Telematics and Informatics 28 (2011) 22–31

Contents lists available at ScienceDirect

Telematics and Informatics

journal homepage: www.elsevier.com/locate/tele

Mobile application market: A developer's perspective

Adrian Holzer [a,*], Jan Ondrus [b]

[a] Information Systems Institute, University of Lausanne, CH-1015 Lausanne, Switzerland
[b] ESSEC Business School, 95021 Cergy, France

ARTICLE INFO

Article history:
Received 15 June 2009
Received in revised form 24 December 2009
Accepted 18 May 2010

Keywords:
Mobile platforms
Mobile application development
Market analysis

ABSTRACT

Major software companies, such as Apple and Google, are disturbing the relatively safe and established actors of the mobile application business. These newcomers have caused significant structural changes in the market by imposing and enforcing their own rules for the future of mobile application developments. The implications of these changes do not only concern the mobile network operators and mobile phone manufacturers but also bring additional opportunities and constraints for current mobile application developers. Therefore, developers need to assess what their options are and how they can take advantage of these current trends. In this paper, we take a developer's perspective in order to explore how the identified trends will impact the mobile application development markets. Our preliminary analysis leads us to suggest eight propositions which summarize our findings and can be the starting points for future empirical research.

1. Introduction

Mobile computing has caught the attention of the research community for quite some time (Forman and Zahorjan, 1994) and has also reached the commercial industry and mainstream consumers via smartphones and PDAs. The constant improvement in hardware related to mobile computing, such as better processing power, larger wireless network bandwidth, enhances capabilities of mobile devices. More than ever, mobile devices can run rich stand-alone applications as well as distributed client–server applications that access information via web gateways. Lately, the development of mobile applications has generated more interest among the independent and freelance developer community. This has opened up new avenues for future mobile application and service development. The potential of the mobile application market is expected to reach $9 billion by 2011, according to Compass Intelligence.[1]

For many years, the development of mobile services was mostly controlled and managed by the mobile network operators (MNO), phone manufacturers, and some mobile application and content providers. Traditionally, in the mobile application industry, there are several actors intervening along the value chain (Adrian, 2002; Ballon et al., 2008; Barnes, 2002; Buellingen et al., 2004; Funk, 2009; Maitland et al., 2002; Tsalgatidou et al., 2001), in which each actor has its own importance. Recently, this has changed with the arrival of software companies with new mobile phones and platforms such as the iPhone and Android. The market structure and value chain are evolving (de Reuver and Haaker, 2009; Feijóo et al., 2008). Roles are changed, combined and exchanged. Some actors lost control on the device (i.e., mobile network operators – MNOs), others got new revenue streams (i.e., portal providers), and some became more seamlessly integrated into the platforms (i.e., financial institutions, content providers).

* Corresponding author. Tel.: +41 78 689 87 68.
E-mail address: adrian.holzer@gmail.com (A. Holzer).

[1] <http://www.compass-intelligence.com/content.aspx?title=PressRelease04>.

0736-5853/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tele.2010.05.006

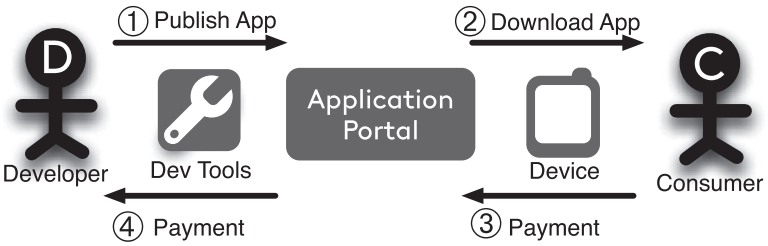

Fig. 1. Mobile application distribution process.

The current mobile development market is dominated by five big platform providers (i.e., providers of operating systems and development tools) namely: Nokia with its Symbian OS (52.4%),[2] RIM with its Blackberry OS (16.6%), Microsoft with its Windows CE OS family (11.8%), Apple with its iPhone OS (8.2%), and LiMo Foundation with its Linux Mobile operating system (8.1%). In addition, Google launched its Android operating system and is already expected to become part of the big players rapidly in the industry. Gartner predicts that it will become the second most popular platform behind Nokia by 2013 (Cozza, 2009).

With the current evolutions observed, it is not yet clear who will be the winners and losers. However, we can already notice that the situation will not go back to its previous state. We perceive new opportunities and threats for incumbents while the market structure is changing. There is a real need to better understand the current context to take advantages of these changes from a developer's perspective.

In this paper, we propose to analyze the current mobile application market from a developer's perspective. We start by examining the mobile application development mechanism.[3] It is essential step in order to better understand how the different elements are intertwined and interact with each other. Using the different identified elements, we analyze and discuss some trends that are affecting the mobile application market. We continue our research effort by examining the implications of each determined trends for developers. In order to enhance our contribution, we suggest eight propositions derived from the discussed implications. These propositions summarize our analysis and could open new avenues for further empirical research. In conclusion, we summarize the findings and discuss the remaining issues.

2. Mobile application market mechanism

To structure the description of the current practices, we propose to examine the current mobile application development mechanism from a developer's perspective. We analyze the application distribution process, and classify platforms in different categories depending on their approach.

The mobile application distribution process is a process by which an application is developed, brought to the market and purchased by customers, and used on mobile devices. This process involves three main components as depicted in the model in Fig. 1.

First, the developer uses *development tools* to build its mobile application. Second, the developer publishes its application on a *portal*, from which the consumer can download the application onto its *mobile device*. This model, follows a mediated approach as described in Adrian (2002), where a third party (i.e., the application portal) plays an intermediary role between the service provider and the customer. This approach is different from the walled garden approach which was popular until recently where MNOs where in charge of being the interface between customers and service providers.

The application distribution model depicted in Fig. 1 represents a typical two-sided market (Parker, 2005; Rochet and Tirole, 2003; Shapiro and Varian, 1998) with developers on one side and consumers on the other. In such a market, an increase or decrease on one side of the market induces a similar effect on the other side. In other words, in our specific case, as the number of consumers increases for a given platform, portal, or mobile device, the number of developers attracted to this platform, portal or device will also increase. Similarly, as the number of developers, and thus the number of applications increase, the platform, portal, or mobile device will attract even more consumers. So on the one hand, developers have an incentive to develop for the most popular mobile devices using the most popular platform, and to publish their applications on the most popular portal in order to reach the largest number of consumers. On the other hand, consumers have an incentive to buy devices running a platform with many applications. This mechanism creates a positive feedback loop, as depicted in Fig. 2.

Successful two-sided markets can lead to a high volume of transactions and can thus be interesting for a middleman to charge a fee per sale. In the current market, application portals play this role and charge 30% of the application retail price for each transaction.

[2] Percentages represent worldwide smartphone sales by OS in 2008. *Source*: http://www.gartner.com/it/page.jsp?id=910112 (June 2009).

[3] We analyze the current mobile application market by comparing the major commercial platform providers in the European and North American markets. These differ from East Asian markets such as the Japanese and South Korean ones, where MNO have more power to set their own standards and specifications.

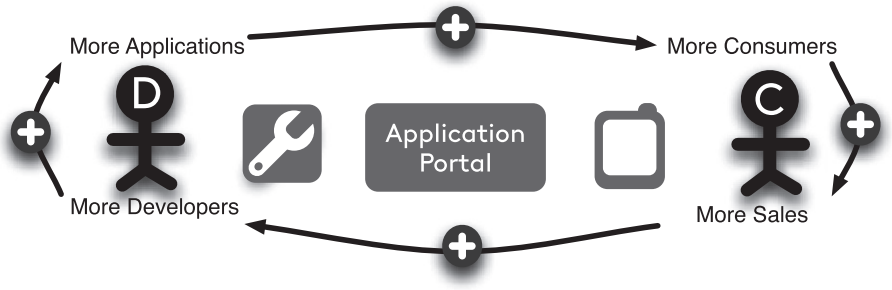

Fig. 2. Positive feedback loop in the two-sided mobile application market.

In order to trigger the feedback loop, typical two-sided markets, such as videogames or credit cards, treat the consumer side of the market as loss-leaders, which means that they make little or no profit in bringing the consumer on board by selling game consoles or credit cards. As far as the mobile industry is concerned, the potential revenue stream from a two-sided application market seems to have been widely ignored until the introduction of Apple's AppStore. The impacts of the arrival of new portals will be analyzed in the next section.

3. Analysis of the mobile application market and trends

In this section we analyze the current state of the mobile application market as well as trends with respect to the three components of the application distribution process model presented in Fig. 1. We examine the approaches used by platform providers towards (i) development tools, (ii) portals and (iii) devices. Furthermore, we analyze how these three components are or can be (iv) integrated together.

3.1. Mobile application development tools

Central to every development platform, software development kits (SDK) enable third-party developers to build applications running for the platform. These kits usually include libraries, debuggers, and handset emulators, among other useful development tools. Often platforms provide integrated development environments (IDE) in order to facilitate the development process. Existing platforms have taken different approaches when sharing their SDK with developers. Some have chosen to restrict access as much as possible, whereas others have chosen to disclose the entire source code of their SDK and OS. As presented in Capiluppi and Michlmayr (2007) and Raymond (1999), we distinguish *closed technology* following a *cathedral* model from *open technology* following a *bazaar* model.

3.1.1. Closed technology

In a *cathedral* model, platform providers keep control over all strategic decisions about the platform. By nature, these platform providers must hire developers. They cannot leverage on a freelance developer community to support the development process. Currently, three platforms (i.e., Apple, Microsoft and RIM), representing roughly half of the total smartphone users, chose the proprietary software path.

3.1.2. Open technology

By contrast, in a bazaar model, platforms grant developers access to all or parts of the SDK and OS source code. There is no central architect responsible for the platform. The use of open technology by platform providers encourages the gathering of a community of developers. As a result, development and maintenance costs of OS and eventually mobile applications are reduced. In the current market, Nokia, Linux and Google have chosen to provide open platforms.

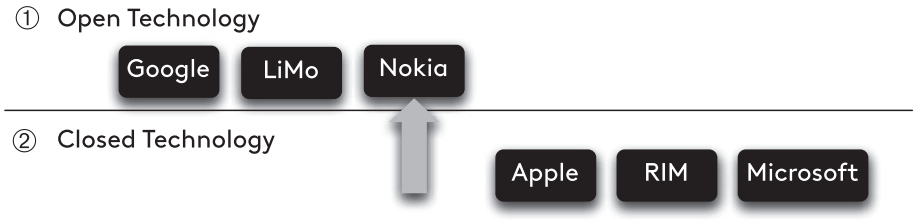

Fig. 3. Technological trends.

3.1.3. A trend towards open technology

Among the major mobile platforms, LiMo has been the only player in the open-source field. Since then Google has adopted the open-source ideology when launching Android OS. Later, Nokia also joined the team after acquiring and making Symbian OS open-source. The shift of major players towards openness had a significant impact. The market moved from a majority of closed systems to a small predominance of devices running open-source systems (Fig. 3).

3.2. Mobile application portals

In the distribution process, an application is developed and made available to customers through an application portal. The mobile application portal is an essential component in the mobile application distribution process. Portals play the role of intermediary between developers and consumers. Some scholars predicted that the number of portals would increase (Buellingen et al., 2004), whereas others predicted that the portal market would consolidate over time (Barnes, 2002). In the current market, both phenomena are observable. Several platforms use a *centralized* single point of sale strategy, while others use a *decentralized* multiple points of sale strategy.

Application portals can leverage on the long tail (Anderson, 2006) distribution of their sales. As there is no storage cost, portals can generate revenues not only from their best selling applications but also from a plethora of applications only sold a few times. Even though the revenue from the long tail might not represent a substantial part of the overall revenue (Elberse, 2008), this revenue stream should not be neglected. This is especially relevant as there is no extra cost in carrying lame-duck applications. As a potential effect, the size of the tail will grow with the number of developers attracted to the portal.

3.2.1. Decentralized portal

With a decentralized portal, developers can freely upload and distribute their applications on any third-party portal, as there is no centralized policy. In this model, all portal providers compete in order to attract customers and applications. The downside for the consumer is that the great variety of portals does not provide a comprehensive overview of existing applications. An advantage for platform providers is that there is no need to put in place and maintain a centralized solution. The application market is handed over to third parties. Players such as Nokia, Microsoft, and LiMo have mostly used a decentralized portal approach so far.

3.2.2. Centralized portal

In this model, one portal is proposed as the main portal on which all applications are published. This approach gives the main portal provider a competitive advantage over others. Consumers can easily find and download applications. It also facilitates the job of developers by offering a single point of sale. Centralized portals can benefit from network effects by attracting more consumers and more developers. Apple and Google propose a single point of sale with the AppStore and the Android Market. However, these two platforms have different approaches. On the one hand, Apple pushes a unique and exclusive portal with a strict application review process. Google, on the other hand, does not restrict the publication of applications to its portal. Moreover, there are no plans to review applications prior to publication as it counts on consumers' feedback to exclude inappropriate and low-quality applications.

3.2.3. Towards portal centralization

A centralized application portal is an ideal middleman between consumers and developers. However, prior to the introduction of Apple's AppStore and more recently Google's Android Market, platforms did not operate a central portal. With the introduction of its AppStore, Apple has proved that a mobile application market should not be underestimated and can represent an important revenue stream. According to CEO Steve Jobs, the AppStore has generated a revenue of a million dollars a day in its first month of existence.[4] There are currently over 100,000 applications on the portal and there have been well over two billion downloads so far.[5] Following Apple's lead, traditional platforms like Nokia, RIM and Microsoft are moving in this direction. Nokia is pushing its OVI portal and RIM has developed its App World. Microsoft is also planning to launch its own version of the AppStore called SkyMarket with the next version of Windows Mobile (WM7). The trend toward centralization can be seen in Fig. 4.

3.3. Device set

The device used by customers is more than ever of central importance. New technical features, such as wireless broadband internet connection, precise location sensors (i.e., GPS), high-performance digital camera, large displays, better user interfaces, enhanced the development of more advanced mobile applications. Platforms can have different approaches. A platform could dedicate itself to one type of device or a set of varied devices.

[4] Wall Street Journal, August 11, 2008: http://online.wsj.com/article/SB121842341491928977.html?mod=2_1571_topbox.
[5] *Source*: http://www.apple.com/pr/library/2009/11/04appstore.html.

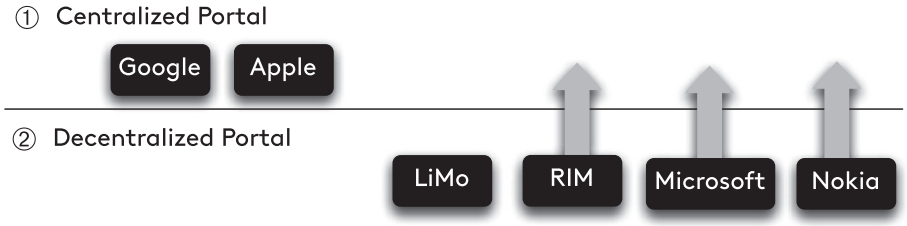

Fig. 4. Portal trends.

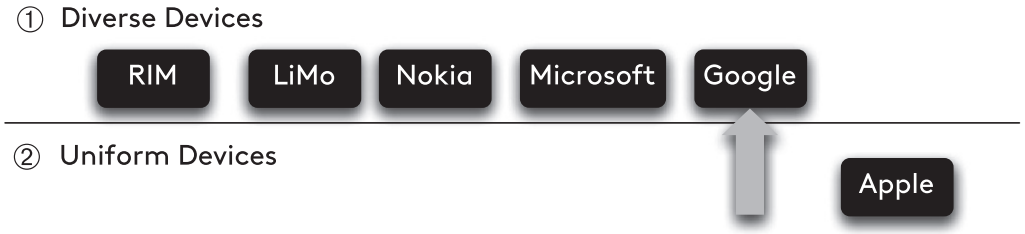

Fig. 5. Device set.

3.3.1. Device uniformity

Platforms can chose to target only one or a few devices with similar technical features. This approach allows platform providers to have an extensive control on devices and standardized features. Moreover, OS can be more easily fine-tuned to the uniform devices. The main drawbacks reside in a strong reliance on the device manufacturer and possibly a smaller user base. A good illustration of this strategy is Apple which offers its OS only to four relatively similar mobile devices (iPhone, iPhone3G, iPhone3Gs and iPod Touch). This approach is not new for Apple as it has already been adopted for its personal computers. This strategy enhances device standardization of the devices and lower dependance on various device manufacturers (e.g., Samsung, LG, Motorola). The platform provider has more control on the devices and lower compatibility issues of its OS.

3.3.2. Device variety

Platforms can chose to allow their OS to run on a variety of mobile phones manufactured by different vendors. Through this strategy, platforms can reach more potential customers. This approach also allows to be less dependent on individual manufacturers; the pressure is on manufacturers rather than on platform when it comes to prices and features. The main drawback is the customization for the specifications of different devices and different manufacturers. Java is a good example of a platform that was split in several subplatforms to accommodate different mobile devices.[6] It has lost its original purpose of compile once and run everywhere. Currently all platform providers, except Apple, follow this approach.

3.3.3. Towards device variety

When looking at the set of targeted devices, commercial platforms were traditionally targeting a variety of devices. Apple and Google both began by targeting uniform devices. However, Google shifted its approach to target a plethora of different devices and manufacturers, leaving Apple alone in the uniform category as depicted in Fig. 5.

3.4. Platform integration

Some platforms focus on their core business, which is to provide an OS with programming support for developers, whereas others integrate some or all elements of the distribution process. Hereafter, we classify platforms according to their level of integration similarly to Gereffi et al. (2005), but instead of taking into account the whole value chain, we focus on the elements of the distribution process. We identified four different types of integrations, namely *full integration*, *portal integration* as well as *device integration* and *no integration* (see Fig. 6).

3.4.1. Full integration

Platforms with a full integration have a strict control over every step of the distribution process from device manufacturing to application publishing, as depicted in Fig. 6.①.

[6] First, in order to accommodate small devices, a subset of the Java Standard Edition (Java SE) was developed. This subset is called Java Micro Edition (Java ME). Java ME has been further divided into two sub-configurations, namely Connected Device Configuration (CDC) and Connected Limited Device Configuration (CLDC). Further, there are special profiles for these configurations, such as the Mobile Information Device Profiles (MIDP 1 and 2). Furthermore, others have implemented there own non-compatible version of Java ME such as Docomo's DoJa.

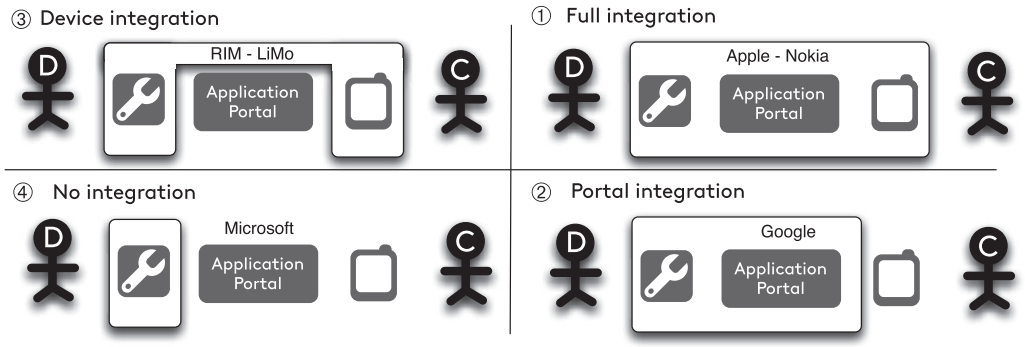

Fig. 6. Platform integration.

A fully integrated platform can take advantage of the two-sided market in the following way. Since the same company controls all elements of the distribution process it can decide to reduce the revenue from one of them in order to attract more customers, or more developers, which will create a positive feedback loop and increase the company's overall revenue. Reducing the price of one element, such as the mobile device, will attract more customers which in turn attract more developers which will make the platform more attractive for customers and increase the number of application sales, and so forth. The revenue generated by the application sales and the higher number of devices sold compensates the reduced revenue for the individual device sales. Such an intervention is more difficult when several players are involved in the distribution process.

Apple and Nokia exhibit a strong integration. Apple produces the device on which its OS runs, namely its iPhone, and it owns the unique authorized portal for mobile applications, namely the AppStore. Furthermore, Apple also plays the role of content provider with the iTunes store available on the iPhone. Similarly, Nokia manufactures its phones and provides an application portal as well as other content via its OVI[7] portal.

3.4.2. Portal integration

Platforms with portal integration focus on application development and application sale by integrating a portal, as depicted in Fig. 6.②. Google provides such an integration with its Android Market. Contrary to Apple, Google does not manufacture mobile phones on which its OS runs.

3.4.3. Device integration

In the device integration model, platforms also manufacture devices but are not in the application portal business, as shown in Fig. 6.③. RIM and LiMo are such platforms. RIM manufactures its Blackberry mobile devices but does not provide a portal. The LiMo foundation can also be considered to follow such a model since it is composed of handset manufacturer such as Motorola, NEC, Panasonic and Samsung.

3.4.4. No integration

Platforms with no integration focus only on their core business as depicted in Fig. 6.④. For example, Microsoft does not manufacture mobile devices, nor provide an application portal.

3.4.5. Towards full integration

Another trend is the emergence of more integrated platforms, as shown in Fig. 7. Some scholars have also hinted that an intermediary could play an integrating role in the mobile development industry (Barnes, 2002; Buellingen et al., 2004; Maitland et al., 2002; Tsalgatidou et al., 2001). A surprising observation is that mainly phone manufacturer companies and software development companies have played this integration role and not so much MNOs as was the intuition of most of these scholars. Before the introduction of Apple's platform, major mobile platforms were not *fully integrated*. Moreover, there was no platform with *portal integration* before the introduction of Google's platform. Symbian OS is an example of the trend towards integration since it started as a platform with no integration, before it was integrated by Nokia to become a *device integrated* platform and finally by launching OVI, it became *fully integrated*. RIM is also expected to soon become *fully integrated* with the introduction of its App World. Furthermore, with Microsoft moving towards *portal integration* there will be no major platform left without integration.

[7] More information about OVI can be found on: http://www.ovi.com.

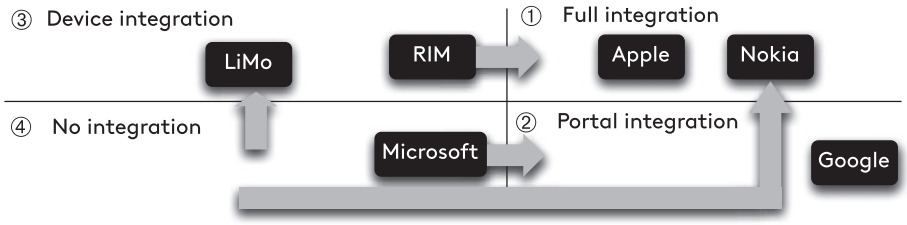

Fig. 7. Integration trends.

Table 1
Overview of the platform providers.

Platform name	Technology	Portal	Devices	Integration
Apple	Closed	Centralized	Uniformity	Full
Google	Open	Centralized	Variety	Portal
Linux	Open	Decentralized	Variety	Device
Nokia	Open	Decentralized	Variety	Full
RIM	Closed	Decentralized	Variety	Full
Windows	Closed	Decentralized	Variety	Portal

3.5. Overview of the platform providers

Table 1 shows an overview of the different platforms with respect to the dimensions presented previously. Interestingly, none of the platform providers have the same pattern. However, according to our previous analysis we can conclude that platform will tend to be more integrated in the future. Their marketplaces will become more centralized and they will mostly target heterogeneous devices. Some will offer an open-source OS and others will offer closed technology. So far no strong conclusions can be made about possible further trends in this dimension.

4. Implications for developers

The different market trends, presented in the previous section, can be seen as not only opportunities but also threats for developers. Therefore, it is crucial that developers have a good understanding of the possible implications of each trend. Hereafter, we analyze these trends and propose eight possible implications for developers. These *propositions* could represent starting points for further research.

4.1. Implications of portal centralization

Portal centralization is an major shift for developers. The fact that platforms providing centralized portals count on application sales to increase their revenue encourages them to promote application downloads heavily. This promotion is done through advertising but more importantly through greatly enhanced user interfaces. Before the emergence of centralized portals it took an expert user to download and install third-party applications, usually involving an internet search and a credit card payment, on a personal computer and then a file transfer via Bluetooth. Now it has become a "one-click" operation directly executable on the mobile device. This has made it much easier for the everyday user to download applications and therefore has led to an increased pool of potential consumers. Other than through the increase in consumers, portal centralization impacts developers in the following ways:

Proposition 1. *Portal* centralization makes access to customers easier.

The fact that only one portal offers applications implies that customers shopping on a single portal have access to *all* applications developed for the platform. Thus, developers have an easier access to the whole consumer base. Conversely, in a decentralized portal model, an application on one portal can only be downloaded by the subset of users familiar with this particular portal. Further research should investigate whether this enhanced access to the consumer base is translated by higher sales, or whether it has the drawback of drowning applications in the sea of competing applications.

Proposition 2. *Portal* centralization lowers distribution costs.

Distribution costs include the cost of uploading applications on the portal, with the attached registration fees, the maintenance cost of updating applications on the portal, and the billing cost. Having to publish an application only on one portal allows to decrease their distribution and maintenance costs, since uploads and updates must only be made on one single portal. Furthermore, a centralized portal also takes care of the administrative tasks, such as billing and advertising. However,

this service comes at a price. Centralized portal typically take a 30% commission on every application sale. Developers must evaluate whether the reduced distribution cost covers the portal fees. Personal advertising might still be needed to promote an application in order to stand out of the competitors.

Proposition 3. *Portal* centralization limits the freedom of developers.

A negative side of portal centralization is that developers might have to conform to rules defined by portal providers. This hurdle has raised with Apple's AppStore, which rules over which applications will be sold or not. To overcome these restrictions, the developer community can rely on alternative portals (Installer, Cydia) where developers can publish their applications. Unfortunately, only tech-savvy customers shop on such "black markets", since phones must undergo a "jailbreak" procedure. This drawback does not occur with Google's open publishing approach, which counts on consumers' feedback and ratings to discourage and ban low-quality applications.

4.2. Implications of technological openness

Moving towards open technology aims at reducing the development cost of OS. This possibly reduces its retail price, thus making it more attractive for customers (Riehle, 2007). Developers can indirectly benefit from the increase in potential customers. More directly, developers can benefit from the two following opportunities presented in the next propositions.

Proposition 4. *Open* technology lowers the development costs of application.

Open-source platforms can encourage code sharing within the open-source developer community. Thus, for many common routines, existing code and libraries can be reused. Decreasing the rewrite code has a major impact on development time and cost. Further investigation should quantify this impact and should find what incentives, such as different licensing schemes, can lead to greater resource sharing among the community.

Proposition 5. *Open* technology offers more job opportunities for developers.

An open-source project can provide career opportunities for developers willing to contribute to the development of the platform (Hertel et al., 2003; Lakhani and von Hippel, 2003; Riehle, 2007). In an open-source project, developers can be involved on different levels, as simple *users* when they develop applications, as *contributors* when they develop the platform itself or as *committers*. Committers are developers who have the authority to decide whether or not to integrate contributions into the platform. It should be assessed how many developers end up working for the platform provider for which they developed applications.

4.3. Implications of device variety

Targeting a variety of devices has some advantages for platform providers, previously exposed in Section 3.3. It can also provide interesting opportunities for developers. Similar to the previous trends, targeting a wider set of devices might increase the consumer pool and with it the potential application market. Furthermore, device variety has also two direct implications for developers.

Proposition 6. *Device* variety increases freedom for developers.

Developers interested in some particular feature will have a greater chance of finding at least one device supporting it in a large pool of devices. Conversely, when developing for platforms that target only a uniform set of devices, developers might be limited by the smaller amount of features supported. For example, one can recall the lack of Bluetooth connection in the early versions of the iPhone, while it was a standard feature in many other devices on competing platforms.

Proposition 7. *Device* variety increases customization cost.

A negative aspect of device variety is that it might raise development costs through an increased need for customization. Developers who desire to target all devices of a given platform have to remove features which raise incompatibility issues. Furthermore, they will have to fine-tune their applications for almost every device. Features such as screen size and input type (touchscreen or keyboard) are typical examples of aspects for which an application must be customized. The customization cost should be compared to the benefit of having a greater potential market.

4.4. Implications of platform integration

Fully integrated platforms offer an ecosystem where every element works together seamlessly. As mentioned previously, in such a system platform providers can leverage on a positive feedback loop of the two-sided application market. This can be achieved by reducing the profit margin on one side of the market. For example, reducing the sale price of mobile devices can potentially increase the consumer base. As a result, it will attract more developers, who benefit from such an environment in the following way.

Trends	Implications
Portal Centralization	P1: easier access to consumer P2: lower distribution costs P3: less freedom
Technology Openness	P4: lower development costs P5: more job opportunities
Device Variety	P6: more technical features P7: higher customization cost
Platform Integration	P8: easier development process

Fig. 8. Summary of the trends and implications.

Proposition 8. *Platform* integration facilitates the flow through the distribution process.

The emergence of a fully integrated end-to-end distribution process seems to create an ecosystem with reduced compatibility issues. This provides a smoother flow for the mobile application from its development to its installation and usage. Such ecosystems help developers to reach their customers using a rather frictionless solution. It also lowers market entry barriers for new developers by paving the way to the market.

To conclude this section, we provide an overview of the trends with their implications on developers (Fig. 8). We hope that these propositions might lead to some new research efforts in the mobile application development area.

5. Conclusion

In this paper, we described the implications that different market and technology trends have on the mobile application development market. The current evolutions show that the game has changed dramatically for developers. There are many new opportunities for them to develop, distribute, and generate significant revenues with the emerging mobile application portals.

Since the mobile application development landscape has substantially changed over the past several years, mobile development platforms have become more integrated and generally play the role of application portal, device manufacturer, or even both. As discussed in the paper, application portals tend to become more centralized, facilitating the link between developers and consumers. Moreover, several new platforms entered the open-source community to lower their development costs and possibly extend their consumer market by lowering prices and as a consequence increase their developer pool. In this changing environment, choosing for which platform to develop reveals to be challenging.

This assessment conveys that there will probably be an aggressive competition between platforms in order to get developers on board, which will hopefully lead to both further opportunities for developers and increased user-experience for consumers. Despite all benefits of these new platforms, a general issue can be seen in interoperability of the developed applications. Currently, developers have to choose their preferred platforms and develop a separate applications for each platform. Eventually, this will complicate the work of developers and increase the overall cost of mobile application developments.

We hope that the propositions formulated will offer some guidance for further academic research. We are conscious that this paper does not address all the identified issues in depth. An empirical study is needed in order to collect first-hand data and validate the implications. In addition, other similar developments in different industries could be relevant to put into perspective in order to compare the different developments. In conclusion, we strongly believe that this topic is timely relevant and need further investigations.

This paper also aimed at providing some contributions to the industry. We hope that practitioners can benefit from our analysis in order to better understand the current trends in the mobile application development market.

References

Adrian, B., 2002. Overview of the mobile payments market 2002–2007. Gartner.
Anderson, C., 2006. The Long Tail. Random House Business.

Ballon, Pieter, Walravens, Nils, Spedalieri, Antonietta, Venezia, Claudio, 2008. The reconfiguration of mobile service provision: towards platform business models. In: Proceedings of the ICIN'08.
Barnes, Stuart J., 2002. The mobile commerce value chain: analysis and future developments. International Journal of Information Management 22 (2), 91–108.
Buellingen, Franz, Woerter, Martin, 2004. Development perspectives, firm strategies and applications in mobile commerce. Journal of Business Research 57 (12), 1402–1408. Mobility and Markets: Emerging Outlines of M-Commerce.
Capiluppi, Andrea, Michlmayr, Martin, 2007. From the cathedral to the bazaar: an empirical study of the lifecycle of volunteer community projects. Open Source Development, Adoption and Innovation, 31–42.
Cozza, Roberta, 2009. Forecast: smartphones by operating system and end user segment, worldwide, 2007–2013. Gartner 23 September.
de Reuver, Mark., Haaker, Timber., 2009. Designing viable business models for context-aware mobile services. Telematics and Informatics 26 (3), 240–248.
Elberse, Anita, 2008. Should you invest in the long tail? Harvard Business Review.
Feijóo, C., Maghiros, I., Abadie, F., Gómez-Barroso, J.L., 2008. Exploring a heterogeneous and fragmented digital ecosystem: mobile content. Telematics and Informatics.
Forman, George H., Zahorjan, John, 1994. The challenges of mobile computing. Computer 27 (4), 38–47.
Funk, Jeffrey L., 2009. The emerging value network in the mobile phone industry: the case of japan and its implications for the rest of the world. Telecommunications Policy 33 (1–2), 4–18.
Gereffi, Gary, Humphrey, John, Sturgeon, Timothy, 2005. The governance of global value chains. Review of International Political Economy 12 (1), 78–104.
Hertel, Guido, Niedner, Sven, Herrmann, Stefanie, 2003. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. Research Policy 32 (7), 1159–1177.
Lakhani, Karim R., von Hippel, Eric, 2003. How open source software works: 'free' user-to-user assistance. Research Policy 32 (6), 923–943.
Maitland, Carleen F., Bauer, Johannes M., Westerveld, Rudi, 2002. The European market for mobile data: evolving value chains and industry structures. Telecommunications Policy 26 (9–10), 485–504.
Parker, Geoffrey G., 2005. Two-sided network effects: a theory of information product design. Management Science 51 (10), 1494–1504.
Raymond, Eric., 1999. The cathedral and the bazaar. Knowledge, Technology, and Policy 12, 23–49.
Riehle, Dirk, 2007. The economic motivation of open source software: stakeholder perspectives. IEEE Computer 40 (4), 25–32 (article 25).
Rochet, Jean-Charles, Tirole, Jean, 2003. Platform competition in two-sided markets. Journal of the European Economic Association 1 (4), 990–1029.
Shapiro, Carl, Varian, Hal R., 1998. Information Rules: A Strategic Guide to the Network Economy. Harvard Business School Press.
Tsalgatidou, Aphrodite, Pitoura, Evaggelia, 2001. Business models and transactions in mobile electronic commerce: requirements and properties. Computer Networks 37 (2), 221–236. Electronic Business Systems.