DOCUMENTING THE MOBILE INTERNET FRAGMENTATION

MOSQUITO

Project Deliverable

Project Number: 258067	**Project Acronym:** MOSQUITO	**Project Title:** Mobile software and services, Standardisation, Quality, Interoperability, Testing, Open source

Instrument: Coordination and support action	**Thematic Priority** Internet of Services

Title Final report

Contractual Delivery Date: January 2013	**Actual Delivery Date:** September 2010

Start date of project: September, 1st 2010	**Duration:** 24 months

Organization name of lead contractor for this deliverable: INNO	**Document version:** V1.0

Dissemination level (Project co-funded by the European Commission within the Seventh Framework Programme)

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group defined by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Revision History

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V1.0	January 2013	Creation	F. Le Gall (inno), F. Clari (inno)
V1.1	January 2913	Updates	F. Le Gall (inno), F. Clari (inno)

Table of Content

1 Publishable executive summary

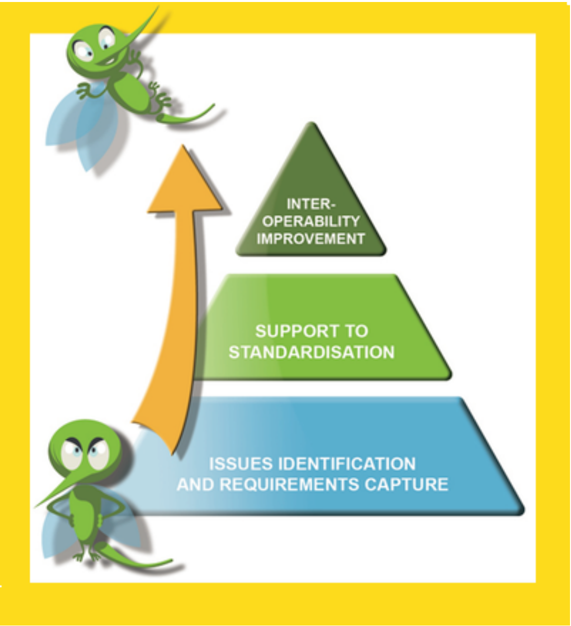
1.1 Executive summary

MOSQUITO is a project co-financed by the European Commission under the 7th Framework Program (FP7), under the ICT thematic Internet of Services, Software and Virtualisation. The project started on the 1st of September 2010 and last for 25 months. Six partners, from Europe and China, compose the MOSQUITO consortium.

MOSQUITO aimed at documenting mobile application (namely applications for mobile devices) fragmentation, which actually forces developers to build several versions of the same application in order to increase the potential target users and prevents taking full benefit of the development of Internet of services.

MOSQUITO intended to achieve this aim through:

- ✓ **Detailed identification and capture of requirements** (within the **work package WP1**),
- ✓ **Support of standardisation and dissemination in the field** (within the **work package WP2**),
- ✓ **Address interoperability improvements** (within the **work package WP3**).


INTER-OPERABILITY IMPROVEMENT
SUPPORT TO STANDARDISATION
ISSUES IDENTIFICATION AND REQUIREMENTS CAPTURE

The principal outcome is **a full documentation of the mobile Internet fragmentation**, covering not only most used operating systems and devices, but also all layers of the mobile Internet architecture (e.g. app stores, mobile networks, content delivery…). As part of this work, the MOSQUITO project also liaised with standardisation bodies involved in the mobile Internet ecosystem.

Obtained results

The analysis of the numerous symptoms of fragmentation has enucleated a series of causes for it:

The manufacturers produce a lot of new versions of their devices, improving at a rapid pace both the hardware and software. This causes older hardware rapidly not being actively supported by new versions of the operating system, so that also the applications developed for an older operating system (depending on an older hardware) cannot be run on more recent devices. This can be defined as **internal fragmentation**.

Different mobile platforms are still co-existing. Two of them are currently mainstream (**Apple iOS** and **Google Android**), while other are declining but still present (e.g. **BlackBerry**, **Symbian**, and **Windows Phone**).

Different platforms and different versions in the field of the same platform cause the need of having different **development tools**. Also the use of very different **programming languages** contributes in further fragmenting the development landscape (Java for Android and, in a different flavour, for BlackBerry; C/C++ and Objective-C for iOS; again C/C++ for the new BlackBerry 10; C# for Windows).

A **set of concrete steps** to take to reduce fragmentation has been identified and described in the Mosquito documents.

Fragmentation is **not always a cost**, for some players it is a key aspect for differentiation, business and market positioning.

A possible solution against some kind of negative fragmentation is possibly the emerging of the development of **web applications as "first class citizens"**, basing on the capabilities of the on-board browsers. In this scenario it would be theoretically possible to write an application only once using well know technologies like HTML, CSS, and JavaScript, and building an executable application for every single platform. A high level of standardisation will be anyway demanded.

While the number of platforms is diminishing, due to the dominance by Android and iOS, **fragmentation related to the content** remains an issue not to be underestimated.

Standardization activities should be encouraged as can help reducing fragmentation, but effort should be put in **providing standards quickly** and as soon as they are needed, otherwise market will find workarounds and will be filled with different partial implementation of drafted standards, creating an even more fragmented landscape.

1.2 Summary description of project context and objectives

1.2.1 Introduction

Mobile applications and services are important components of the future internet in particular when more broadband is going to be mobile. This field was for years considered as one of the most promising fields but the critical mass of services and applications was not realised due to Standardisation, Interoperability, Quality and Convergence issues which have been at the heart of MOSQUITO considerations.

Internet is going to be massively "mobile" and therefore delivery of efficient and growing internet of services has also to take into account issues of this "mobility". The same is valid for the mobile devices and their applications. If, at the origin, mobile internet solutions depended on the power of mobile operators, we should note today that the take-off of the iPhone® and its mythical platform of applications "Appstore" upset the balance of the forces. Therefore future of **Mobile Internet of services is increasingly dependent on mobile applications**

Since the beginning of the decade, it has become possible to install applications on mobile phones, past their point of sale; those phones are said to be open. In Europe, the ratio of open phones currently ranges around 80% of the number of units sold and around the same ratio of units in use. Applications for mobiles phones are called mobiles applications, the majority of them are downloaded over wireless networks, therefore those application are often called downloadable over the air (OTA). Downloadable mobile application is a fast growing market. This trend was first been observed in Japan in the beginning of the decade, using NTT DoCoMo Doja technologies; later and to a more limited magnitude in the US, using Qualcomm Brew technologies. It is now spreading globally using Apple iPhone® technologies.

It can take up to nine months to deploy an entertainment (mobile) application, But that's the duration of

Unfortunately, the mobile application business is plagued by a critical and recurrent issue: fragmentation. Fragmentation, in the wider sense, shall be understood as anything forcing developers of mobile applications to build several version of the same application in order to increase the number of mobile phone users capable to use the application. Fragmentation has an extremely large number of origins, including, but not limited to: phone hardware differences (such as screen size, screen layout, screen touch capability, memory size, Bluetooth capability), platform differences (such as operating system or interpreter), implementation differences (such as bugs, standards inconsistencies or standards omissions), middleware differences (such as codecs, shared libraries, helper applications), network differences (such as routing of certain kinds of traffic), territorial, linguistic or cultural differences; or distribution channel requirements differences (such as branding, legal disclaimers or contracts). To be fair, the causes of fragmentation are not fully understood in fine detail, at least not in widely publicly available technical literature. Fragmentation is striking standards based mobiles platforms more severely than proprietary; the interim list sketched earlier shall help the reader understand that proprietary platforms are not immune against fragmentation. How sensitive mobile application markets are against fragmentation is definitely an insufficiently documented topic.

Mosquito helped documenting all issues of fragmentation which prevent full development of Mobile.

Fragmentation has severe consequences. Mobile application developers always have to make a business limiting trade off between the size of the market they can address, which represents their potential gains and the number the of versions they can afford developing, which represents their costs. The more the market is fragmented, the larger the number of versions is required, the greater the cost. Mobile developers using the Java platform often spent sixty to eighty percent of their development investments building a number of versions frequently ranging above one hundred. It must be noted that building numerous versions of an application makes corrective maintenance almost impossible, tracking bugs implies identifying the exact version of the application as well as the exact version, settings and configuration of the device, which is complex and expensive. Therefore, until recently, mobile application were mainly restricted to games, where a fire-and-forget logic applies, most mobile games are released and sold as short lived, disposable, products, quickly replaced by newer. No maintenance ever takes place. This is unacceptable for application as service enablers. Services are made to be used, not consumed. **The number of services successfully implemented over fragmented platforms is extremely small.**

Mosquito addressed the fragmentation of applications which prevent the development of services which would take full profit of existing broadband technologies (3G, Wireless, LTE, Wimax)

It must be noted that fragmentation is not a phenomenon likely to disappear, as it is the result of natural diversity of a healthy market. One may claim that the desktop application market never fragmented because Microsoft acquired a shocking monopoly in desktop operating systems. Fragmentation allows the device manufacturers to select mobile platforms from different vendors, which helps reducing the costs. There is also a form of fragmentation caused by technology evolution, at any given instant in time, the market addressable by a mobile application developer is a snapshot of technology evolution composed by states of new, mainstream and older devices, embedding the technologies used at time of shipping of the device. Last but not least, it must be noted that fragmentation is also one the strongest protections against malicious applications. Malicious applications often take advantage of bugs or design flaws called security holes, in presence of fragmentation these holes cannot be exploited in large number of devices, to present; this has efficiently deterred most people to write malicious mobile applications.

1.2.2 Context

1.2.2.1 Fragmentation and standards

The relationship between mobile standards and mobile fragmentation is complex. As said earlier, standard based platforms are often said to be more sensitive to fragmentation than proprietary ones. However, counter arguments against this claim are solid.

① First, there is currently only one real example of mobile platform standard: Java ME, however it has been established that Java is more a shared proprietary technology rather than a standardized technology[1]. All the deliverables of a workgroup of the standardization like process defining Java specifications, (the JCP - Java Community Process) are owned by the leaders of the workgroup. This obviously includes the specification but also includes a reference implementation and most importantly the unique test suite used to check for compliance of candidate implementations. In essence, the Java test suites are not used to insure any level of quality or interoperability, there are the tools allow the few leaders company to divide

[1] T. Egyedi "Why Java Was Not Standardized, Twice").

among them the control over the evolution of Java, Sun being the main leader and owner of the key specifications.

② Secondly, proprietary platforms are sensitive to all the non platform specific fragmentation introduced by hardware, network, business and operational environment. The only standards based alternatives to Java are the so called Web-based applications widgets. However, because the W3C and ECMA are lenient against bad business practices against standards such as sub setting (a.k.a. quick and dirty) and super setting (a.k.a. embrace and extend). Web based technologies are already fragmented in the desktop browser space, where web developers have to design their services defensively against the various browsers in order to be compatible with all of them. This suggests that fragmentation of web based applications in the mobile space has the potential to further fragment the mobile internet of services. Even more interestingly, this also illustrates another bizarre side effect of fragmentation. In the browser space, web-applications use defensive programming approach, this is extremely rare in the mobile space. This is extremely intriguing because JavaScript, the language of the web, is well known to be weakly capable for defensive programming while Java, the language of the mobiles is far more capable. This is explained by the differences of business models. In the web, the applications are service enablers, therefore monetization takes place elsewhere, therefore handling fragmentation is a pure cost, and therefore it is handled by the successful using state-of-the-art techniques, visible today with the rapid spread of html5/CSS3 standards. In contrast, in the mobile, the applications are content; monetization takes places in sale of the application itself, and therefore knowing how to manage fragmentation become an extremely high value asset and an entry barrier against new entrants. As a side effect of this, there is a fair amount of good quality technical literature on how to cope with fragmentation in the web while there's nothing on the same topic for mobile, however there are companies selling porting tool and porting service.

Fragmentation is also a dynamic phenomenon. This is better understood, first in the case where fragmentation is sufficiently[2] contained. In this scenario there's a consistent fleet of devices companioned by developers tools and an application shop. This forms a business opportunity which attracts developers. Applications are shipped to the users through the store. Then by viral effect, the users begin to buy phones capable to run the applications, precisely because of the availability and appeal of them. Therefore the number of compatible devices in operation is increased, therefore the market addressable by the developers is larger, and therefore more developers begin coding for that platform. The reader will have recognized the example Apple iPhone®. But this is no different of any content and player sales mutual driving. In presence of fragmentation, the positive feedback turns into a negative one and the business either reach a plateau or collapses. **Developers are frustrated by differentiated platforms, they waste their resource porting rather than innovating and finally, since a user can not share their successes using applications among them, the market remains a niche, if it has even appeared.**

1.2.2.2 Fragmentation and convergence

Convergence between wired and mobile networks focused on vocal, tv, music, and others multimedia services is a very important topic for the development of mobile internet services.

The trend introduced by Google Android is a very important step: Google Android can run on a mobile phone of course, but also on a PC or an eePC. It's clear the strategy of Google was to spread their operating

[2] The definition of what is a sustainable level of fragmentation is an open question

system on all kinds of (embedded or not) systems. This has been followed by window within its latest version of Window 8. In others words, we can consider that in the future we will be able to run the (almost) same application on a Google Android mobile phone and your PC or your TV.

Another example of trend is the Samsung TV. On these screens, we can display Yahoo widgets. So, we can also consider in the future we will see more and more "embedded small softwares" in embedded systems. But widgets are very heterogeneous and can't be run in the same way on a PC, a connected TV and a mobile phone.

1.2.2.3 Fragmentation and Heterogeneity of the applications stores

A recent evolution is the applications stores. It’s not a recent concept but this concept has been developed by Apple with its iPhone. Since the AppStore of Apple, a lot of new stores appeared: Google Market, OVI, Blackberry App World, Microsoft Windows Market, LG Application Store, and SamsungApps... And some portals evolve, like Hangango InHand. These stores are very heterogeneous in terms of supported platforms, revenue sharing model, billing, description, restrictions on logic of application, manufacturers, geographic zone supported, devices, etc.

For example, a portal could bill by the mobile operator in some countries, by credit card in the others (if the mobile operator billing is not supported). So, the revenue sharing rule will change. Another example: stores are in general open on a limited list of countries, not worldwide.

The trend is that each manufacturer launches its own store, and some independent stores appear. For developers, it’s very difficult to have a global strategy and to address each (or not each) store.

1.2.2.4 Ecosystems and value chain

The smart phone is critical - it represents a key growth sector for the mobile Handset Manufacturer and promises an associated increase in service revenues for the Mobile Network Operator (MNO). With a new generation of improved smart phones, with large size touch screen, powerful processor, dual wireless/mobile connectivity and an improved user interface, the smart phone is a key element in providing user access to the mobile internet. If smart phones are essentially handheld computers with wireless connectivity, the embedded middleware is critical.

There is a race taking place between the handset manufacturers and the content provider to become the leading provider of mobile middleware for the smart phone. One example is Nokia, the owner of the Symbian Foundation, which had made the company into a non for profit organisation and had recruited a number of companies to join as members and is converting various mobile OS into one unified version based on open source technologies. In terms of worldwide licences for mobile OS, it is Symbian which had traditionally held up to half of the world market, largely thanks to the dominance of Nokia in European markets. But over time Symbian lost market share to other platforms including Google Android, Apple iPhone and RIM Blackberry.

Open source mobile OS services are being promoted by both Google (through Android) and Nokia (through Symbian) as a lower cost alternative to proprietary mobile OS standards and Microsoft's licence-based Windows Mobile OS standard. But the goal of the new mobile OS standards is to become the centre of a sustainable mobile ecosystem, which includes handset, applications, content and provide recurring revenues to the handset or content company.

These developments affect the key foundations beneath the mobile telecoms industry, namely handsets and services. More opportunities will arise given the commitment by major players to support this arena. The digital commercial landscape is becoming more fractured with competitive activity increasing, indicating a proliferation in opportunities for players across the value chain.

1.2.2.5 Point of view of developers

The number and diversity of mobile devices is exploding. There are about five billion mobile phones in the world. In addition to phones, there are billions of additional devices that are becoming wireless-enabled, such as laptops, netbooks, mp3 players, e-readers, game consoles, digital cameras, printers and appliances.

From the viewpoint of a mobile developer, writing software that works on a large number of wireless devices is extremely challenging. Excluding non-phone devices, there are several major smartphone platforms that must be supported. In addition, there are dozens of feature phone platforms. Each platform consists of its own operating system and environment, file system, development environment and tools, communication protocols and more. Although there are mobile standards such as SyncML, creating apps for these phones must still be performed on a platform-by-platform basis.

One consideration for developers is building mobile web vs. native apps. The advantage of mobile web apps is they can run on a variety of devices. The disadvantage is they are not very usable or compelling. Developers today do not build mobile web apps as their interfaces and performance are not rich.

At the same time, it has become a nightmare to build mobile native apps for more than a handful of devices. This is due to the learning curve for each platform as well as maintaining the required expertise, as well as owning the devices and testing them. This device fragmentation problem is exasperated as mobile devices behave differently on different networks. As a developer, it's not just a matter of writing software but then testing it on potentially thousands of devices on hundreds on networks in hundreds of countries. This combinatorial explosion makes the task of creating innovative mobile software prohibitive for most developers. Most smartphone platforms now also have an app store in which developers must post their apps, which is a non-trivial process that is the latest extension of the device fragmentation challenge.

1.2.2.6 Fragmentation and market

At present, without disruption, the forces of the market are likely to make the mobile application ecosystem evolve in three directions: (i) One takes all. There is nothing but a single application platform surviving the fragmentation. That would have dramatic consequences on many economical sectors. (ii) As a variant, few non-interoperable application platforms would survive. (iii) The mobile Internet of services just does not happen. Because fragmentation remains at an unacceptable level, innovation and service just do not reach the consumers.

1.2.3 Project objectives

The Mosquito project provided key and focussed support, mainly to overcome mobile application fragmentation which today prevents the development of internet of services. To overcome fragmentation, an important task is to document all issues which need to be addressed, including aspects of convergence, standardisation, validation and interoperability, all topics covered and addressed by specific workpackages within the Mosquito project.

The high level objective of the project has been to promote the virtuous win-win development circle as opposite to the current loose-loose one in order to give the right condition for a booming development of Mobile Internet of services

For that purpose, the operational objectives of the project have been:

- The identification of barriers to prevent development of Mobile Internet of services (i.e. the mobile internet application fragmentation)
- The support to standardisation and collaboration for mobile services
- The support of actions relating to interoperability for mobile application and services
- The support to Cross-sector convergence of IT, telecom and media in the content area

1.3 Main S&T results and foregrounds

1.3.1 Overall methodology

1.3.1.1 Project organization

MOSQUITO activities have been articulated around 3 complementary workpackages

The first work package (WP1) identified all the causes, symptoms, characteristics, consequences and issues related with mobile platforms fragmentation. It also identified the requirements against fragmentations which are not fulfilled by current mobile platforms and the relationships between these items in order to structure them.

The issues related to fragmentation have been collected by looking along three different perspectives, each documented in a public deliverable:

- ✓ *Mobile platform*s (deliverable D1.1)
- ✓ *Services and networks (*deliverable D1.2)
- ✓ *Applications and content* (deliverable D1.3)

The second workpackage (WP2) analysed the worldwide standardisation landscape in order to reduce fragmentation and boost the development of internet of services. Purpose of this workpackage was not to do standardisation on its own but to discuss and liaise with all standardisation related groups in order to

- ✓ *Provide a gap analysis of on-going standardization activities* (deliverable D2.1)
- ✓ *Promote standardisation best practice and cross fertilisation between the groups* (deliverable D2.2)

Standardisation have been looked at for the base standards as well as for the testing standards often lacking whereas and are important pieces of the overall interoperability. This work package organised market driven workshop to ensure large audience and participation of market players in particular small developers and SMEs

The third workpackage (WP3) provided hands-on support to interoperability issues through the organisation of interoperability test events in both Europe and China.

1.3.1.2 Data collection and analysis process

A consultative approach has been chosen to establish project deliverables. Each step allowing both to collect additional data and to assess the results obtained up to that point.

Figure 1: Androïd interoperability event (Beijing)

- Firstly, a desk research have been done, looking at public documents as available on the web, company reports available within partners premises and limited access literature such as standardisation working groups reports.
- Then, more than 30 interviews have been conducted, addressing the different stakeholders of the mobile internet ecosystem. These included service providers, content providers, platforms providers and mobile developpers.
- Three consultative workshops have been organised in Europe and China. These workshop have been organised each with a central theme being reposectviely: platform perspective (China), business perspective (UK) and mobile applications development (FR)
- Finally, two interoperability events, gathering industrial and research stakeholders allowed participants to compare and evaluate the interoperability of their implementations withnio these 2 days events.

1.3.2 Fragmentation analysis

The issues related to fragmentation have been collected through different channels: platform industry perspective (programming language and platform),[3] service and network,[4] application and content.[5]

To better produce a more structured overview we are going to list some of the most important kinds of fragmentation that are analyzed in the deliverables produced by the project, along with a quick explanation. This structured overview is meant for quick reference only, each of these items is deeply analyzed in the other deliverables produced by the project and it is recommended to reference the proper deliverable to get a better insight in each item.

External fragmentation has unquestionably decreased in the past years due to the success of iOS and Android

1.3.2.1 External fragmentation

External fragmentation is the actual separation between platforms (like, Apple iOS and Google Android). External fragmentation has unquestionably decreased in the past years due to the success of iOS and Android, currently (2013) representing the vast majority of the market. The number of platforms was reduced when other players' market share got eaten by the two leading platforms. A deeper analysis of the external fragmentation is available in D1.1, including the list and history of various mobile Operating Systems in Chapter 5.

Internal fragmentation is caused by different version of the same operating system

1.3.2.2 Internal Fragmentation

The increased complexity of the platforms and the frequency of releases of new Android and iOS versions caused a high **internal fragmentation**, i.e. the fragmentation caused by different versions of the same operating system present on the market. This kind of fragmentation is partially limited by auto-upgrade

[3] See deliverable D1.1
[4] See deliverable D1.2
[5] See deliverable D1.3

functionalities, but as different stakeholders have different needs and goals, auto-upgrade is not always the path of choice (device manufacturers do not always want the device to run latest version of the operating system due to technical or commercial reasons). iOS and Android have very different approaches to limiting the internal fragmentation. Their competitors will need as well to limit fragmentation inside their platforms, if they want to attract developers. Apart from what reported in the following paragraphs, more information on internal fragmentation is available in D1.1, especially in chapters 8 and 9.

Though in different ways, users consider the number, quality, and price of applications available on a given platform when they choose which device to buy. This means that the success of a platform depends also on how the platform is able to attract application developers.

Despite its complex history, as of the second half of 2011, the mobile platform landscape is easy to look through. There are basically **two groups**: **leaders** (Apple **iOS** and Google **Android**) and **outsiders**.

As Google is providing Android for free, it leaves very little room for platforms licensed by vendors (e.g. **Windows Phone**), unless the ongoing patent disputes increase the cost per Android device. Single vendor platforms (e.g. **Symbian**, **BlackBerry**, **Bada**) may have higher chances; however, their appeal to developers is low. The only alternative which may arise is the **mobile web**, a platform based on W3C/ECMA standards such **HTML**, **CSS**, and **JavaScript** and open source projects such as WebKit, Firefox, and Chromium.[6]

1.3.2.2.1 Leading Platforms Approach Toward Fragmentation

The overall perception is that Android represents a more fragmented ecosystem, while the iOS platform is perceived as less affected by fragmentation.

Although this is true, research in this field has shown evidence that iOS is more subject to fragmentation than what is normally perceived and, on the other hand, that Android cultivates successful strategies that help the platform to deal with fragmentation.

Apple iOS

Apple has some key aspects that allow better fragmentation handling: since the company builds both the hardware and the operating system, the number of devices to support is smaller. Moreover, the targeted market is for high-end devices: there are not Apple low-end devices, so the operating system can count on solid (and well known) hardware specifications.

Apple builds mobile hardware and OS, this is a strength in fighting internal fragmentation

Over-the-air upgrades are handled once again by Apple, with no third-party modifications in the middle: a very large number of devices can be updated to the latest operating system version in a short time frame.[7]

It is interesting to study the way the market shaped itself concerning versions of the same app for tablet and mobile phones. iOS supports the so called **Universal Binary**: a single package that can contain two versions of the same application: one for iPhone and one for iPad.

[6] See deliverable D1.1, §§ 5.5 and 5.16, and deliverable D1.3

[7] See Chris Sauve at http://pxldot.com/18754186750: "*iOS devices have, on average, reached 10% version share 300 times faster than Android versions, 30% share 19 times faster, and 50% share 7 times faster*"

It is important to observe that universal binaries have been mostly adopted by developers of applications which are available for free, while developers who are selling their application tend to produce two versions (one for iPhone and iPod Touch, another for iPad). Developers distributing their applications for free have a service oriented business model: their applications are enablers, while the development is pure cost. Therefore, they tend to minimize it, and to consequently build a single binary. Developers who are selling their applications often began having a single version, for iPhone and iPod Touch.

This example shows that despite being complained about by developers, some fragmentation is actually introduced by the developers themselves. This fragmentation is painful for users owning an iPhone and an iPad who bought both versions of an application. When syncing their devices, the iPad version does not go to the iPhone, while the iPhone version goes to the iPad, were it uselessly duplicates the iPad version.[8]

The already mentioned **Universal Binary** (see ci-dessus) is facing an additional issue related to the introduction of the new Retina display in the iPad 3: a single universal binary may now need to include **up to four versions** of a single image to support the different displays.

Android

Android is probably the most successful mobile open source project, but at the same time it is also one of the most 'closed' open source projects.

In 2011 VisionMobile published a research on the openness of various open source projects, based on an Open Governance Index computed taking into account thirteen metrics across the four areas of governance.[9]

The results are as follows:

Open Governance Index	**% Open**
Android	23%
Qt	58%
Symbian	58%
MeeGo	61%
Mozilla	65%
WebKit	68%
Linux	71%
Eclipse	84%

[8] Deliverable D1.1, § 8

[9] *A new way of measuring Openness, from Android to WebKit: The Open Governance Index* (http://www.visionmobile.com/blog/2011/07/the-open-governance-index-measuring-openness-from-android-to-webkit)

This closed governance model is the result of Google's effort of keeping ownership and control over the Android platform, and is used to keep fragmentation under control.

Google has developed the **Android Compatibility Program** (**ACP**) that defines hardware, software, and functional requirements for manufacturers to get the Android brand and to grant access to closed source apps delivered by Google (such as the ones delivered on the former Android Market - now Google Play - and the Gmail app).[10]

Closed governance model is the result of Google's effort in keeping ownership and control over the Android platform

The ACP is designed to allow device manufacturers to differentiate themselves from competitors by allowing some freedom in hardware and software customization, but at the same time guarantees a shared set of specifications and interfaces, limiting hardware and software fragmentation.

New releases of the Android operating system are frequent: four in 2009, two in 2010, two in 2011. Actually, this may even be an excessive release frequency.[11]

1.3.2.2.2 Competing Platforms' Approach to Fragmentation

Followers need or will need to keep fragmentation into account as market requires fast and easy development of apps.

Research In Motion (RIM) BlackBerry

Historically, RIM devices have been a quite non-fragmented reality.

BlackBerry devices supported Java ME in a quite consistent way (which is fragmented by itself, but not inside the RIM family) and the custom BlackBerry Java APIs.

Things started to change when RIM introduced BlackBerry devices with a touchscreen (starting with the BlackBerry Storm in November 2008.) The need to support both touch screen and keyboard-based devices caused fragmentation problems and decreased the consistency of the platform.

Moreover, development on the BlackBerry Playbook tablet and in Blackberry 10 can be accomplished using C/C++ natively, HTML5, Adobe AIR, Java Android Runtime, Blackberry Java. While this wide support of development tools can attract developers, the risk of fragmentation is increased.

Nokia and Microsoft

In February 2011, Nokia announced an alliance with Microsoft and the consequent replacement of Symbian and MeeGo with Microsoft's **Windows Phone** operating system.

Further development of the Symbian-based S60 device family was discontinued, while S40 is still alive and will continue to be used on Nokia feature phones.[12]

In such an ecosystem, Nokia has the devices, but not a committed developers community (as proven by the failed tentative to go open source with Symbian and MeeGo), while Microsoft has a new mobile operating

[10] See deliverable D1.1, § 9.2

[11] See deliverable D1.1, § 9.1

[12] The platform does not expose any API, so developers cannot build native S40 applications. Developing for S40 must be done via Java ME or Flash Lite.

system and a large pool of developers who can potentially be converted to develop for Windows Phone, but lacks the devices.

As of May 2012, two versions of the Windows Phone have been released: **Windows Phone 7.0** and **Windows Phone 7.5** (codename *Mango*). The release of the **Windows Phone 8**, codename *Apollo*, is planned for Q4 2012.

Upgrade from Windows phone 7.0 to Windows phone 7.5 took place automatically.

Windows Phone: still at an early stage, already fragmented

There were discordant opinions on whether Windows Phone 7.5 devices would have been upgraded to Windows Phone 8. Microsoft stated that applications built for Windows Phone 7 will run on Windows Phone 8, but conflicting reports emerged regarding the upgradeability of current phones.[13] At the end, users with Windows Phone 7.5 will get upgrade to Windows Phone 7.8, but not to Windows Phone 8. This cause an installed base fragmentation.

1.3.2.3 Fragmentation in programming languages

The most popular programming languages currently adopted to develop mobile applications are **Java**, **C**, **C++**, **C#**, **Objective-C**, **JavaScript**. This variety of languages already represents per se an example of fragmentation, since each single platform is more or less bound to the support for one of these languages in the development tools provided by the manufacturers.

The lower level and strictly defined semantics of **C/C++** make it almost immune to fragmentation. The **Objective-C** language is de facto controlled by Apple for its products (but the open source **gcc** compiler does support it). As of Objective-C, the corporate control and the limited ecosystem around **C#** reduces the probability of fragmentation emergingS. Few portability issues across operating systems and implementations of the runtime environment have been reported.[14]

JavaScript is mainly used to develop web applications executed by an interpreter hosted by the device's web browser. Fragmentation issues caused by the language are well known, since the code may execute differently in different browsers.[15]

More information on the evolution and adoption of programming language in the mobile space is available in D1.1, chapter 4.

1.3.2.4 Form factor fragmentation

Mobile devices come in different shapes and sizes and kind (e.g. tablet, phones, netbook) with different screen ratio and preferred orientation). Differences in the form factor may require heavy changes in design of an application (especially in the UI). Consequences of the form factor fragmentation and discussion on how a mobile OS can make easier to support different form factor is available in D1.1, chapter 9.7, some information on content adaptation for different devices is available in D1.3, chapter 6 and 7.

[13] See http://www.pocket-lint.com/news/45331/apollo-upgrade-windows-phones-uncertain
[14] Deliverable D1.1, § 4.3
[15] Deliverable D1.1, § 4.2

1.3.2.5 HW fragmentation

Groups the fragmentation aspects related to hardware differences such as different input methods (touchscreen, keyboard, keypad) or availability of hardware components (WIFI, memory card, 3G connection...) and sensors (GPS, orientation sensor, light sensor...). This kind of fragmentation includes also performance-related fragmentation: differences in CPU speed, memory, graphic capabilities etc. cause require developers to produce different content or experiences based on what each device is able to achieve. More details on Hardware Fragmentation are discussed in D1.1, Chapter 6 and 7.

1.3.2.6 Service Level Fragmentation

1.3.2.6.1 App Stores

The revenue of an application sold on an app store is often shared between the **application developer** and the **app store operator**. The popularity of app stores and some strong constraints of some of the mobile industry players (manufacturers) on the developers have made the different app stores the de-facto only wide distribution channel for an application developer.

Fragmentation at App Store level is caused by multiplicity of actors interested in having their own store

The different app stores offer different conditions for mobile applications developers: they address **different devices and platforms**, offer **different revenue sharing schemes**, require **different entry fees** and application **validation processes**, authorise **different types of content**, offer **different** advertisement **opportunities** and **different development tools**.

This multiplication of distribution channels increases the fragmentation for mobile application developers, as the choice of the distribution channel impacts the development of the application.

The market of the app stores is characterized by the different industry actors operating them:

- **App stores run by device manufacturers**: mobile handset manufacturers are operating app stores that offer applications for their handsets. Their interest (differentiation from competitors) increases directly the mobile application fragmentation. Examples of manufacturers' app stores include: **Apple App Store**, **BlackBerry App World**, **Nokia Store**, **Samsung Apps**.
- **App stores run by platform makers**: producers of mobile software operating systems are running app stores that offer applications supposed to be available for every device running their platform. As for the manufacturers' app stores it is to be noted that such app stores have also some interest in the fragmentation of the mobile world. Examples include the **Google Play** store and the **Windows Phone Marketplace**.
- **App stores run by network operators**: network operators have low interest in the fragmentation of mobile application development, as they aim to offer as much as possible a unique experience to their customers, regardless of device and platform. They have also the ability to offer interesting billing solutions to application developers: for the end-user, the possibility to pay for application directly on their carrier bill. Examples are **Verizon V CAST**, **Orange App Shop**, **AT&T AppCenter**, **Sprint Digital Lounge**, **T-Mobile web2go**, **Vodafone 360**, or **China Mobile Market**.

- **App stores run by independent players**: some examples of this kind of app stores are represented by Amazon Appstore for Android, Appitalism, OpnMarket, AppCentral.

The conflicting goals of the different app stores, along with the natural market competition, push them to try to differentiate on several points:

- **Different Terms of Conduct and content policy**: the most important consequence is that the developer will have to multiply the signing process operations (certificate management, code signing) for each app store.
- **Different audiences and rules for visibility**: this increases the developers' effort on marketing to get their applications visible in app stores that differently advertise, rank, and classify applications.
- **Different rules for revenue sharing**: it increases the difficulty of predicting the potential revenues that an application will generate, as not only the revenue sharing varies from one store to another, but also the behaviour of the users visiting a particular type of store.

1.3.2.6.2 Security

Securing a mobile network is not limited to providing an encrypted connection between devices and services. Mobile network operators are expected to use their networks as a platform to aggregate services and products also from different partners, and to provide them to their users.[16] This is an obvious source of fragmentation.

1.3.2.6.3 Accounting (In-App Billing)

Billing is defined for both the monthly subscription paid by the end-user, and for paid services that users can buy from their device.

Payment APIs are software components made available by mobile network operators. Various payment systems exist and cohabit, and developers have to deal with all of them and the resulting fragmentation:

- if developers have an application available in multiple versions for various devices, then they have to make the purchase working with various app stores
- end users have to provide their credit card information to another stakeholder
- as remarked above, developers have to be compliant with each app store's license

1.3.2.7 Network Level Fragmentation

Telecom operators (carriers) sit on mounds of legacy switches and cores that still remain the backbone for most of the telecommunications across the globe. Hence, there are several moving parts in the complex values chain, involving several technologies and components that are held together by interoperability and standards.

As the technology landscape evolves, some of these components or the parties controlling them become redundant, outdated or left unsupported in spite of an active users base. Since different carriers apply their own approach to their network evolution, this creates a fragmented environment, often asking for mediation services that can sustain several carriers' operational requirements.

[16] A. Prasad, H. Wang, P. Schoo. *Network Operator's Security Requirements on Systems Beyond 3G*, Conference Paper, WWRF8, April 2003 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.3075&rep=rep1&type=pdf]

This already fragmented scenario is complicated by several additional factors:

Network level fragmentation is manily caused by legacy hardware components that need to interoperate

Service requirements: while a lot of services are evaluated in the context of current smartphone requirements, it is important to realize that operators are also eyeing a lot of **connected services** markets.

Communication between stakeholders: an example is **roaming**, where end-users have the flexibility to make or receive a phone call or use their email irrespectively of the network operator.

Moving towards all IP-based networks: several operators across the globe are moving towards **all-IP networks**, which will allow interoperability between a variety of services and content since it would be independent of the underlying radio access technology.

IPv6: while **IPv4** is still the most used protocol on the internet, large **IPv6** availability will take time, especially on mobile networks, but will happen in the next years. IPv6 support has to be validated for each of technologies involved, such as the language used, additional libraries, etc. **Carrier-grade NAT** (**CGN**) as an approach for mitigating IPv4 address exhaustion is not a good option, since it provides a solution to postpone IPv6 deployment that will happen anyway.

1.3.2.8 Fragmentation at the Content Level

The goal of mobile applications is usually to deliver some form of content to the user. This can be short snippets of information for a traveller, eBooks or video content to pass the time or something else. A lot of issues can arise with any type of content when you wish to deliver it to a variety of devices.

Every mobile application delivers content to the user

The support for types of content and the way it is delivered vary between platforms. The device's hardware, e.g. the screen size, plays an important role.

1.3.2.8.1 Applications and Content Types

Applications can be seen as conglomerates of presentation logic, views flow, and actual content. The bare bones content types that can be distinguished from the other application components are

Text: the most basic type.

Pictures: due to the large fragmentation in screen size and types of devices, mobile platforms still have to deal with lower bandwidth and data limits, and content should be optimized for the device. Social media applications like **Facebook** and **Google+** provide immersive experiences showing images optimized for the width of the device, in order to avoid an overconsumption of bandwidth.

Video: For using video elements, several big players try to push their own preferred standards, fragmenting the mobile experience in this area. This afflicts the usage of different **codecs** and **streaming protocols**, and impact developers as well.

Apple poses some severe restrictions. Videos need to be encoded with the **H.264** codec and streamed with Apple's **HTTP Light Streaming** (**HLS**) protocol. This means that the more commonly used **MPEG** transport protocol (**MPEG-TS**) is not allowed.

Android is more flexible. Videos should still be encoded with **H.264**, but both **MPEG-TS** and **HLS** as streaming protocols are supported.

Windows Phone also uses the **H.264** codec for videos.

Audio: Audio is less complicated then video, since most platforms support the standardized **Advanced Audio Coding** (**AAC**) encoding scheme, next to supporting other commonly used codecs such as **MP3**.

Because they are fixed and immutable objects also the following can be considered as content types:

Books and Newspapers: e-books come in a variety of formats, despite the fact **interoperable formats** actually exist and have been agreed upon. Some of the more famous formats with widespread support are: **EPUB**, **HTML**, Apple's **iBook**, Amazon's **Kindle Format 8** (**KF8**), and **PDF**.

For several application types,[17] it is impossible to extract content types and look at them separately. The most important of these are

Games: fragmentation is a barrier to the quick growth of this emerging market. The variety of operating systems, APIs and the lack of cross-platform libraries are affecting mobile games as happened in the early days of desktop gaming, before the advent of OpenGL. Famous game developers such as **Gameloft** even go as far as only allowing a game to be played on a limited number of supported devices. Hardware fragmentation, with increasing performance gaps between devices, makes even more difficult to provide a consistent gaming experience.

1.3.2.8.2 Content-Related Fragmentation Handling

The Web as an Application Platform

The use of **web technologies** for application development purposes has proven to be a viable long-term candidate solution. Running on top of the Internet infrastructure, the web application ideology is rapidly gaining momentum amongst developers. In particular since this approach offers a number of important advantages over device-dependent development.[18]

A web-based application development approach has been explored from various perspectives. Developers can opt for pure web applications, running in a standard browser environment. In turn, a **hybrid web application approach** was introduced providing developers access to a richer API set, whilst still maintaining most of the cross-platform advantages from pure web applications.

Context-Aware Engineering

Five contextual base categories have been identified in the research:

Device context: it describes the characteristics of the **target device** that is being used to access the application, in order to be able to promptly react to the requirements.

[17] See deliverable D1.3, § 4

[18] See deliverable D1.1, § 5.5.1

User context: this model comprises knowledge regarding the user's specific abilities and disabilities, in order to enable accessibility requirements for providing support to elderly people, and people with disabilities.

Environment, Time, Historical context: they define where, how, and when the interactions between the user and an application are taking place (typically basing on sensors available on the device), in order to accordingly adapt the content being served.

Design

An approach that can be taken at design time to circumvent fragmentation issues and which is gaining momentum and popularity is **Responsive Web Design** (**RWD**). Here the actual layout of the application and the content is based on the viewport used to see it. It allows more easy reading and navigation with a minimum of resizing, panning and scrolling across a wide range of devices.

Runtime

A different type of approach requires **the content to be adapted at runtime**, as the application is in-use. One thing to consider is whether to perform the adaptation server-side through a proxy, or on the client device itself.

When delivering specific media types, specific content **transcoders** can be used as well. They are mainly used for video services, because of the large fragmentation issues.

1.3.3 Fragmentation as perceived by different actors

Fragmentation affects the vast majority of the aspects of the mobile industry, but it is perceived in different ways by different categories of stakeholders. We can say that also the way fragmentation is perceived is fragmented. This is due to the fact that different kinds of fragmentation exist and that each actor is exposed directly and thus more aware, only to a subset of these different kinds of fragmentation, even if indirectly affected by other kind of fragmentation. For some categories fragmentation and differentiation is part of their core business (e.g. hw fragmentation is what device manufacturer use for positioning in the market)

1.3.3.1 Stakeholders

In this chapter we will try to group stakeholders in categories and to analyze the way different kinds of fragmentation interact with each category of stakeholders.

1.3.3.1.1 Software Companies

This category groups industries producing mobile apps.

> *Software companies are ultimately the ones required to find solutions to issues caused by fragmentation.*

This is probably the category that is **more heavily impacted** by all kind of fragmentation, especially for people in charge of interacting directly with code such as developers, QA, people porting application from one platform to another etc. Software companies and developers are, ultimately, the ones required to find solutions to issues caused by fragmentation. Needless to say that being the ones in charge of solving problems, a deep knowledge of all the aspects of fragmentation is required.

The **service** and **network** fragmentation issues are somehow more distant from the daily work of such companies a developer, but knowledge of these kinds of fragmentation is required to quickly identify issues that could be seen as inexplicable if fragmentation is not taken into account (e.g. network filtering on some mobile network can lead to unexpected behaviours that are not easy to understand without fragmentation-oriented thinking).

It is interesting to remember that for software companies a **positive aspect of fragmentation** is that the know-how produced by addressing fragmentation in the past is a competitive advantage over new companies that does not have yet addressed fragmentation related issues.

1.3.3.1.2 End users

On the other side of the scale there are end users: this category is **less impacted directly by fragmentation** as effort has been put in by the whole market to reduce negative fragmentation impacts on end users and to help them to benefit from the different options.

End users benefit of the effort put by the whole market to reduce negative fragmentation impacts

For a non technical end user willing to make an informed choice for a new device it is often enough to be aware of the existence of different operating systems, different versions of each operating system, an overview of the form factors and hardware differences. In this sense, interaction with some kinds of **fragmentation produces a positive interaction** as it allows the user to access different kind of devices and have different options to choose from.

It is worth spending some words on the **app store fragmentation**, as this is often cause of unexpected (and often not perceived) issues for the end user.

The majority of the users are often not aware of the existence of multiple app stores providing different sets of application (even when this happens on their own device) and may think that an app is not available for their device while they just used the wrong app store. Samsung, for instance, installs the Samsung App store on the majority of his android devices in a prominent way. Often the users are not even aware that they can look for applications even in the Google Play app store (usually reachable in a less intuitive screen). If an app is available on the Play store and not on Samsung App, a user may have a suboptimal experience without even knowing.

1.3.3.1.3 Service Providers

The service providers' category includes very different kind of stakeholders, so it is not easy to analyze the impacts and interaction with fragmentation. Depending on the focus of service provided (and often depending on the department of the same service provider) interaction with the same kind of fragmentation can be positive or negative. In the majority of the cases, anyway, fragmentation has negative impacts as the same services should be provided; obviously this is not true for the service fragmentation as this is the core business and the differentiator for service providers.

1.3.3.1.4 Device Manufacturers

Fragmentation and differentiation in devices is one of the key aspects of the device manufacturers' core business. **Form factor** and **HW differentiation**, plus customization of OSs (such as the HTC Sense UI for Android) or Manufacturer specific application (Samsung App store, Samsung MusicHub...) is what allow device manufacturer to win (or lose) in the market. Basically DM are **one of the main sources of fragmentation** (and innovation). **External fragmentation has mixed impact**, as the choice of which OS should be installed on device produced by the DM is a choice that has a specific positioning meaning, allowing differentiation, but at the same time providing a Manufacturer-specific experience across different OSs has a cost (e.g. the aforementioned HTC Sense UI has been developed for Android, Brew and Windows Mobile). More information on such impact is available in D.1.1

Device Manufacturers are one of the main sources of fragmentation and innovation

Also **service fragmentation** is part of a DM strategy, specifically for what concerns the **app stores** that are source both of differentiation and income. This aspect and impact of app store fragmentation is well explained in D.1.2

1.3.3.1.5 Content providers

This category groups a wide range of professionals figures, including graphic designers, video and media providers, authors of articles or research, people in charge of adapting content provided by a third party (localization, summarization, conversion, media integration etc.).

Depending on the specific task, **deep knowledge in the specific domain is required** (e.g. producing or converting a video for a specific usage requires knowledge of the right video resolution and codec to use for the given device). With the exclusion of the too-technical programming language fragmentation, all other kinds of fragmentation have impacts on the way content should be designed and produced.

1.3.3.2 Fragmentation impacts perceived by stakeholders

From the deliverables produced within WP1 of project MOSQUITO, a map of impacts of fragmentation issues over the stakeholders is proposed, using the classification used in the present deliverable.

A evaluation of the importance of impacts is proposed. In some cases, both positive and negative impacts are identified. It is important to understand that some of the stakeholder gain benefit from the fragmentation. As an example, part of the content provider industry has **business depending on fragmentation**: graphic and video conversion between formats and adaptation of content is a cost for someone in the industry, but a business opportunity for other, so even if we report the fragmentation interaction for content provider as a negative interaction, this is not true for all the players in the space.

Table 1 - Interaction between different kind of fragmentation and different actors

Stakeholder Fragmentation	Software companies	End user	Service providers	Device manufacturers	Content provider
External	---	-/+	--	--/++	--
Internal	---	-	--	--	--
Programming language	---	/	--	--	/
Form factor	---	-/+	--	+++	---
HW	---	-/+	--	+++	---
Content	--	/	--	--	---
Service	--	--	+++	--/++	--
Network	--	-	-/+	-	--

Legend

- +++ **Very positive interaction** with this kind of fragmentation
- ++ **Positive interaction** with this kind of fragmentation
- + **Small positive interaction** with this kind of fragmentation
- - **Small negative interaction** with this kind of fragmentation
- -- **Negative interaction** with this kind of fragmentation
- --- **Very negative interaction** with this kind of fragmentation
- / **No significant interaction** with this kind of fragmentation

Analyzing this table, it is worth to spend some additional words on **internal fragmentation**:

Although it has negative impacts on all the players, still it is currently unavoidable due to the way the market is shaped.

Mobile OS providers need to release new versions of their OS to provide new functionalities to end users and new APIs to developers.

On one side, they want to limit internal fragmentation by providing automatic upgrades of the OS, but this is possible only at some extent as they also have to take into account device limitations (older phones may not have sufficient performances to provide a good user experience with new versions of an OS) and the need to push the users to buy new devices (this is true for instance for Apple, that provides both HW and SW).

When we add third-party players such as device manufacturers or carriers to the equation, **internal fragmentation is increased** because every upgrade to a new version is a cost that device manufacturers or carrier have to handle because of customization, porting and branding.

Every upgrade to a new version has a cost for Device Manufacturers

Similarly device manufacturer have to balance between making customers happy by upgrading their device to latest OS and encouraging users to buy newer devices, and releasing new devices with new functionalities provided by a newer version of the OS is often an effective way to encourage users to buy a new device. This kind of interactions and impacts are well explained in D1.1, chapters 8 and 9.

A similar scenario is valid for the **programming language fragmentation** that has historical and business related reasons to exist (and sometime also forms of emotive bounds of developers that are fond of a specific language).

Large companies such as Google and Apple have their entire development department oriented to use specific languages or tools and are not inclined to change this to reduce programming language fragmentation, so there is no proof of market moving in the direction of a reduction of number of programming languages.

The market is more inclined, instead, to provide ways to make it easier to share part of code, libraries or routine, and to adopt **cross platform tools and languages**.

In this sense we see increasing effort in adoption and promotion of technologies such as HTML5, web based apps and JavaScript oriented libraries. More information on such aspects is available in D1.1, chapters 4 and 5.

1.3.4 Standardisation gaps analysis and support

1.3.4.1 General approach

The figure below (Figure 2) can present a technical overview of the different technical areas on the left side and the different solutions present on the market while Figure 3 presents the associated standardisation organisations.

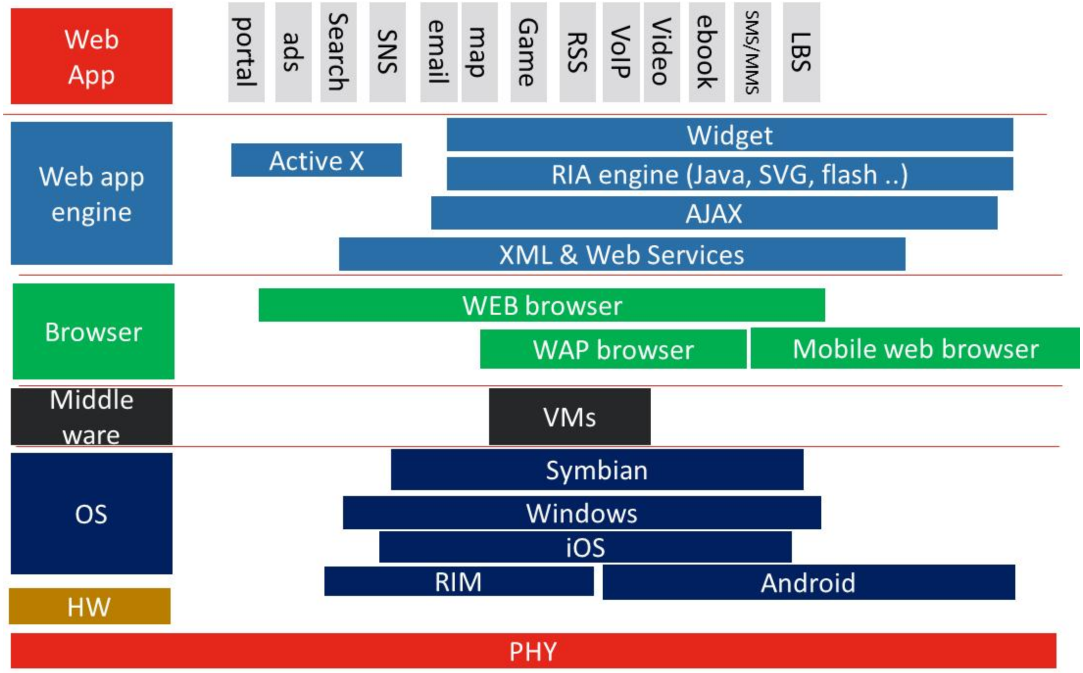


Figure 2: Mapping of mobile internet related technical areas

The same overview can be presented with the major standardisation-related organisations:

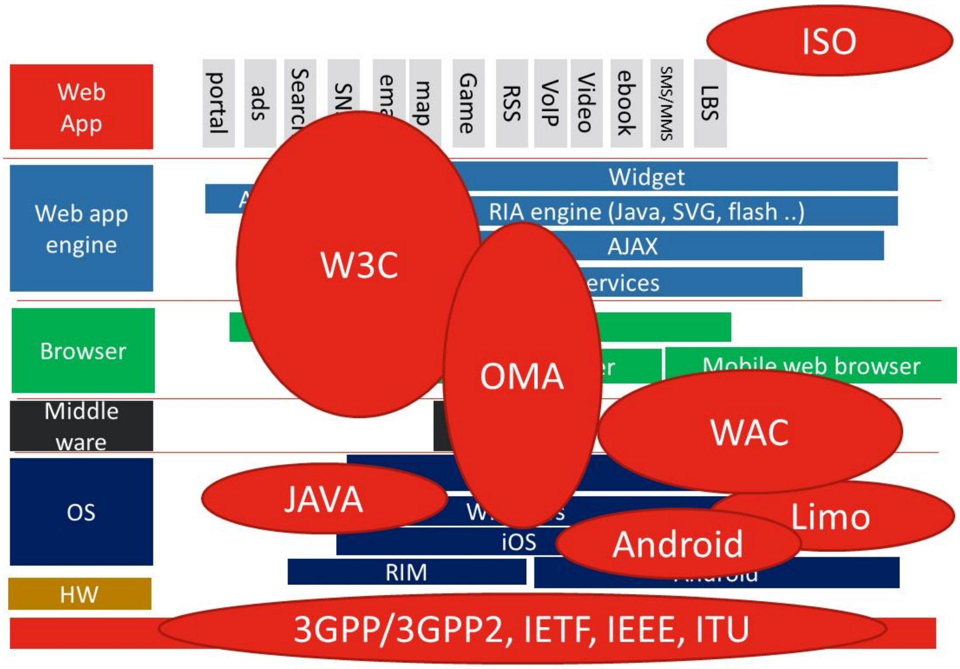


Figure 3: major standardisation organisations related to mobile internet

To efficiently deal with standardization , one has to be pragmatic as

The Magic bullet doesn't exist:

- One of the worst myths is the "unifying technology" that will make things "simpler and easier". Some have claimed that Java (J2ME) was the answer, then Flash Lite, then Webkit browsers, and most recently HTML5.
- While each solution has its merits, **there will *not* be any unification anytime soon**. Even as HTML5 richness has improved substantially, browser support will still vary and many phones will not support HTML5 comletely for years to come.
- Anyone who is waiting for a *single* silver bullet to solve fragmentation issues in mobile will be waiting a very long time, especially if they want to go after the global mobile opportunity.

There is a real politic: common standards is commodity standards for many

- The real politic is that Mobile is truly global and serves an extremely wide range of countries and users. There will naturally be a wide breadth of technologies, from CDMA vs. GSM protocols, J2ME vs. BREW, Mobile Apps vs. Mobile Web, xHTML vs. HDML, SMS vs. MMS and others to serve this market.
- Common standards = commodity standards for many players in this industry. It will be hard to drive any single set of worldwide standards given the different economic incentives of the many players, despite how good it would be for developers.

Therefore the Mosquito project evaluated where it can be influential depending on the motivation of market forces and how it can really improve the situation in the mobile market.

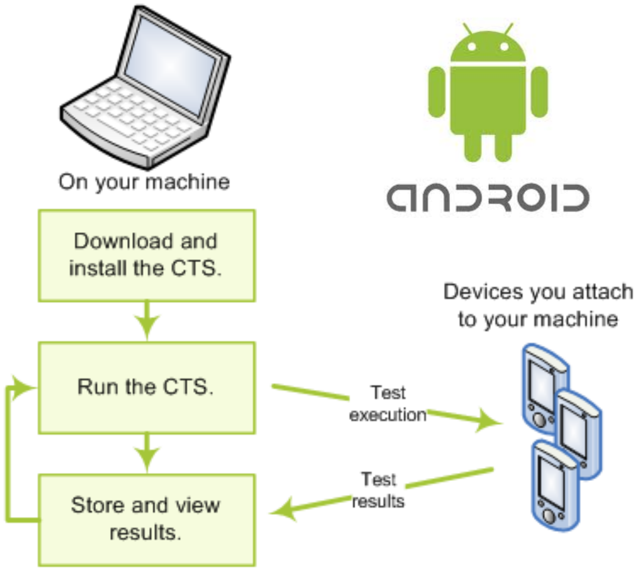
Therefore, to address standardisation and fragmentation, instead of looking at all the variety of issues in the blue sky, the project decided to look at the market trends and market figures to identify where actions on standards or market-accepted specifications are likely to have a major influence.

1.3.4.2 Gaps analysis

1.3.4.2.1 Testing environment can also create fragmentation and interoperability issues

Some issues are currently reported

- Android CTS -> **too much tests , may be a lot missing**
- Android versions -> too much versions
- As a developer, it's not just a matter of writing software but then testing it on potentially thousands of devices on hundreds on networks in hundreds of countries



Actions in **standardising** ANDROID test cases including using market proof test methodologies will greatly help to reduce android fragmentation

As mentioned within the deliverable D2.1, among the problems that surfaced which still to be addressed:

- **Device fragmentation**. 56% of Android developers said that operating system fragmentation among the various Android devices was a meaningful or "huge" problem, a percentage that actually increased over the past three months.
- **Store fragmentation**. Several developers expressed concern over Android app store fragmentation. "Generally," Baird reports, "developers seem to prefer a unified, single store experience like Apple's App Store."
- **Ease of development**. iOS outscored Android, but both were considered far easier to develop for than, say, Research in Motion's (RIMM) BlackBerry OS or Nokia (NOK) Symbian.
- **App visibility**. "iOS continues to lead," Baird reports, "followed by Blackberry, with Android still receiving poor marks in this category." Developers are particularly concerned about the level of "junk" apps in the Android ecosystem.
- **Ability to get paid**. iOS leads here too, followed by BlackBerry.

1.3.4.2.2 Web apps: need to pursue market supported standardisation activities on HTML5

As mentioned in the deliverable D2.1, HTML5 is a key area to focus on. Current activities where gaps are detailed in previous chapter are:

- **Graphics Forms** (SVG Tiny 1.2; HTML Canvas 2d context, CSS background and borders, CSS 2D transform model L3, CSS animation module L3, WOFF files 1.0
- **Multimedia** (HTML5, media capture API, HTML canvas 2d)
- **Forms** (HTML 5 attributes, etc)
- **Users interactions** (touch event, web notification, ..)
- **Data storage** (webstorage, file API, contacts API)
- **Sensors and hardware integration** (geolocation API, IS API, ..)
- **communication** (messaging API,HTML5 web browsing..)
- **Packaging** (HTML5 app apache, Widgets packaging..)
- **Performance and optimization** (web worksers, mobile web application best practices, ..)

As reported by Jonas Lind, A mobile area expert in an article called : The HTML5 hype – time for a reality check : The hype created by the promise of HTML5 has almost reached fever pitch during the last quarter. With HTML5/CSS3 it will be possible to run most types of applications directly in the "browser" and the need to install apps that execute native code will be a thing of the past. HTML5 can run from a cache in your smartphone/tablet/PC even if you are offline and the app can access the phone's GPS, compass, accelerometer, touch recognition and native video/audio control. App developers will no longer need to develop separate native versions for iPhone, Android, WP, Blackberry, Samsung/Bada, and WebOS. Just write once in HTML5 and run everywhere.

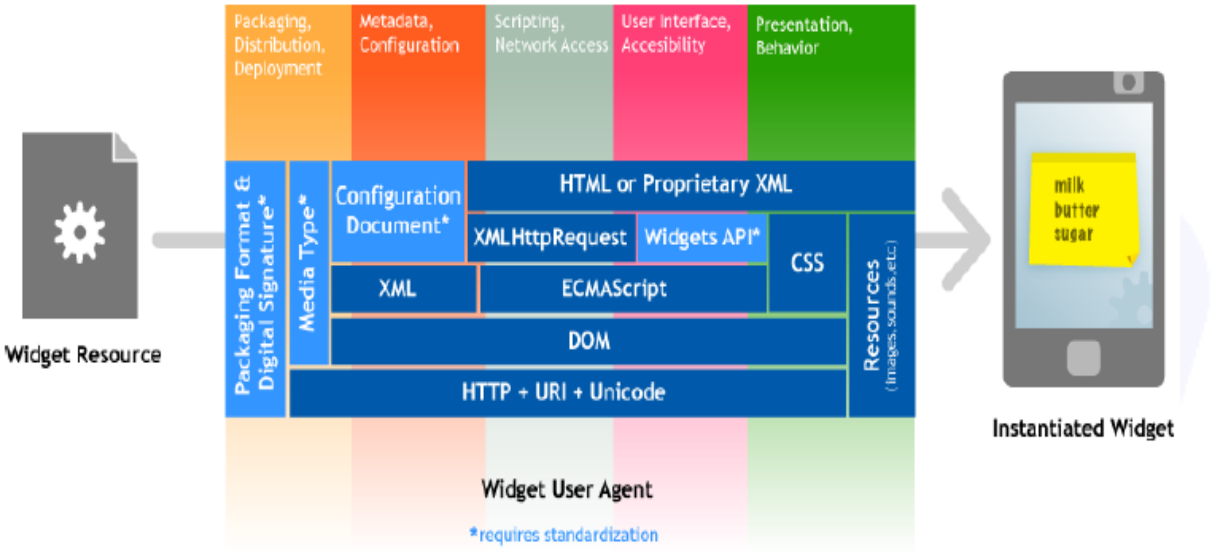
There is truth in all this and HTML5 is a great technology. But as usual during the peak of inflated expectations people tend to forget the limitations. HTML5 is still an immature technology. The final draft will be finished in mid-2011 and the W3C recently stated that the formal standard decision will be delayed until 2014. When people actually start using HTML5 the experience will most likely be underwhelming as developers are faced with the limitations of the technology. (This view is supported by comments from industry conferences.) Older handsets will most likely not be able to run full HTML5 web apps, which kind of defeats the vision of universal access, at least for the near future.

Native apps will always offer better performance, better UI/UX, and better integration with the device hardware. For example, HTML5 does not support augmented reality. HTML5 will, over time, be able to close the gap but if we assume that the native app platforms continue to develop, the goalpost will be moving as well. Ecosystem owners (Apple, Google, etc.) will of course work to make their native development environment as pleasant as possible to work with. In addition, cross-platform tools in the native environment will reduce the effort of porting from one platform to the other.

What the market tends to forget is that the fundamental trade-off between standardization and flexibility will not go away. By complying with the HTML5 standard, handset makers and web app developers will be unable to differentiate outside the limits set by the standard. It is inevitable that one global standard will not be fully capable of adapting to a highly heterogeneous base of various screen sizes, handsets, tablets, etc. Once a committee-based standard is finalized, innovations and new product features that are introduced after that point will not be included until the next upgrade of the standard. Apps will be better at taking full advantage of device variation and new functionality.

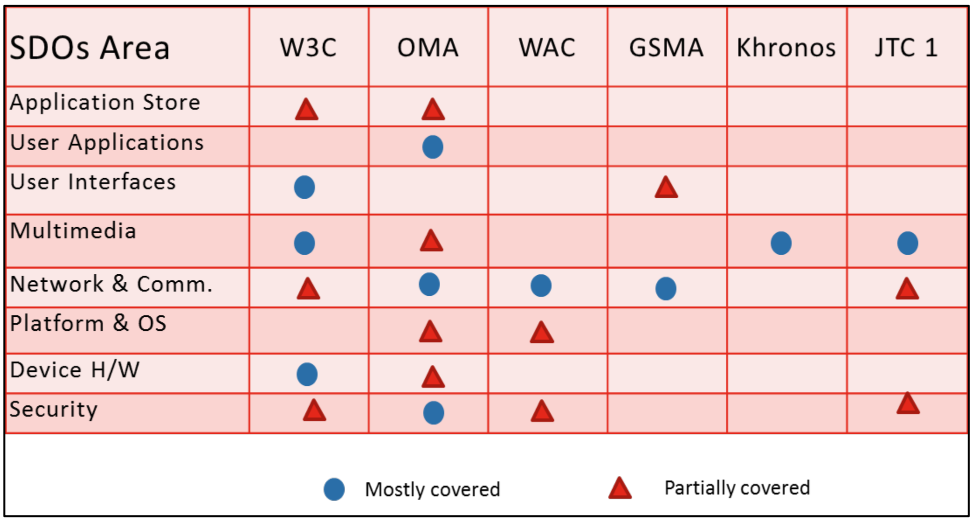
1.3.4.2.3 Interests and trends in Widget standardization

The continued proliferation of a class of software application known as *widgets* onto desktop computers and mobile phones has resulted in incompatibilities across most widget-related software. It is often reported how these incompatibilities currently affect distribution and deployment, accessibility, security, metadata, internationalisation and the device-independence of widgets.



1.3.5 Recommendations and roadmap

After the deep analysis of the standards work further explained in Deliverable D2.1, we then look at the impact of those standards on the future of app development. We took into account the market vision and then explains the business and technical requirements for future standards work. The overall driver behind proposing recommendation is to address pain points that plague the app development ecosystem as well as the emergence of new technologies. In preparing this report, we interviewed several industry people at several meetings, conference and one-to-one discussions to gather their understanding and requirements. Since much of the discussion revolves around standardisation, we closely analysed the standardisation process itself to find its shortcomings and whether it can be improved further. We have then broken the standardisation work as per each of the layers in the smartphone ecosystem that a developer comes across. Looking further into these layers, we have looked at gaps and identified the use-case, market adoption and issues with technologies where standardisation is required. A summary recommendation along with action required has been suggested for organisations to explore each of those topics and further build their activities, using this report as a reference. The report concludes that is important to have standardisation not just for present needs but also to serve the future evolution of smartphone industry. We are just at seeing a pass of the gradual evolution of the internet infrastructure industry and it is important to identify key technologies that will serve as a base for future application needs.



SDOs Area	W3C	OMA	WAC	GSMA	Khronos	JTC 1
Application Store	▲	▲				
User Applications		●				
User Interfaces	●			▲		
Multimedia	●	▲			●	●
Network & Comm.	▲	●	●	●		▲
Platform & OS		▲	▲			
Device H/W	●	▲				
Security	▲	●	▲			▲

● Mostly covered ▲ Partially covered

Figure 4: Overview of main standardisation activities for mobile apps with some level of coverage per technical areas as documented by a JTC1 report

We have identified 9 recommendations for standardisation

Recommendations for standardisation actions in :	
REC 1	Bar Codes
REC 2	Proximity Technologies
REC 3	Browser security
REC 4	App distribution
REC 5	App Server and frameworks
REC 6	Widget Development
REC7	UI Development
REC 8	Security and Certificate Management
REC 9	QA processes

1. **Bar Codes** - Open Mobile Alliance has listed Mobile 2D Bar Codes in 2012 standardisation release plan[19]. However, it will be very difficult to implement a standard bar code management platform until all the major major code registry owners do not have implement a mechanism to allow interoperability. This would require a format that can ensure that data be interchanged between these platforms without having to implement uniquely for each of the platforms.

Action Required - Define standards for implementation of bar codes on mobile devices supported interoperable bar code management registry. For e.g. GS1, the body that manages 1-D bar codes can lead this effort

[19] http://www.openmobilealliance.org/comms/pages/oma_2011_AR_release_plan.html

2. **Proximity Technologies** – There are several organisations working on defining the standard for NFC. While the technology standards have been well defined, there isn’t a standard that can allow multiples business processes to function using NFC. Hence, there is a need for standards organisations to lay down reference architecture for specific industry business processes, where the security and operational model of NFC can be clearly understood.

   Action Required – Standardise the access model for both client- and server-side NFC services on mobile devices. Need a broader consortium that is non-partisan and can help in delivering to the requirements of banks, OEMs, operators and services providers like Google

3. **Browser security** - The web is the biggest vector for distribution of malware and we can expect cyber criminals to continue their focus here but look to exploit new methods. As web technology merges with conventional thick client capabilities and is deployed across a wider range of devices the security model will need much revision and diligence from the security community to identify new attack vectors. Separating individual sessions and the rights of different authors, publishers and sites within the content rendering environment becomes all the more important. Hence, standards bodies should define a robust permissions model that allows apps to trust-check other apps. There is also a need for the browser to determine which kind of apps can access what device API’s based on its characteristics or trust score.

   Action Required – Ramp up the security and permissions model within the browser. Security vendors such as Symantec, Mcafee and chipmakers can supplement the efforts of browser makers

4. **App distribution** – Even as companies are exploring mobile web apps they face questions around discovery and monetisation opportunities. Hence, standards bodies need to define an approach to distribute apps, certify them and provide payment and advertising services around them. Mozilla and W3C is currently working on these however, with the recent split of W3C and WHATWG, there are two camps that will eventually define a split standard thus contributing to fragmentation.

   Action Required – Have a coherent set of distribution and certification policies in order to simplify the process of app distribution. Vendors like Verisign who maintain the global DNS registries should look closely at the standardisation effort

5. **App Server and frameworks** – There are several vendors that provide the app server runtime for apps and there are even more open source initiatives. However, majority of these cater to a specific use-case or business process. Since much of the data that apps use is common across apps, for example, personal information data or device data, it should be abstracted via a service bus with a well-defined nomenclature. It should be seen similar to other initiatives, where interoperability of data between systems is allowed, for e.g. SWIFT or OASIS standards. A standard is required where each of these variables can be defined and conveniently accessed through an app server.

   Action Required – Assess how a standard service structure could be designed, which can allow apps to use a common set of backend services for common workflows such as login etc.

6. **Widget Development** – Since widgets are client-less applications that run outside of the typical Web browser interface, they could be called a subset of the typical hybrid app in the context of mobile devices. Widgets can be downloaded, installed and customized in very much the same way as any other client-side application, as long as a compatible host runtime is present. There is a need to standardize these host run-time components on mobile platforms such that any server-side widget can be accessed by end-users. Currently the standard is well defined for desktops and there is a need to develop a similar standard for mobile devices.

Action Required – Need a standard interface that can allow server-side widgets to run across all kinds of devices

7. **UI Development** – There is no such global standard to develop natively for different form factors and screen sizes. It is also difficult to arrive at such as standard since OEM's prefer to add their own proprietary interface controls in order to differentiate against competitors. With mobile web and HTML5, it has possible to access native components as well as render server-side objects on devices. There are several frameworks that are address this need by componentising for both client and server services. Hence, a standard, within the context of HTML5 is required to define such adaptive UI development models.

Action Required – W3C is best placed to lead such a standards work. Organisations like Coremobweb, Google and Mozilla are already doing some work here.

8. **Security and Certificate Management** – Security remains the prime concern of majority of enterprises interested in developing apps. However, as discussed above, both the browser and the multi-platform world, makes it difficult to ensure a harmonised implementation of a security infrastructure. Several organisations are tiding over this problem by going in for app store-style mobile app management tools, in addition to their existing Mobile Device Management systems. Organisation who were content with BES as their de-facto device management standard area now coming to terms with the invasion of BYOD. In spite of several standards in the areas of cryptography, key management, certificate management and identity management there is no well-defined standard that provides a joined up view of all these systems put together.

Action Required – Government agencies such as NIST who typically define such standards should actively provide specifications for such kind of requirements

9. **QA processes** – Developers use several kinds of tool, programming languages and frameworks to develop apps for target mobile devices. Tools focus on providing the right libraries and classes while app stores actively monitor rogue apps from access to the protected app ecosystem. One critical component that is often missed is the QA process itself that can ensure bug-free apps and prevent potential vulnerabilities. Except for certain independent QA and testing vendors providing static and dynamic analysis for apps, there is no such standard, which defines how an app development should take care of issues like privacy and device API access.

Action Required – System Integrators and independent organisations like ISO or BCS who define testing specifications should actively look at defining QA processes for mobile apps

1.3.5.1 Summary of recommendations

	RECOMMENDATIONS	WHO CAN ACT BEST (Only suggestions)	WHEN -TIMING IMPORTANCE
1	Bar Codes	GS1	Low but should help developing new business
2	Proximity Technologies	NFC Forum ETSI	High as a major technologies to support new businesses
3	Browser security	W3C Security vendors	High – urgent as confidence is at stake and can prevent new business to take off
4	App distribution	Certification policies stakeholders	Medium but should help avoiding fragmentation is giving trust and confidence
5	App Server and frameworks	OASIS	Medium but should help avoiding fragmentation is giving trust and confidence
6	Widget Development	Open	Medium as interface to customers will also give confidence and support innovation through independance to platform
7	UI Development	W3C	low
8	Security and Certificate Management	e-gov agencies	High to ensure worldwide adoption
9	QA processes	ISO Quality committes, CEN	Medium

1.4 Expected impact

Should the market and the stakeholders be able to follow recommendations from the MOSQUITO project (see 1.3.5), some negative impacts of fragmentation could be reduced. In particular recommendations from 1 to 5 address very specific gaps, thus impacting the stakeholders directly working in specific fields, while recommendations from 6 to 10 could benefit a wider set of stakeholders.

Rec 6 and **Rec 7** will have positive impacts on **reducing the development effort** and can made less severe impacts related to external, internal and programming language related fragmentation.

Rec 8 will reduce service fragmentation impact

Rec 9 will make easier the development of mobile application (QA is a key aspect of development) and will **reduce the iterations** on the same section of code.

Rec 10 will help both development and content related fragmentation

We present here a table with the same structure as Table 1 with highlight on the cells that can be positively impacted directly by following the suggestions described above.

Stakeholder Fragmentation	Software companies	End user	Service providers	Device manufacturers	Content provider
External	---	-/+	--	--/++	--
Internal	---	-	--	--	--
Programming language	---	/	--	--	/
Form factor	---	-/+	--	+++	---
HW	---	-/+	--	+++	---
Content	--	/	--	--	---
Service	--	--	+++	--/++	--
Network	--	-	-/+	-	--

For the end user, positive impact is perceived as reduced effort is required to the entire market to produce software, innovation and services

It is important to notice that suggestions are meant to benefit directly the technical stakeholders, so although the End user column seems not to be impacted directly, the positive impact is due to the **reduced effort by the entire market** to produce software, innovation and services. Less costs and effort in the mobile chain will turn into better and cheaper products for the end user, and this is for sure a benefit, although an indirect one.

1.4.1 Conclusions

The analysis of the numerous symptoms of fragmentation has enucleated a series of causes for it:

The manufacturers produce a lot of new versions of their devices, improving at a rapid pace both the hardware and software. This causes older hardware rapidly not being actively supported by new versions of the operating system, so that also the applications developed for an older operating system (depending on an older hardware) cannot be run on more recent devices. This can be defined as **internal fragmentation**.

Different mobile platforms are still co-existing. Two of them are currently mainstream (**Apple iOS** and **Google Android**), while other are declining but still present (e.g. **BlackBerry**, **Symbian**, and **Windows Phone**).

Different platforms and different versions in the field of the same platform cause the need of having different **development tools**. Also the use of very different **programming languages** contributes in further fragmenting the development landscape (Java for Android and, in a different flavour, for BlackBerry; C/C++ and Objective-C for iOS; again C/C++ for the new BlackBerry 10; C# for Windows).

A **set of concrete steps** to take to reduce fragmentation has been identified and described in the Mosquito documents.

Fragmentation is **not always a cost**, for some players it is a key aspect for differentiation, business and market positioning.

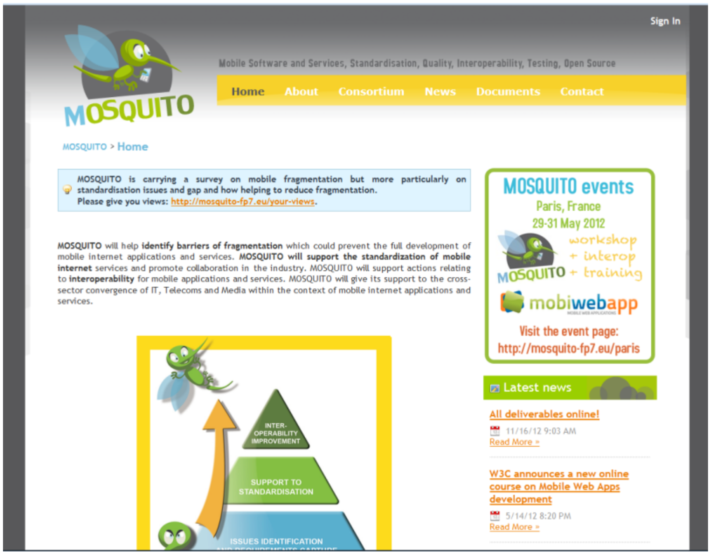
A possible solution against some kind of negative fragmentation is possibly the emerging of the development of **web applications as “first class citizens”**, basing on the capabilities of the on-board browsers. In this scenario it would be theoretically possible to write an application only once using well know technologies like HTML, CSS, and JavaScript, and building an executable application for every single platform. A high level of standardisation will be anyway demanded.

While the number of platforms is diminishing, due to the dominance by Android and iOS, **fragmentation related to the content** remains an issue not to be underestimated.

Standardization activities should be encouraged as can help reducing fragmentation, but effort should be put in **providing standards quickly** and as soon as they are needed, otherwise market will find workarounds and will be filled with different partial implementation of drafted standards, creating an even more fragmented landscape.

1.5 Project website

Project website is available at http://www.mosquito-fp7.eu/. All public deliverables produced by the project can be downloaded from there.



MOSQUITO is carrying a survey on mobile fragmentation but more particularly on standardisation issues and gap and how helping to reduce fragmentation.
Please give you views: http://mosquito-fp7.eu/your-views.

MOSQUITO will help **identify barriers of fragmentation** which could prevent the full development of mobile internet applications and services. **MOSQUITO will support the standardization of mobile internet** services and promote collaboration in the industry. MOSQUITO will support actions relating to **interoperability** for mobile applications and services. MOSQUITO will give its support to the cross-sector convergence of IT, Telecoms and Media within the context of mobile internet applications and services.

All deliverables online!
11/16/12 9:03 AM
Read More »

W3C announces a new online course on Mobile Web Apps development
5/14/12 8:20 PM
Read More »

Figure 5: project website homepage

2 Use and dissemination of foreground

2.1 Section A

2.1.1 List of all scientific (peer reviewed) publications related to the foreground of the project

No scientific articles (peer reviewed) have been published during the MOSQUITO project.

2.1.2 List of all dissemination activities

The dissemination activities included: publications, conferences, workshops, web sites/applications, press releases, flyer, articles published in the popular press, videos, media briefing, presentations, exhibitions, thesis, interviews, films, TV clips, and posters.

No scientific publication has been done by the MOSQUITO project.

2 Use and dissemination of foreground

2.1 Section A

2.1.1 List of all scientific (peer reviewed) publications related to the foreground of the project

No scientific articles (peer reviewed) have been published during the MOSQUITO project.

2.1.2 List of all dissemination activities

The dissemination activities included: publications, conferences, workshops, web sites/applications, press releases, flyer, articles published in the popular press, videos, media briefing, presentations, exhibitions, thesis, interviews, films, TV clips, and posters.

No scientific publication has been done by the MOSQUITO project.

TEMPLATE A2: List of dissemination activities								
NO.	Type of activities	Main leader	Title	Date	Place	Type of audience	Size of audience	Countries addressed
1	News on aW3C mailing list (public-test-infra@w3.org)	INNO/W3C	Invitation to the “Mobile Web App interoperability event on 6-7 December 2011 in Sophia-Antipolis, France »	23 novembre 2011	Internet	Developers, Browser vendors		
2	News on a W3C mailing list (public-test-infra@w3.org	INNO/W3C	Invitation: Mobile Web App interoperability event	14 novembre 2011	Internet	Developers, Browser vendors		
3	Press release	INNO	Press release send to MEP	October 2010	N/A	Member of European Parliament	100	European countries
4	Participation in conference	INNO	EC collaboration event (Presentation of call 1 and call 5 projects and networking)	19-20 October 2010	Brussels	Industry, research	500	European countries
5	Participation in conference	IBBT	Apps World	29/30 October 2010	London			Europe

6	MOSQUITO flyer	INNO	Creation of the MOSQUITO flyer, highliting projects' objectives	December 2010	N/A		N/A	Europe
7	MOSQUITO posters	INNO	Creation of the MOSQUTO poster	December 2010			N/A	Internet
8	News: mobile worlds congress	INNO	Announcement of the MOSQUITO participation in the event	31 January 2011	MOSQUITO website			
9	Participation in conference + flyer distribution	ETSI	Mobile World Congress	14 February 2011	Barcolona		3000	Worldwide
10	News: first MOSQUITO event	INNO	Announcement of the first MOSQUITO event	14 February 2011	MOSQUITO website			
11	Second MOSQUITO workshop	INNO	Announcement of the second MOSQUITO event	5 April 2011	MOSQUITO website			
12	D2.1 published!	INNO	Announcement of the publication of the D2.1 deliverable	3 May 2011	MOSQUITO website			
13	News section re-organized	INNO	News to inform website users that the news section has been redesigned	3 May 2011	MOSQUITO website			
14	Interop event flyer	INNO	Publication of the first workshop/interop event flyer	3 May 2011	MOSQUITO website			

15	Google I|O 2011	INNO	Announcement that the MOSQUITO project will be present, through IBBT at the Google I|O event.	3 May 2011	MOSQUITO website			
16	Participation in conference	IBBT	Google I|O	15/16 May 2011	San Francisco, CA	Industry, research	5000	Woldwide
17	Organisation of first international MOSQUITO workshop.	All (led by TMC)	First MOSQUITO workshop	5 July 2011		Industry, research	40	
18	Organisation of first international MOSQUITO interop event.	All (led by TMC)	First MOSQUITO interop event	6/7 July 2011		Developers	30	
19	Participation	inno	EC concertation days event	28/29 September 2011	Brussels	Industry, Research 500 attendants involved in FP7 call 1 and 5.	500	
20		INNO	MOSQUITO deliverables published online	18 October 2011	Internet			

	News on a W3C mailing list (public-test-infra@w3.org	INNO/W3C	Mobile Web Applications Interoperability Event	18 October 2011	Internet	Developers , Browser vendors		
21	News	INNO	Incoming event: second MOSQUITO interop event!	6 December 2011	Internet			
22	Publication	INNO	SOFI Newsletter	March 2012	Internet	ICT stakeholders		
23	Participation	ETSI	EMMEA	8 March 2012	Internet			
24	News	INNO	Launch of the MOSQUITO survey	9 March 2012	Internet			
25	Participation	FT	Mobile Mondays	May 2012	Paris	Developers		
26	Participation and organisation of a session on HTML5	ETSI / inno	Aalborg FIA	10 May 2012	Aalborg	Services provider, developers, open source communities, …		
27	News	INNO	W3C announces a new online course on Mobile Web Apps development	14 May 2012	Internet			

28	News	INNO	MOSQUITO in Paris: in two weeks!	14 May 2012	Internet			
29	Participation	ETSI	ETSI colloquium	6 July 2012	Sophia Antipolis	Standardisation supporters		

2.2 Section B (Confidential[20] or public: confidential information to be marked clearly)

Part B1 - applications for patents, trademarks, registered designs, etc..

No applications for patents, trademarks, registered designs, etc during the MOSQUITO project

Part B2 – exploitable foreground

Type of Exploitable Foreground	Description of exploitable foreground	Confidential Click on YES/NO	Foreseen embargo date dd/mm/yyyy	Exploitable product(s) or measure(s)	Sector(s) of application[21]	Timetable, commercial or any other use	Patents or other IPR exploitation (licences)	Owner & Other Beneficiary(s) involved
Tests plan	The MOSQUITO test plan proposes a methodology for testing mobile applications (especially Android mobile applications) (D3.1)	NO	N/A	N/A	Mobile internet industry	N/A	N/A	European Commission, mobile internet stakeholders (developers, browser vendors…)
Tests suite	Set of HTML5 tests covered various HTML5 specifications (inputs, server-side events and DeviceOrientation) (available on the MOSQUTO website and submitted to W3C)	NO	N/A	N/A	Mobile internet industry	N/A	N/A	European Commission, mobile internet stakeholders (developers, browser vendors…)
Recommendations	Recommendations for standardisation actions in the field of Mobile Internet applications and services (D2.2).	NO	N/A	N/A	Mobile internet industry	N/A	N/A	European Commission, mobile internet stakeholders (developers, browser vendors…)

[20] Note to be confused with the "EU CONFIDENTIAL" classification for some security research projects.

[21] A drop down list allows choosing the type sector (NACE nomenclature) : http://ec.europa.eu/competition/mergers/cases/index/nace_all.html

3 Report on societal implications

A General Information *(completed automatically when **Grant Agreement number** is entered.*	
Grant Agreement Number:	258067
Title of Project:	MOSQUITO
Name and Title of Coordinator:	Franck Le Gall, Director

B Ethics	
1. Did your project undergo an Ethics Review (and/or Screening)? • If Yes: have you described the progress of compliance with the relevant Ethics Review/Screening Requirements in the frame of the periodic/final project reports? Special Reminder: the progress of compliance with the Ethics Review/Screening Requirements should be described in the Period/Final Project Reports under the Section 3.2.2 *'Work Progress and Achievements'*	***No***
2. Please indicate whether your project involved any of the following issues (tick box) :	
RESEARCH ON HUMANS	
• Did the project involve children?	
• Did the project involve patients?	
• Did the project involve persons not able to give consent?	
• Did the project involve adult healthy volunteers?	
• Did the project involve Human genetic material?	
• Did the project involve Human biological samples?	
• Did the project involve Human data collection?	
RESEARCH ON HUMAN EMBRYO/FOETUS	
• Did the project involve Human Embryos?	
• Did the project involve Human Foetal Tissue / Cells?	
• Did the project involve Human Embryonic Stem Cells (hESCs)?	
• Did the project on human Embryonic Stem Cells involve cells in culture?	
• Did the project on human Embryonic Stem Cells involve the derivation of cells from Embryos?	
PRIVACY	
• Did the project involve processing of genetic information or personal data (eg. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	
• Did the project involve tracking the location or observation of people?	
RESEARCH ON ANIMALS	
• Did the project involve research on animals?	
• Were those animals transgenic small laboratory animals?	
• Were those animals transgenic farm animals?	
• Were those animals cloned farm animals?	
• Were those animals non-human primates?	
RESEARCH INVOLVING DEVELOPING COUNTRIES	
• Did the project involve the use of local resources (genetic, animal, plant etc)?	
• Was the project of benefit to local community (capacity building, access to healthcare, education etc)?	
DUAL USE	
• Research having direct military use	

• Research having the potential for terrorist abuse	

C Workforce Statistics

3. Workforce statistics for the project: Please indicate in the table below the number of people who worked on the project (on a headcount basis).

Type of Position	Number of Women	Number of Men
Scientific Coordinator	0	1
Work package leaders	1	3
Experienced researchers (i.e. PhD holders)	0	0
PhD Students	0	1
Other	9	26

4. How many additional researchers (in companies and universities) were recruited specifically for this project?	**0**
Of which, indicate the number of men:	N/A

D Gender Aspects

5. Did you carry out specific Gender Equality Actions under the project?

○ Yes
● No

6. Which of the following actions did you carry out and how effective were they?

	Not at all effective				Very effective
❑ Design and implement an equal opportunity policy	○	○	○	○	○
❑ Set targets to achieve a gender balance in the workforce	○	○	○	○	○
❑ Organise conferences and workshops on gender	○	○	○	○	○
❑ Actions to improve work-life balance	○	○	○	○	○

○ Other:

7. Was there a gender dimension associated with the research content – i.e. wherever people were the focus of the research as, for example, consumers, users, patients or in trials, was the issue of gender considered and addressed?

○ Yes- please specify

● No

E Synergies with Science Education

8. Did your project involve working with students and/or school pupils (e.g. open days, participation in science festivals and events, prizes/competitions or joint projects)?

○ Yes- please specify □

● No

9. Did the project generate any science education material (e.g. kits, websites, explanatory booklets, DVDs)?

○ Yes- please specify □

● No

F Interdisciplinarity

10. Which disciplines (see list below) are involved in your project?

○ Main discipline[22]: 2.2

○ Associated discipline[22]: - ○ Associated discipline[22]: -

[22] Insert number from list below (Frascati Manual).

G Engaging with Civil society and policy makers

11a Did your project engage with societal actors beyond the research community? *(if 'No', go to Question 14)*	● ○	Yes No

11b If yes, did you engage with citizens (citizens' panels / juries) or organised civil society (NGOs, patients' groups etc.)?

● No
○ Yes- in determining what research should be performed
○ Yes - in implementing the research
○ Yes, in communicating /disseminating / using the results of the project

11c In doing so, did your project involve actors whose role is mainly to organise the dialogue with citizens and organised civil society (e.g. professional mediator; communication company, science museums)?	○ ●	Yes No

12. Did you engage with government / public bodies or policy makers (including international organisations)

○ No
● Yes- in framing the research agenda
○ Yes - in implementing the research agenda
○ Yes, in communicating /disseminating / using the results of the project

13a Will the project generate outputs (expertise or scientific advice) which could be used by policy makers?

● Yes – as a **primary** objective (please indicate areas below- multiple answers possible)
○ Yes – as a **secondary** objective (please indicate areas below - multiple answer possible)
○ No

13b If Yes, in which fields? Information Society

Agriculture
Audiovisual and Media
Budget
Competition
Consumers
Culture
Customs
Development Economic and Monetary Affairs
Education, Training, Youth
Employment and Social Affairs

Energy
Enlargement
Enterprise
Environment
External Relations
External Trade
Fisheries and Maritime Affairs
Food Safety
Foreign and Security Policy
Fraud
Humanitarian aid

Human rights
Information Society
Institutional affairs
Internal Market
Justice, freedom and security
Public Health
Regional Policy
Research and Innovation
Space
Taxation
Transport

13c If Yes, at which level?

○ Local / regional levels
○ National level
● European level
● International level

H Use and dissemination

14. How many Articles were published/accepted for publication in peer-reviewed journals?	**0**
To how many of these is open access[23] provided?	**0**
How many of these are published in open access journals?	
How many of these are published in open repositories?	
To how many of these is open access not provided?	**0**
Please check all applicable reasons for not providing open access:	
❑ publisher's licensing agreement would not permit publishing in a repository ❑ no suitable repository available ❑ no suitable open access journal available ❑ no funds available to publish in an open access journal ❑ lack of time and resources ❑ lack of information on open access ❑ other[24]: ……………	

15. How many new patent applications ('priority filings') have been made? *("Technologically unique": multiple applications for the same invention in different jurisdictions should be counted as just one application of grant).*	**0**

16. Indicate how many of the following Intellectual Property Rights were applied for (give number in each box).	Trademark	**0**
	Registered design	**0**
	Other	**0**

17. How many spin-off companies were created / are planned as a direct result of the project?	**0**
Indicate the approximate number of additional jobs in these companies:	

18. Please indicate whether your project has a potential impact on employment, in comparison with the situation before your project:

❑ Increase in employment, or	❑ In small & medium-sized enterprises
❑ Safeguard employment, or	❑ In large companies
❑ Decrease in employment,	● None of the above / not relevant to the project
❑ Difficult to estimate / not possible to quantify	

19. For your project partnership please estimate the employment effect resulting directly from your participation in Full Time Equivalent (*FTE = one person working fulltime for a year*) jobs:	*Indicate figure:*
Difficult to estimate / not possible to quantify	●

[23] Open Access is defined as free of charge access for anyone via Internet.

[24] For instance: classification for security project.

I Media and Communication to the general public

20. As part of the project, were any of the beneficiaries professionals in communication or media relations?

○ Yes ● No

21. As part of the project, have any beneficiaries received professional media / communication training / advice to improve communication with the general public?

○ Yes ● No

22 Which of the following have been used to communicate information about your project to the general public, or have resulted from your project?

●	Press Release	●	Coverage in specialist press
❑	Media briefing	❑	Coverage in general (non-specialist) press
❑	TV coverage / report	●	Coverage in national press
❑	Radio coverage / report	●	Coverage in international press
●	Brochures /posters / flyers	●	Website for the general public / internet
❑	DVD /Film /Multimedia	❑	Event targeting general public (festival, conference, exhibition, science café)

23 In which languages are the information products for the general public produced?

❑	Language of the coordinator	●	English
●	Other language(s)		