Platforms in the Palm of Your Hand: Mobile Value Chain Evolution

Richard Mortier, Catherine E.A. Mulligan
Horizon Digital Economy Research,
University of Nottingham,
Nottingham, UK
email: *richard.mortier@nottingham.ac.uk*

***Abstract*—The last decade has seen the telecommunications industry responding to a process of revolutionary change. Infrastructure, technology, supply chains and customer demands have evolved more in the last 5–10 years than in the previous 30. In this paper we explore how technological changes have contributed to the evolution of the mobile network value chain and consider how the latest developments in mobile and cloud computing are likely to continue the revolution. Finally, we investigate the impact of this on the business models of the telecommunications industry.**

I. Mobile Network Value Chains 1980 – Today

The mobile industry has changed radically since its inception in the 1980s, with recent dramatic incursions by the computer industry in the shape of Google, Apple and Microsoft. Continual technology development has radically changed the nature and capabilities of the mobile handset, and thus the use and purpose of the mobile network. As a result, the value chains of mobile networks have evolved rapidly over the past three decades [1]. The result has been redefinition of bargaining relationships between consumers, operators and network vendors that have been well-established since the beginning of the mobile industry. The rapid evolution of these bargaining relationships has led in turn to redefinition of the associated business models [2], [3]. Following a brief overview of this process to date, we will discuss a development on the horizon with the potential to continue this rapid evolution.

A. First Era: 1980s–mid 1990s

A high-level view of the traditional mobile networks value chain, the *first era*, is shown in Figure 1. The defining characteristic of the value chain of this era is the extremely close relationship between handset manufacturers and operators. Substantial research and development investment by both operators and vendors was used within telecommunications standards forums to establish requirements for interoperability between handsets and networks [2]. The key intellectual property in the handset was in the hardware, particularly in the radio, network and service interfaces.

The joint development of this voice services platform allowed the operators both to specify requirements to the handset manufacturers, and to exercise tight control over the services available to consumers. This value chain was maintained by the constraints of capacity of component technologies: the low bandwidth and processing capacity of handsets meant that much functionality *had* to remain within the network.

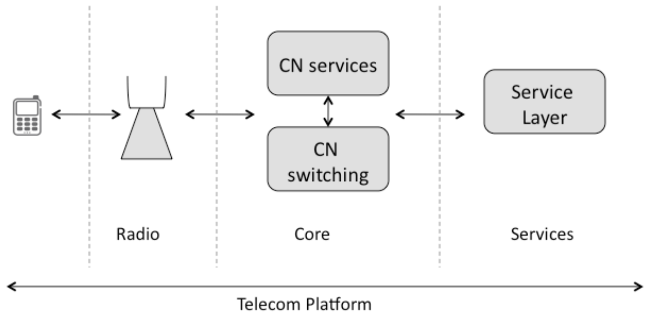

Fig. 1. Mobile Value Chain: First Era.

The inability of the handset to perform significant local processing meant that value-add had to be provided predominantly by the different network operators. Coupled with the esoteric and non-standard handset APIs, this meant there was little incentive for a developer community to grow around handset application development. Such software development that did happen was either handled in-house by the handset vendors, or was undertaken by specialist development firms often in partnership with the handset vendor.

As a result, network operators were able to extract a lot of value from the market by providing these services. Network equipment manufacturers, meanwhile, earned money from the sales of products to support these services [2]. The important bargaining relationships were thus between handset vendors and network operators.

B. Second Era: late 1990s–2000s

The open standards on which GSM was built meant that the first era was wildly successful for vendors and operators. The GSM mobile phone handset rapidly became one of the most popular consumer electronics products in the western world: over 120 million people in Western Europe, and a third of the UK's population, owned a mobile phone by October 1999. This growth was driven by several factors, including reduction in device costs, substantial subsidies from retailers and operators, the introduction of pay-as-you-go schemes, and the sale of handsets in supermarkets.

By about 2005, the situation changes radically and we enter the *second era*, shown in Figure 2. The dramatic success

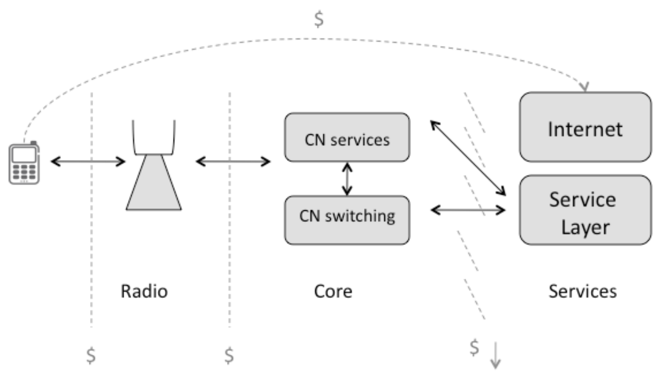

Fig. 2. Mobile Value Chain: Second Era.

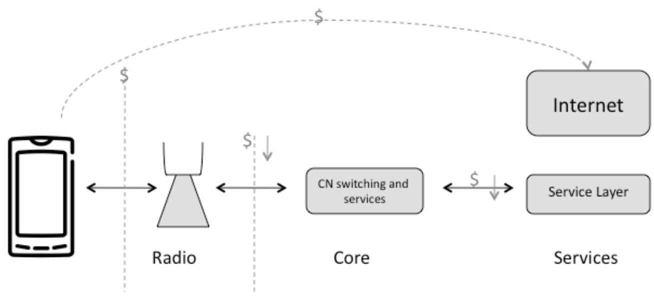

Fig. 3. Mobile Value Chain: Third Era.

of GSM led to a number of new entrants to the handset market, e.g., Samsung, LG, Panasonic. These companies built on the substantial investment represented by the GSM standard specifications to produce handsets at relatively low cost. Reusing experience and components development by companies such as Ericsson, Motorola and Nokia, they could focus on system integration, licensing many hardware and software components.

At the same time, the pace of progress in the semiconductor and associated electronics industry allowed handsets to become far more capable in terms of processing capacity. Not only did this increased capacity make them more capable application platforms, it also enabled native support for packet switched (IP) networking. Other developments also led to increased storage capacity, and potential for much richer interaction through large touch-screens, cameras, and so on. This resulted in the software on the handset becoming increasingly important – differentiation was now between the music player, the web browser, and the email client, rather than the raw radio performance.

This increased important of software was made very visible by the industry's response to the deployment of Microsoft Windows CE on handsets. In response to this, Ericsson and Nokia bought 60% of Psion to gain access to its Epoc mobile OS, which was named *Symbian*. This provided a far more familiar development environment than previous phones, supporting standard protocols such as WAP and Bluetooth, and enabling client software to be implemented in the Java programming language. Platform vendors had to begin to produce open application programming interfaces (APIs) for use by developers outside the owning corporation.

As a result, where operators and vendors had previously made substantial profits across all the boundaries shown in Figure 1, the importance of the service layer and corresponding core network support began to diminish. While operators and equipment manufacturers still created and captured value from both core and radio networks, the value capture in the service layer of mobile networks began to shift to the new entrants. The industrial structure begins to move away from the strong *vertical* integration of the first era, to a more *horizontal* integration. The core services of voice and data still relied on operator services, but applications and content available to consumers increased. From the manufacturers' point of view, entire platforms could be licensed from suppliers such as Ericsson, obviating the need to develop their own expertise in the underlying technologies.

C. Third Era: 2000s onwards

The *third era*, depicted in Figure 3, has continued the developments of the second, at "Internet pace." The core network has evolved into an "all IP" platform, while improvements in silicon technologies have continued to dramatically increase the processing, storage and bandwidth capacity of mobile devices. Coupled with the use of more familiar software platforms, e.g., webkit, iOS (iPhone), Linux (Android) and Microsoft Windows Phone 7, the established value chains of the mobile industry have essentially broken down.

This is also challenging the established role of handset manufacturers whose main value-add is currently in hardware [3]. This has allowed new entrants such as HTC to present a significant challenge to the established players, causing a reconfiguration of the mobile handset industry. All three operating systems currently available on the mobile handset market are developed by computing companies rather than traditional telecommunications manufacturers.

New features and services are no longer introduced by the operator in the core network, after long and rigorous compliance testing, but deployed in minutes after days or weeks of development by individuals or small teams of developers, often in open source communities. The significant companies in this space are now traditional computer software and service companies such as Google, Apple and Microsoft. Their significance derives from their control and development of the open APIs in use, and perhaps even more importantly, their control of the distribution channels for consumer applications, e.g., the Google Android Marketplace, the Amazon Android AppStore and, of course, Apple's iOS AppStore. These value chains have little or no place for the traditional handset and hardware vendors, or mobile operators with their expertise having been standardised and encapsulated within a set of relatively simple to use hardware packages and APIs.

In short then, the three decades of the mobile phone have seen a dramatic shift in power away from handset vendors

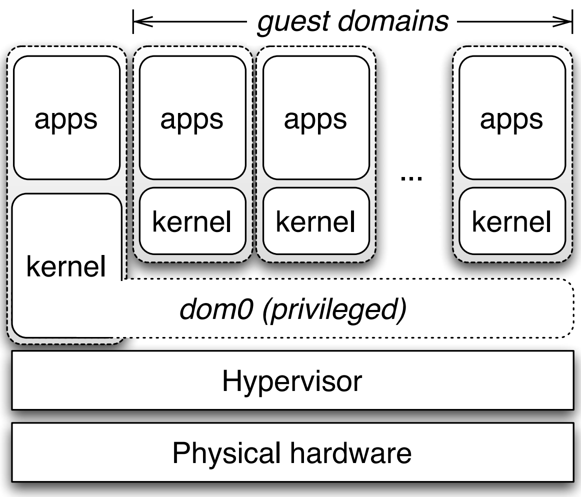

Fig. 4. Structure of a paravirtualized system.

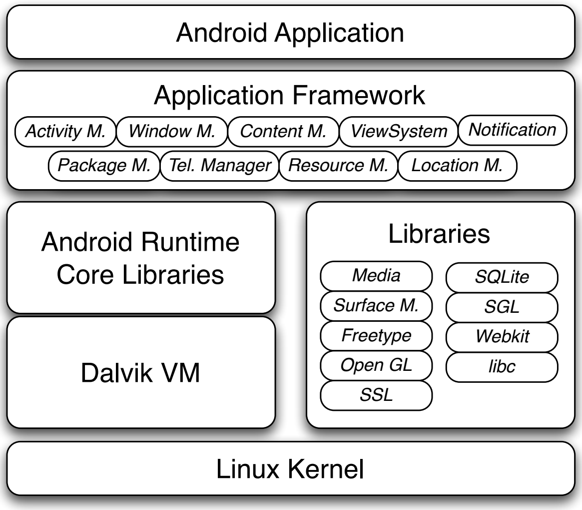

Fig. 5. Overview of the Android architecture, based on *http://blog.zeustek.com/wp-content/uploads/2010/11/android-architecture.png*.

and mobile operators toward certain elements of the computer industry, particularly software and service companies. In the following section we discuss an inevitable technology development with the potential to further open up the mobile platform to development from all quarters, placing its evolution even more firmly on the Internet curve.

II. Continuing Collision: Xendroid

Currently Internet service companies are gaining ground in the mobile telephony value chain and the open standards, extensible protocols and innovation regime of the Internet are increasing in importance. Current trends might suggest that this is a simple reconfiguration of bargaining relationships, with new entrants merely taking the place of the slower-to-innovate incumbents. We believe this view to be shortsighted however: changes taking place now within the computer industry have the potential to further disrupt the mobile network industry.

A key technical development in the computer industry over the last few years has been the widespread adoption of virtualization techniques. Companies such as VMWare and XenSource have demonstrated the value in the many possibilities that arise when a software layer is interposed between the physical hardware and the operating system on which applications execute, e.g., resource isolation, enhanced security and privacy, and very high degrees of multiplexing of virtual machines onto physical hardware.

Figure 4 depicts the general structure of a so-called *paravirtualized* system. The *hypervisor* is a thin layer of system software that abstracts the physical interfaces of the hardware sufficiently that other operating system kernels can be ported to target the abstracted hardware interface. In conjunction with the privileged *dom0* domain, the hypervisor provides basic separation between running *guest domains*, multiplexing them onto the physical hardware. Each guest domain contains its own kernel, typically Linux or one of the Microsoft Windows

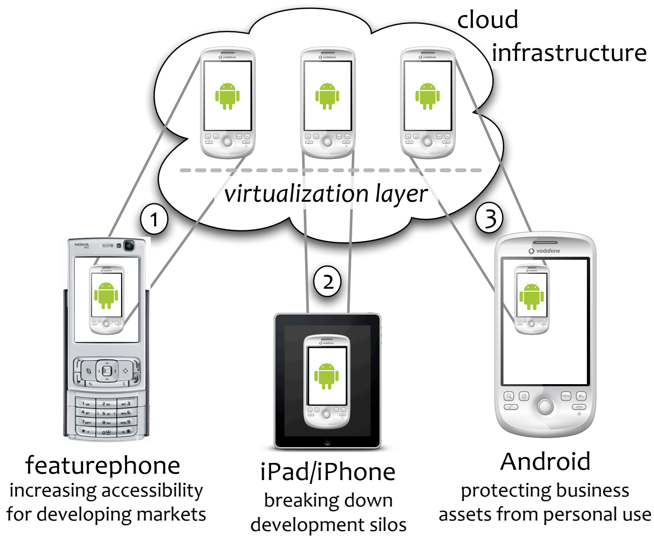

Fig. 6. Overview of *Xendroid*.

Server operating systems, on top of which unmodified applications execute. The flexibility and system-wide efficiency gains that this multiplexing provides are substantial and have had a significant impact on the IT industry at all levels.

In particular, the cost savings and management efficiencies available through this technology have led directly to the creation and dramatic success of cloud computing facilities such as Amazon's EC2. We believe that application of this technology to take mobile phone platforms into the cloud could have another profound effect on the evolution of the mobile handset value chain.

Specifically, consider the Android smartphone platform depicted in Figure 5, enabled to be hosted on Amazon's EC2 cloud computing platform. This is not a far-fetched proposition: other authors have reported systems that migrate code between Android phones and the QEMU emulator running on x86 hardware [4] and customised Dalvik virtual machines running on VMWare's ESX server [5].

Our own foray in this area, *Xendroid*, also leverages the cloud and the Android OS but takes a different approach as depicted in Figure 6. We have used the Android-x86[1] port to run Android directly on Xen using the VNC[2] framebuffer to access the image. Hosting Android as a guest OS over the Xen hypervisor as supported by, e.g., Amazon's EC2 cloud, Xendroid aims to provide both strong sandboxing of resources and access to personal data, and to support access by non-native devices, i.e., iPhones, legacy featurephones.

We believe this will provide a very wide range of benefits, while presenting a range of interesting challenges.

A. Benefits

a) Streamlined management: Of course, the Android image in the cloud need not be simply a clone of the Android device in the user's pocket. For example, business wishing to provide mobile devices to their employees could either provide them as an app to access a specific cloud-hosted VM that they retain the capability to manage; or could permit their employees to migrate their personal device onto the business-provided device by allowing them to access a cloud-hosted VM that clones their personal device. Indeed, the VM image could even be hosted on a standard desktop machine via a hypervisor platform such as XenClient.[3]

b) Increased inclusion: Of course, the client software used to access the cloud-hosted image need not be limited to run on the same platform as the image to which it provides access. For example, client software that provides access to cloud-hosted Android images could equally well run on the iPhone or even legacy feature phones, albeit with a reduced fidelity user experience. This allows for greater shake up of the value chain of the mobile handset industry – Apple may potentially lose part of their control over the applications that end-users are able to run on their iPhones. This would potentially create a new dynamic to the app store phenomena. Also, although mobile phone penetration is growing rapidly in the developing world, the technology typically still lags that commonly found in the developed world. Giving legacy devices the ability to interact and use the capabilities of the latest software should help users in developing nations engage at an earlier stage of technology development, improving the chances of technologies being relevant to their needs. This could both greatly accelerate penetration of smartphone technologies into developing markets, and break down the walls developing between the different app stores/marketplaces.

c) Reduced platform fragmentation: Perhaps the most intriguing benefit of this approach is that it opens up the possible unification of a number of app markets. By presenting a simple interface that would allow incompatible devices to install and run applications from other app stores, e.g., iPhones accessing applications from Google and Amazon's Android marketplaces, the problems of platform fragmentation that mobile device developers currently suffer might be reduced. These problems are already becoming visible, even on tightly controlled platforms such as Apple's iOS.

d) Capability enhancement: Most smartphone and tablet platforms use low-power ARM processors for obvious reasons. By hosting the phone image in the cloud allows them to make use of higher power Intel x86 CPUs, making many more processor intensive applications feasible. The higher speed processing in the cloud also makes possible savings in energy consumption, lengthening battery life: it can be lower energy to use the network to move data and results to and from the cloud for computation [6].

e) Future-proofing: A problem with uptake of modern smartphones is their high cost, which leads them often to be offered subsidised on high duration contracts. In turn, this means users are slow to upgrade, retarding uptake of new technologies and the applications they support. The cloud hosted images enable users to upgrade their experience without having to change contracts or even upgrade their handsets.

B. Challenges

Several interesting challenges arise when trying to virtualize the mobile device in this way. As noted above, these are inherent in the goals of Xendroid, namely providing access to remotely hosted virtual machine image over an intermittent and variable network connection.

a) Remote access: Providing efficient remote access to applications and data stored in a sandboxed virtual machine in the cloud implies some protocol that runs between the device and the remote image. Any such protocol must ensure that data remains synchronised, while avoiding introducing extra latency to user interactive operations and being energy efficient.

b) Heterogeneous clients: In order to support heterogeneous clients, consideration must be given to how user interaction can be obtained and represented. For example, capturing events via keys that are available on Android but not the iPhone. An even greater challenge arises from the desire to support legacy devices such as featurephones: it is likely that existing applications will not map well into such situations. However, approaches such as HTML/CSS support for a range of media, from screen, to print to screen readers, might prove relevant. Finally, typical cloud services, Amazon's EC2 included, are x86 based while most mobile devices use ARM cores. In the long run it is expected that ARM cores may become used in server installations for reasons of power efficiency;[4] the Xen hypervisor is already available to run on such systems.[5] In the meantime, emulation tools are available to cope with native libraries and apps available on the devices.

c) Sensor data management: Xendroid must support the transport of sensor data from the device to the cloud-hosted

[1] http://android-x86.org
[2] http://www.realvnc.com/support/index.html
[3] http://www.citrix.com/xenclient
[4] http://gigaom.com/cloud/a-sneak-peek-at-calxedas-arm-based-servers/
[5] http://wiki.xensource.com/xenwiki/XenARM

image. For reasons of efficiency this had best be done so as to allow multiple apps running in the image, and even multiple distinct images, to share a single stream of sensor data. Further, there is a clear tradeoff to explore between the timeliness and accuracy of sensor data, and the energy consumed in transferring it. Finally, protocols developed here will need to handle intermittent and variable connectivity, and the receiving software in the VM image will need to present this variability in a meaningful way to the consuming applications.

Allowing end-users to store or contribute their sensor data externally to the established value chain regime, however, has the potential to impact greatly on the innovation regime of the mobile industry. An open question in this area is whether this takes control out of the hands of mobile OS providers and into the cloud-based computing operators?

d) Personal data management: Finally, the wide range of sensors available on many mobile devices means that they collect a great deal of very sensitive, personal data. It is quite likely that users will be reluctant to allow at this data to be stored freely in the cloud without protections being put in place against malicious exploitation. An alternative approach would be to keep personal data on the device, instead providing access to it via, e.g., code mobility as in our Dataware system [7].

This again challenges the value chain even though it has only existed in its current form for a few short years. The management of personal data is currently controlled by companies such as Google and Facebook, who are able to store, manipulate and sell such data from users across the globe. Giving end-users the potential to handle this themselves creates a new dynamic within the value chain, whereby they are able to collect micro-payments for their personal data.

III. Approach

An overview of the the Android platform is depicted in Figure 5. It consists of a modified *Linux kernel* with device drivers to handle particular mobile device hardware; on top of this run a number of standard native *libraries* (WebKit, libc, SQLite, etc.) and the Dalvik VM plus some core libraries collectively known as the *Android Runtime*; on top of those run the Android *Application Framework*, which manages activities, packages, content, etc.; and finally on top of that run the *applications*, ranging from the phone to the browser, to any others the user has installed. There are a range of possible approaches to take in realising Xendroid, which we briefly explore below.

Perhaps the most obvious is simple screen remoting, and indeed, this has already been pursued by several commercial players, e.g., Wyse,[6] using standard desktop remoting protocols such as RDP. Their focus is firmly on providing access via mobile devices to existing *desktop* applications, rather than virtualizing the device itself. While straightforward both in terms of implementation (e.g., using an Android VM image running on Xen using the built-in VNC framebuffer support) and in terms of use (e.g., a simple VNC client application the device); and capable of supporting a wide-range of clients, this approach does not address most of the challenges above: the efficiency of the protocol, its ability to handle intermittent connectivity, management of personal and sensor data, or access by legacy devices.

[6] http://www.wyse.co.uk/products/software/pocketcloud/index.asp

The Android platform itself could be extended, in at least two places. First, the platform as instantiated on the mobile device could be customised by, e.g., a custom build of the Dalvik VM, which would be able to interpose on interactions between user, device and sensors. This would enable the most complete low-level control over user and cloud interaction including interception of all events signalling system state, control of private and sensor data, custom protocols to shadow state with a cloud hosted image or images, and so on. Unfortunately it would require obtaining root access to the device to replace such low-level components.

A second approach would be to provide a simple application on the device which could act as a client to a cloud-hosted VM image that ran a customised Android platform. This would enable detailed interception and control of all application interactions in the cloud image, as well as use of customised protocols to manage access to personal and sensor data on the device, and presentation of applications running on the cloud image. These protocols would also be responsible for enabling presentation of cloud-hosted application behaviours on legacy devices, e.g., by using cloud telephony services to generate screen-reader output as a phone call, or simpler text based output as SMS messages.

This latter approach allows many of the challenges of Xendroid to be addressed, the only notable weakness being the inability to easily manage or port existing applications installed on the device to the cloud. As a further extension, one could imagine building applications specialised to such a remote-device cloud-hosted environment. Doing so would allow developers to better specialise applications' behaviour and presentation on non-standard hardware, e.g., legacy phones, and may also enable remoting protocols to be made more efficient by providing hints to current and future use patterns.

IV. Overall Impact on Value Chain – System Integrators and Surplus Value

Platforms such as Xendroid and Dataware challenge the established value chain in a few ways. Increasingly, the value chain with the mobile industry becomes one of a series of value networks co-ordinated by large-scale system integrators (SI) who act to co-ordinate these value networks across a specific value chain. This is illustrated in Figure 7, where the radio, core and service layer network technologies are formed from the basis of a multitude of different firm that are co-ordinated by the System Integrators, for example Ericsson, Alcatel-Lucent or Huawei in the radio network or Google and Facebook in the Internet Services section.

How does Xendroid affect the value chain of the mobile handset industry, therefore? Through allowing the end-users to more effectively control their selection of mobile services and

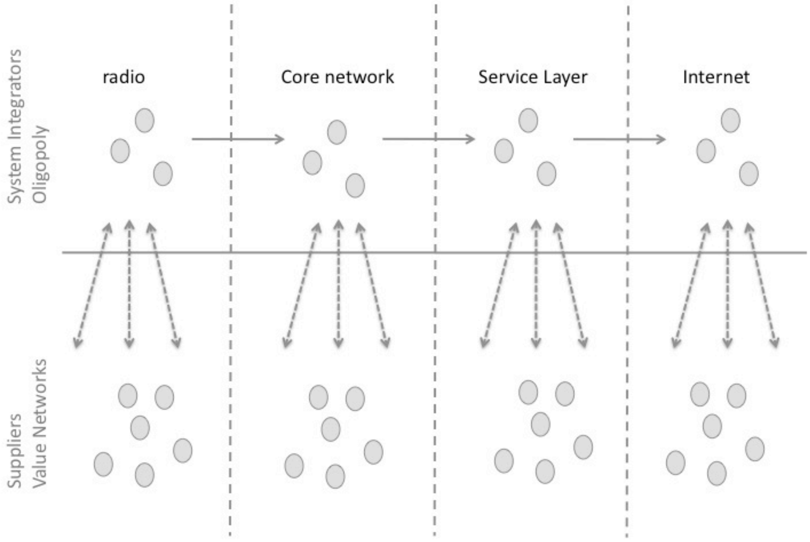

Fig. 7. System Integrators and Ecosystems.

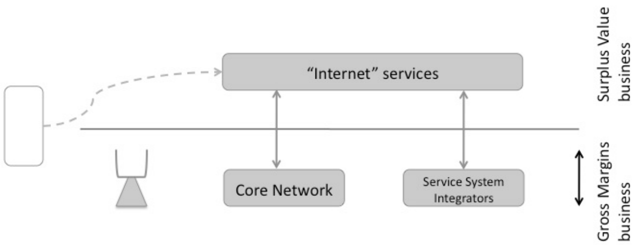

Fig. 8. Impact of a Participatory Value Chain on the Mobile Handset Industry.

applications, the value chain, formed of a multitude of value networks, the value chain becomes participatory. Companies such as Apple are no longer able to execute strict control over the applications that an end-user has on their mobile device. Google may effectively lose access to precious search data that they need to improve their information products on a continuous basis [8].

A further impact, however, is the fact that the mobile industry from handset to Internet service provision becomes an industry driven into two directions. Firstly, those SI developing products such as radio, core and service network technologies are oligopolies driven by lowering cost bases and achieving as high volumes on their products as possible. Secondly, those companies working with technologies that allow end-users greater control over their applications, services and data are able to capture the surplus value of the products created by the oligopoly SI as illustrated in Figure 8.

ACKNOWLEDGMENT

The research on which this paper is based was funded by the RCUK supported Horizon Hub, EP/G065802/1.

REFERENCES

[1] H. Sabat, "The evolving mobile wireless value chain and market structure," *Telecommunications Policy*, vol. 26, no. 9–10, pp. 505–535, 2002.

[2] Value Chain Dynamics Working Group, "Value chain dynamics in the communications industries," 2005, MIT Communications Futures Program.

[3] J. Peppard and A. Rylander, "From value chain to value network: Insights for mobile operators," *European Management Journal*, vol. 24, no. 2, 2006.

[4] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid android: versatile protection for smartphones," in *Proc. 26th Annual Computer Security Applications Conference (ACSAC)*, Austin, Texas, 2010, pp. 347–356. [Online]. Available: http://doi.acm.org/10.1145/1920261.1920313

[5] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: elastic execution between mobile device and cloud," in *Proc. 6th ACM EUROSYS*, Salzburg, Austria, 2011, pp. 301–314. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966473

[6] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Proc. 8th International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, San Francisco, California, USA, 2010, pp. 49–62. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814441

[7] D. McAuley, R. Mortier, and J. Goulding, "The Dataware Manifesto," in *Proc. 3rd IEEE International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, Jan. 2011, invited paper.

[8] C. E. Mulligan, *The Communications Industries in the Era of Convergence*. Oxford, UK: Routledge, 2011.