 offers not only as a platform, but with its current implementation in each mobile OS. We will talk about last of them, let go of desktop operating systems, a sector that is still present but it does the scope of this project.

Thus, we will avoid that Chrome OS is an operating system based on HTML5, and Windows 8 or Ubuntu have a SDK for this platform. And also that any application developed for Firefox OS can be opened as an application (installation included desktop icon) in both Windows, Linux or Mac as long as they have the Firefox browser installed. Something that also occurs with Chrome apps.

4.1. Android

An operating system based on Linux and developed by the Open Handset Alliance, a conglomeration of manufacturers and developers led by Google. Android, or rather AOSP (Android without owning part of Google) is an open source project. And it is important to note the difference, as the Android that comes installed by default in the majority of terminals is far from following the open source philosophy.

Android architecture follows a very similar pattern to which we defined earlier in this document.

A reduced version of the **Linux kernel** acts as an abstraction layer between the hardware and other software layers.

The **libraries**, a set of libraries of C / C + + used by other upper layers. Are delivered to the developer in the framework calls application work.

Android Runtime is a set of libraries written in Java necessary to operate the processes within the Java virtual machine using Android, and called Dalvik (in the new Android L will be exchanged for a new version). An Android device can run multiple virtual machines at the same time quite efficiently, and managing different processes on different machines, modularizing the resulting workload and completing a simple enough to continue with the rest.

The **Application Framework** is the work environment developers, with all necessary to access the different elements of the device APIs.

The upper layer is the **application environment**, where we find the entire interface and user interaction with the device.

Android is a mobile operating system designed primarily for tablets and smartphones, but has more versions specifically developed for specific environments, such as Android Wear for wearables, Glass Glass development kit for Google Glass, Google Android TV and a future that seems to be still in production .

Android supports most HTML5 standards present, allowing you to enjoy an adequate web browsing. However, things get complicated when we install applications developed on this platform.

4.1.1. HTML5 Applications ported to native language

Titanium Appcelerator provides a development framework based on JavaScript that compiles native language (Android, iOS, Blackberry, ...). That is, the developer program in HTML5, but what finally goes to market is a native app.

Of course, it has a downside, is that the Framework comes to the best, not having all the expected freedom to configure your applications the way you want.

4.1.2. Hybrid applications

In this group, we collect all those developments based on using HTML5 Frameworks to upload your application to the appropriate market. As in the previous section, the best known is *PhoneGap*, a suite of developer tools that handles communication with Java. And again, we find the same limitation, with the addition that the functions that you can access depend solely Framework compatibility with native implementations of the system. Thus, you are forced to have continuously updated and not only the application, but the intermediary.

4.1.3. WebApps

The third and last point brings all those applications developed in HTML5 language, and through different strategies that allow their use within an operating system like Android.

Currently the WebApps in Android are deeply hit by the system, which prevents access to most resources of Android (discounting basic as access to geo I upload a photo, ...).

However, the strategy followed by Google with its Chrome browser predicts a near future in which the WebApps have a far more democratic treatment. The first movement we saw recently with the possibility to use Chrome extensions and apps by Android (after all, an extension of Chrome is a WebApp). Meanwhile, Firefox proposes exactly the same, packaging an application for OS on a Firefox browser tab, and access to the necessary permits, provided they fall under the very app for Firefox. In this way, the result is a user's eyes more application "installed" (actually it is a shortcut to a WebApp) along with the other applications.

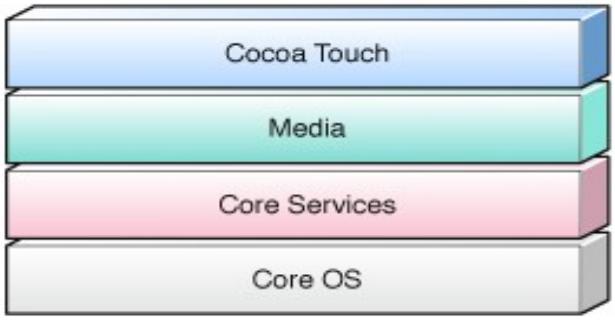
4.2. iOS

iOS (formerly known as iPhone OS) is one of the operating systems developed by Apple. Today is present only on iPhone, iPad, iPod Touch and Apple TV devices not being available for use in devices other than the company.

The main advantage of this philosophy is that the hardware of these devices is deeply optimized operating system to be used, so it is not uncommon for devices with Apple technical characteristics worst work just as well as end caps in Android.

However, having a proprietary software and a very restrictive rights management precludes any further interaction with the operating system that is not native.

According to their architecture, iOS has four layers:



Core OS, the operating system kernel. An evolution of OS X, the desktop operating system for the company, which in turn is based on Darwin BSD and therefore UNIX.

Core Services, contains the major services necessary for applications.

Media, responsible for providing the necessary tools for graphical interfaces and applications.

Cocoa Touch, which contains the Frameworks group needed to run applications on the system. This in turn formed by two elements: *UIKit* (UI) and *Foundation Framework* (core classes for handling objects and system services).

Regarding integration with HTML5, certainly this is a much more restrictive environment, as in the case of Android, has no SDK for this type of technology. Both applications ported to HTML5 native language as hybrid applications belong in the same range observed in the previous section (multi-character limitation in exchange for use of native APIs and possibilities offered by the Framework that acts as an intermediary). In WebApps, and due to the closed nature of the platform, all iOS navigation service must use the Safari engine, which makes Firefox is not present in the system, and therefore, there is no interoperability with applications.

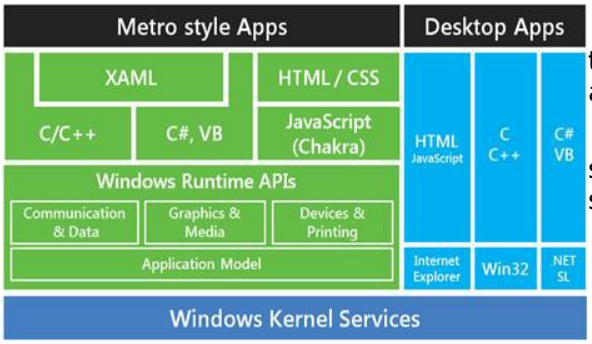
Google has a version of Chrome that does allow these applications generate "packed" from WebApps, but considering that today the access to most of native iOS APIs are closed to web applications.

4.3. Windows Phone

Although there is still terminals with Windows Phone 7 and 6 in the market, focus this paragraph on Windows Phone 8/8.1, the latest version of Microsoft's mobile platform, with backward compatibility between applications developed for earlier versions (although the architecture change) and increasingly more support for convergence of developments with his older brother, Windows 8/8.1.

In fact, part of its architecture (based on Windows NT) is shared by the two, such as kernel, networking, support for graphics and multimedia file system.

This is a mainly based on gestures, operating system so you might consider new generation, whose interaction is purely asynchronous operation. The decision to adopt as UI style block (I hesitate to call Metro since each bit will change the name ...), supported in choosing a good contrast and typography, makes Windows Phone a differentiating system very fast in loading and different compared to the market.



Course, has a very disadvantage to consider, is that compared to Android and iOS, the number of applications available is very low.

In the picture accompanying this text we can see the different languages used by Windows systems. Make it clear that one of the strengths of

Microsoft development suite is that it allows both work at the same level with HTML/JavaScript, and with C/C++ or C#/VB.

In addition, you can always use Frameworks like *Sencha*, *Cordoba Apache* or *PhoneGap*.

The WebApps are contemplated within the formal market, IE10 and support (both mobile and desktop version), but as usual, do not have all the permissions you may need.

Curiously, there is an application called WebApps that acts as a market of WebApps, using as a basis the application itself, and thus facilitating its discovery and installation. And besides, since November 2014, Visual Studio 2013 Community is open source (can be used for any development on any platform).

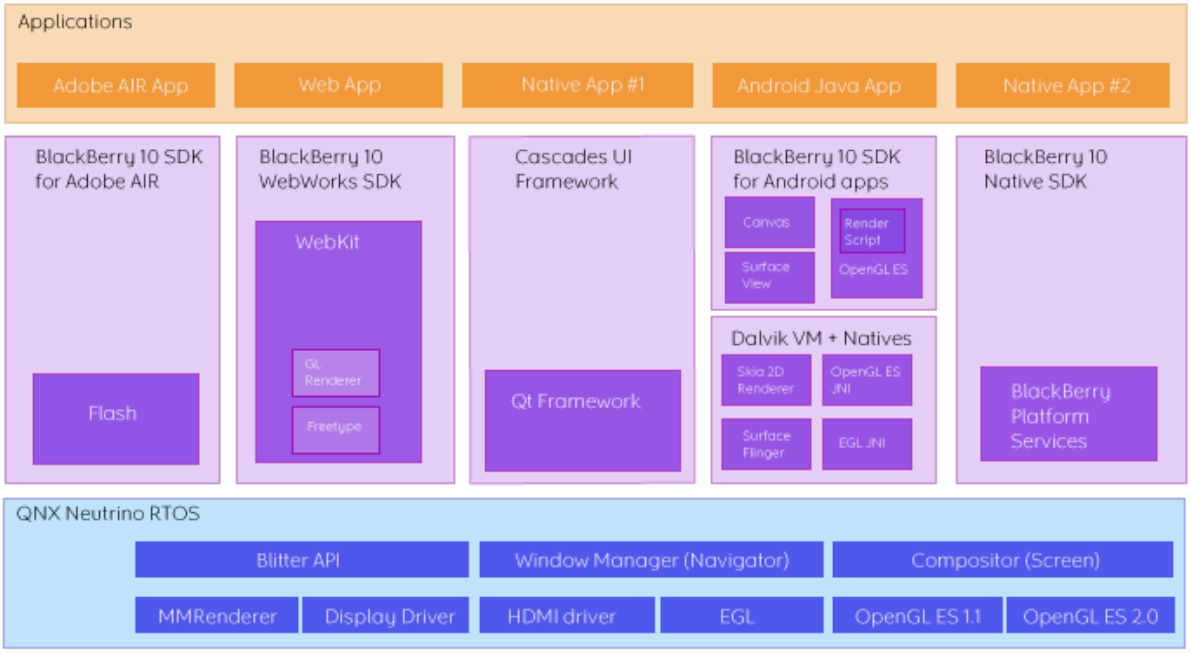
4.4. Blackberry

The version analyzed in this section is the last, BB10, by this mean, as happened in Windows Phone, completely change the paradigm of previous versions.

Blackberry 10 is a proprietary mobile operating system, developed by Blackberry (RIM above), and at first target was to clear the business world.

The blackberrys were an intermediate step between the figure of the mobile and smartphone. A device with a QWERTY physical keyboard and a large group of attractive applications added to the system.

However, a bad decision by the company pushed out of the way, relegated to what it is today.



Blackberry 10 comes to fight with this situation, they made eliminating those mistakes at the time with an operating system is based on interaction whose gestures, and with the opening of its SDK to multiple languages.

The kernel is an optimized version of QNX present in previous versions of the OS on which runs the rest of the system.

In blackberry 10 can install native applications developed with Adobe AIR technology WebApps applications and even Android apps, of course, after having adapted to their environment.

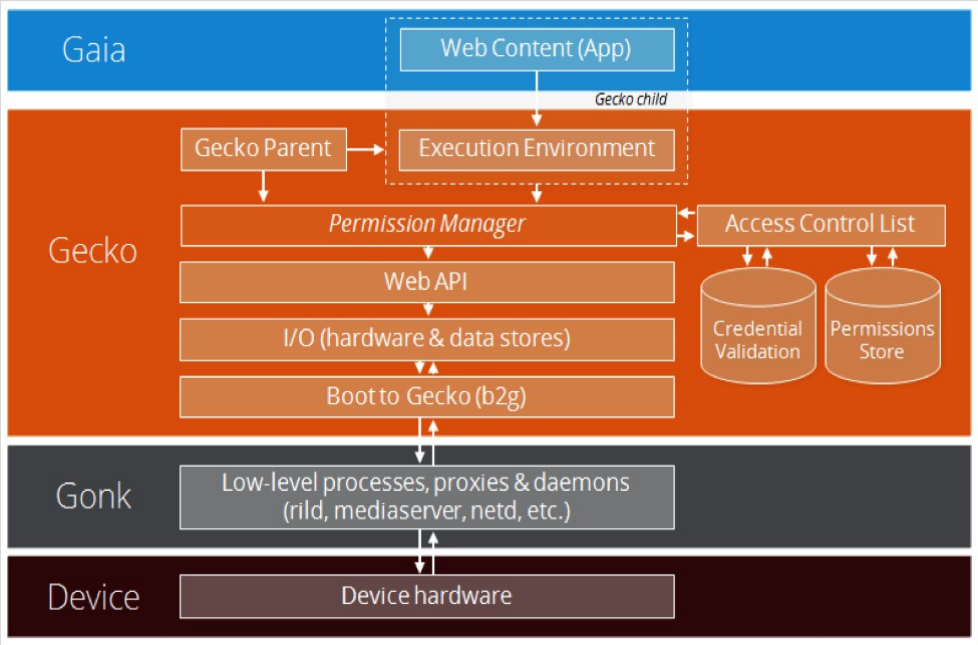
In the case that concerns us, the official documentation on HTML5 application development for Blackberry is the best in the sector.

Applications developed using *Cordoba Apache* covers, within the official SDK itself, so that the level of access to native permissions is secured.

4.5. Firefox OS

Mozilla's proposal is the most interesting of the market, and not only for being the only run by a nonprofit foundation, but also has a historical background to support net neutrality admirable.

Firefox OS is an open source operating system based on HTML5, which is based on technology developed by Mozilla, and thus points to the platform.



Firefox OS architecture is divided into three layers:

- **Gonk:** we are in the lowest layer, a sort of distilled Linux kernel, and acquires body thanks to a large number of open source libraries.
- **Gecko:** is the most popular rendering engine, and we can find in all Mozilla products. This means that an application can run Firefox OS in a terminal with Firefox OS, but also anywhere you can install the Firefox browser.
- **Gaia:** The visible layer, responsible for providing the interface, and based on HTML5. The biggest advantage that Firefox provides OS in this regard is that all communication with the different elements of a device (GPS, connectivity, SMS, ...) is made by and standardized APIs and other pending standardization. Therefore, for the first time in history, we have web tools for communicating with hardware elements being already covers the other browsers.

Needless to say, HTML5 support is beyond question. HTML5 is the native language of Firefox OS, the native language of the web, and therefore, will be present wherever there is a browser that can interpret.

As discussed in other sections, the applications developed for Firefox OS can be used as apps without making any changes in your code both in desktop operating systems like Android. Moreover, the idea behind the Mozilla foundation is that this system of democratization application does not have why pass only for them, but its interoperability is allowed with other markets and even (taking into account the appropriate security measures) is accessible from the own web service.

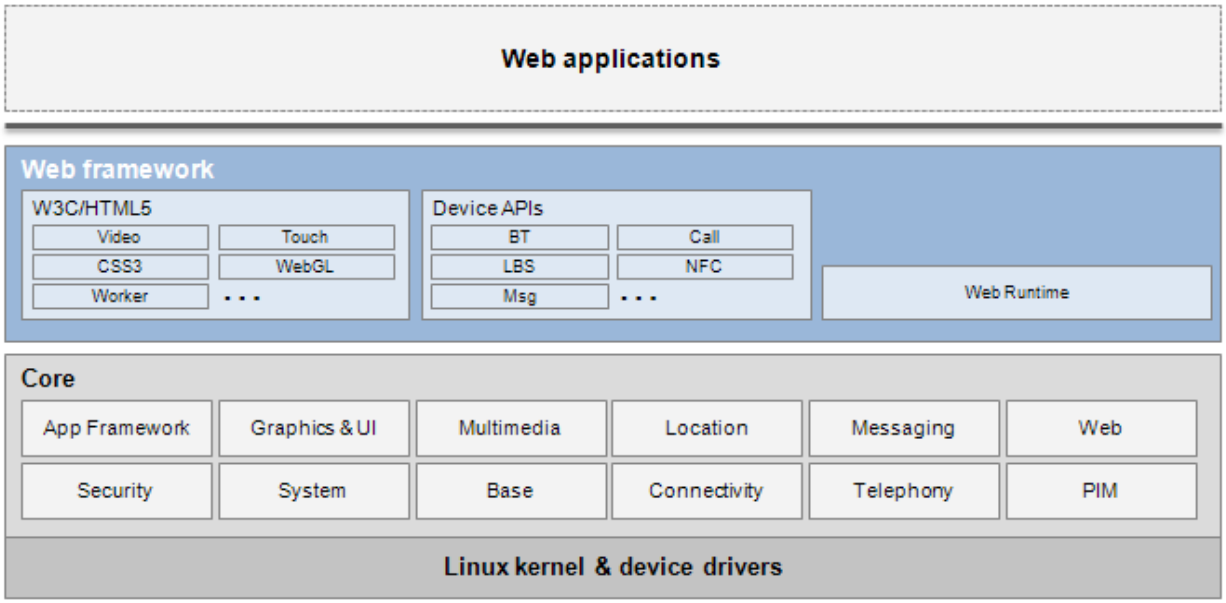
The ultimate goal is that HTML5 is the default platform market, breaking the current walled garden internet as once did with closed networks.

With this OS devices currently available in 13 countries: Brazil, Colombia, Venezuela, Peru, Uruguay, Mexico, Germany, Poland, Hungary, Greece, Spain, Serbia and Montenegro, and its expansion is expected for the rest of Latin America and Asia in the remainder of the year.

Mozilla's proposal is to attack the market entry in developing countries, with a very low cost devices (the latter presented, $ 25). Although the system has been initially designed for smartphones, and it is testing in tablets and even Smart Tvs and DIY plates.

4.6. Tizen OS

Tizen is an open source mobile operating system based on Linux which comes under the purview of the Tizen Association (formerly called LiMo Foundation), led by Samsung and Intel, and sponsored by the Linux Foundation.



Although it has a distant relationship with Meego, the operating system developed by Nokia and Intel primarily for netbooks, the truth is that Tizen has followed a quite different way, and as discussed below, Meego has finished as external project (and supported by Nokia) under the name of Sailfish OS.

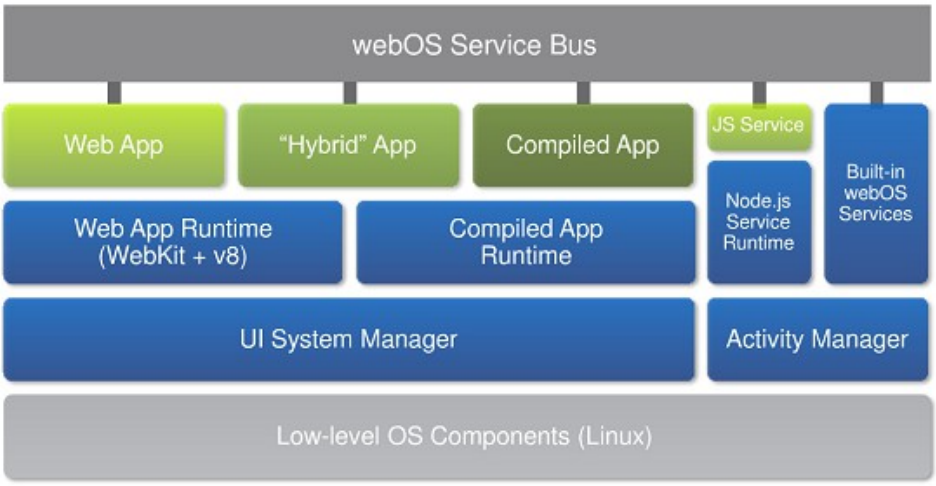
The operating system is aimed to cover different topologies devices currently still available in the new version of smartwatch Samsung and in some smartphones and tablets that have not yet been released (the first named Samsung Z, released in Russia along this year).

Among its strengths is having behind an industry giant like Samsung, basing its development of apps in HTML5 (the rendering engine is Webkit, Firefox versus Gecko OS) and be compatible (after some small changes OpenMobile ACL) with Android applications.

As happened with Firefox OS, HTML5 apps are native to the system, so you just know how the SDK to develop. It is therefore necessary to resort to intermediaries, although it is possible to use the port (for example if you already have a project in PhoneGap).

4.7. Web OS

The history of Web OS is the story of wannabe. Initially developed by Palm as a multitasking operating system based on Linux, was sold to HP, which eventually released its code and sell it again, this time LG, who has recently been used to smart TVs.



Like Tizen OS, has a Linux kernel and an interface developed in HTML5 that makes it easily exploited for different uses, as evidenced by the long list of products that has been used.

The truth is that your purchase for LG and interest to use it as a TVs SO close enough market to third parties. There is not much updated project documentation, some otherwise normal vision as this new product has been introduced in January of this year.

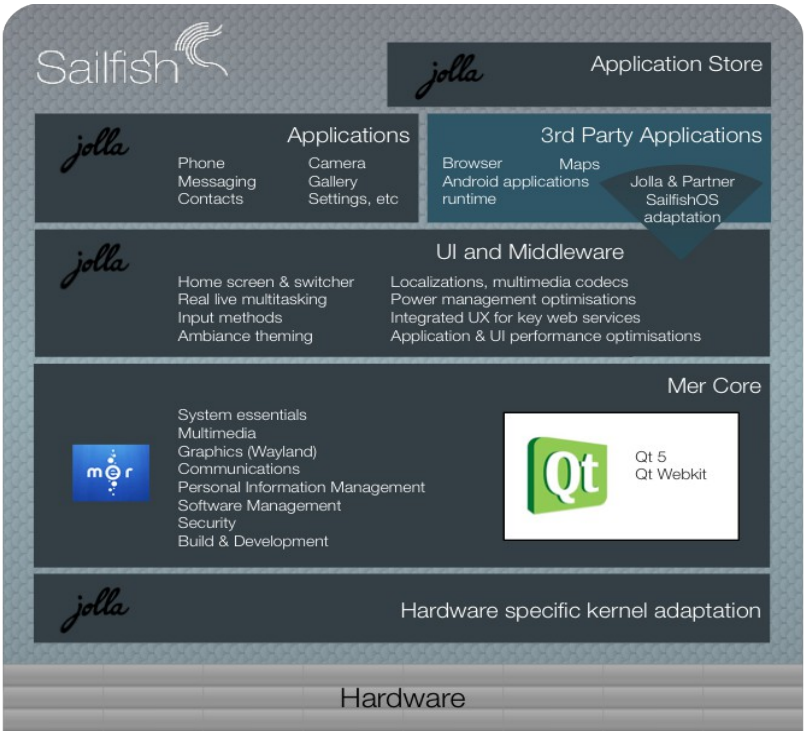
4.8. Sailfish OS

Sailfish OS is the logical evolution of MeeGo, an operating system developed by Nokia, based on Linux and in order to attack the market of smartphones and tablets.

When Meego was dropped, some of their workers left to form Nokia Jolla, one covered by the Finnish company, which has renamed the system as Sailfish OS.

The operating system has its own programming language (Qt / QML), although accepted through their respective sdks developing apps in HTML5 (Cordoba Qt) and Java (Android apps).

The architecture is similar to what we have seen in most mobile operating systems: Linux kernel, a layer of components (based on Mer), with the interface middleware and applications.



4.9. Ubuntu (Phone)

The promise of Canonical (the company behind this distribution of Linux) was to reach the true mobile convergence / desktop with the new Ubuntu.

Of those words until now already last year and a half, and the project is still in alpha stage.

The operation, now available for Nexus Terminal system provides gesture-based system that inherits the look and feel of its desktop counterpart. The interesting part of it was this supposed convergence in which a user could use your smartphone, get home, connect it to the screen and keyboard, and use Ubuntu as such, something that today is not yet possible.

For initial testing, also lacks polish multitasking because it is impossible for the user to remove apps from memory, just to make a dent in the battery of the device.

Regarding integration with HTML5, is assured. Ubuntu has its own development framework, but allows using sdk perform the same in web language.

It is therefore to be patient and hope that the project will not be forgotten.

5. Conclusions

As we have seen, all proposals mobile market (even those that are still to come) have in one way or another support HTML5 developments.

The idea behind projects like Firefox OS is to free the sector, creating a community based on standards, able to tailor the OS to their possible different uses. The end user you can access information wherever you are, regardless of the device you have on hand.

Thereby enrich human knowledge, accessibility and communication.

The current market has two big winners: Android and iOS, based on the same paradigm (control market applications). The next step would be to decentralize the markets, so that the user has more options to choose from and more freedom to carry your tools from one to another system.

And the most immediate way we are using HTML5, a platform that has everything that society needs to manage information. There will still be proprietary developments, and closed operating systems (in fact the more the more competition and therefore more market developments), but ultimately, enable, as today most systems are enabling the development of web services ensures that all the value is not lost when a fall or other platform.

The instead of two, there are four or five, even more force to HTML5 is the engine of communication between them. A set of platform technologies, with a medium / low curved learning and the tools to interpret the scope of any device today.

The objective of this study is not dogmatic in the benefits of HTML5, but display a market reality. The same that are leading companies like Microsoft or Canonical, a priori, very interested in the benefits of the web to promote their booming systems. At that foundations like Mozilla or Linux Foundation are behind standards-based HTML5 platforms. By showing interest in Latin America by betting on the web to reduce costs and make the leap to smart devices in developing countries.

The market calls for convergence, and the only platform that is in power to offer it universally is HTML5.

Communication that tomorrow can be done in the same way from any device, keeping it to a necessary level of anonymity and privacy. Companies must adapt their business model to this new paradigm, perhaps offering value with a service layer with advantageous customer agreements.

It is not, therefore, that all use Firefox OS or OS Tizen, but all have access to the benefits of HTML5 as a platform. Because society is so demanding, and the market understands is the right output for this new generation of technology.

6.- References

APPIO. «Tipos de Apps: Nativas, híbridas y WebApps». Retrieved June 8, 2014.

Blackberry Developer. «Native SDK for Blackberry 10». Retrieved June 15, 2014.

Comscore. «SO Mobile market». Management & Productivity. Retrieved June 8, 2014.

Developer Economics. «Developer Economics Q1 2014». Retrieved May 8, 2014.

F. Iglesias, Pablo (July, 2013). «Plataformas móviles y lenguajes web: Realidad del mercado». PabloYglesias. Retrieved June 8, 2014.

F. Iglesias, Pablo (March, 2013). «Vivimos una democratización de sistemas operativos». PabloYglesias. Retrieved June 8, 2014.

F. Iglesias, Pablo (April, 2013). «De camino a un mundo donde el usuario no será dueño de sus dispositivos». PabloYglesias. Retrieved June 15, 2014.

F. Iglesias, Pablo (April, 2013). «Google en busca de un único ecosistema». PabloYglesias. Retrieved June 15, 2014.

F. Iglesias, Pablo (December, 2012). «Fastbook, demostrando que HTML5 sí le planta cara al lenguaje nativo». PabloYglesias. Retrieved June 15, 2014.

F. Iglesias, Pablo (January, 2013). «Desarrollando para Firefox OS: Primeros Pasos». PabloYglesias. Retrieved June 15, 2014.

Fried, Ina (February, 2014). «Mozilla says new chip $25 smartphone». Recode. Retrieved June 15, 2014.

Gaia Program. «Enterprise Structure & Organization». International Business & Change Management. Retrieved June 12, 2014.

Game Developers Conference. «GDC Q1 Mobile Development». Retrieved May 18, 2014.

Grieve, Andrew (January, 2014). «Run Chrome Apps on mobile using Apache Cordoba». Chromium. Retrieved June 8, 2014.

Kantar. «Mobile SO 2014». Retrieved June 15, 2014.

Osmar, Gabriel (2012). «Monografías: Sistemas operativos en dispositivos móviles». Northeastern University. Retrieved May 8, 2014.

Pro-Chile, New York (2013). «Estudio de mercado plataformas móviles». Retrieved May 19, 2014.

Pastor, Javier (March, 2013). «Desarrollo en plataformas móviles: Así está el mercado». Xataka Móvil. Retrieved June 8, 2014.

PhoneGap. «Make easily apps with HTML, CSS and JavaScript». Retrieved June 8, 2014.

Salesfore. «Native or Hybrid app». Retrieved June 15, 2014.

Sencha. «The State of HTML5 Developer in Enterprise». Retrieved May 14, 2014.

Shapiro, Matthias (February, 2013). «Gettin started with Windows Phone 8 HTML5 Apps». MSDN. Retrieved June 15, 2014.

Titanium Appcelerator. «Appcelerator Platform». Retrieved May 14, 2014.