RESEARCH ARTICLE OPEN ACCESS

Mobile Platforms And Apps Cross-Platforms Development Tools

Serena Pastore*
*(Astronomical Observatory of Padova, INAF, Italy)

ABSTRACT
Apps are the new model of application software that originally developed for the software distribution to mobile devices, is emerging also in the other categories of devices. User finds the apps in a custom website called the store, whereas apps in the form of an auto-installing software package can be downloaded freely or after paying a fee, and easily installed and configured. The mobile platform environment shows a plethora of systems considering the large number of mobile hardware and software manufactures. The apps regardless of the type of category they belong (i.e., social, educational, games), are distinguished in native, web or hybrid apps depending on the tools and technologies used to develop. The paper analyzes mobile operating systems from a software developer perspective focusing on cross-platforms tools in order to propose actual and future platforms where to deploy apps. The work is part of a research aimed at designing and developing apps for science outreach and education. The selection of mobile platforms target influences the tools needed to develop apps. Cross-platforms development tools allow greater distribution to different platforms, but have some limitations and cannot be the only methodology adopted. The choice of mobile development framework depends both on the type of the app and the mobile platform.

Keywords **-** mobile apps, mobile platform, software development kits, apps stores, mobile cross-platforms tools

I. INTRODUCTION

Since the explosive growth in mobile device adoption [1], the specific application software targeted to mobile devices known as apps are gaining popularity. With the term mobile device, we refer mainly to tablets and smartphones considering them as the best-selling devices, even if the term encompasses a number of different devices. Mobile devices have become the primary entry point for all types of users, who are using smaller and light devices to do the same or more as they do on the desktop system. Internet connection or mobile Internet [2] from such devices is more efficient thanks to wireless data transmissions both through wireless networks and cellular networks that allow higher band rate. Enhanced computing and storage facilities allow mobile software to include complex multimedia content as well as audio and video content. Some apps could take advantage of Internet connection and use the web platform as an environment for apps execution. When using the app by means of support distributed computing platforms, apps are known as web apps [3]. A primary categorization of apps lays on technologies involved in their development. Apps can be distinguished in web, native and hybrid meaning that they use web technologies, programming languages used by mobile platforms and specific cross platforms tools that are able to mediate on these two methods. The distribution of mobile software passes through the "app store" a custom kind of an e-commerce site where a user could find the software he/she needs distributed with different types of licenses both payment and free. Given the presence of a single container, apps are categorized by scopes (entertainment, educational, social) and the mobile platforms [4] (iOS, Android, Windows, and Blackberry OS). Regardless the scope and the underlying operating systems, mobile apps have some peculiarities since targeted to computer systems that show limited capacities, are equipped with some type of Internet connection and could show specific hardware as sensors, camera, and Global Positioning System (GPS) receivers. Usually mobile devices, even if different as regards the scope and functionalities (e.g., tablets can be seen as processing devices while smartphones' main purpose is telephony), offer limited computing and storage features, small screen size, input device with a touch and without the usual pointing devices and the integration with sensor devices. A plethora of mobile operating systems [4] has been developed for managing the different mobile hardware for the two main computer systems categories (tablets and smartphones), most of them tied to mobile devices manufactures (Nokia-based systems such as Symbian OS or Blackberry-based systems). But mobile platform providers include software companies or organizations such as the case of Android OS, Windows Phone or the new proposals for operating systems (e.g., Tizen OS or Firefox OS, Ubuntu mobile) that developed mobile platforms suitable for different categories of hardware. The paper analyzes mobile platforms from a software developer perspective examining the various existing and development platforms, the type of apps that could be developed for such environments and the tools available for such purpose that could help to develop for a cross-platform purpose. In fact the presence of many mobile platforms, even if the actual market reduces the number to two systems (iOS and Android

OS) is characterized by numerous frameworks (i.e., Windows-based, Blackberry OS, Bada OS). Developing an app for a single system excludes, although with different percentages, the number of potential users. The software developer needs to understand to what mobile devices it is worthwhile to develop an app. Moreover if the choice regards more than one category, the developer should consider the design of native or web apps meaning the use of mobile platform developer software rather than web technologies. Since the presence of mobile frameworks helping the cross-platform development, advantages and disadvantages of these software should be examined.

This is a research carried out in order to understand what mobile platforms and tools are to be taken by considering actual and future platforms and choices to make in offering educational and outreach app. Our background is a research institute involved in Astrophysics and this paper describes the study made, to choose a platform where to develop education and outreach Astronomy apps.

The remainder of this paper is organized as follows: Section 2 gives an overview of apps features according users' behavior recent statistics from many business companies. Section 3 describes the various, mobile operating systems. Section 4 presents the methods of using cross-platforms tools in app development. Conclusion and future work are given in the last section.

II. MOTIVATION FOR APPS' DEVELOPMENT

Apps are a kind of software programs that perform specific tasks for the mobile users. Usually an app is downloaded by a store to be installed and run in a handheld device. Sometimes for the specific type of apps (e.g., those related to social networks) the execution could refer to the web platform. Initially, most famous apps were born in the context of social network as an easy way to access to social tools like Facebook or Twitter from mobile devices, but are becoming a kind of software that could perform every kind of task both for personal or work activities. From a design point of view, an app is characterized by a graphical element or widget that could be easily started with a touch. This feature allows the user to use the application software in the same way despite the mobile platforms he/she uses. But apps' structure is generally rigidly defined. For example, an Android app has a pre-defined structure with code and resource organized into a number of folders, and the layout is described by different XML files that help to define the widgets (buttons, text fields and so forth) that combine the application. Anyhow an apps should be designed to solve a real problem and not as a simple add-ons. In developing an app, the developer should consider the scope, the technologies and the mobile platforms. The app is becoming the new form of application software considering that mobile devices market (known also as ultra mobile market) that includes the combined market for laptops, tablets, and smartphones, are constantly growing. According to several market analysts companies, including Gartner, this market is still growing at the expense of the traditional market of computing systems. It is estimated, for example, that the world market for tablets the smartphones devices total a 2.4bn units in 2013, an increase of 9% compared to 2012. The estimate of the sale is shown in Fig. 1.

Device Type	2012	2013	2014	2017
PC (Desk-Based and Notebook)	341,263	315,229	302,315	271,612
Ultramobile	9,822	23,592	38,687	96,350
Tablet	116,113	197,202	265,731	467,951
Mobile Phone	1,746,176	1,875,774	1,949,722	2,128,871
Total	2,213,373	2,411,796	2,556,455	2,964,783

Fig 1: Estimates of mobile devices sales in thousands of units (Gartner April 2013 Source)

Software applications typically have an essential role in the sale of electronic products, as a product makes sense if there is application software that allows to use it. Even in the case of mobile devices, the success or failure of a product depends on the number and type of apps that are available for that specific hardware.

Considering these estimates, the apps are the new frontier of application development. An app differs from traditional software applications to both the media which is used for the conception that embodies. It is an application that needs to be performed on devices that have unique features and therefore must be light, essential and fast.

Considering the fragmented landscape, initially a developer focuses on the major player in the market since usually only installed apps get used. Regardless of the operating systems, the basic features of the app are that should be simple, with a clear navigation that interacts with the touch, and when there is the need of Internet connection, the user should be advised since a lot of users use Wi-Fi connections. The technical approach is to build a quick prototype of the app that could give a mobile-optimized experience.

By considering the goal of the app, next to a classification according market categories that, in Fig 2, we can see associated to an Android-based system (e.g.., business, tools, education), we can see terms such as social apps, smart apps, and so on that could identify the role of an app.

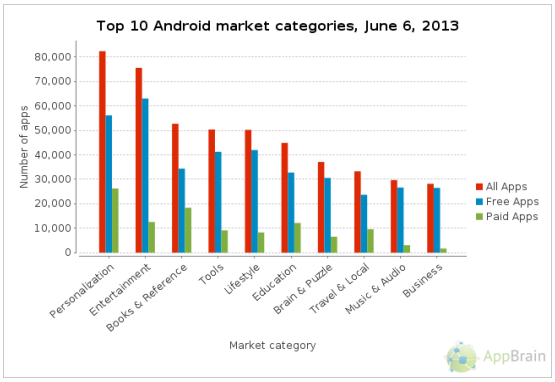

Fig 2 Top Android Apps category (AppBrain Source)

The new class of apps which embody the definition of "smart" has the meaning that the apps could be personalized, contextual and proactive. Smart apps don't all have to be a virtual assistant of personal productivity apps. Fig. 3 shows the major activities carried out by mobile Internet users in the world. We can see how users use the mobile device when connected to the Internet for a lot of activities that range from accessing news to use email or navigating on the web.

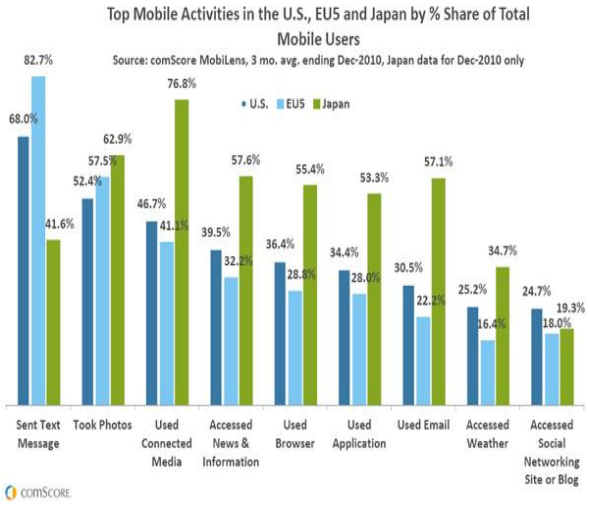

Fig. 3: Mobile activities (comScore Source)

The role of apps that not fall between these major activities, must be able to attract users. Apps should have such features of customization for users since executed in devices the use of which is for personal use rather that professional. All the apps should consider that mobile devices have some limitations as regards processing power, memory and storage compared to the desktop computer that cannot be forgotten or ignored otherwise apps perform badly. Furthermore, the specific presence of sensors (e.g., the GPS receiver and the camera) could be exploited to develop apps that combine, for example, the recognition of a location with the specific interests of the user. An example of a type of apps that combines these two aspects are the so-called augmented reality [5] app. Such app takes advantage of sensors in mobile devices to enrich the real world seen through the devices with the digital world and thus what can be derived from a user with the application software that uses. Augmented reality is a term that describes the enhancement of real-world objects or views with computer generated actions. It is a technology that combines virtual reality with computer graphics. An augmented reality app can include several functions, be they for interaction or display. There are different kinds of apps also in this context [6] (i.e., those called augmented browsers that consist in using a rich browser able to display information about a real object). These apps seem to be very interesting in the context of educational or outreach apps. We could, in fact, describe a particular real object (i.e., a telescope or an instrument both ancient that modern) by mean of a code (bar code or similar) on the base of the object. The app developed could show a picture of that object with a fully interactive description. This could be extended to show a map highlighting similar objects allowing to follow the map to their locations. Such app displays lots of data, moves it continually around the screen, and resized components, resulting in using the device's processing power to the maximum. It needs a lot of time and efforts in coding to minimize the amount of processing required. But it allows having a custom user experience. These types of apps need a specific type of development that depends both on the mobile platform of reference and the types of apps.

2.1 Apps Types: native, web and hybrid

Apps are categorized in native, web apps and hybrid. The benefit of the hybrid architecture, which combines the portability of web apps that mean apps developed with web technologies such as those related to the HTML5 framework [7], with a native container that facilitates access to native device features, will appeal to many developers. The need of context awareness in mobile apps has increased the capabilities of mobile devices, causing developers to consider both hybrid and native architectures. For applications to leverage location information, notification systems, mapping capabilities and even on-device hardware such as the camera, some forecasts from research firm such as Gartner [8] say that, by 2016, more than 50 percent of mobile apps deployed will be hybrid. Native apps offer higher performance, but the need of various versions of the app to serve the users because different versions must be made for each type of device or mobile platform, drives developers to consider a hybrid approach. HTML5 seems to provide interesting capabilities such as offline apps execution and animation-rich tools that allow a web app a good user experience, especially for some categories of apps.

Given the current proliferation of mobile devices, an app developer should create mobile roadmaps, and strategies for the content. Table I summarizes the main features required by the app and the different methods. It shows how the method chosen depends on the type and the purpose of the

app. Since the plethora of mobile platforms, even if the market seems to reduce mobile operating systems to few platforms (Android and iOS, Blackberry OS or Windows Phone), there is a necessity to find a way to build content or functionality once and deploy it on a variety of different mobile platforms.

Apps types and Features			
	Native	*Mobile web*	*Hybrid*
Internet access	Not required	required	required
Performance	v	x	x
Hardware access	v	x	x
OS access	v	X	x
Hw and platform dependent	V	X	x
Installation	Must be installed	URL link	Local or store installation
Distribution	App store	Not required	Not required
Updated	reinstallation	simple	reinstall

Table I: Comparison between different app development methods

There are many solutions that you can take to develop a cross-platform app. The different frameworks can be more or less complex, and have from low to high adoption rates and costs also linked to the technical skills required. However, it is necessary to decide when is the best time to use cross-platform solution, since this option has some requirements respect a simpler approach that is developing for a native environment. In any case before this choice, it would be better to know the fragmented ecosystem of the mobile platforms in order to understand the features of the existing and developing mobile environments.

III. THE ECOSYSTEM OF MOBILE OPERATING SYSTEMS

The mobile environment is a fragmented landscape. Mobile operating system (OS) are composed by software modules that should be customized in order to manage mobile devices features. Different kind of devices could be categorized as mobile regardless their primary functionalities: tablet, laptop, mobile phone, smartphone, game portable console, e-book readers and so on.

Fig 4: Mobile categories devices

We could define a mobile device as a system that should be hand portable and usable in mobility situations "everywhere, anywhere and anytime". Usually mobile devices are equipped with reduced processing capabilities (e.g. ARM Cortex CPU), low storage capabilities even if some of them could be externally expanded, small screen sizes, touch input device and no keyboard or other pointing devices. Furthermore, they are equipped with added functionalities such as sensors (especially in the smartphone category) or other features such as cameras, GPS receivers. They offer a network connectivity that according the main function of the device could allow Internet connectivity by using a Wi-Fi adapter or a phone card. Fig. 4 shows a categorization of mobile devices according the main function: general purpose devices, dedicated devices and phone devices. Even if there are these different categories of devices, actually mobile devices are identified with two main types: tablets and smartphones that are the users' most used devices both for Internet, personal and business activities. All of these devices is equipped with operating systems that share main functionalities with a computer operating systems, but should be optimized to the specific hardware of mobile devices. In this way, operating systems are composed by a kernel, a runtime framework for the execution of the application and user interface as the high-level OS, but each layer is optimized for the specific hardware and the main functionalities of mobile devices.

Mobile operating systems are described as layered with specific software module that could manage the hardware and the critical features of mobile devices such as power capabilities in order to give access to them in a secure way for the apps. Among the plethora of mobile operating system, in our study, we distinguish three categories of mobile OS as showed in Table II: those that are actually supported by devices and with different percentages occupy the real market (i.e., iOS, Android, Blackberry OS); those that are disappearing or are already disappeared as having non supported devices and those who are appearing in the market but have not yet a mobile device or are not totally available all over the world (e.g., Firefox OS). The success or otherwise of a mobile platforms depends on the support of the other main drivers of the mobile environment that are hardware devices manufacturers, mobile network operators and mobile application developers.

Hardware support	*No longer hw support*	*Not yet hw support*
iOS (iPhone, iPad)	Symbian OS	TizenOS
Android	OpenwebOS	FirefoxOS
Windows Phone /Windows 8	BadaOS	Ubuntu Phone
Blackberry OS		Salfish OS

Table II: List of mobile operating systems

Mobile OS as those born for phone devices offered a closed environment related to the device manufacturer. But with the market explosion of such devices, also software vendors and open source companies have developed operating systems versions customized on mobile devices. This is the case of the Windows Phone or of Android OS that is a system that in few years has reached an incredible success. Actually there are operating systems that:

- shows an unique hardware and software owner (Apple or Blackberry products and in part Nokia products);
- are equipped with proprietary software systems (e.g., Windows) even if on different hardware platforms (Nokia, HTC);
- are equipped with open source software (e.g., Android) and supported by various hardware manufacturers (Samsung, LG).

Market researches in mobile operating systems, prove as the Figure 5 shows in a research taken from IDC, that Android and iOS occupy the market reaching the 92,3 percent of both tablets and smartphones operating system. In any case, the market is quite volatile, considered that the success of Android OS is exploded in just two years also thanks to the hardware support by Samsung. There are systems that face in the market as well as resist niche solutions such as Blackberry. In these market sector, for example, Windows Phone has surpassed Blackberry, and Android OS and iOS exchange rates when you consider smartphones or tablets devices. Each operating system has its features, however, when developing apps it is necessary to consider all these environments even if most users actually prefer Android or iOS system-based.

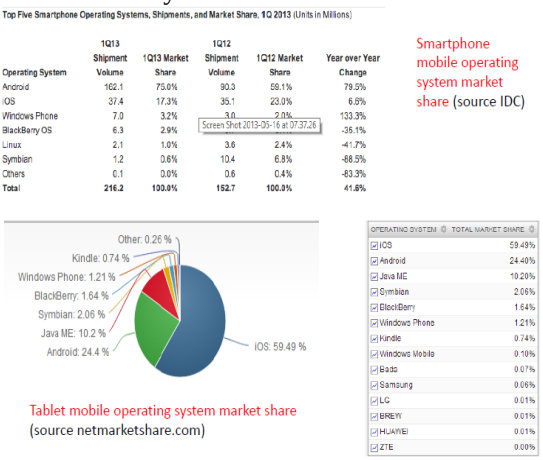

Fig 5: Mobile operating system market share

The situation can change quickly. It is so interesting to examine the new proposals of operating systems such as TizenOS, Firefox OS or Salfish that seem to be promising. We consider the main features of such operating system in order to understand the underlying environments and the tools needed to develop an app for these environments. Each platform support both native apps that are developed considering the operating systems modules and web apps that are apps based on HTML5 framework. When analyzing the main features of mobile platforms we made a comparison between different systems.

3.1 The iOS, Android OS and Windows 8

We made a first comparison between the software architectures of Apple iOS [9], Android OS [10] and Windows 8 OS [11] in the mobile version. These three mobile platforms are operating systems currently in constant development and are supported by different hardware manufacturers both in tablets and smartphones categories. Fig. 6 shows how the main components of such operating systems are similar. Mobile OS can be structured in three main layers, as the Fig. 6 shows. The lower layer is the kernel that operates in direct contact with hardware managing it and providing utilities for the upper layers. The kernel is the core of the operating systems and act as an intermediate between the hardware and the application software in order that applications could use the hardware in a secure way. The intermediate layer includes a framework with libraries and runtime environments for app executions, that interacting directly with the kernel, could take advantage of the hardware. Finally, the last layer is the application layer where resides application software both as native software that web-based software. iOS and Windows 8 are proprietary systems developed respectively by Apple and Microsoft.

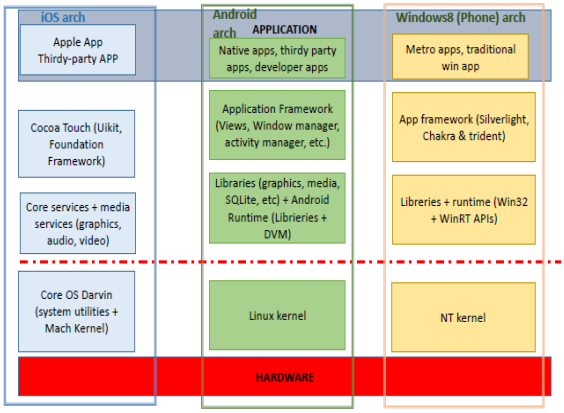

Fig 6: A comparison between iOS, Android and Windows 8/Phone operating system architecture

Luckily, also to promote apps development, such companies provide the software development kit and tools. iOS were developed by Apple as an OS adapted to different Apple devices (the so-called i-device that is iPad or iPhone) that is, however, a customized version of the OS X system that equips Apple computers. It presents an open source Unix-based kernel called Mach kernel and a proprietary graphical interface (Aqua). Android OS represents the first open source mobile platform that includes not only the operating system but a middleware and applications. It was developed by a group of companies, of which the most famous is Google, as an open source based on the Linux kernel. The interesting

layer is the application runtime layer. Finally Windows Phone and the Windows 8 version for tablets, are the customized version of the main Microsoft Windows 8 for mobile devices and with this version of the OS share most of their functionalities. From apps developer perspective, the most interesting layer is the application framework that presents the runtime environment where applications are executed and could be linked to the various libraries related to the core and media services. In the iOS system, the native apps that could take full advantage of the hardware are written in the Objective C language, a proprietary object programming language that is, however, compatible with the standard C language. This layer includes all useful services and libraries (e.g., the graphical libraries such as OpenGL [12]), and these are the software tools to exploit the hardware features of mobile devices. The runtime Android that is used by all types of Android apps, is composed by the core libraries and by the Dalvik virtual machine (DVM) where apps are executed as in a secure sandbox. The DVM is similar to the Java Virtual machine (JVM) even if it is optimized for mobile devices. Native Android apps should be in Java language, and yet all apps are executed in the DVM container. Also, Windows Phone offers various runtime environments both for mobile or metro apps (the so-called Win Runtime or WinRT) and Windows traditional desktop-based apps or win apps (the Win32 framework) together with the web-based framework app that includes the engine for rich web applications (e.g. Silverlight, Microsoft JS-based applications such as Chakra and Trident). As explained in Fig. 5, all the three systems show an application layer that includes the apps as distinguished as native or web or third-party apps, each one taking advantage of the underlying mobile operating system according the approach used for development.

3.2 The interesting mobile operating systems: OpenwebOS and Bada OS

Other interesting solutions, adopted however on few devices, are the OpenwebOS and Bada OS. The main structure of this two systems, shown in Fig. 7, present the same three main layers of the previous systems. WebOS was a system implemented on HP devices that gave rise to two projects: OpenwebOS [13] now sponsored by LG see as a web operating system and Enyo that is a Javascript library.

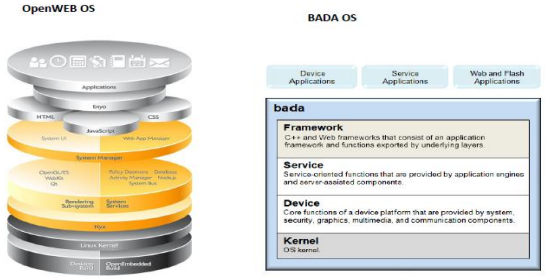

Fig 7: WebOS vs. Bada OS architectures

Open webOS is based on Linux kernel and has a layer of portability of the platform above the kernel (known as Nyx) that isolates the rest of the system dependencies hardware and core system. The application layer is based on the main web technologies (HTML, CSS and JavaScript), on top of which stands the Enyo application framework.

Despite the good features of this operating system that allows you to create web applications, HP has preferred to focus on Android, and then pay for licenses for their Android devices and sell the license to LG that has invested in the platform as an open source project. Probably so, given its limited distribution, it will remain a niche OS or for specific devices. Bada OS [14] is an operating system for low profile smartphones developed by Samsung and released as open source since 2011. The devices with Bada take the name of Wave, as well as Android-based devices get the Galaxy brand. Bada architecture (Fig. 6) consists of four layers. A kernel layer includes either a Linux kernel and a real-time operating system (RTOS) or the Linux kernel; the device layer provides core functions such as graphics, protocols, telephony and security and the service layer, that is similar to an application framework layer, offers service-oriented services (there is a server called Bada Server that supports them) that are provided by an applications engine. The final framework layer provides the C ++ and web frameworks that offer the features for apps life cycle management, event handling and apps controls. Samsung has, however, announced the end of the Bada operating system in 2013, bringing together some of its features in a new system called Tizen OS.

3.3 The new systems: TizenOS vs Firefox OS vs. Salfish OS and Ubuntu Phone

The current market for mobile OS devices sees the duopoly of Apple iOS and Google Android, but considering that mobile devices market appears to be a growing market, new mobile platform providers are trying to conquer the shares. Many experts wonder if the market is actually ready to receive new operating systems for mobile devices. To survive in the market, and OS should be attractive to users and competitive than the others. In any case, the excitement in the business allows the consumer a wide range of possible options. We will examine four mobile platforms, Salfish OS, Tizen OS, Firefox OS and Ubuntu for mobile OS that are appearing in the last period, even if in some cases not yet supported by mobile hardware. A comparison between the

architectures of the three first OS is shown in Fig. 7.

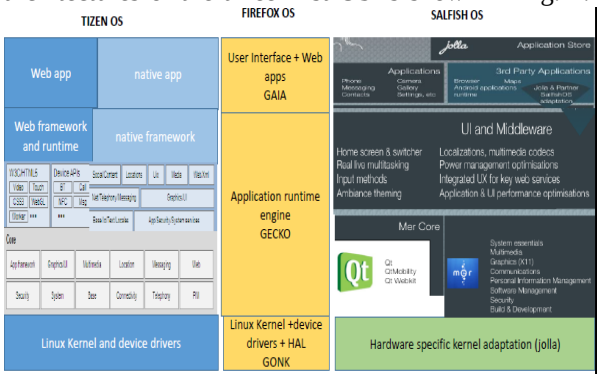

Fig 8: TizenOS vs.Firefox OS vs. Salfish OS architectures

Sailfish OS [15] is platform created by a team who now works at La Jolla, a Finnish company founded by Nokia employees. It is proposed as a modern, independent and alternative mobile operating system. Its architecture is based on the MeeGo [16] open source OS, a Linux-based platform developed to be executed on different categories of devices. The same MeeGo was created as a function of two projects: Nokia's Maemo and Intel's Moblin. Maemo. The first is the development platform of Nokia mobile devices (tablets and smartphones), while Moblin is an open source project started in 2007 dedicated to applications for mobile portable devices. The use of these technologies, includes in Salfish OS some phone's features such as power management and connectivity that are built-in and optimized for embedded environments. The core of the OS is based on the Mer project [17] that is based on other frameworks such as the Qt [18] and its language (Qt Modeling Language, QML) for the graphical user interfaces and the HTML5 framework. It is interesting to notice that Salfish OS includes a runtime environment for the execution of Android applications based on open source libraries and web development tools. Sailfish has a minimal and clear user interface that Jolla owner intends to use to differentiate its mobile devices from other competitors. Jolla's first device running Salfish OS was announced to be available probably at the end of the year: actually such smartphone is available for preorder.

Tizen OS [19] is an open source software platform based on standards consisting in an OS and some apps. It is created to be supported on several devices that range from mobile devices (smartphones and tablets) to specific devices such as Smart TV. The project is developing under the Linux Foundation, the non-profit consortium founded in 2007 dedicated to supporting Linux development. Tizen OS shows the same architecture structured in different subsystems: a Linux kernel, the core system that provides a set of open libraries and an application framework to support web and native applications. Samsung has confirmed that it will launch on the market different devices based on this new operating system. A prototype has already been presented and probably the first devices will come out towards the end of 2013. Many analysts [20] argue that Tizen OS might cover only a 2% market share by the year 2013, but a 5% by the end of the following year.

Firefox Mobile OS [21] is the operating system developed in early 2013, born to be supported on different mobile devices by Mozilla Foundation, a non-profit international organization that also developed one of the most used web browser. Its architecture consists of three main components (Fig. 7): a base layer codenamed Gonk that includes a Linux kernel, the hardware abstraction layer (HAL) and devices' drivers; an application runtime layer (Gecko) that is the engine used by other Mozilla software such as Firefox or Thunderbird and provides an implementation of open web standards; and the web application layer. Some hardware manufacturers (Alcatel, ZTE, Huawei and LG) are working with Mozilla developers, and the first smartphones equipped with such OS have been launched into the market initially only in some countries (e.g., Spain, Poland, Hungary) supported by leading mobile network operators. The main goal of such OS is to break down the differences with native applications using web standards to make the operating system software. The last layer is Gaia, the graphical interface of the system is, in fact, entirely programmed as a web application.

The last emerging mobile platform taking into consideration is Ubuntu OS for phones and mobile devices. Ubuntu Linux is a GNU/Linux-based distribution born in 2004 as a customization of the Debian distribution. Ubuntu become famous for its ease of use and its user friendly user interface. Originally geared towards desktop use, it is then especially for other classes of computing systems. The software is free under the GPL, but there is a support company (Canonical), which gains on additional services related to operate systems and software in general. Some analyzes [22] show that the distribution Ubuntu is among the most widely used Linux distributions on the market. Ubuntu provides customized versions of the distribution depending on the platform and in specific there are released for mobile phones (Ubuntu Phone), for tables, Ubuntu for Android and Ubuntu TV. Canonical, therefore, aims to create a unique operating system that can be used by all mobile devices thanks to the "micro variants".

Ubuntu for phones has been officially unveiled in 2013, as the version for tablet, with the support of entry-level and high-end level hardware support. All Ubuntu distributions are based on the Linux kernel being of derivation of the Debian distribution and thus represent an ecosystem for which the writing of applications is made for each type of device having the same system software. Like other systems, the architecture is represented in layers with the upper one consisting in native and web apps. To date there is still in the market for devices with these operating systems: it was stated that will go on sale early next

year (2014) or probably by the end of 2013. In the meantime, a version was released for developers installed on some devices of the series Galaxy Nexus. Later it will be released for another 20 devices (e.g., Samsung Galaxy, HTC One, Galaxy Nexus, Sony Xperia). Canonical is also developing a version called Ubuntu for Android (a term that causes confusion because Android is another mobile platform) meaning that this OS will allow devices with Android hardware to run its Ubuntu system. The idea is to have a co-existence of the two systems that, based on the same Linux kernel, can share it. The user could then use Android and its applications using the device as a phone and Ubuntu when using the device such as a PC. The problem of groped to capture part of the market remains connected to the need to obtain the support of at least one of the major producers of mobile devices and the need to ensure the support of the largest possible number of telephone operators. This is needed to obtain a commercial success and not remain a niche business for server systems or users connected to the world open.

IV. THE ROLE OF MOBILE FRAMEWORKS

Apps play an essential role in the use of the OS and next to apps provided by mobile OS providers, there are those offered by third-party companies that could contribute to the success of a platform. It is believed that the center of gravity of the software industry will be allocated on mobile applications, given the sales figures and margins for growth in this area. There were a constantly increasing of the number of companies that make use of apps on mobile platforms to stay closer to their customers and groped so keep active in the competition. As mobile devices become everyday objects of users/consumers, developments of applications designed for the mobile world are the fastest growing area of the market. The type of application would depend on the activities that the user performs regularly with such devices. A peculiar characteristic of mobile applications is their method of distribution. Unlike what happened with the desktop world, where the individual producer of software applications found its own distribution model (i.e., through external media or by downloading from the network by logging into a website), for mobile applications have been created especially containers, called Application Store, which exploit the features of online commerce (e-commerce). The store may be specific to the manufacturer of the mobile device (e.g., Apple, Ovi Store), or producer of the software (e.g., Google Play / Android, Windows MarketPlace). Most of these distribution models accept both native and web apps and hybrid apps, even if with some constraints. When choosing more a mobile platform for apps development, we should necessary taken into account cross-platforms tools to avoid the high cost of native development for different platforms. We have highlighted some features that must be considered in the apps design project before making a decision about which tools to use for development.

4.1 The requirements of cross-platform tools

When considering a cross-platform pool, there are some things that need to take into account: the design, the technical limitations related to the mobile platforms update and the continuous support to the platform, and the context in which the app is located. These are all questions we need to ask to assess the risk associated with using any cross-platform tools before embarking any project. In the app design, it is necessary to analyze what are the functionalities required by the app in order to assess if that app requires native code or plug-ins that could not be supported by the tools. And the time spent hiring, designing and coding around the issue, could turn out to be very costly. At the same time, the benefits of cost savings and time to market when cross-platform mobile tools, can also be very real.

4.1.1 Issues in apps design for mobile platforms

When developing for different mobile operating systems platforms, the developer should consider that each platform has its own peculiarities. There are design differences among mobile platforms (e.g., in iOS the tabs go on the bottom, in Android the tabs are on top and in Windows 8 everything scrolls horizontally). Custom icons are made at different sizes, and there are many screen resolutions and aspect ratios to consider as well as many screen densities and image sizes to account for. There are different app's requirements if the app needs to fit the iPad screen right up to the edges (i.e., to support the iPhone including retina and non-retina graphics) or if the app must also fit perfectly on a Samsung Galaxy smartphone and tablet of varying sizes and ratios. When the requirement is the cross-platform installation, the designer for that app needs to be skilled in both iOS and Android design, and the designs could conceivably be different enough to warrant two completely separate code streams.

4.1.2 Mobile platforms technical limitations

Another aspect to consider is the evolution of the mobile platforms' functionalities. Each cross-platform solution can logically only ever support a subset of the functionality included in each native platform. When new features come out, developers must wait until the mobile frameworks could incorporate them into the API. Without the control of being able to incorporate new capabilities as soon as they are introduced into each native platform, a developer could risk to lost an opportunity (e.g., being first to use a new feature is compelling for the app success). In addition, a new functionality could require the development of "plug-ins" for each platform, which can become an unanticipated, costly and time-consuming part of the project. The only way to ensure an apps is built within the appropriate, and most

efficient, technological parameters is to build it natively.

Since an app is usually distributed by means of stores, the developer should be sure that that store support also cross-platform-based apps. Actually, for example, Apple accepts some hybrid apps (e.g., PhoneGap-based apps), but it could be that with later versions of operating systems, they are no longer supported. In such cases, a developer should evaluate what is the cost relative to the "port" of the cross-platform code into a native app.

4.1.3 The apps context

Finally, there are scenarios to consider when it might be appropriate to use cross-platform tools or to use a native approach. For example apps for "online Store" or educational app should have a presence on as many platforms as possible (i.e., iOS, Android, Windows Phone and BlackBerry). If the app is not deemed to be graphically-intensive, and the initial design wireframes need a conventional mobile app layout even if responsive, this app probably could be developed by means of a cross-platform such as PhoneGap or Titanium. The decision is, however, a decision based on costs, time, and performance of the app.

If the app has a social network content and needs to incorporate location-based functionality, camera usage, graphical animations and sound interactions, and needs different design for tablets of smartphones, a careful approach must be taken. Location-based functionality or camera usage is available in most mobile frameworks, yet each OS truly has different ways of handling location accuracy and performance and as regards the camera usage, there are advanced functionality, such as augmented reality or built-in editing features, that the cross-platform approach could not support.

Finally, especially for educational and outreach aims, the video game app category is more important for an app. In these scenarios, the Unity approach allows the development for various mobile platforms, but is expensive. Another approach could be using open source free framework like cocos-2d or cocos-3d to build an app on one platform and then port it to the other. The downside, of course, is multiple code streams in multiple languages, which essentially makes for many products.

4.2. A list of cross-platform tools

There are some technologies that allow for cross-platform development of mobile apps each one with its features that make it appropriate or not for an app or business. In Table III, we propose a list of used tools, available in open source and commercial licenses that could be used to develop hybrid apps. Among the different tools probably the most famous are PhoneGap and Titanium tools that use a different approach in the hybrid development.

PhoneGap, an open source project now under the Apache Software Foundation (ASF) projects known as Apache Cordova, is a tools for apps development for different platforms (e.g., iOS, Android, Windows Phone, Bada). It uses the HTML5 framework to create a mobile "web app" that sits inside a native application wrapper. The web code is packaged with a library that bridges web code to native functionality. It has the benefits of a low barrier to entry (HTML/JavaScript/CSS is the only skill needed to get started), a single code base for all platforms and rapid testing and deployment. But the app could report poor performance since it works on a WebView that is it deliver the web app as a part of a mobile client application. But it lacks of supports for native user interfaces elements and widgets. For this issues, this tool is not probably good for graphically-intensive apps.

Cross-platforms tools	
tools	***features***
PhoneGap	Open source framerwork (uses standardized web APIs for the platforms) – Apache License, 2 – part of ASF under the name Apache Cordova
Sencha Touch	HTML5 mobile app framework (commercial license free-of-charge under the agreement and also available under the GPLv3 license for open source projects).
mono	Platform based on .NET framework based on ECMA standards and C# for the common language infrastructure. Sponsored by Novell. Uses open source licenses: GPL for the C# compiler, tools and runtime libraries, MIT X11 for class libraries, ASP.NET, MEF, DLR, under open source Microsoft Permissive License + Aapche 2. There are also commercial licensing options.
titanium	Open extensible development environment for creating native apps across different mobile devices and OSs including Android, windows and Blackberry as well as hybrid and HTML5. It includes an open source SDK (Studio).
Unity	A game developer ecosystem with commercial licenses even if there is a free version for windows with publishing support for iOS, Android, desktop and web
Corona	2D development platform for apps and games with commercial licence even if it is offered a starter kit for free.
appMobi	Cross-platform push messaging, app promotion, in app purchasing for all apps and deployed in any environment
LIVECODE	Live app creation too for multiple platforms (community & commercial editions).
MoSync	Cross-platform SDK tools and HTML5 tools for mobile app development (MoSync mobile sdk and mosync reload with GPL2 licenses)

Table III: List of mobile cross-platform tools

Appcelerator Titanium is a mobile development environment that helps to create all types of mobile apps: native iOS, Android, Windows and Blackberry apps, HTML5-based apps and hybrid apps. It uses JavaScript libraries to wrap the code and transform it in an app that closely resemble the look and feel of a native app. It is offered as an eclipse-based integrated development environment (IDE) that

simplifies the development process. The package includes the open source JavaScript-based SDK with devices and mobile operating system APIs, Studio that is the IDE, Alloy a model-view-controller (MVC) framework useful to the app design and cloud services for ready-to-use mobile backend. Alloy is a development framework to facilitate the rapid development of high quality mobile apps that follows a MVC architecture using XML and CSS to provide a simple model for separating the app user interface, business logic and data models.

While Appcelerator compiles the JavaScript code into a native binary, converting the JavaScript into native class and object files, PhoneGap simply renders a WebView with the code being interpreted inside. Since more close to the native way, Appacelerator-based apps are faster to execute and more compact in size. Conversely Phonegap-based apps are faster to develop.

Within the presented list, if we focus on apps developed as games, we can cite Unity. Unity focuses on gaming and other graphically-intensive applications. It currently supports mobile platforms and also gaming platforms (e.g., Nintendo, Wii, PlayStation 3 and Xbox 360). Unity provides a development environment with which graphics, physics and sound can be programmed. The framework is built on Mono, the open source implementation of the .NET framework. Programmers can use different programming or scripting languages like UnityScript (a custom language with JavaScript syntax), C# or Boo (a Python-inspired syntax). The downside of Unity is that it tends to be expensive by comparison particularly when you add in the cost of plug-ins used to build on multiple platforms. The choice of this tool must be a tradeoff between the required performance and the actual cost of development.

V. CONCLUSIONS

Apps are the new way of software development considering that sales of mobile devices are surpassing those of desktop computers. But, while desktop platforms are an established world with few solutions (e.g., Windows-based, Mac OS or Linux distribution), the ecosystem of mobile platforms is highly fragmented and in constant evolution. There are different types of mobile devices with different functionalities and platforms that in many cases are proprietary. If we focus on Internet-enabled mobile devices such as tablets or smartphones, the platforms are surely reduced, however, the market, even if actually seems stabilized on two OS (iOS and Android OS), shows the introduction of different mobile platform. It is worth considering the emerging mobile platform (e.g., Tizen OS and Salfish OS) that probably could become the "Android" of the future. So then developing an app, it is necessary to take into account the different mobile platforms, each one with its peculiarity. Focusing on one platform, iOS or Android, could be very limited for the specific type of apps. And when deciding to develop for more than one platform, the problem is the selection of the best framework. Developing native apps could be very expensive and take a long time, so the use of cross-platform tools in some cases could be a good solution. When considering whether to use a cross-platform tools instead of native mobile development, some important deciding factors include:

- Platform requirements—which platforms must the app support and which platforms are considered "nice to haves";
- Technology requirements—which mobile device features must the app support and which are "nice to haves";
- Performance requirements—are there aspects of the conceptual apps where performance, especially graphical or computational, a consideration;
- Design constraints—how disparate are the designs for different device types.

Probably for most types of apps in our context as regards education and outreach, we could select a cross-platform strategy apart from games development. Many of our apps can be designed as quizzes and require no special graphics or high performance. A cross-platform approach is helpful to develop our apps for different types of mobile platforms, since it allows to reach a large number of potential users. We are therefore considering and trying after this study, the different mobile frameworks and then Phonegap and Appcelarator to figure out which is the best tool for us from the point of view of skills required, to be taken to develop this type of apps. The aim is that type of development can still be used immediately in the new mobile platforms such as TizenOS or Ubuntu Phone as soon as there are devices on the market that support such new operating systems.

REFERENCES

[1] R. C., Basole, J. Karla, On the Evolution of mobile platform ecosystem structure and strategy, Business and Information systems engineering, Springer, Oct. 20122, Volume 3, Issue 5, pp. 313-322, 2011

[2] A. Jamalipour, The wireless mobile Internet. Architectures, Protocols and services, Wiley, 2003

[3] S. Pastore, Website development and web standards in the ubiquitous world: where are we going? WSEAS Transactions on Computers, Issue 9, Volume 11, September 2012. pp. 309-318. E-ISSN: 224-2872

[4] A. H. Lashkari, M. Moradhaseli, Mobile operating systems and programming: mobile communications, VDM Verlag Dr. Muller, 2011

[5] S. Cai, F-K Chiang, X. Wang, "Using the augmented reality 3D technique for a convex

imaging experiment in a physics course ," International Journal of Engineering education, vol. 29, No. 4 pp. 858-865, 2013
[6] G. Kipper, J. Rampolla, Augmented Reality: an emerging technologies guide to AR, Elsevier, 2012.
[7] M. David, HTML5 mobile website: turbocharging HTML5 with jQuery mobile, Sencha touch and other frameworks, Focal press, 1 ed. 2011
[8] R. van der Muelen, J. Rivera, Garter Recommends a hybrid approach for business-to-employee mobile apps, press release, Gartner site, April 2013.
[9] iOS technology overview, Apple developer site, Apple, 2012 .
[10] C. wang, W. Duan, J. Ma, C. Wang, The research of android system architecture and application programming, Proc. Of Int. Conference on Computer science and network technology (ICCSNT), 2011, pp. 785-790.
[11] Microsoft Software developer website, Concepts and architecture for Windows Phone, MSD Microsoft library, 2013.
[12] Openwebosproject website, Open webOS architecture, openwebosproject.org website, 2013
[13] SkyTeckgeek.com blog, Understanding Samsung Bada OS and its development architecture, 2011
[14] B. Fling, Mobile Design and Development: practical concepts and techniques for creating sites and web apps. O'Reilly media, 2009
[15] Salfish OS website, at url:http://salfish.org
[16] MeeGo website, at url:https://meego.com
[17] Mer project website, at url:http://merproject. com
[18] Qt project website, at url:qt-project.org
[19] Tizen OS project, at url: www.tizen.org
[20] Ubuntu statistics at url:http://www.chimerare vo.com/steam-linux-distribuzioni-statistiche