The Mobile Common:

A guide to mobile open source and its effects on mobile device manufacturers

By

Ofri Markus

SUBMITTED TO MIT SLOAN SCHOOL OF MANAGEMENT AND SCHOOL OF ENGINEERING
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR SYSTEM DESIGN AND MANAGEMENT PROGRAM FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT

AT THE

MASSACHUSETTS INSTITUE OF TECHNOLOGY

APRIL 2008

[June 2009]

Signature of Author: _____

OFRI MARKUS
SDM FELLOW
MIT SLOAN SCHOOL OF MANAGEMENT AND SCHOOL OF ENGINEERING

Certified by: _____

MICHAEL A. M. DAVIES, THESIS SUPERVISOR
SENIOR LECTURER, MIT SLOAN SCHOOL OF MANAGEMENT

Accepted by: _____

PAT HALE
DIRECTOR, SYSTEM DESIGN AND MANAGEMENT PROGRAM

This page has been intentionally left blank

Abstract

At the beginning of 2009, the global mobile industry is in the midst of a revolution, characterized by a trend towards openness, driven by technological advances, evolving consumer demands and increasing competition.

Open mobile operating systems, which allowed third party software development, have existed since 2001, but they haven't had much effect on the industry's business ecosystem or the customers' expectations and demands. The release of the iPhone in 2007 had profound impact not only on dominant design of smartphones and on consumers' expectations, but also on the balance of power in the mobile industry amongst network operators, mobile device manufacturers and software developers.

The trend continued with the subsequent announcements of leading industry players that they will develop open source operating systems for mobile phones, and make them accessible to everyone. Today, top mobile device manufacturers are on the verge of releasing dozens of new devices running open source operating systems, which has the potential to completely change the current market share of the players in the market.

The purpose of this thesis is to guide product managers in the mobile industry. It examines the historical and current structure of the mobile business ecosystem, reviews latest developments in open operating systems and open source platforms and their affects on industry players today, and identifies potential influences and developments for the future based on analysts' reports assessment, industry experts' viewpoints study, and interviews with industry executives.

Thesis Supervisor: Michael A. M. Davies

Senior Lecturer, MIT Sloan School of Management

This page has been intentionally left blank

Acknowledgements

I would like to thank my thesis instructor, Michael Davies, for introducing me to the fascinating world of the mobile industry, for helping me focus my research, and for providing ideas and insights that enhanced my work.

I'm grateful to Pat Hale and the SDM office, for giving me the opportunity to be a part of MIT and the System Design and Management program, and for their support, kindness and help during the last two years.

I would like to thank my SDM class fellows, which I learned a lot from, and enjoyed spending time with, inside and outside of the classrooms. Special thanks to my friends Daniel and Shimon, for providing insights, challenging my thoughts, and connecting me to great people.

Lastly, I would like to thank my dear family - my parents, my wife Liron, and my wonderful daughters Tamara and Daniella, for being there for me and reminding me of the things that matter most.

This page has been intentionally left blank

Contents

This page has been intentionally left blank

List of Figures

This page has been intentionally left blank

Capter 1 - Mobile industry background and analysis

1.1 Technology description and trajectory

1.1.1.1 The beginning

The first public telephone call placed from a portable device was made on April 3rd 1973 by Martin Cooper, then general manager of Motorola's Communications Systems Division, who is considered today to be the "father of mobile phone". It took six years for NTT (now DOCOMO) to launch the world's first commercial cellular network in Japan, in 1979. In the following years, cellular networks were also launched in Denmark, Finland, Norway and Sweden using the NMT-450 technology. Those networks are nowadays referred to as 1st generation networks, which supported devices about the size of a briefcase.

The underlying technological principle of all cellular networks is identical. Any given geographical area is divided into sections (or "cells"). Each cell is serviced by a radio antenna (traditionally installed on a cellular tower) that can receive and transmit radio signals to mobile devices within that cell. All cellular towers of a specific cellular network are interconnected and "know" which devices are served by each cell. When an inbound call reaches the network, the network first establishes the target device's location. The cellular tower managing the cell the device is in signals the device that it has an incoming call and directs it to a specific frequency onto which the device must switch to connect the call. When the device is on the move between cells, the tower signals it to switch to an adjacent tower signals in order to ensure continuous connectivity.

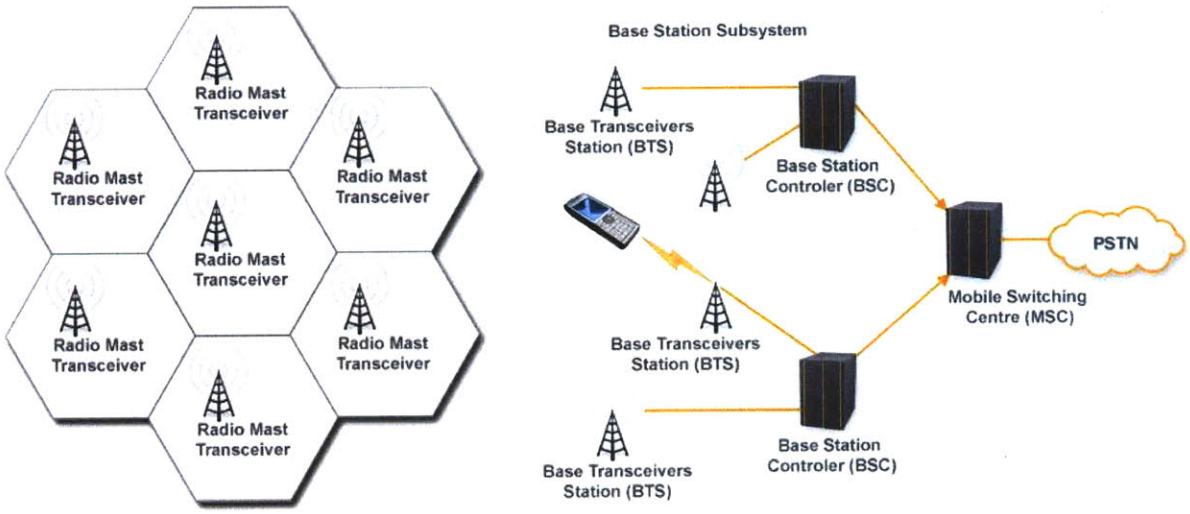

Figure 1- How a cellular network works. Source: http://www.allaboutmobile.org.uk

Cellular technology symbolized a sea change in the concept of telecommunication. Unlike landlines, people no longer communicated with a particular location (e.g., a house or an office), but rather had direct access to another individual. The first mobile devices, bulky as they were and limited to voice communication only, were the first truly personalized electronic communication devices. This seemingly minute observation, which may not seem important in today's world of personalized Web content, multiple email accounts, Instant Messaging and social networking, was the key driver behind the mobile industry's evolution in terms of technology, products and services.

At first, consumers were hardly lured by the promise of personal communications. The penetration rates of 1st generation mobile phones was very slow. In the year 1985, there were still less than one million mobile subscribers in the world.

1.1.1.2 2G

The next generation of cellular networks was characterized by a shift to digital technology. The main advantage of digital technology was improved resource utilization: a given allocation of radio frequencies (commonly referred to as "bandwidth") could now support 3-10x more calls compared to a single analog cell thanks to multiplexing. It also enabled over time the cost of mobile phones to be reduced enormously, as these devices could exploit Moore's Law, which has no equivalent for analog electronics. Digital technology significantly enhanced the potential economic value of cellular networks, allowing mobile network operators to optimize the usage of their allocated frequencies and significantly expand their subscriber base. Some of the cost savings associated with the increase in network capacity were passed on to the end-user, making the technology more cost effective and more accessible to a larger percentage of the population. The following figure shows the continuing increase in the number of mobile subscribers worldwide, including predictions through 2010.

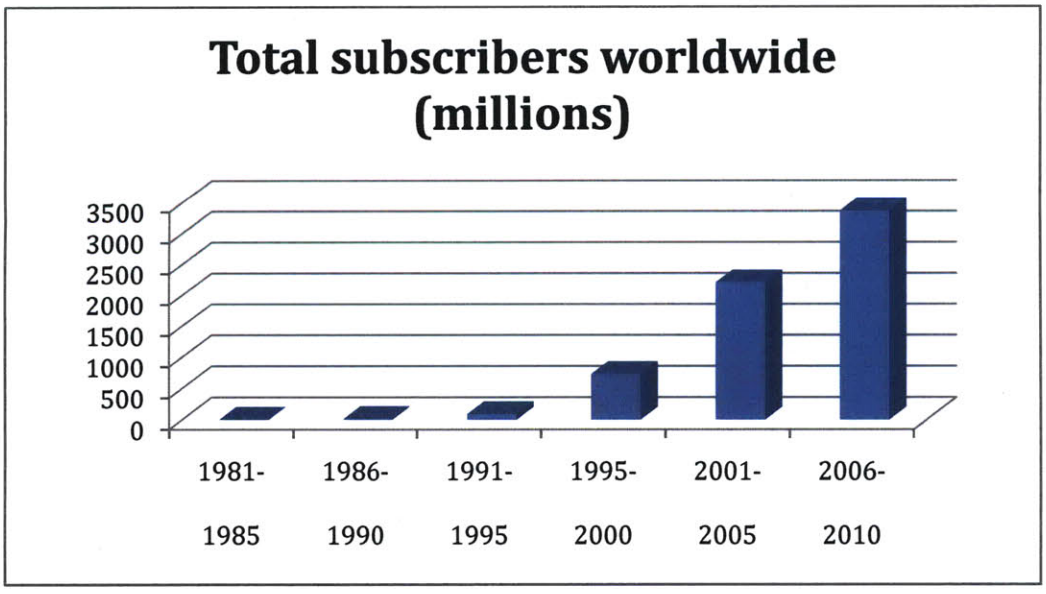

Figure 2 - Mobile subscribers worldwide. Source: eMarketer

The most tangible effect of 2G networks was probably their immediate and perceptible value-add to end-users, providing the first behavior-altering benefits of the "digitalization of communication". The move to digital devices enabled better displays, and devices with easy-to-use electronic address books for their contacts in the phone. In the past, people relied on maintaining physical address books or memorizing phone numbers and subsequently manually dialing them to place a call. The digital mobile phone eliminated many of the inconveniences associated with the human-machine interaction by maintaining all our personal contacts on a single device, enabling users to communicate at the click of a button. This important change, accompanied by ubiquitous access to caller ID and Short Message Service (SMS, which allowed the exchange of short text-only messages between cellular users) amounted to a revolution because it increased users' attachment to their communication devices, personalized telecommunication, and ultimately increased our dependence on mobile devices.

1.1.1.3 2.5G-3G

The next generation of cellular networks, often dubbed "2.5G", was characterized by the introduction of packet data connectivity, such as General Packet Radio Service (GPRS) for GSM-based networks, that enabled additional services beyond voice calls, such as Multimedia Message Service (an extension to SMS that allowed the sharing of pictures and short video clips), and, in particular, limited access to Internet content. 2.5G networks highlighted users' interest in expanding the functionality of mobile devices beyond voice – but also exposed the inherent limitations of existing cellular standards. The transition to 3G embodied a shift in the popular perception of what was known up till then as a "mobile phone". The introduction of data-aware cellular standards increased the availability of Internet content on mobile devices, allowing them to support a plethora of services, from text messaging, to billing systems, email

access, downloading and consumption of multimedia content such as music and videos, taking and sharing photographs, location based services, Internet browsing, and ultimately – using the mobile device as a pocketable computation device capable of running a wide variety of software applications that were previously limited to the realm of personal computers.

Today, as the global market continues to grow, mobile devices fall into three main categories:

1. ***Smartphones*** - Sophisticated high end devices that include all-in-one capabilities: 8.4% of the market in 2007, growing rapidly
2. ***Feature phones*** which includes additional features like built-in camera, Bluetooth communication device, and so on
3. ***Basic phones*** with voice and text only

1.1.1.4 Future technologies

The next (or 4th) generation of mobile networks is expected to be implemented primarily using a set of network technologies called LTE (Long Term Evolution), Given the growing customer expectations for richer mobile service, with data traffic demands that could overwhelm even the capabilities of today's fixed broadband connections. The 3G Partnership Project (3GPP), a group of international standardization organizations and mobile technology companies, began discussions on setting the requirements and standards for the technology in 2004. The key characteristics that were defined for the LTE networks are:

- Higher bandwidth of data, targeting 100Mbps for downlink, and 50Mbps for uplink
- Increased capacity, a 3x gain over 3G networks
- IP based only network, without separate voice circuits

- Simplified, flat network architecture, which allows reduced installation and maintenance costs and higher efficiencies compared to 3G

Today, even though the standard is not fully set yet, hardware and chip manufacturer began shipping LTE equipment in early 2008. Many mobile device manufacturers have by now demonstrated LTE-capable devices and data rates of 100Mbps. Most mobile carriers in the world are expected to upgrade their networks to LTE in the next few years.

1.1.1.5 Competition to cellular technology

There are two other main competing Radio Frequency (RF) technologies that are used for mobile communication today:

- Wi-Fi is a short range RF technology, designed to be mainly used in buildings, or populated urban defined areas. Its main advantage is its data rates, averaging up to 54Mbps in its 802.11n form. Its main disadvantage is its range, averaging from around 32 meters inside buildings, and up to 100 meters in open spaces. Wi-Fi is mainly used as the technology for home or office Local Area Network, with the central base station of the network, a device called Wireless Access Point, usually privately owned by the end customer. Many public Access Points can be also found in many public places around the world, and are usually maintained by Wi-Fi service providers. Although Wi-Fi networks overcome cellular networks with regard to features, it is usually not considered a competing technology, but a complementary one. Wi-Fi is strong in closed spaces where cellular signal is weak, and cellular is much more useful for wide area communications, anywhere outside where it's hard to implement good Wi-Fi coverage.

- WiMAX is a medium range RF technology, designed to complement the capabilities of Wi-Fi networks. Wi-Fi access points are usually connected by cables or fixed lines to Wide Area Networks (WAN), and operators are having problems to be cost efficient when connecting remote rural areas. The WiMAX technology can help covering the "last mile" and be complemented with in-house technologies like Wi-Fi. Since WiMAX is a data network, it is a possible competitor to the future cellular LTE networks. In the US, a consortium of companies called Clearwire, consists of Sprint, Intel, Google, Comcast, Time Warner Cable and Bright House networks, is aiming to deploy the first, nationwide, US WiMAX network, competing directly with the other cellular operators over providing 4G services. The company already lunched WiMAX networks in several US cities, including Baltimore, Maryland, and Portland, Oregon, and planning to expend its network. WiMAX is also being deployed in some developing countries.

1.2 Business ecosystem – players, leaders, business models, standards

1.2.1 The value chain

The mobile industry consists of the following key groups of players:

1. Mobile operators (carriers) –the owners of the cellular networks. National regulatory bodies usually award operating licenses to a handful of mobile operators, creating de-facto oligopolies. Since the carriers control the network, they also have the power in many cases to determine what types of devices will work on their network and what types of services will be available to their customers, either through technical control or because of their paying subsidies and controlling the distribution channel. The operators

compete with each other on quality (often measured by calls dropped and geographical availability of high-speed data services, as well as customer support), price, and features (i.e., types and features of devices, accessible data and information services). The largest mobile operator in the world is China Mobile, operating in China, which has more than 443 million subscribers (as of January 2009).

For years, mobile operator wielded tremendous market power and owned an inordinately large piece of the mobile industry pie. Stringent regulatory environments and high initial capital costs of infrastructure erected high barriers to entry for new players, leaving a handful of carriers in each market to compete for the business of several millions of customers. The figure below shows the world's top mobile operators by their number of subscribers.

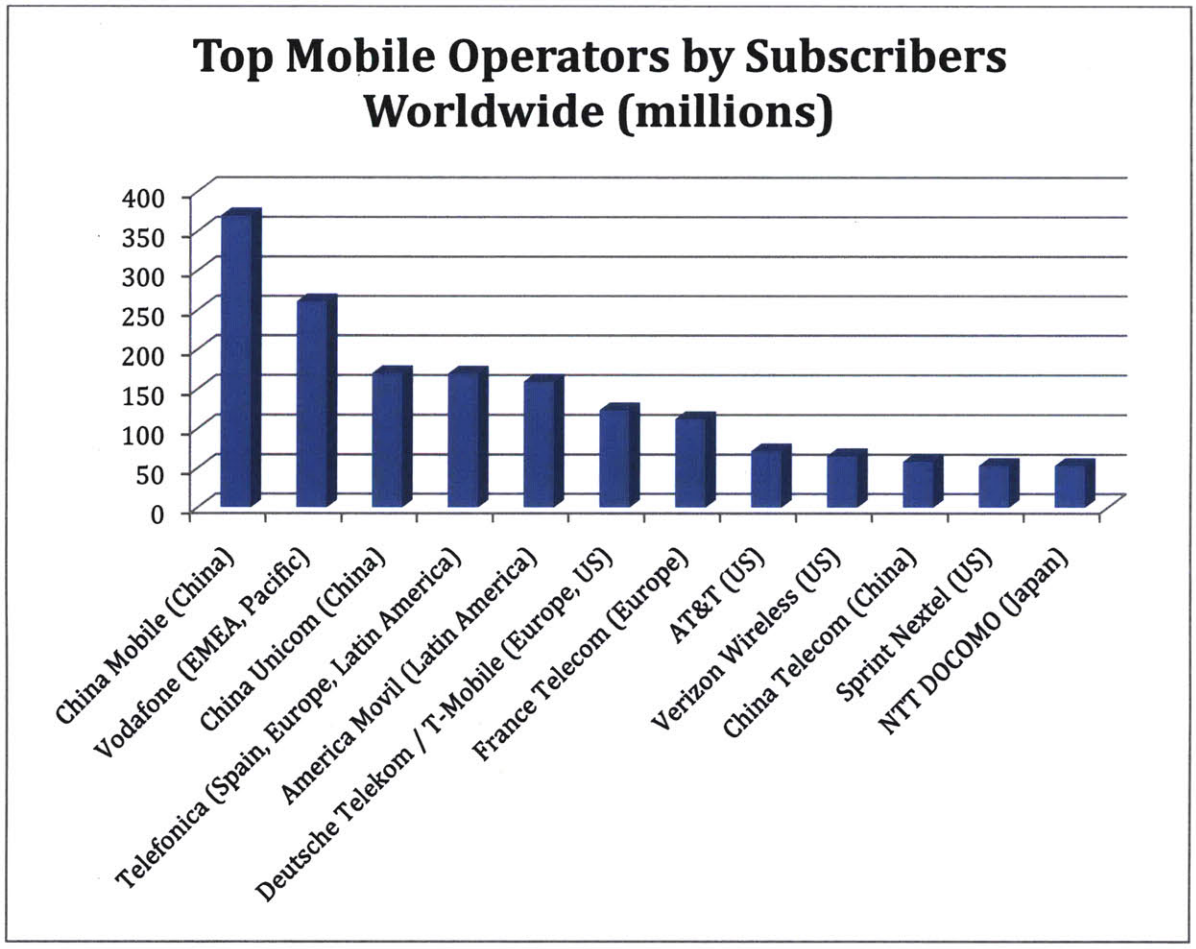

Figure 3- Top mobile operators by subscribers worldwide. Source: Plunkett Research

2. Manufacturers of Mobile Devices –the companies that make mobile phones and devices. They have the financial ability and the technical expertise to combine software and hardware components, including operating system, radio frequency transmitters, and processor shells to create highly integrated devices that provide users with the desired mobile phone features. Mobile devices manufacturers compete with each other on the quality of the devices that they produce, their feature set, their durability, and the user friendliness of their user interface. The biggest challenge facing mobile device manufacturers is that in most cases, they do not sell their devices directly to the end users.

Instead, they have to interact with carriers who control the access that devices have to their network. It is not uncommon for carriers to sell mobile devices directly to their customers, oftentimes offering subsidies in exchange for long-term commitments to certain service packages. This common business practice established a balance of power that highly favored the operators, often at the expense of end-users who did not necessarily have access to the best devices but rather to those made by manufacturers who have established good relationships with the operators. A classical example is Nokia, the world's largest device manufacturer, who has a measly market share in the United States.

The world's largest device manufacturers by volume have been five global players: Nokia, Samsung; Sony Ericsson; LG; and Motorola. Apple, RIM and HTC have also become significant players at the high end of the market; although their share of device volumes is lower, their share of value is much higher because of the high average selling price (ASP) of their devices. Indeed, in a recent quarter following the launch of the iPhone 3G, Apple achieved the #3 slot for market share by ***value.***

The figure below shows the device manufacturers worldwide market share based on the number of phone shipments.

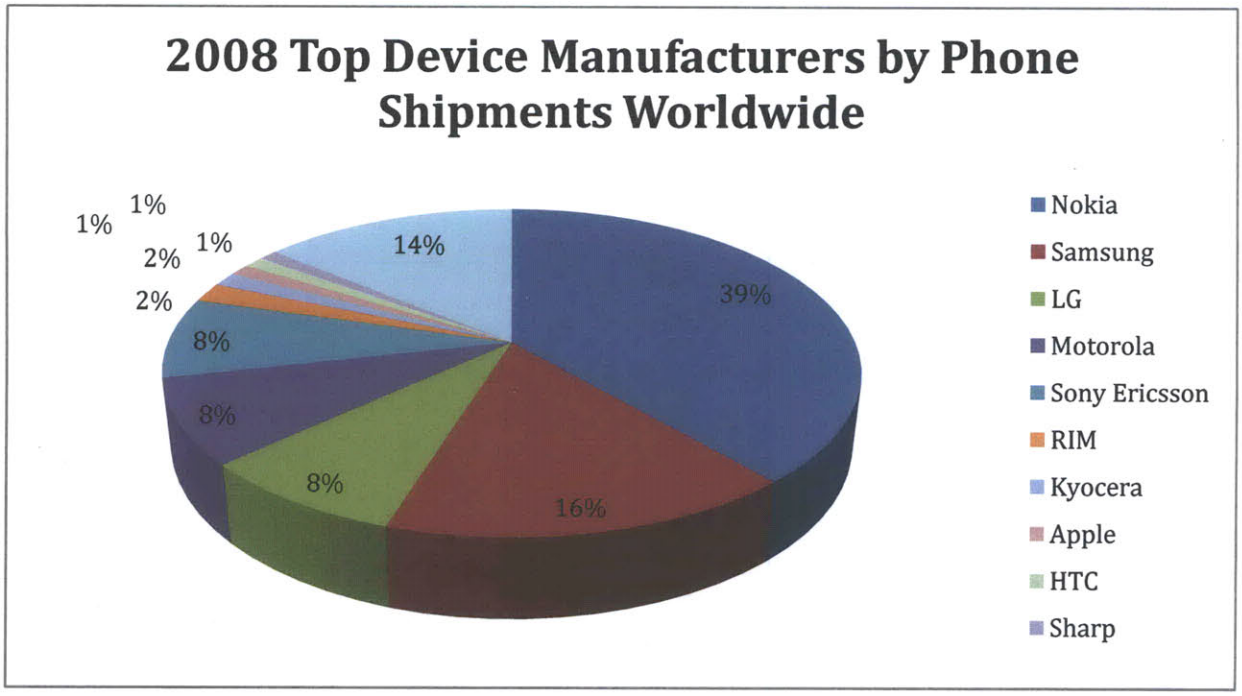

Figure 4 - Top device manufacturers by phone shipments worldwide. Source: eMarketer

3. Semiconductor Companies – These are the companies that make the core hardware components of mobile phones, among the most important ones are the Microprocessors, Digital Signal Processors. Those companies are also referred to as "The chip manufacturers", or the "Microprocessor companies". Semiconductor companies compete with each other on the base of cost and quality. Price is particularly important for general purpose chips, and firms with lower cost structure are in better position to compete. Quality is also a major factor as many products that use semiconductors rely on high level of performance, reliability, low power consumption, etc. Major semiconductor companies, developing integrated circuits for mobile devices are: Intel, Texas Instruments, Freescale, Qualcomm, AMD, Broadcom, NSC, and Cree.

4. Core software components suppliers – These are the companies that provide the software platforms that operate mobile devices. The platforms in the market consist of *proprietary platforms* that are made by device manufacturers to be used only on their devices, *generic platforms* that are made by device manufacturers but are sold to other manufacturers (sometimes competitors) to be used on their devices, and *third party platforms* that are often developed not by device manufacturers but by software companies such as Microsoft (Windows Mobile) and Google (Android). Software platforms compete on various dimensions – ease of use, applicability to a broad range of hardware devices, and more recently – "customizability", namely the ability to configure and change the properties of the platform. In the past, mobile software platforms were closed environments where users could only benefit from the pre-installed features and applications. Competition was restricted to the cleanliness and simplicity of the user interface and the availability of simple but popular applications (such as the mobile phone game "Snake", which generated strong customer loyalty to Nokia devices). Subsequently, due to pressure from mobile operators, software platforms increasingly introduced some degree of openness, such as the ability to run third party Java applications. These very limited API's allowed the creation of simple programs that carriers could pre-load into the devices. Ultimately, this led to the emergence of open operating systems, allowing device manufacturers to introduce substantial changes and manipulations, programmers to write software to, and users to install third party software applications directly on the device. The next generation of software platforms clearly involves a wholesale opening of the platform's software code.

5. Application developers – In the early days of mobile phones, when software platforms were closed, application developers did not exist since they had little economical incentives to participate in an industry that allowed them limited access to mobile devices (thereby limiting their financial opportunities). In addition, third parties seldom had access to APIs that enabled writing software for those devices. With the introduction of Java APIs, more developers could create mini-programs for mobile devices, but these were limited by two main factors – the APIs were very limited (i.e., disconnected from the operating system and could therefore not access core resources of the devices such as contact lists or the voice system). Secondly, there was a limited market for such applications, mainly because customers could not directly install applications on their devices; they would have to be pre-installed by the carriers, with which application developers had to develop relationships. With the introduction of open software platforms, the application developers segment has been growing exponentially. The introduction of Apple's iPhone second OS in 2008 marked a revolution for mobile application developers. The introduction of the App Store allowed application developers to use a very advanced API and create aesthetically appealing, useful and highly integrated applications that can be easily sold through the App Store and monetized through direct sales to millions of iPhone users.

1.2.2 Standards

There are two, leading, competing standards for cellular networks - Global System for Mobile Communication (GSM), and Code Division Multiple Access (CDMA). Each mobile operator uses a different standard for its network and compatible devices. In the US, for example, AT&T

and T-Mobile are GSM networks, and Verizon and Sprint are currently CDMA networks, although Verizon is now on the point of converging with the GSM path as it has announced its intention to go to LTE for its 4th generation network.

- GSM - The standard was created by the GSM association, with the first specification released in 1990. The motivation behind it is to create a standard that could be used across countries. GSM is implemented using a transmission method called Time Division Multiple Access (TDMA), in which the band is split time-wise into slots. GSM pioneered the Short Message Service (SMS) which allowed subscribers to send text messages to each other instead of calling. Today, GSM is the most popular standard in the world with estimates of over three billion subscribers (a market share of 80%). Due to its wide adoption, it's very easy for GSM users to use their phones in different countries.
- CDMA – A proprietary standard by Qualcomm, which uses a different method for data transmission. Here, the information is spread across the entire available bandwidth, which each call assigned a unique sequence code. CDMA currently has better coverage than GSM in some parts of the world (including the United States), though GSM is gradually catching up to speed in those areas also. There is no 4th generation standard for CDMA; Qualcomm had proposed Universal Mobile Broadband (UMB) but has recently abandoned this standard and thrown its weight behind LTE.

1.3 Evolution of customer demands

As in many other industries, competition drove technological evolution, which in turn increased customer expectations.

In the early days of the industry, customers' main concern was the voice feature of the phone. Devices provided a basic user interface that primarily facilitated the reception and placing of calls. Over time, more and more features were added to and standardized on mobile devices. The evolution of customer demand is very much in line with that of cellular networks. The first generation included basic voice and text features, the second generation introduced basic multimedia capabilities such as color displays, cameras, and access to a limited number of applications. The third generation represents full-blown data connectivity and access to online information using the mobile device.

1.4 Value Creation, Capture

1.4.1 Carriers

The basic value the carriers are creating is through the network - the infrastructure which enables mobile communication. On top of the network, each carrier has built its own set of services that are offered to customers. Those services often include music downloads, text, information and billing services, etc.

Carriers capture value by charging a wide variety of fees across the value chain:

- Fees from subscribers for basic voice or text usage of the network. Some carriers charge a monthly flat rate, other package a predetermined amount of voice minutes and text messages
- Revenue sharing with device manufacturers for the right to sell devices on their network
- Subscriber fees for data packages which enable Internet surfing and email usage on the mobile device

- Subscriber fees for additional services such as multimedia downloads, payments, etc.

1.4.2 Device Manufacturers

According to a product manager at Nokia, the main value added by device manufacturers is the understanding of customer needs, understanding the operating systems with which they have to work in, and the manufacturing of an integrated the device that will meet customer needs. An example for such complexity is that each mobile device product line contains hundreds of integrated components that need to work seamlessly and support dozens of countries, each with different sets of standards, frequencies and regulations, often supporting at least two carriers per country, which may in turn have different network configurations and standards. The ability to deliver all this while providing a unique, state of the art user experience is the essence of the value created by device manufacturers.

Traditionally, device manufacturers capture value by selling their devices to consumers through carriers. In order to do that, device manufacturers had to work hard in order to develop devices that complied with stringent requirements set forth by the carriers, and negotiate deals to co-promote and sell their devices. In some countries, recent deregulation decreased the market power of carriers with regard to selling devices. This allowed device manufacturers to sell their devices directly to the customers, regardless of the package the customer buys from the carrier. It should be noted that this is primarily applicable to second generation phones and not applicable to advanced third generation phones which allow access to data services.

This page has been intentionally left blank

Capter 2 - The rise of open source and open innovation

2.1 Definitions

When we speak about "openness" of mobile software, we should define some basic definitions to the levels of openness a system can be, from bottom-up:

1. Closed system – this is a system that is pre configured by the manufacturer. It is designed to run on a limited number of devices. The architecture of the design is typically integral rather than modular and there is a strong coupling and interdependence between the internal components. There is little or no option to change the look and feel of the device or the UI experience; it may be possible to change the 'wallpaper' or background screen, but only cosmetically. It is very hard to develop 3rd party applications for the device because APIs are rarely publicly available and any application that will be created will be able to run only on specific devices made by a specific manufacturer. There is no option for the end user to install additional applications on the device and only a small set of parameters is configurable.

2. Open operating system – This is a system that has built in capability to allow 3rd party development of applications. By its definition, it's more modular than closed systems, and has clear distinction between "core" components that cannot be "touched" or changed by developers, and "application" elements that allow developers to access resources like files, databases and hardware components in order to write their software.

3. Open source system – This is a system that its structure, components and code is completely open and visible to everyone depending on the licensing agreement. Due to the nature of open source software that has distributed development rather that single location or company, it's the most modular. Each component is carefully isolated from the others in order to ease the process of development and maintenance by sometime thousands of people that are often spread around the globe. With open source software, each person can usually take the software, view its code, change any part of it and develop 3rd party applications to it.

2.2 Background and rationale

As customer preferences grew higher and higher, the mobile phone began to be perceived as not just a device for communicating using voice and text, but a multimedia device, used for multiple usages. The category of those devices with open operating systems, also known as smartphones, began to grow and gain market share.

At this point, the industry's main players, such as carriers and device manufacturers, have realized what their colleagues at the PC industry realized years before– that the competition over the heart of the customer will be overpowered by platforms that will provide the biggest variety of features to its users, the ability to choose between multiple applications to handle different types of tasks. In order to achieve this variety, there was a massive need for innovation and understanding of the user needs. The problem was, that in this completely new territory, it was very hard to understand what the user needs are.

In his book "Democratizing Innovation", MIT Sloan professor Eric Von-Hippel discusses the trend of shifting from manufacturers' innovation to users' innovation. He claims that the development of the internet allowed formation of platforms that create new processes of innovative design. In these processes, companies have abandoned their efforts to understand the users' needs accurately and in detail. Instead, they outsource many innovative tasks to their users and equip them with appropriate toolkits.

The same trend affected the mobile industry. The manufactures were the platform developers, and the users were the application developers, who wanted to create useful and fun applications for their devices. A user innovation toolkit for a mobile platform is a programming API (Application-Programmer Interface), which allows developers to write software code that, can run on the platform, and interact with the platform resources, like calendar, contacts database, and hardware components.

In June 2001, Symbian released the first "open" operating system, which runs on the Nokia 9210 Communicator. In 2002, the first Smartphone running Windows Mobile debuted. Both platforms provided open APIs for software developers. Over time, attempts to further open the operating system and make it more appealing to software developers brought the first open source projects to the world.

At the beginning of 2009, open source is the hottest trend in the mobile industry, with platforms like Android having already been embraced by many of the industry's device manufacturers, leading industry leaders from all segments members in at least one open source consortium, and with mature platforms like Symbian have announce a transformation from their hitherto proprietary business model to open source.

2.3 The world's leading mobile platforms

The diagram below shows the world leading mobile operating system platforms and their market share at the end of 2008:

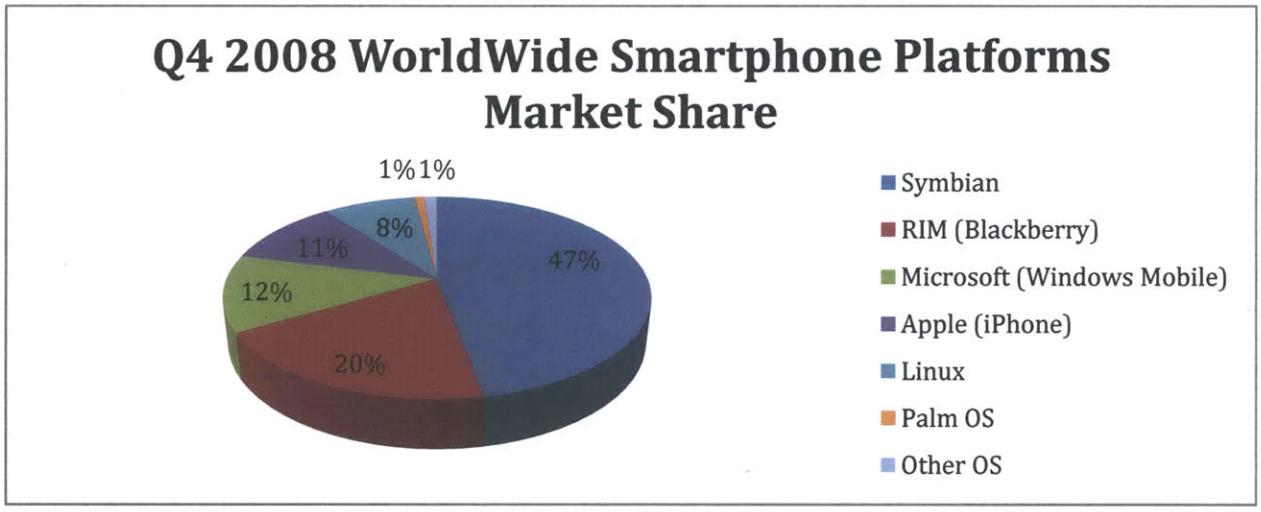

Figure 5 - Worldwide Smartphone platforms market share. Source: Gartner

2.3.1 Android

2.3.1.1 Background

The Android project was announced by Google at the end of 2007, along with the establishment of the Open Handset Alliance, a consortium of 48 hardware manufacturers, software developers and network operators. The purpose of the project is to develop an open source operating system for mobile device, based on the Linux kernel. Since then, Google released the Software Development Kit for developers and announced a developer competition in order to attract talented programmers to create applications for the new platform. In October 2008, the first device powered by Android operating system was release under the name of T-mobile G1. Close to the release of the G1, Google released the Android code to the public and it is now available as open source under the Apache License.

The incentive for Google behind Android is expending their business into the mobile arena. Google, as a company which its main source of revenues is from paid ads in their search engine, would like to ensure the sustainability of their business beyond the world of personal computers, extending this into the mobile domain which already has billions of devices. Unlike the PC market, which is dominated by a single operating system today, the future mobile operating system has not been set yet, and Google would like to ensure the availability of their services by providing device manufacturers a free, open platform to power their devices, and by that being able to prevent another player, such as Microsoft, establishing control of this bottleneck, and also to promote adoption and usage of its own integrated set of applications and services such as search, email, maps, documents, and so on.

2.3.1.2 Architecture

Android is designed from the perspective of the end users of the phone, and how to provide them with a rich experience of using multiple applications on their device. Because of that, Android features a modular architecture which distinguishes between the core operating system elements and the application elements, and yet maximizes the ability to share resources among running applications to the benefit of the application developers and the end users.

The Linux kernel and the libraries are the core elements of the system, and the access to them is provided by a layer called the 'application framework'. On top of that there is the layer of user applications, which use the application framework to access the system core resources. Although the Android platforms comes with default applications like a Web Browser, Phone and Contacts, third party developers can add applications, but using the same resources as the default applications. Indeed one of the key features that distinguish Android from other platforms is that it is possible to replace the default core applications.

Below is a diagram of the Android architecture.

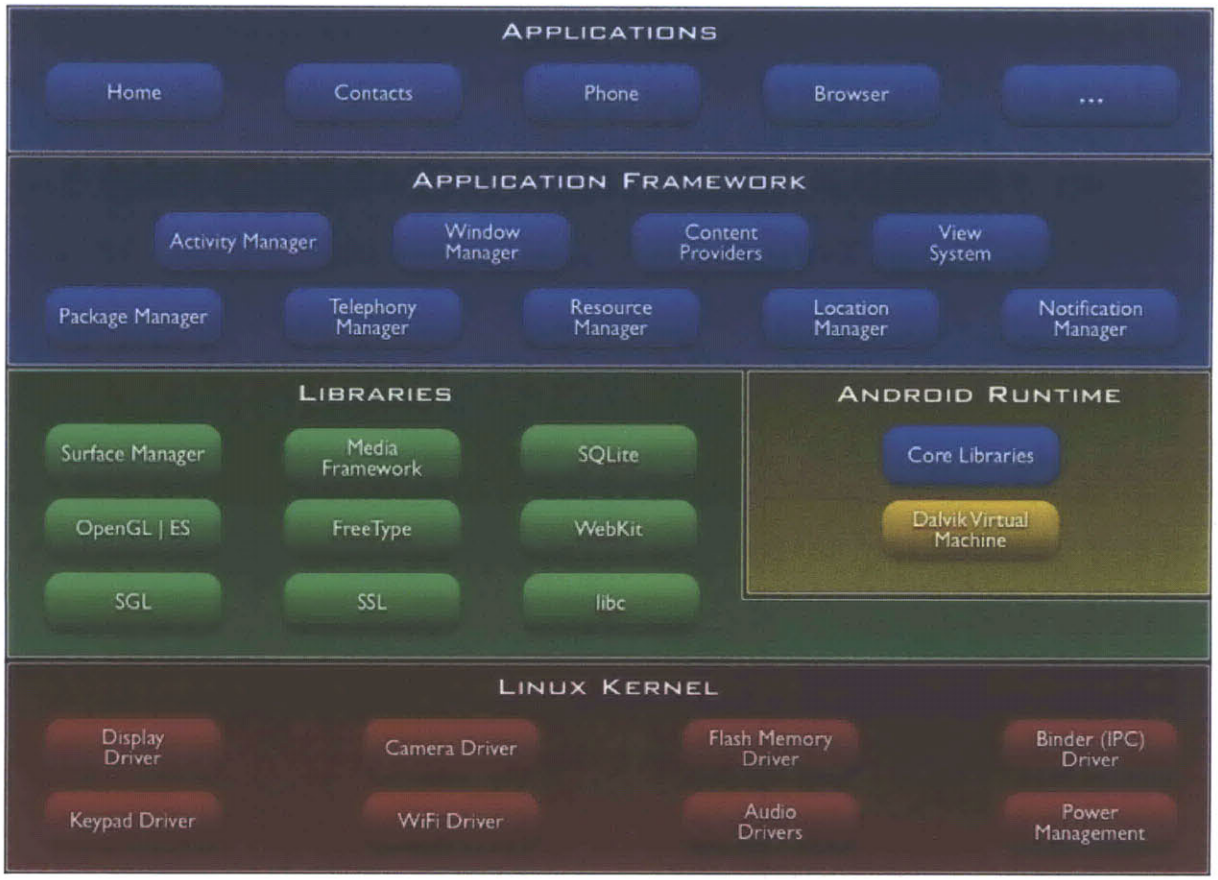

Figure 6 - Android Architecture. Source: Android Developer Website - http://developer.android.com

2.3.1.3 Licensing Model

Android embraced the Apache Software License (ASL), which is a permissive license and is conducive to commercial development and proprietary distribution. Code that is distributed under the ASL can be integrated into proprietary products and redistributed under different licenses. This license encourages commercial adoption of open source software because it makes it possible for companies to make profits by further development and enhancement of code that was first distributed as open source and gives the adopting company a chance to differentiate itself from its competitors. The decision to distribute the Android code under ASL was criticized

by many of the open source community because of the claim that it encourages defragmentation of the open source code and by that hurt the principal of "openness".

2.3.2 LIMO

2.3.2.1 Background

The LIMO foundation fits the definition of "co-opetition", in which companies find themselves both working together in cooperation to create value, and competing with one another to capture their share of the value that they have created; cooperation and competition at the same time, hence co-opetition.

The LIMO foundation was founded at the beginning of 2007 by leading device manufacturers like Motorola and Samsung, together with leading carriers like NTT DOCOMO and Vodafone. The mission of the consortium was to develop an open, Linux-based operating system for mobile handsets. The project was meant to develop the OS up to the level of middleware only, without getting into content or UI in order to avoid conflicts with carriers and to allow each of the device manufacturers to differentiate with a different UI. LIMO is built as an ecosystem in which industry players cooperate in order to build a competitive platform, through a process of shared leadership and decision making. The existence of key players from all levels of the value chain should provide a strong industry backup and insure committed engagement.

Unlike Android, which was developed by an external player to the industry in order to set foot in the industry and promote its business, LIMO was initiated by existing players, in order to improve the current development process of each one of the members, reduce costs, and provide a competitive product which fit the evolving customer expectations.

2.3.2.2 Architecture

Due to the incentives of the LIMO Foundation members, the LIMO platform is not designed first and foremost by thinking about the end customers (or, like the case of Android, how to get them to click on more Google ads), but on the middleware that should be able to run and can be easily adapted to run on a large variety of devices. On the other hand, it should allow multiple user interfaces to control it (and by that allowing the device manufacturers to differentiate), should allow different applications to run on it, according to the choice of the end customer, and should allow different content to be accessed through the device, based on the choice of the carrier.

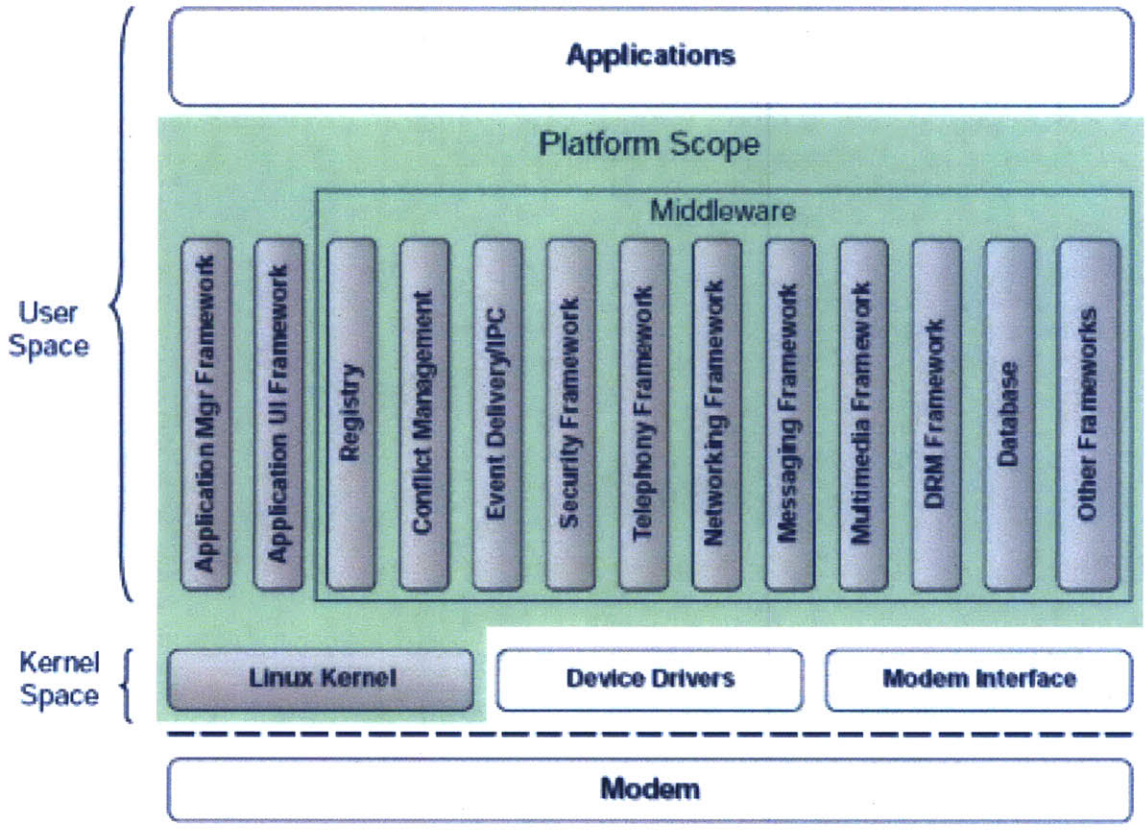

Figure 7 - LIMO Architecture. Source: linuxdevices.com

2.3.2.3 Licensing model

The licensing model that the LIMO Foundation chose to use with its platform is different from the one Google are using with Android. It is called General Public License Version 2 (GPLv2), and its key characteristic is that code that was released under this license cannot be used modified, defragmented or changed in whole or in part without contributing all the changes back to the community. That license makes it harder to allow companies to make commercial use of this code, and by that making it only relevant to develop shared components under this license. Companies that use this code are part of a community that share information and cooperate in order to benefit all.

2.3.3 Symbian

2.3.3.1 Background

Symbian was founded in 1998 by four device manufacturers including Nokia, Motorola, Ericsson and Psion, building on Psion's EPOC operating system developed for its handheld personal computing devices. The core business of the company is to develop and license an operating system with open APIs. Over the next 10 years the company signed contacts with device manufacturers and carriers around the world to deliver the Symbian OS, added additional members to the consortium (like Panasonic and Samsung), and managed to pass the 200 million phones sales mark. It comes in a variety of different UI flavors, such as S60 developed by Nokia, UIQ promoted primarily by Sony Ericsson, and MOAP, promoted by the Japanese mobile network operator NTT DoCoMo, which exerts more technical control over its device vendors than any other network, primarily for its FoMA 3rd Generation service.

In June 2008 Nokia announced that it is going to acquire the outstanding Symbian shares from the other owners of the company, unite the code of the different versions of Symbian (S60, UIQ and MOAP(S)) and contribute the code to the newly founded Symbian Foundation. This business decision and announcement aims directly to compete with the new Google Android platform which has been founded on open source principles. Nokia had realized that users wants an ecosystem in which their device is an integrated unit which gives them seamless and easy access to the world, and the best way to achieve that is by opening it and building an ecosystem around it in which developers have incentives to write software to it.

The Symbian foundation plans to release the first unified version of the Symbian OS during 2009, and release it as open source not later than June 2010 (two years after the Symbian Foundation announcement).

The biggest question behind the Symbian initiative is whether or not it is too little, too late. The iPhone and Android platforms are far more modern than the current Symbian architecture, which is an evolution of the EPOC operating system developed in the 1980s for devices with very limited resources, and they have already managed to gather huge community of developers around them. Developers have embraced them because of their power, as they are scaled down versions of what were originally mainstream computing platforms, and because of how easy it is to develop applications for them.

On the upside, Symbian has the advantage of its large installed base and its maturity, but its challenge is to lose the image of the "old" and "non trendy" system and create an excellent platform that is appealing to device manufacturers as well as for developers.

2.3.3.2 Architecture

Since the process of unification and transformation of Symbian into open source system is at its beginning, three are still no available details about the architecture of the system.

2.3.3.3 Licensing model

The licensing model that was chosen is the Eclipse Public License which, like the Android license, allows for proprietary developments. Like Android, Nokia was also criticized by many within the open source community this decision, claiming that the ecosystem that it will create will not be truly cooperative and therefore will not allow innovation to blossom and will hurt the efforts of the Foundation to create a truly advanced and fully competitive platform. This may be true, but Symbian's decision should be evaluated based on the resources that they have. At the time of the establishment of the foundation, Nokia believe that the best way to encourage innovation is to make the platform reachable to as much audience as possible. Unlike that LIMO Foundation which already had a strong industry backup and support, if Nokia would have taken the efforts of developing the open source Symbian alone, it may not be embraced by the broader mobile industry. Since the foundation is only at the beginning of the way, the future will tell if that was a good decision.

2.3.4 iPhone

2.3.4.1 Background

In 2007 Apple revolutionized the global mobile industry with the introduction of the iPhone, the world's first full touch screen mobile phone. As in other Apple products, the device was closed software and closed sourced, and did not allow modifications beyond the few configurable items on the phone. The new mobile phone was extremely successful, and sold more than 3.7 million

units in 2007. But the revolution was only a small scale compared to what happened with the introduction of the second version of the iPhone operating system in March 2008.

The unique thing about the announcement was the release of the iPhone SDK which allowed developers for the first time to write applications for the iPhone. The API was relatively advanced, and allows developers to access different components of the phone, like the positioning system, camera, and sound devices. When the new version of the iPhone came out, it was equipped with the new operating system, and was integrated with Apple's App Store, which allowed users to download applications written by 3rd party developers directly to the phone. The process of developing software applications, uploading them to the App Store and selling them globally was relatively simple, especially compared to the accepted standards of mobile 3rd party applications in those days. Within weeks, thousands of application were written and uploaded to the App Store, and a whole new ecosystem was created around it. In 2008, the device sold more than 13.6 million units.

Even though the iPhone is not an open source device, it revolutionized the mobile domain in three ways:

1. It was the first time that a device manufacturer successfully dictated the conditions for the carrier. Unlike the traditional model, where manufacturers did everything they could to comply with the restricted condition of the carriers, Apple was able to dictate the design of the phone to AT&T. In addition, it was the first device manufacturer ever that was able to make the carrier to share its revenues from data and voice packages of iPhone users, although it has now given this up in favor of AT&T providing subsidies but Apple

keeping the content and application revenues that it gets from the iPhone users who buy through the iTunes Music Store (iTMS).

2. It was the full touch screen device, and marked a new dominant design for smartphones. By the time the iPhone came out, there were many designs for smart phones but not a single dominant one (for example, Nokia has been trying for years to market the design of its Communicator but it did not quite took off). RIM has been relatively successful with the design of the Blackberry with a full keyboard and a screen above it. But iPhone was the first device that employed a full touch screen, which almost other device manufacturers are now seeking to copy.

3. The App Store ecosystem. As Jeremy Wright, Global Director of Brand Solutions at Nokia said at the Boston Mobile Monday gathering in April 2009, "There was practically no 3rd party mobile application development before iPhone, at least not one that is worth mentioning". Instead of writing software that will work on very few devices, using a limited and poor API, with very low chances of being able to sell it to the carriers, iPhone application developers are able to relatively quickly and easily develop beautiful applications for iPhone, and sell them instantly to iPhone users all over the world. Followed by this revolution, many other companies and device manufacturers have announced their own App Stores including Google for the Android platform, Nokia, RIM for its BlackBerry devices and Samsung.

2.3.4.2 Architecture

Since the iPhone OS is not open sourced, its detailed architecture is not publicly available. An architecture overview, however, is available at the iPhone developers' website, in order to guide and help developers create iPhone applications. According to the guide, the architecture of the iPhone is similar to the basic architecture of Apple's personal computer operating system, the Mac OS X, in which the operating system serves as a mediator between the applications and the hardware. The iPhone OS can be viewed as a set of layers. Developers writing their code are encouraged to use high as possible level frameworks, since they provide abstractions and reduce the number of lines of code needed. The Cocoa Touch Layer includes key frameworks for basic applications. The Media Layer provides access to graphics, audio and video technologies to create multimedia applications. The Core Services Layer provides access to fundamental system services such as Address Book. The Core OS Layer encompasses the kernel, drivers, and basic interfaces of the operating system. Below is a diagram of the iPhone OS architecture:

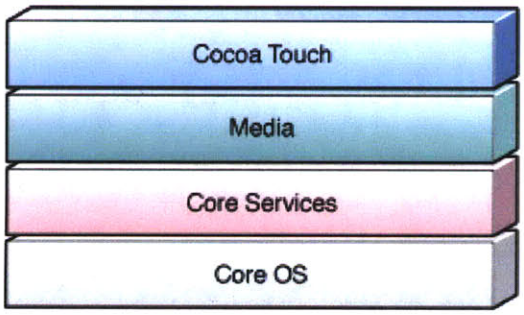

Figure 8 - Layers of iPhone OS. Source: iPhone Dev Connection

2.3.4.3 Licensing model

The only code which is available in the iPhone is the Software Development Kit. The SDK includes interfaces and methods that allow 3rd party developers to use the resources of the

iPhone. It is released under GPLv3 license which enforces anyone who modify it to distribute it under the same licensing model.

2.3.5 Windows Mobile

Windows Mobile is Microsoft's operating system for mobile devices. Its first version, called Pocket PC 2000 was released in the year 2000. Microsoft's incentive for developing this operating system was to enter the market of mobile devices, which at that time were mainly Personal Digital Assistants (PDAs). It perceived these as both a potential threat to its dominance of the personal computing landscape, and as a business opportunity for it to extend its franchise into from the desktop. It was supposed to run on Pocket PC devices, and compete with the popular Palm Pilot devices. Over the years the operating system evolved and included new features, such as multimedia, connectivity, and of course, phone.

According to industry specialists, the Windows Mobile platform today requires a fundamental upgrade and adjustment to the new and evolving customer requirements. According to a recent Gartner report, the platform has lost its second place market share position behind Symbian to Apple and the iPhone on the consumer side, and to RIM and the Blackberry on the business side.

Microsoft is expected to release the next version, Windows Mobile 6.5, in May 2009, which is mainly an incremental upgrade to the previous version. Its next major release, Windows Mobile 7, is expected only in 2010, and will probably include the popular multi touch and multi motion features that are already standard features of the competitors. Another possible direction for Windows Mobile is some form of merge with two other Microsoft's mobile platforms:

- The Danger platform, engine of the popular multimedia device T-Mobile Sidekick. The platform was developed by Danger, which was purchased by Microsoft in early 2008
- The Zune platform, engine of Microsoft's popular product line of MP3 players

2.3.6 RIM and BlackBerry

The BlackBerry is a wireless communication device, developed by Canadian company Research In Motion (RIM). The first device was introduced in 1999, and included a two-way pager. Over the years, they introduced more and more functionality into the device, including email connectivity, mobile telephone, text messaging, internet faxing and web browsing. The BlackBerry was the first to incorporate a Push Email capability into a mobile device and because of that became popular within enterprises. The BlackBerry uses a proprietary operating system called BlackBerry OS. In October 2008, in response to the Apple's App Store, RIM announced their new application store – The BlackBerry App World. Since BlackBerry originated as a corporate device, is uses a slightly different system architecture than the other platforms that makes use of a dedicated server to control the devices of each organization. The dedicated server allows users of BlackBerry devices enjoy a full, real time, synchronization between their device and the corporate email environment. On the administrator side, it allows IT administrators the ability to fully control all features of the BlackBerry device. This unique client-server architecture has made the BlackBerry the current leading enterprise mobile device, although competing platforms like the iPhone are beginning to close the gap, and appeal to corporate customers, by introducing features like Push Email and real time synchronization using services such as Apple's MobileMe.

This page has been intentionally left blank

Capter 3 - General effects on the industry

3.1 General changes in the ecosystem

3.1.1 The power of carriers

These are tough times for the world's mobile carriers. Over the last 10 years they enjoyed:

1. Steady growth in the number of mobile subscribers
2. Control over the devices that can use their network and revenue
3. Exclusivity in providing services though proprietary networks (not the web)

Increasing competition from other carriers, wire line communication services, VOIP services, and potentially disruptive technologies such as WiMAX have forced the carriers to invest heavily in technology, and provide better services and products to their customers. They have had to do this, however, while trying to keep as much control as possible on the devices, services and content that runs on their network. An example for that is the service offered by the Japanese carrier NTT DoCoMo. As part of the i-mode service, it includes a mobile payment capability also known as the "E-Wallet" that integrates many cards normally in one's wallet (ID cards, credit cards, membership cards, room keys, commuting passes), into a mobile phone. As of January 2006, the number of subscribers using i-mode surpassed 10 million. Customers using the i-mode service typically are much less likely to switch to other carriers.

By providing these services, the carriers attempt to overcome their greatest fear, that of becoming a dumb pipeline that only delivers basic network services. They are able to keep large percentage of the pie of the mobile industry, such as revenues from selling devices, revenues from providing services, and also manage to retain their customers.

The opponents to that approach claim that carriers are not supposed to influence anything in the industry except providing the infrastructure and the network. The situation can be compared to connecting to a cable television company. In that case, the cable company will not tell the customer which TV to buy (and the customer would definitely not by it from carrier), they will not have influence on the features that TV sets will have. They will not decide which movie rental services the customer should use, and will certainly not tell the customer which programs to watch.

In spite of the carriers' best efforts to push back anything that might promote their becoming a dump pipe, the increased competition described above and the increased customer expectations, forced them to invest and promote technologies that will increase the openness of their network and their vulnerability. There are a few clear examples for that in the last few years:

1. AT&T and the iPhone – in 2007 AT&T signed an exclusive contract with Apple to distribute their new and anticipated device, the iPhone, in the US. Apple locked the device to work only in the AT&T network, and in return, AT&T paid Apple for every new customer that joined AT&T because of the iPhone. That contract was a turning point in the history of the mobile industry. For the first time, the bargaining power was in favor of the device manufacturer, and not only the device manufacturer did not pay the carrier for selling its device, the carrier paid the manufacturer for joined customers. In addition to that, Apple opened an application store, where iPhone users could choose from a variety of more than 25,000 3rd party applications and download them directly to their device. By that, AT&T gave up also the power of dealing directly with the applications developers, determining which applications will go into the devices, and charging fees for

that. Instead Apple maintains these relationships with end users, and captures for itself the resulting revenues.

2. Open source consortiums – carriers openly embraced the development of mobile open source platforms. For example, NTT DoCoMo is a member of both the Open Handset Alliance (developing Android) and of the LIMO Foundation. By that, they are supporting and financing the development of operating systems that will decrease the power of the carriers even more. According to Dr. Kiyohito Nagata, Senior Vice President and Managing Director of Product Department at NTT DoCoMo, his company is well aware of the risk of becoming a dumb pipe. Supporting open source platforms is DoCoMo's "opportunity to create differentiate services which can be accepted by current customers", and by that, hope that third party software developers and device manufacturers will gather around DoCoMo.

Alongside the carriers' actions to open their networks and provide variety and freedom to their customers, we still see signs of resistance to certain services and content, especially things that might be significantly disruptive to the carriers' business. The best example for that is the carriers' relationship with the global VoIP service Skype.

Even though Skype is a well established internet service with more than 400 million users as for April 2009, its penetration to the mobile applications space is still very limited. The implications of availability of Skype applications on millions of 3G devices around the world are potential revenue losses of voice calls to the carriers. Because of that, carriers have tried in the last few months to limit or block the penetration of Skype to new platforms and markets:

1. In February 2009 two UK carriers, Orange and O2, are threatening Nokia not to carry its latest N97 handset, due to the inclusion of Skype in its built-in software

2. In early April 2009 the iPhone version of Skype was added to the Apple App Store, but due to the pressure of mobile carriers around the world, the application only works on Wi-Fi networks, and not using 3G connectivity

3. German carrier, Deutsche Telekom announced that it would block the use of Skype on iPhone completely

The mobile carriers around the world explain the move not by a will to limit the customer's choices, or by potential loss of revenues, but by the potential to hurt their network. Skype, on the other hand, published a message on the company blog that condemn the decision to block Skype on mobile devices, refute the carriers' claim of technical concerns, and encouraged customers' to apply pressure and seek regulatory actions against the carriers.

According to Ajit Jaokar, author of the open technologies blog Open Gardens, any carrier that will not allow its customers to use Skype opened, will simply loose customers that will switch to other operators.

The future, according to Mr. Jaokar, therefore looks inevitable for mobile carriers. They will keep losing control over their network, and will not be able to keep making revenues from services and content that can be provided by other players and not simply because they own the network. If they want to retain their customer, they will have to provide them with the freedom

and to use the devices they want, to download and use the applications they want, and access the content they want. Differentiation will be in the form of innovative added value services (like i-mode) that will give real value to their customers. It is still unclear whether he is right or wrong.

3.1.2 Application developers

This industry segment is the one that had the most drastic change in the last few years, and specifically since the launch of the iPhone SDK, which substantially increased the number of mobile application developers worldwide.

The main changes that the iPhone brought are:

1. A standard environment to create applications which is similar across millions of devices (more than 17million devices as for January 2009)

2. An advanced, accessible, and relatively simple SDK

3. A new, revolutionary ecosystem to distribute the applications (the App Store)

After the introduction of the iPhone, more and more companies are announcing similar intentions to do the same, like the Google Android Market, the BlackBerry Application Store, Nokia's Ovi Store for Symbian applications and Samsung Mobile Applications website.

It looks like this is just the beginning with regard to 3rd party development, but with the great opportunity, there are also some risks:

1. Temporary hype – the iPhone App Store and the Android Market contains tens of thousands of applications, many of them very simple and primitive which do not offer

substantial value to the user except few hours of interest, as evidenced by the figures below, displaying retention rates for paid and free iPhone apps. According to the data, we can see the percent of the users, going back to use s certain application after the first use s rapidly decreasing.

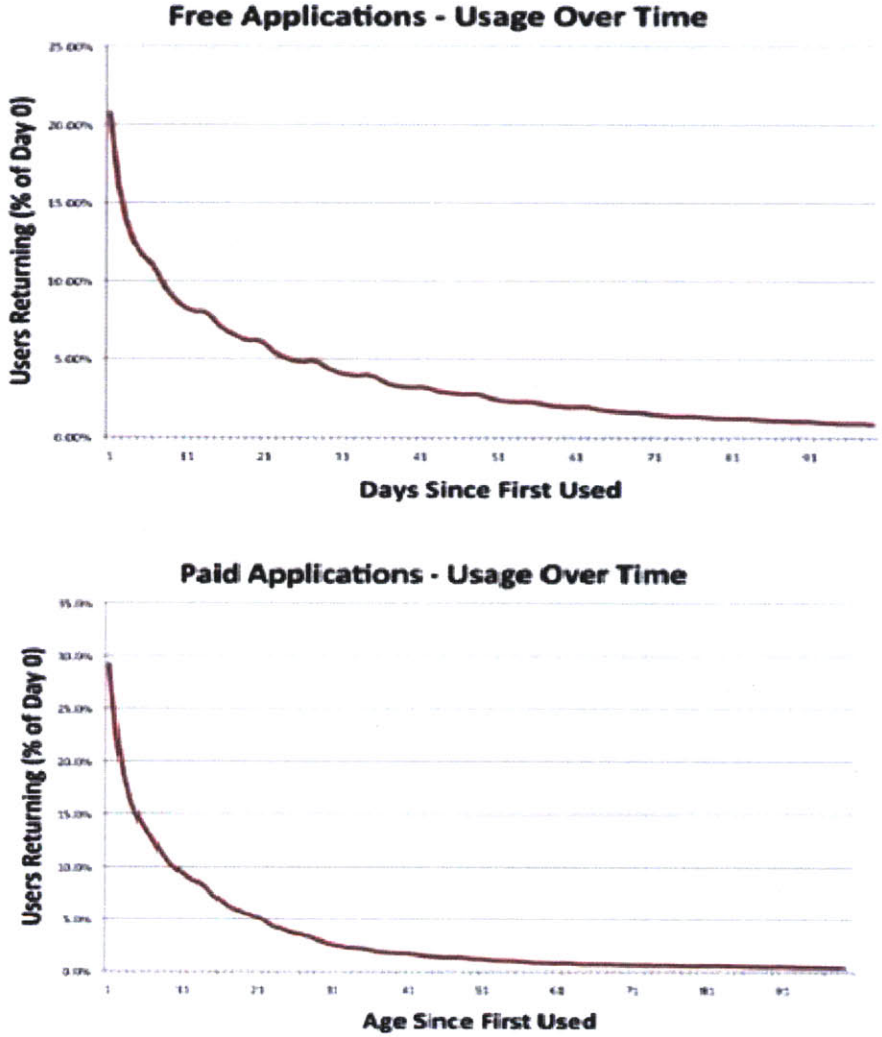

Figure 9 - Retention rates of iPhone applications. Source: Pinch Media, February 2009

According to these figures, it is possible that the latest rise of 3rd party mobile applications is nothing but a temporary hype that does not indicate anything on changes

in the balance of power in the market. On the other hand, it may be just the beginning as more and more real value applications are being developed in parallel to the development of the devices APIs, with access to more of the devices resources and capabilities.

2. Platform fragmentation – the iPhone is only the first (and until the Android market came out, the only) platform using the App Store model, and therefore it made a simple choice for developers to build their applications for the iPhone. But as the market evolves, more operating systems will offer the same model (as the Android today). Furthermore, the Android operating system is not targeted at just one device. On the contrary, it is meant to be a platform for multiple devices with multiple interfaces, features and hardware. When the Android will be fragmented, how will their App Store be managed? Is it possible what we are again facing deep fragmentation in the market, as in the pre-iPhone area, which will make it harder for application developers to exist? On the other hand, it may be possible that the fear from too much defragmentation and incompatibility of devices to existing Android applications will be the driving force behind non-defragmentation of platforms.

3.1.3 New industry entrants

Now that mobile operating system will be available to everyone for free, the question is whether it will encourage new device manufacturers to enter the market. In other words, it's the same as asking whether developing a mobile operating system today is a significant barrier to entry. To answer that question as part of this research project, I interviewed several people from all of the

leading device manufacturers in the market. They all answered the same – that developing the core operating system is only a small stage in developing a mobile phone.

A mobile phone contains hundreds of different components that need to be fully integrated to deliver a seamless functionality. The operating system has to work with the different components of the system like cameras, chips, and network connectivity. Those components upgrade very rapidly. A small upgrade like increasing the mega-pixels of the camera can major changes in the architecture of the device due to more data traffic on the internal phone data channels. Additional stages in the development stage of a phone include adapting it to different carrier networks in many countries. Symbian, for example, supports 12 lines of phones, more than 200 countries, and a few carriers in each country. There is huge amount of work of adaptation and configuration of phones in these countries. Mature device manufacturers like Nokia, Motorola, Samsung, etc. has acquired experience in the development process of such devices, including the complex operations of upgrading devices, relationships with carriers, distribution, and are able to lower their developing cost to a minimum required to make profits. A new player with only an operating system in hand will not be competitive enough to support the whole developing and distribution process.

This approach is given reinforcements, by the latest industry apathy towards the potential new Dell mobile phones, running the Android OS. According to CNET News' report from April 2009, the device that Dell came up with so far hasn't been very exciting, and was rejected by carriers. The reporter, Erica Ogg, claims that in order for Dell to compete with Nokia, Apple and RIM in the smartphone market, it will have to offer a significant jump over what is currently being offered, in addition to being able to make decent margins.

In any case it seems that it will not be easy for new devices manufacturer players to enter the market and create a profitable business. According to industry estimations, it took Apple 3 years to develop a single mobile device (the iPhone). Developing a complete line of products creates additional challenges.

3.1.4 New source of revenues – Advertising

As some mobile phone components are provided free, companies might look for new revenue generating sources. One of those potential models is mobile advertising. The idea is to provide products (like applications) or services (like content viewing) for free, in return for display of advertisement as an integral part of the user experience. The advertising company pays the advertising platform for each user click on an ad. The challenge is to create an advertising platform that will best target ads to user needs, and that way increase the average click rate of its ads. Targeting of ads was very successful in the PC industry, with services such as Google's Ad-Words, but is still premature in the mobile industry.

According to a report by eMarketer, 2008 was the year that advertising on mobile devices really took off. The turning point, according to the report, was the increasing popularity and availability of smartphones, heightened by the iPhone launch in 2007. New advertising strategies for these devices include banner, display and video ads, and application marketing.

eMarketer's forecasts predict tremendous growth for mobile advertisement in the next few years, as seen the diagram below:

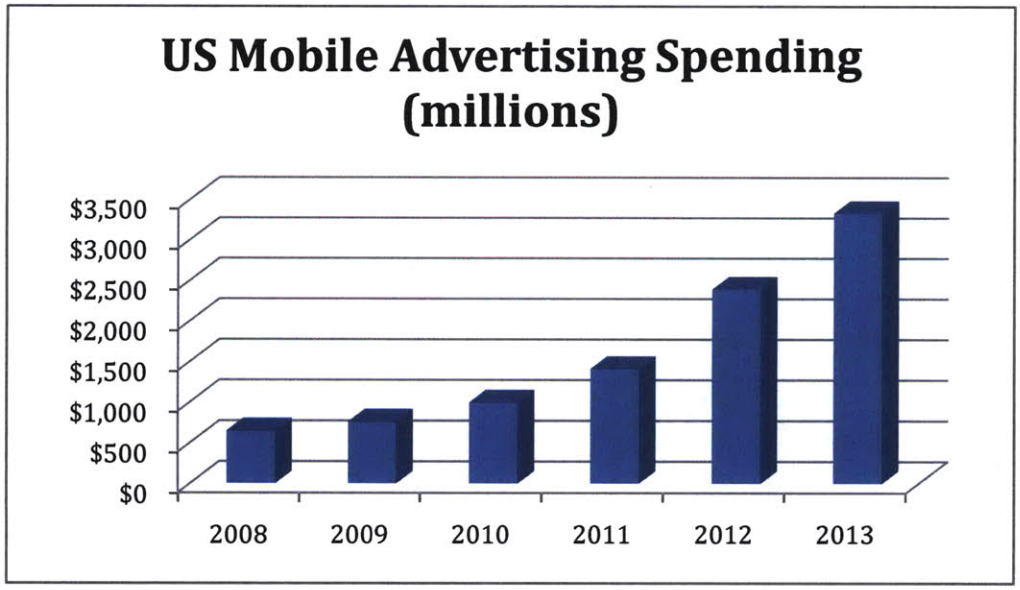

Figure 10 - US mobile advertisement spending. Source: eMarketer

The question is whether mobile advertising has the potential to make profound changes to the current business value chain of the mobile industry. Will revenues from mobile advertising be high enough to support ad-funded business like application development?

According to data gathered from application developers and advertising companies Admob and Pinch Media, it seems that for the next few years, mobile advertisement will not be the next best thing. Below are some of the reasons why:

- High development costs of mobile applications even thought they are lower than before the iPhone area, requires high returns
- The application stores are designed for maximum turnover, which means, there is an increasing competition between developers
- Short application life cycle, as seen in the retention rates of iPhone applications

- Hard to target the ads to the users. The carriers are the only ones with information about the users. Applications can only know the type of device.

This page has been intentionally left blank

Capter 4 - Effects on device manufacturers

4.1 Strategy

4.1.1 Should companies use or not use open source?

The question whether or not to use open source depends on the individual strategy of each player. It seems that the vast majority of the companies in the market have decided to incorporate at least one open source platform in their product portfolio. An evidence for that is with looking at the new member lists of open source consortiums. Samsung, LG, Motorola and Ericsson are both members of the Open Headset Alliance (Android) and of the LIMO Foundation. In addition Samsung and Motorola already officially announced that they are going to release Android phones during 2009. With many other players already officially adopting it, it seems that Android have managed to find a sweet spot with very high rates of adoption, both from developers and device manufacturers. While talking to device manufacturers about their reasons for embracing or even switching to open source, the most compelling answer is that most of them did not do it because they wanted to, or because it gave them something they didn't have before. It was more of a forced move - consumers are demanding it. Below is a breakdown of the potential effect open source might have on a company and for each one, what industry players had to say about it:

1. Substantially improving current platform – With the appearance of the iPhone, most of the industries mobile platforms became awkward, out of date, complex to use and poorly fitted to the new customer expectations that it has now set. Even companies that were long known for ease of use of their product and for their advanced features like Nokia,

found themselves lagging behind a state of the art design the users simply loved. In this situation, strategic moves were needed for each device manufacturer to decide how to close the gap. By improving the current proprietary platforms, or teaming with the new ones. Nokia chose to improve their own platform, Symbian, but decided not to do it by themselves. They established the Symbian Foundation, and began a process of rebuilding their platform on an open source foundations. Motorola, which were renowned for their innovative hardware designs (i.e. the StarTAC and the RAZR), but notoriously known for their not-so-friendly software and user interface, decided to place their bets on completely new open source platforms.

Figure 11 - The Motorola StarTAC and RAZR

Figure 12 - The soon-to-be-released rumured Motorola's Android phone. Source: anonymous

2. Cost saving – Although one of the core missions of the LIMO Foundation is to reduce the development costs for its members, that argument was dismissed by most of the mobile devices manufacturers representatives I interviewed. In their opinion, the basic operating system, like the one Android will provide for free is a very minor step in developing a mobile phone. The most complexity is in connecting the operating system to the many hardware components that exist in a mobile device, and which are different from model to model, country to country, and even carrier to carrier, and also being upgraded very rapidly. Moreover, any switch a company does to a new operating system requires heavy initial investment, which requires sometime to completely wind down capabilities and perhaps close existing facilities and open new ones.

3. Attracting developers – while all players have realized the latent power in user innovation, 3rd party developers and applications stores, it remains unclear how each of

the players will go about implementing it. Nokia is making substantial changes in the Symbian architecture even before turning into open source, in order to make it more appealing to developers. Despite being so new, Android already has a large developer's community, with a working application store, and users downloading software to its single device. It is unclear, however, how introduction of new devices from more manufacturer will be integrated into this model. The biggest question is how much exactly the platform will be fragmented, and if current Android applications will support new phones and in what level. Since the ASL license under which Android is being released on allows endless fragmentation, it is impossible to know what device manufacturers will do with it. Motorola, for example, has not yet released any information to consumers or developers about their prospective Android phones. We can assume that any device manufacturer will not turn its back against the large community of Android developers Google managed to build, and will allow this great resource to contribute to its success.

4. Break the Monopoly of Nokia + Symbian – Additional reason for a device manufacturer to support a new open source platform which is available to all the players is the fear from competition that will risk their current ecosystem. For example – in the past, players like Samsung and Sony-Ericson relied heavily on Symbian as their smartphones platform, although it was controlled by Nokia, their biggest competitor. Other players relied on Microsoft to supply the Windows Mobile operating system, but there was also a great fear that Microsoft would decided to go into the device manufacturing business, as it did with video gaming consoles and MP3 players. The decision of going to an open platform

which is not controlled by any major device manufacturer holds a major strategic risk reduction to the company.

4.1.2 Basis for differentiation

Companies focus on different things to achieve differentiation. Some companies are great at innovating, some are good in designing user interface, and some have mastered industrial design. The differentiation depends first and foremost on the company's resources, the things that they have and others don't. Good resources are usually hard to achieve, and often evolve over time. Examples are:

- Intellectual Property
- Loyal Customers
- Brand name
- Size
- Employees

Owning a mobile operating system is a resource companies have invested a lot to have. In the past, companies have based their business models on mastering the development of operating system and selling them to mobile device manufacturers (i.e. Symbian and Microsoft). Other companies developed their own proprietary operating systems, and each manufacturer created its own UI on top of the core software, and integrated it to its unique hardware, services, and content.

Open source operating systems, provided free by Symbian, Android or LIMO, have changed that picture. For the first time, it's almost impossible to make money by simply selling operating

system licenses or by selling devices that embody a closed proprietary operating system. The meaning of this is that companies that their core competence in the past was developing mobile operating system must find other ways in which they can differentiate themselves.

Interviews with product managers and R&D managers at device manufactures raised different opinions with regard to what their companies should do in the new situation:

- According to Interviewee A, a product manager at a leading device manufacturer, we should distinguish between two main markets:
    - The developing world, targeted by simple voice/text devices and feature phones. In these markets the open source operating systems has no major effect. Customer's expectations in those markets are still relatively low, and the key base for differentiation is low cost of development and manufacturing, achieved by size of the company, knowledge and expertise of its employees, and efficiency of its supply chain
    - The developed world, targeted by smartphones. Here there is a huge impact of open source on the base for differentiation. According to the interviewee, customers no longer care about a standalone phone, but on the complete set of content and services accessible by that phone, the challenge of every devices manufacturer is to think holistically about how to deliver this customer experience, and moreover, how to fit it into an ecosystem that will benefit all players and therefore be possible. For example – the iPhone and App Store benefit the carrier, AT&T, as the provider of the infrastructure, the developers, as they are able to sell their software to millions of customers, and Apple, which sells the device but also share the revenues of both the carrier and the developers

- According to interviewee B, CTO at leading device manufacturer, differentiation at the area of open source will come from a few possible sources:
    - Hardware components – New types of screens, Memory chips, batteries, etc. are tied directly to the user experience, and are a major target for innovation. Companies can innovate a lot in this area and also protect their innovations with IP. An example for that is Apples Multi Touch screen which cannot be copied by any player since it is patented.
    - Rich UI – Above the layer of core operating system, device manufacturers will compete over creating the best user interface experience. Innovative features like cubes and figure scrolling. The figure below shows an example of an innovative cube user interface, developed by Haansoft.

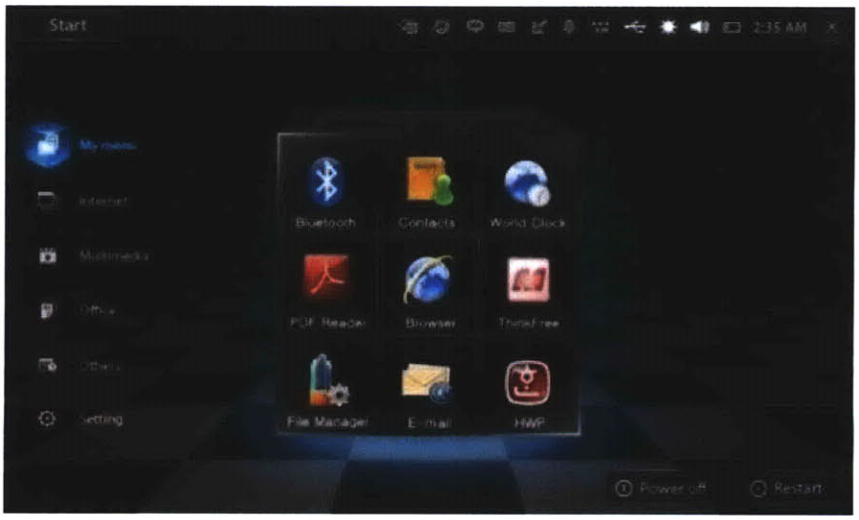

Figure 13 - 3D Cube interface. Source: http://www.ubergizmo.com

- According to interviewee C, an R&D manager at a leading device manufacturer, the open source operating system should not change anything in the base for differentiation for this company. Developing the core OS is only the first part in developing the device, the most

work is to integrate the platform into the hardware, make it compatible to all the devices. Changing to an open source platform might cause heavy operational influences (which will be discussed later) but for differentiation issue it does not change a lot because before open source, players could purchase operating systems and modify it.

4.1.3 Chosen platforms

In his blog Open Gardens, Ajit Jaokar claims that Android and LIMO are like Apples and Oranges – you cannot compare them because they are not the same thing. Android is an ecosystem (i.e. containing not just the core OS, but integrated with additional services and content), and LIMO and Symbian are simply operating system. It is unclear at this point how each ones of those operating systems / ecosystems will evolve. Android might get fragmented into dozens of types and Symbian might be integrated with Nokia's Ovi service. Perhaps because of this uncertainty, device manufacturers are betting on more than horse and releasing phones on multiple platforms, waiting to see where the market go to.

4.2 R&D

4.2.1 Changes to the development process

According to interviewee C, who is an R&D manager at a leading device manufacturer, the decision to work with a certain open source platform poses huge changes to the development process and methodology. However since open source platforms are relatively new with the company, it is still too early to tell exactly which effects they will have. Below are some possible examples of the effects it might have:

- Developing a new line of phones using a new operating system is a huge investment. It requires hundreds of people to be directed to focus on this effort, developing knowledge and expertise. Due to that, choosing a new operating system, generally means discontinuing an older one, and consolidating the company's efforts.
- Switching to work with a new OS maker requires redesigning the R&D processes to accommodate tight synchronization with between the two companies. Simple activities like upgrading a driver to support an upgraded hardware component should be built into the development cycles in order to make sure new models will be released on time and with the desired quality.

Other considerations when working with open source OS might include:

- Allocating some of the R&D team to work on the core operating system, and by that, share the effort and costs of developing the core elements of the OS to the benefit of the whole (members of the community or the general public). Similar approach can be found in the PC industry. There are many commercial companies who offer Linux distribution.

However the core component of the Linux operating system, the Kernel, is the same for each one of the Linux distributions, and each one of the companies that develop those, are taking part in the effort to develop the Kernel, and allocating some of their workforce for that task and for the benefit of all.

- Quality Assurance – Since mobile phones running open source code will be more open to changes and fragmentation, and will be able to run thousands of third party applications, it may be argued that they will require much more quality assurance. This is going to be even harder to implement due to the distributed development of all the software of the phones. Therefore, companies should be capable of organizing the development process and their operational process to support this. One example for this type of change is a startup company called Nexperience, aiming to provide remote testing solutions to hundreds of mobile phone models though different countries and carriers. According to their solution, a developer, a device manufacturer or a carrier, can facilitate quality assurance tests for its phones even when development teams are in different locations in the world. For example, a software development team in India, can integrate the software with the hardware manufacturer who is located in the US, without the need to travel or relocate people. Nexperience are reporting their selection by Vodafone to support the delivery of mobile applications on new handsets.

4.2.2 Changes to internal application development

According to interviewee B, who is a CTO at a leading device manufacturer, open source and open innovation have changed the role of internal company application development. In the past, companies employed internal development teams to create a variety of mobile application, from

utilities and tools, to games and entertainment. Those applications created value for the users, and added to the differentiation of the devices. An example for such application is the classic mobile phone game "Snake", which run on Nokia phones and their proprietary operating system only and was a huge success among customers.

Today, the governing model for developing mobile applications is the App Store model, in which the platform maker provides the developers community with a toolkit, an API which they can create applications with, allow them to distribute the applications through the online store. This model changes the role of internal application development teams. Now the job of these teams is to create much more advanced and complex applications. The advantages of this approach are:

- They are not restricted to the rules of the application stores ecosystem and therefore do not have to target the market of individual, private consumers. Instead, they can think more broadly about developing applications and services that will fit into the big picture of their company's strategy
- They have access to phone resources that are not accessible by regular developers through the API (like the camera, GPS antenna, and core operating system components).

These advantages will direct them to develop mainly two types of applications:

- Compound applications with heavy algorithms and use of hardware like speech recognition, and video streaming, picture recognition
- Applications which are part of client-server systems, which may not be appealing to private customers, but can be very relevant for enterprises.

4.2.3 Methods of supporting new hardware components

One of the biggest questions behind the future of open source mobile platforms is: who is going to provide support for the platforms?

According to interviewee A, a product manager in a leading device manufacturer, the technology requirements for mobile devices are advancing much faster than the ones of the PC industry. Components like cameras, processors, memory chips are being upgraded very rapidly. An operating system manufacturer must support that so its platform will stay up-to-date. According to him, there is a big question mark about the support capabilities of companies like Google, which were able to develop the base platform and team up with industry players to deliver the first device, but their ability to provide continuous support for new devices, and new hardware components is not yet clear.

Application developers are also not very clear about the level of support will receive from the open source foundations. According to interviewee E, a CEO of a mobile application company, he is not sure if his company's products will work on new versions of Android and on new devices that will run the platform.

The question of support occupies not only device manufacturers and developers, but goes deep into the makers of the basic components of mobile phones. According to Interviewee D, who is a product manager for a leading chip manufacturer, in the past they had a clear address to report compatibility issues and submit bugs. Developers of operating systems would help the chip manufacturers to integrate their components with the software because it was their incentive to make sure that the products will work seamlessly together. With open source things are different.

The chip manufacturers are expected to do much more in the integration process and there is no single point of contact to address bugs to.

4.2.4 Licensing considerations

Open source licenses can be divided into two types:

- Permissive licenses like ASL (Used by Android and the Open Handset Alliance) and EPL (used by Symbian) – This type of license allows devices manufacturers to integrate the code under it into their closed-sourced proprietary products and redistribute them under a variety of other licenses, without having to turn proprietary enhancements back to the open source community. Even when they don't have to, companies might choose to work with the community which develops the core operating system components, and allocate R&D resources in developing software that will be published to the public. Possible incentives for doing that are:
    - Help promoting the platform – the idea is that making the platform more stable and full featured, will encourage consumers and developers to adopt it, which will benefit all. Interviewee E, a product manager at leading chip manufacturer, said that they open fix bugs in the open source operating system their chips are working with, and then contribute the code back to the community
    - According to blogger Ajit Jaokar, open source communities are actually standard bodies which are working bottom-up and driven by solution specific needs. Based on this approach, a company can promote its own technology, business model, or application, but contributing an open source code to a platform and by that insuring that the platform will work best with their proprietary products.

- Non-permissive licenses – This type of license does not permit proprietary enhancements of the code released under it. Any modification or enhancement must be turned back to the community and be released under the same license to the benefit of the members. Incentive to use this type of license are:
    - Cost reduction and co-opetition – Open source communities like the LIMO Foundation are cooperation between companies for the purpose of sharing information and knowledge that can benefit all members of the community. A company can choose to be a member of such consortium if the product or technology that the community is developing is not its core competence, and cooperating with other companies to develop it can reduce its costs, give them a better product than the one they had before, and allows them to differentiate based on other things.
    - Setting the standard – like the case of the permissive license, becoming a member in a community with high industry power, can help a company to promote its technologies and standards

In his post in the Ars Technica website, blogger Ryan Paul discussed the importance of Google decision to release Android under the permissive license ASL. This type of license is usually preferred by companies, and Google's choice to use it was motivated by their desire to insure that Android will be adopted by as much companies as possible, which will build on top of it without having to expose their proprietary technologies, which can result in completely changing their base for differentiation and their business model.

So far, this strategic decision was proved right for Google, whom their platform is already widely adopted by many device manufacturers, and with dozens of new devices expected to be released later this year.

This page has been intentionally left blank

Capter 5 - Summary

The mobile industry is going through major changes, possibly equivalent in magnitude to the effects of the release of Windows 3.1 on the PC industry, or to the effects of MP3 format on the music industry. In this work I tried to analyze and asses how big is that change, and how it affects different players in the industry, particularly manufacturers of mobile devices.

Chapter One provided background and history on how the mobile industry evolved to these days. It explained the structure of the ecosystem and value chain that drove the industry, and how customer demands grew over time.

Chapter Two discussed the rise of open software platforms, and how it led to the current availability of free, open source mobile operating systems for all players in the market. It reviewed today's leading mobile platforms and how they fit into the trend of openness.

Chapter Three discussed the general effects of mobile open source on the industry today and in the future. The key takeaways from the chapter are:

- Mobile carriers are at risk of becoming a dumb pipe. Due to their awareness of the risk (and in spite it), the world leading mobile carriers are openly and broadly supporting open source platforms, in hope to gather consumers and manufacturers around them, and to create opportunities to provide unique content and services
- Application developers are currently enjoying an explosive demand for their services, as creators of potentially killer applications for device manufacturers and carriers. There is a risk, however, that this increasing demand is nothing but a temporary hype and a bubble that may burst

- New entrants are expected to enter the mobile industry as device manufacturers using open source platforms. Industry's experts, however, are skeptical about their ability to create and capture value, due to the high barriers to entry beyond the cost of developing an operating system, and most device manufacturers do not see this as a threat
- Potential new revenue sources are emerging in the form of mobile advertisements. Although this field is still at early stages and has not proved to be a substantial source of revenues for companies, advanced technologies such as location awareness can have positive influence on the targeting of ads to users, and can transform mobile advertising into a highly profitable business

Chapter Four discussed the effects of mobile open source on device manufacturers. Key takeaways from the chapter are:

- Most device manufacturers do not consider mobile open source as a driver for changing company strategy or base for differentiation. For the companies that have been in the mobile device making business for many years, creating a mobile phone is much more than taking an operating system and installing it in a handset. It requires years of acquired knowledge and expertise to seamlessly integrate hundreds of components into dozens of lines of phones, manufacture and distribute them in a way that will allow profits. For those companies, even with the availability of mobile open source, their core business stays the same
- While high level strategy does not change, device manufacturers are making extensive changes in their development processes in order to embed new open source operating systems into their products, and adjust for rapid cycles of changes and upgrades to match the consumer demands

Given the effects on the mobile industry, we can also discuss the implications on the work of the industry's product managers. The research and interviews have indicated a few key issues in which the job of product managers was affected:

- Design of products – when considering different design alternatives, working with open source components requires careful modularization of the products. This is due to the need to simplify the development and maintenance process, which can be, in case open source is used, decentralized between thousands of people, and distributed over multiple locations. Modular designs help managing the development work and division of responsibility between different development teams. Also, in case of frequent upgrades and changes, modular designs make quality assurance easier.
- Complexity of products – As the researched indicated, consumer expectations at the age of open operating systems have risen to such a level in which users want solutions, not devices. Therefore product managers must not think of their products as standalone devices or services, but as parts of larger, highly integrated systems that deliver value to users and fit into business ecosystems which supports value creation and capture of value by players at all levels.
- Interaction with people – product managers have indicated that one of the key characteristics of working with open source products is that in many cases there is no single point of contact to work with, and no single company or organization to take responsibility for the product. When there is a need to integrate a system with a 3rd party proprietary product, the other party has a strong incentive to make sure that the integration will succeed. That is not always the case for open source products, and product managers must take that into account.

- Licensing issues – Careful attention should be directed to licenses under which the open source products are released. Although access to the code is usually granted to the public, using the code as a part of proprietary products may be illegal under some types of licenses. Therefore the product managers should understand the rights and restrictions that are associated with each open source product they use.

In the next two years companies will unveil dozens of new devices, new application stores will be opened, and new services will be announced. Those changes have the potential to completely change the picture of the leading industry players and way the leading players are differentiating themselves from the others. It will be interesting to repeat this research two years from now, check how the industry evolved in the light of the effects mentioned in this work, and draw a new picture of the mobile industry and its leading players.

Abbreviations

- API – Application Programmer Interface
- UI – User Interface
- SDK – Software Development Kit
- ASL – Apache Software License
- GPL – Gnu Public License
- EPL – Eclipse Public License
- IP – Intellectual Property
- OS – Operating System
- R&D – Research and Development
- GPS – Global Positioning System

Sources:

1. "China Telecommunications Report Q1 2009", Business Monitor International LTD, 2009
2. "Japan Telecommunications Report Q1 2009", Business Monitor International LTD, 2009
3. Open Gardens Blog: www.opengardens.com
4. About.com website: http://inventors.about.com/cs/inventorsalphabet/a/martin_cooper.htm
5. How Stuff Works website: http://electronics.howstuffworks.com
6. Plunkett Research Reports: http://www.plunkettresearchonline.com
7. OSS Watch website: http://www.oss-watch.ac.uk
8. Ars Technica website: http://arstechnica.com
9. "The LIMO Foundation overview", The LIMO Foundation, May 2008
10. Babylon Dictionary
11. The Symbian OS website: www.symbian.com
12. The Symbian Foundation website: http://www.symbian.org
13. O'reilly Radar Blog, June 2008, Nokia to buy and open source Symbian: http://radar.oreilly.com/2008/06/nokia-to-buy-and-open-source-s.html
14. The Ian Skerrett Blog: http://ianskerrett.wordpress.com
15. "Global Investments for Sustainable Growth in the Wireless Telecommunications Industry", Osamu Matsuda, June 2006
16. "Is Deutsche Telekom playing an April's Fool joke at the expense of Skype users in Germany?", Robert miller, April 2009: http://share.skype.com/sites/en/2009/04/is_deutsche_telekom_playing_an.html
17. Uber Gizmo website: http://www.ubergizmo.com
18. Interview A – January 2009, Product Manager at a leading device manufacturer

19. Interview B – February 2009, CTO at leading device manufacturer

20. Interview C – February 2009, R&D manager at leading device manufacturer

21. Interview D – February 2009, Product Manager at leading chip manufacturer

22. Interview E – April 2009, CEO of a mobile application development company

23. Interview with Dr. Kiyohito Nagata, Senior VP of NTT DoCoMo, April 2009

24. Nokia website: www.nokia.com

25. 3G Partnership website: www.3gpp.org

26. Android website: http://www.android.com/

27. Pinch media website: http://www.pinchmedia.com/

28. Admob website: http://www.admob.com/

29. Nexperience website: http://www.nexperience.com/

30. "Democratizing Innovation", Eric Von Hippel, MIT Press, 2005

31. iPhone Developer Connection website: http://developer.apple.com

32. BlackBerry Users (& Abusers) Community website: http://crackberry.com/

33. "More apathy for a Dell smartphone", CNET News, April 2009:

http://news.cnet.com/8301-1001_3-10229353-92.html