Source: **UMTS Forum**

Title: **UMTS Next Generation Devices**

Agenda item: **11**

Document for:

Decision	
Discussion	
Information	**X**

1 Introduction

The attached UMTS Forum Report is intended for use by the PCG members only.

"This document is produced by the UMTS Forum, an association of telecommunications operators, manufacturers and regulators as well as IT and media industries interested in broadband mobile multimedia who are active both in Europe and other parts of the world. All members of the Forum have supported the development of this document. However, the views and conclusions expressed in this report do not necessarily represent the views of every member and they are not bound by them. It should be borne in mind that since this report was published technologies and markets have moved on and therefore the contents may no longer be uptodate and the UMTS Forum cannot be held responsible for any errors or omissions. Copyright UMTS Forum."

UMTS FORUM

Report Name: UMTS Next Generation Devices
Report Status: Approved, following distribution at GA30 in Prague (27th-29th October 2003)
Report Originated by: ICT Group

'Report 31 from the UMTS Forum'

UMTS Next Generation Devices

UMTS Forum

This report has been produced by the UMTS Forum, an association of telecommunications operators, manufacturers and regulators .The UMTS Forum comprises of IT and media industries interested in broadband mobile multimedia that are active both in Europe and other parts of the world and who share the vision of UMTS (Universal Mobile Telecommunications System). These are key industry members of the Forum and have contributed significantly to this report. In terms of a technology platform UMTS will move mobile communications forward from today's environment to the Information Society incorporating third generation mobile services that will deliver speech, data, pictures, graphics, video communication and other wide-band information direct to people on the move. UMTS UTRA (Universal Terrestrial Radio Access) is a member of the IMT-2000 family of standards.
This report has been generated by one of the UMTS Forum Working Groups, the Information and Communication Technologies Group (ICTG), which addresses the main services and applications enables for the introduction of UMTS / Third Generation.

Report 31 is one of the family members of UMTS Forum reports that deal with the regulatory framework and the vision for UMTS. Other outputs from the Forum cover technical aspects, economic conditions, and licensing issues.
The views, conclusions and detailed recommendations expressed in this Report are purely those found and expressed during the work of creating this document and exempts National Administrations who are UMTS Forum members from being bound to them.

The following UMTS Forum members contributed in the preparation of this report:

Ecaterina Ganga ecaterina.ganga@siemens.com

John Whitehead John.whitehead@seri.co.uk

Roberto Sannion Roberto.sannino@st.com

Jorge Pereira Jorge.Pereira@cec.eu.int

Bosco Fernandes bosco.fernandes@siemens.com

Special acknowledgement goes to STMicroelectronics, who have contributed extensively to this Report and work of the ICTG. Special thanks goes to Nokia Mobile Phones for good comments and contribution as well.

First edition, revision 2.4

Copyright © UMTS Forum, 2001. All rights reserved. Reproductions of this publication in part for non-commercial use are allowed if the source is stated. For other use, please contact the UMTS Forum Secretariat, Russell Square House, 10-12 Russell Square, London WC1B 5EE, UK; Telephone +44 20 7331 2020.

Web: www.umts-forum.org

All possible care has been taken to assure that the information in this report is accurate. However, no warranty of any kind can be given with regard to this material. The UMTS Forum shall not be liable for any errors contained in the report or for incidental consequential damages in connection with the use of the material.

Table of Contents

List of Figures

List of Tables

0 EXECUTIVE SUMMARY

The success of UMTS will depend on, both, applications, services and the device availability and capabilities to meet the users needs and social aspects which are important issues and this creates the market demands. Today devices have derived from a voice phone and yet its capabilities for delivery of data and image services have reached phenomenal results.

Video telephony, video streaming, audio streaming and concurrent use of several media types are examples of services that will benefit from the high data speeds characteristic to UMTS. As mobility will bring an exciting new dimension to games experience, real-time on-line gaming and entertainment in general will constitute an increasingly big share of the future mobile services.

Operators and service providers have been active in launching a wide range of mobile services. Multimedia messaging mass-market rollout has been rapid. Many operators and service providers see video services and presence services as the way to continue the messaging evolution. Almost everyone has been downloading ring tones and screen icons. Java technology has established itself as the leading application development platform for downloadable mass-market applications. Mobile browsing services are experiencing encouraging growth.

Implementing these services results in quite complex service architecture. Products from traditional telecommunications and information technology companies - which often are designed with different assumptions of the overall architecture - are brought together and integrated into one entity, and interworking between terminal and server platforms becomes increasingly important.

Operators and service providers want to bring new services to market quickly and cheaply. The way to achieve this is to design a platform and architecture following the principles of openness where multiple service can run on a single platform. Public open standards reduce technology fragmentation and enhance interoperability. By using public open standards, market inefficiencies can be reduced and several players can provide cost efficient solutions for the global market.

Operators and service providers are in key position to implement Service Delivery Platforms. Operators are looking for several business benefits from the selected Platform like economies of scale, differentiation, time to market and reduced customer service costs.

Over the years, a wide variety of digital-signal-processor (DSP) core architectures has been developed for wireless applications. They have been optimised for ultra-low-power, high-volume, cost sensitive telecom, digital-servo, and speech applications.

Cellular phone manufacturers are on the horns of a dilemma. On one hand, the demand for multi-standard and multimode besides new features, applications and services, multimedia capabilities continues to grow. On the other hand, cellular-phone technology is also changing rapidly. Providing each feature typically requires a separate chip or, in essence, multiple circuitry systems physically joined on a piece of silicon. The additional circuitry adds cost, takes up space, increases power usage in mobile devices, and increases product-design time. Consolidating many chips' functions onto a single chip is good for wireless devices that must be small, lightweight, and energy efficient. However, adaptive chips may have problems in the marketplace for low-cost mobile devices because they are more expensive to develop than fixed-circuitry chips. Handheld manufacturers now release at lease two major new products lines each year compared with only one just a few years ago. And each provide many functions beyond their core functionality, such as Web browsing capabilities, personal digital assistant functions, navigation/ Location Based, short messaging multimedia / services, and even gaming. Shorter product life cycles and increased functionality create a demand that matches well with the ínternational technology roadmap for semiconductor's prediction of 1 billion transistors on a single chip within this decade.

Mobile computing, now a mature and established field, appears destined to become the dominant computing paradigm. They offer ubiquitous accessibility to companions that support activities of daily living. Appearing in many forms and with increasingly diverse functions, mobile devices now are offering ever-increasing functions that cross boundaries e.g. GPS-enhanced PDAs and games or cameras on phones. Communication technologies such as Bluetooth, WiFi and 3G will make it easier for these devices to communicate with other Handhelds, applications, and computers. The next generation of Handheld will be a computer with phone circuitry built in.

Today's PC functionality will be available in the future smartphone and PDA Phones. Smartphone and PDAs designs will be lighter, smaller display with less keys, lower power consumption and always on functionality compared to PC. CPU performance in PCs will be higher with more RAM/ROM compared to mobile devices. All device categories can benefit from faster UMTS data speeds. The device profile and the user's choice of services will determine the device. So, there is a need to research and closely revisit the requirements of the future. It is crucial that both the IC component manufactures and operators work closer to understand the future requirements and assure that the right quantities of the needed components are made available for infrastructure and terminal manufactures to meet the market demands.

The attitude and the acceptance towards new technology depends on a variety o issues that all need to be taken into account. Some of those may be the fundamental bodily needs, concerns of survival and safety, feelings of belong, desire of success, need of control and power, self-expression, order and understanding, etc. What consumers need, feel and purchase has far- reaching repercussions across the economy. Today, technological advancements has heavily influenced and made our life easier. We should not forget that lifestyle and needs of consumers would play a vital role in determining the concept and design of future devices. Hence, new daily life-related services are expected to emerge by integrating existing functions, well beyond our recognition of mere mobile phones.
What was the motivation for this Report?

- In today's evolving scenario, driven by a high rate of technology innovation as well as a big market push, there is a clear advantage for embedded architecture in evolving towards an open model, where interfaces between the key functional components are public and maintained by standardisation organisations like OMA or JCP(Java Community Process).

- As the number of non-voice functionalities that are implemented in mobile terminals grows at a very fast pace, the embedded architecture of the terminal evolves from the old modem-centric cell-phone model, which was optimised for speech processing, to a new application-centric model, where much more complex optimisations have to be devised in terms of faster processing, memory for applications and for data, 3D graphic acceleration, better display resolution, to mention a few.

- The evolution of energy density in batteries does not keep pace with the evolution of the chips' computational capabilities, hence the need for low power consumption has, and likely will have also in the foreseeable future, a crucial influence on design choices at all levels: from silicon technology up to design choices like hardware vs. software partitioning of complex functions.

- A highly innovative contribution to the evolution of UMTS applications will come from communities of independent software developers. Openness of software architectures will enable the market to highly leverage on such contribution while avoiding at the same time to be locked onto proprietary technologies.

- New high level programming tools, or IDEs (IDE = Integrated Development Environment), will have to be developed so to hide the complexity and the diversity of the various terminals. In this way programmers can be focussed on solving the key problems of their applications without getting lost in details such as management of different terminals' screen sizes, or keyboards, or operating systems, etc…

- Components hardware and software functionalities get directly translated into overall terminal capabilities, so in a differentiated scenario for component evolution it will be more and more important that devices profiles are adopted as standard and are exchanged in the dialogue between the terminal and the rest of the network so to automatically manage interoperability and completely hide the complexity of such a problem to the final user.

- As terminals evolve toward new types, and multiple diversity of them will be spread on the market, technology will be more and more challenged to devise solutions which on one side adapt to the needs of the diverse terminals, and on the other side still enable the advantages of economy of scale so to keep a low price for high volume components.

- Application and services provide real consumer/end-user benefits which can be truly be realized on faster UMTS wireless networks. UMTS bring enhanced user experience for services like faster wireless Internet browsing, faster music downloads and faster on-line gaming. As the number of applications implemented in a single mobile terminal grows, new and smart User Interfaces must be devised in order to enhance usability of the terminal. Services with great user experience is a widely recognised driver for a widespread mass adoption of UMTS terminals. A high degree of innovation in this field will be brought by new solutions at component level, such as for instance MEMS-based sensors that could enable the terminal to sense motion, or cameras that could capture information on the surrounding environment.

- Mobile Lifestyle needs:
  Social & behavioural aspects in terms of virtual communities, economic & behavioural aspects, when it comes to person-to-person and economic needs and when it comes to delivery of content. Than all of this is about seeking

balance in life using the instrument called the “Mobile Phone”. The proliferation of devices are a positive driver for the industry.

1 INTRODUCTION

Today, shrinking circuit features and rising transistor density have spawned the astonishing system on a chip (SoC), which most or all of the circuitry required for, say, cell phone fits on a single IC. These SoCs often contain analogue, RF, and mixed-signal components to satisfy the growing demands of communication applications. Nevertheless, SoC designers require new integrated hierarchical design flows that will mold application-specific ICs (ASICs) and SoCs to allow for innovation of large –scale circuitry.

1.1 This Report

Hardware and software technologies upon which electronic devices are built play a fundamental role in the fast evolving scenario of mobile communication industry. If in fact on one side devices technology must enable terminals to support new features at a reasonable cost, guaranteeing an ever-increasing execution speed and acceptably low power consumption, on the other side strategic decisions taken at the level of devices technology can influence the market up to the level of enabling or not entire business models.

In this report we review the fundamental aspects of UMTS devices and we identify and study the economical and technological drivers of their evolution, paying a special attention to the way certain choices made regarding such evolution can affect, and be affected by, the whole UMTS ecosystem.

Chapter 2 outlines the technological challenges that the devices industry will be facing in the coming years in terms of both hardware and software design and development, while chapter 3 outlines what are the major constraints and requirements that the market will dictate.

Chapter 4 explores in great detail the idea of Open Embedded Architecture for UMTS devices and discusses both the technology and market implications of such an idea. The concept of openness referred to embedded architecture is introduced and discussed in 4.1, and various market scenarios disclosed or implied by open and closed architectures are analysed in 4.2. In chapter 4.3 a model of embedded architecture is introduced as a sort of vertical section across hardware and software layers. Such model is used in order to discuss the important issue of interfaces, which is central to the definition itself of an embedded system's openness. The highly strategic problem of partitioning is discussed in chapter 4.4, where a high-level description of the main functionalities of a UMTS terminal is given and used as a reference. Partitioning is discussed in the domain of hardware implementations, in that of software implementations, and at the border between the two domains, where for certain functions a hardware or software implementation is the result of a precise design choice.

2 HOW MUCH MORE DIFFERENT ARE 3G DEVICES?

2.1 Devices

Cellular Devices are extremely portable, self-contained information management and communication terminals. With their increased mobility, limited memory and processing power, and small display sizes offer unique challenges- and opportunities- for design. This just because of their cellular historical in heritage.

Although, today the 2.5G Devices are already very powerful and have sprouting keyboards, cameras and MP3 players integrated, Convergence of services will ask to adhere to open source standards, provide broadband but secure transmission, and be optimised to handle, video, text, and voice simultaneously. And although, Advances in Digital Signal Processors (DSPs), software-defined radios, smart antenna technologies and Embedded microprocessors are evolving to support these market requirements. There is nevertheless, a need to design a real-time operating system by taking advantage in technology at many levels including:

- Improvements in analogue and digital-radio communications including coding, encryption, multiple-access schemes, and modulation methods.
- Enhancements to operating systems and programming languages and software development tools.
- Enhancements in digital-logic integration with the eventuality of migration to molecular/ nano-technologies not just anticipated but expected.
- Dramatic improvements in storage systems
- Evolutionary improvements in energy power systems e.g. New Battery technologies.

The packet transmission interfaces to the network will also call for changes in the Device and support of data protocols such as IEFT “SIP” and “IPv6” besides the complexity of 3G protocols defined by the standards.

3 THE CHALLENGES

3.1 RF

Future UMTS wireless terminals will require innovative RF circuit techniques to meet increased performance demands under existing cost, size, power and regulatory requirements. One of the challenges for RF designers will be to define the optimal architecture of both the radio receiver and transmitter allowing a seamless switch between 3G and 2.5G standard for the user of the terminal.

3.1.1 RF Receiver

Two architectures look to be promising for UMTS RF receivers of mobile terminals: the direct conversion receiver also called zero intermediate frequency (ZIF) and the very low intermediate frequency receiver (VLIF). Both architectures have pro and cons, but in both cases the most important parameters of the receiver to fulfil the UMTS regulatory requirements are sensitivity and selectivity which can be translated from a technology standpoint in noise figure (NF) and linearity of the RF front end (LNA and mixer).

Another relevant parameter is the transition frequency (ft), or the frequency at which the gain of the transistor (beta) is equal to one. Ft depends on the silicon process technology and should be as high as possible.

Silicon Germanium BICMOS process (SiGe BICMOS), which has an ft of about 40 GHz, is now commonly considered as the process presenting the best performance compromise among transition frequency, NF, linearity, and power consumption for RF receiver front-end of UMTS terminals.

The two previously cited architectures also bring the advantage of shifting most of the channel selectivity to baseband (ZIF) or close to it (VLIF) allowing the designers to use dense CMOS processes for this function including also analogue to digital converters.

So finally, choosing the best architecture with the right technology partitioning giving the best cost/power consumption trade off will be the first challenge of the UMTS receiver RF designer.

3.1.2 RF Transmitter

Among the different challenges of the UMTS RF transmitter, the most difficult one is the power amplifier stage. Due to its HPSK modulation scheme, WCDMA signal combines both phase modulation with amplitude variation. So to avoid distortion of the RF signal and fulfil the regulatory requirements, the power amplifier has to operate in linear mode. This mode of operation is reducing the efficiency of the power amplifier at both maximum output power, and even more at lower levels of output power. That's why in the early generations of UMTS terminals the RF power amplifier should be the main contributor limiting the battery lifetime.

In the future, RF designers will have to find clever efficiency enhancement and linearisation techniques to solve this power consumption issue. In the same way as for the receiver, mixing in a clever way different technologies for RF power and RF signal processing to get the best linearity/cost/power consumption trade-off will be a challenge for RF designers.

3.1.3 RF Synthesizers

As already stated, UMTS terminals will have to operate in multiple bands from less than 900MHz to more than 2GHz and in multiple modes (TDD, FDD, …). Development of fast locking frequency synthesizers and integrated voltage controlled oscillators with wide frequency tuning range will be required.

3.2 Baseband

Baseband processing for code-division-multiple-access (CDMA) handsets and base-station systems is extensive and complex. Currently, DSP serve the purpose of providing a Baseband platform for 2 and 2.5G Devices. For 3G Devices more power is required to meet the complexity of the standards. Therefore, Chips that will provide 10 to 100 times the processing power and speed of the best available solution based on legacy DSP architectures and yet reducing the cost while providing the flexibility to allow software upgrades to new revisions of the standard and accelerating development will be needed.

The new ARM's microprocessor core for use in Baseband signal processors for 2.5G and 3G Terminals are expected to increase the capacity of wireless network cells by two and four times.

3.3 Application

As data-rates increase with the introduction of new wireless communication-standards, such as GPRS, EDGE and ultimately UMTS, mobile terminals shift from being voice-centric to being data-centric. This development is fuelled by three factors: ever higher user-expectations with users demanding advanced services, ever more sophisticated technology offering new possibilities and last but not least huge economic profits that are predicted to lie in value added services.

Besides the coding and decoding steps carried out by the Baseband processor, data (i.e. files, video-/audio-data other than digitised voice) that gets transmitted on the air-channel, has to be processed in the mobile terminal. This job gets done by the so-called application processor, a piece of hardware (usually a microprocessor) that runs applications as well as the operating system that manages the phone as a whole, and takes care of I/O-operations. With the growing importance of data processing, it is obvious that also the application-processor assumes a more important role among the components of the mobile terminal.

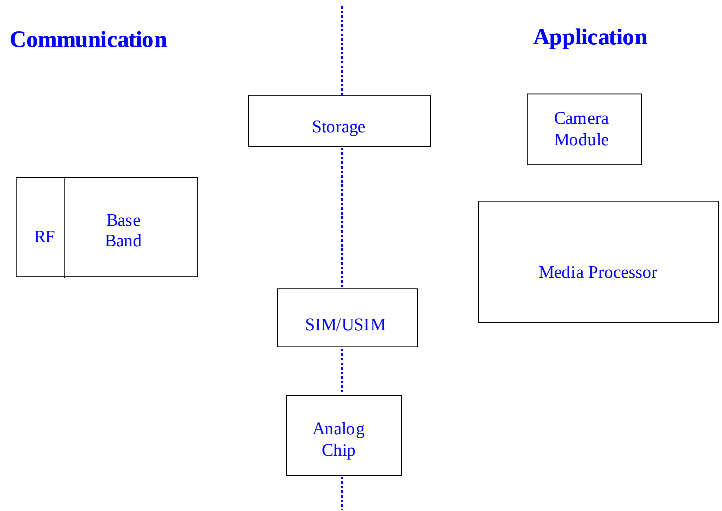

Figure 1 – Split of Applications and Communications

As with any other mobile-terminal component, also in the context of the application processor energy-efficiency is paramount. One of the main challenges is therefore to find a good trade-off between the two antagonists low energy-consumption and high processor-performance. The latter is critical, since future mobile terminals are, e.g., expected to run a Java Virtual Machine (JVM) at reasonable speed and allow users to run ever more sophisticated applications.

Another challenge results from the fact that the costs of a new chip-design have risen significantly due to the complicated process of producing masks for modern low-micron silicon technologies: in order to make profits, chipmakers strive to achieve high-volume sales of components. Hence, the application processor should be designed to offer a high grade of scalability and flexibility in order to satisfy diverse systems' needs and thus maximize reusability of its design.

Finally, there are challenges concerning the application processor's hardware- and software-interfaces. The earlier have to support higher data-rates as processing-power increases and new applications demand, e.g., faster memory access. Various proposals for high-speed inter-component busses have been made and will be implemented in future terminal-generations (see chapter 5.3.4).

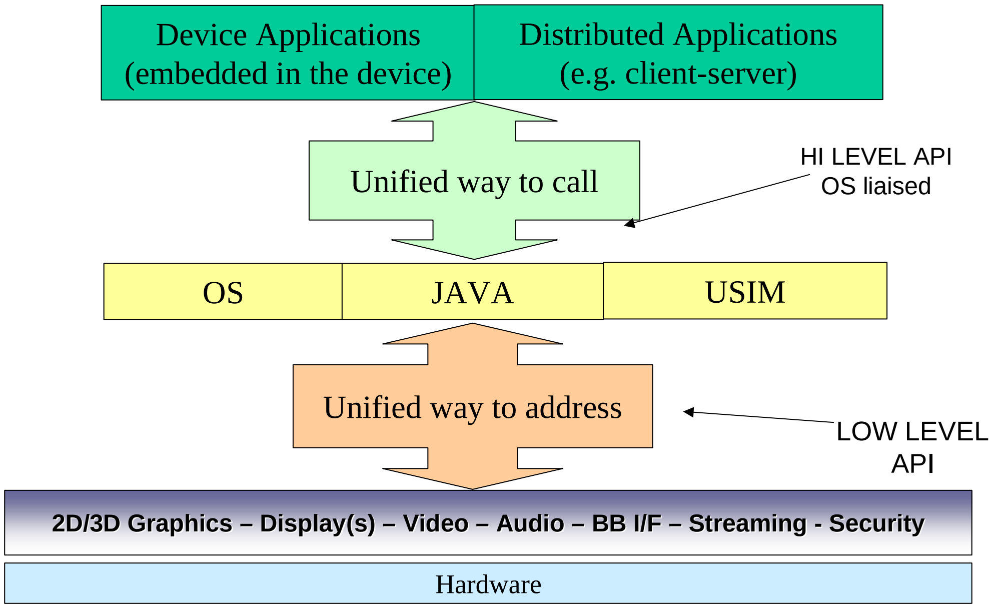

Figure 2 – Software Interfaces

Note: From that the normal application developers cannot write applications directly on application chipsets. Developer tools and APIs running on an operating system like Symbian provides abstraction layer between hardware and software.

It is largely agreed on the fact that for UMTS to be a commercial success, there has to be a broad range of applications that offer functionality and service, which the end-user is willing to pay for. In order to maximize the developer community and thus ultimately the pool of software the end user can choose from, application-processors should appear identical to software, i.e. possess a common programming-interface. Programs would thus be compatible with all terminals rather than just the ones with a certain type of processor and there would be one great market for application-software rather than many smaller ones, allowing software companies to benefit from economies-of-scale (see also chapter 5.2). Another important challenge for the application-processor in this context could be described as ease-of-use: the easier a processor can be programmed, the more people can do it and the less time they need. The challenge is to hide complex structures from the programmer and allow for efficient and fast programmability.

3.4 System On Chip

System On Chip (SOC) devices integrate in a single component various IP blocks, and even different technologies, so to achieve the highest level of optimisation in the silicon implementation of complex hardware/software functionalities. SOCs target high volume market devices like mobile phones and wireless multi-media terminals, where the requirements for low-cost, low power dissipation, and high performance, along with optimal utilization of board space, are particularly stringent. In its 2001 Edition, the International Technology Roadmap For Semiconductors (ITRS) describes a SOC as “a yet-evolving product class and design style that integrates pieces of technology from other system driver classes (e.g., memory, microprocessor, analogue/mixed signal) into a wide range of high-complexity, high-value semiconductor products.”

The principle goals of a SOC-component are low cost and a high level of system integration. A central challenge for SOC-manufacturers is thus to develop and maintain a selection of reusable hardware IP-blocks (IP – intellectual property), in order to minimize the part of a new SOC-product that has to be designed from scratch. The integration of these blocks requires standards and methodologies for IP-description, IP-interfaces and IP-testing. So far, SOC manufacturers have relied on proprietary IP-standards, making it difficult to include IP-cores of different vendors into one SOC. This could change in the future, if a common IP-interface and -description platform emerged, which would create a market for IP-blocks and allow chipmakers to offer SOC's, integrating the best IP-blocks from different vendors. For a general discussion of open interfaces, see chapter 5.

From an economic point of view, the development of a new SOC component amounts to a more complex and risky investment decision than, for example, the development of a new microprocessor. Both systems have to yield high sales-volumes, since producing the masks necessary for the production process is extremely expensive. The functionality however, a microprocessor has to offer, is clearly defined, whereas the functionality of a SOC is – per definition – completely up to its producer. Including a useless functional-block into the SOC can jeopardize the whole product and result in a huge economic loss if remaining unnoticed until it’s too late. Finding the right system partitioning, which gives the customer a maximum-flexibility without including excessive functional overhead, thus results in a further major challenge, SOC-manufacturers have to face.

The integration of system-blocks that need different production-processes results in a further challenge: “the heterogeneous integration of components from multiple implementation fabrics (e.g., RF, re-programmable, MEMS, Opto-electronics, and software).” (ITRS 2001) This is by no means trivial, since production-process complexity inevitably increases, resulting not only in significantly higher component-costs but also in daunting technological challenges. In the end, it is however important that a SOC, that integrates different functional parts, remains more cost-effective than a multi-die solution.
In the context of mobile terminals, it is important that the SOC, which takes care of all application-related processing, costs little in terms of money as well as power-consumption. Future SOC’s should thus also contain an intelligent power management, which is able to selectively switch off different parts of the chip that are currently not needed.

Summing up, the main challenges concerning SOC’s are:

- Reuse of IP-blocks
- Standardizing IP-interfaces
- Partitioning the system

- Integration of heterogeneous technologies
- Power-management

3.5 User Interface

As technology evolves and wireless terminals become ever more powerful, the importance of their user-interface increases significantly. For 2G cellular phones, which offered little more than voice-communication and short messaging, it was sufficient to provide the user with a simple black-and-white display and a traditional telephone-keyboard. Amazed by the new technology of wireless communication and its astoundingly compact and miniaturized devices, users were willing to put up with the inconvenient way of reading text on small displays and entering it "triple-tapping" the letters on their small phone-keyboards.

With the advent of high data-rate mobile communication devices, new kinds of information, like images or big text files, will have to be displayed on and entered into mobile terminals, hence creating the need for a more sophisticated user interface. Ease-of-use must be the mantra while designing the hardware and software components that make up this interface.

The new mobile communications devices user-interfaces range from medium sized smartphones to larger communicator type devices. To better support advanced applications for data and imaging, these user-interfaces will be deployed on open software development platforms based on open operating systems such as Symbian OS, which supports different user-interfaces such as smartphones (Series 60 Platform), PDAs (Symbian UIQ), and communicators.

3.5.1 Hardware

3.5.1.1 Input

There are several ways to enter data into a wireless terminal. Traditionally, users have typed in numbers and characters using the phone keyboard, which offered between 14 and 17 keys. Important factors for such devices, as they can be found on the market today, are cost, feeling and reliability. Especially the latter is becoming more important as user-terminal interaction is occurring more often (games, mobile Internet…).

Operating systems like Symbian OS should provide terminal manufacturers UI and input method (one hand, keyboard or pen) choices so that terminal manufacturers can differentiate and address the needs in multiple and user segments.

Full-fledged keyboard solutions are currently available as add-ons for PDA's. These keyboards can be folded or even rolled up and consume little space, thus offering a true alternative to keypads. In both cases however, internationalisation remains a crucial issue in order not to fragment the market for localized and internationalised content and services.

Already available for PDA's, it is likely that also in future mobile phones touch screens in combination with an input-pen will play a major role. An example for a "smart phone" that incorporates such a pen-input touch screen is the Sony-Ericsson P800.

While this product also offers a traditional phone-keypad, there will also be devices, which allow the user to dial by using his pen to choose the numbers (aka. soft-keys) that appear on the touch screen.

In order to get also visual input-data, next generation mobile terminals will incorporate a camera-module. This module will most likely consist of a CMOS sensor, which consumes much less power than a CCD sensor and still delivers pictures of reasonable quality. First examples of this kind of terminal are the new Nokia 7650 and the above-mentioned Sony-Ericsson P800.

In the simplest case, the visual data captured by the terminal's camera gets treated as an image. More advanced systems, requiring dedicated hardware support, could process the image and eventually read barcodes or handwriting.

Increased signal-processing performance will allow future mobile terminals to rely more heavily on a voice-centric user interface, thus allowing the end user to control his terminal "hands-free" by issuing spoken commands. Simple phones could thus eventually be realized with very small form factors since the keypad and the display could be omitted.

In general, the signal processing related with the above-named video- and audio-applications can also take place on a remote server, thus relieving the mobile terminal of costly (in terms of power-consumption) DSP-calculations. For mobile audio-applications, there already exists a workgroup (called Aurora) within an ETSI Technical Committee (STQ), which develops and standardizes a system for distributed speech recognition (DSR). The standard defines a front-end system (i.e. mobile terminal), which performs audio-signal feature-extraction and transmits only the feature vectors to the DSR server, where the actual speech-recognition takes place or an audio-signal gets reconstructed from the feature-vectors.

A speech-recognition system with the purpose of acting as an operating-system user-interface will however have to be based entirely on the mobile terminal. Hence, a challenge in this field will be to come up with a resource-sparing system.

3.5.1.2 Output

The display clearly is the most important component on the output-side of a UMTS terminal's user interface. The end user will rely on it to consume high information content multimedia and high-bandwidth video imagery promised by the UMTS standard. Display-size and –quality will be a major differentiation-element for terminal-manufacturers and will play a crucial role in the process of making end users accept and consume new mobile services like MMS or mobile Internet. At the same time however, the display is a major power-consumer among the various components that make up the terminal, thus forcing terminal-manufacturers once again to find a trade-off between performance and power-consumption.

As of today, Passive Matrix Displays dominate the market thanks to their lower power consumption compared to active displays. Traditionally, cell phone displays have been realized as monochrome STN LCDs (Super Twisted Nematic Liquid Crystal Display).

Their electrical characteristics (switching speed, voltage response) were sufficient for text and simple graphics under 2G and given that their power consumption is low and that they are very cheap to manufacture, their current prevalence is explained easily.

Colour images and high-bandwidth multimedia content will make it necessary for displays to support colour-graphics and higher resolutions and contrast. Active Matrix Displays (TFT's or TFD's) seem to be able to satisfy these objectives and still stay within the terminal's power- and cost-constraints. In fact, they are already being deployed in current high-end smart phones and PDA's.

A new technology that should bring further enhancement of mobile displays is OLED (Organic Light Emitting Diode), pioneered and patented by Kodak. OLED displays consist of self-luminous pixels that do not require backlights used in LCD's, thus eliminating the need for bulky and environmentally undesirable mercury lamps, resulting in a thinner, more compact display. Besides the improved sharpness and brightness, an OLED display also offers a larger viewing angle (up to 160 degrees) even in bright light.

Besides the traditional display, the future might also bring mobile Terminals with an included projector. Siemens has presented a first prototype of a 'Mobile Beamer Phone', capable of projecting A4-Size presentations directly from stored on the phone, already during the 2002 Cebit.

Another component on the output side is the loudspeaker. Being one of the biggest components of a mobile phone, it currently limits further miniaturization and drastic new design. In order to change this, Siemens has teamed up with NXT plc. (UK) developed a prototype of a 'Display Speaker Phone', which integrates a loudspeaker into the phone's glass display. The integration will open-up more space for creative designs and technology without sacrificing the sound quality: according to Siemens, Nat's solution outplays conventional loudspeakers.

3.5.2 Software

The most important characteristic of a mobile terminal's GUI (Graphical User Interface) is ease-of-use. Future UMTS-terminal users are today's mobile phone users and cannot be expected to have profound experience of desktop computer operating systems. Simply porting a PC-OS GUI to the mobile terminal will thus not result in an acceptable solution.

The GUI of a UMTS terminal should provide for a logical system structure, fast access to the most commonly used features and services, and the possibility to personalize the terminal's look by using wallpapers or skins.

In the short run it seems inevitable that the logical structure, which each PC operating system is based on (i.e. "files" that are stored in "folders" on a "desktop"), shall also be used to organize data on a mobile terminal. In the long run however, a paradigm-shift has to occur, in order to build computing systems that really satisfy the end users' needs. Yale computer scientist David Gelernter states in "The Second Coming – A Manifesto" (published on www.edge.org) that people don't want to be connected to computers, "they want to be connected to information." In his vision of the future, "people are connected to cyberbodies – self-contained, neatly-ordered, beautifully-laid-out collections of information, like immaculate giant gardens." While Gelernter's visions certainly are rather esoteric and transcend current computing reality by far, they are useful reminders that current computing systems' information-structure is not god-given and can thus be altered – a fact, often forgotten by computer users and developers, who are used to work around the flaws of badly-designed software.

The Series 60 Platform is a complete Symbian OS based smartphone terminal software product. The large colour screen size, easy to use interface for single-handed navigation, and suite of applications makes this software ideally suited to support the new mobile services, such as rich content downloading and MMS. In addition, it supports advanced file and data handling.

3.6 Security

Public open standards reduce technology fragmentation and enhance interoperability. By using public open standards, market inefficiencies can be reduced and several players can provide cost efficient solutions for the global market. Security technologies should also be openly standardized, as the most secure solutions are commonly based on security of the technology standards and not to the secrecy of the technologies.

Security is an integral part of the mobile phone and thus cannot be treated as a separate add-on feature. Security affects all parts of the mobile phone and to enable secure mobile phones, the security needs to build into the platform. Even the enhanced security of SIM cards, or other peripherals like secure MMC, cannot be used in their full extent as long as the platform is insecure.

The security in mobile phones means different things to different people. Some people address security from the communication perspective. In this view the focus is on the security of the data exchange. Participants exchanging data need to be authenticated and their actions need to be authorized. Authentication can be based on for example the phone number, user name and password pair or to digital certificates. All data exchanges also need to be encrypted to protect transferred data from 3rd party access.

Different standards and vendor specific APIs specify the ways of how to authenticate other parties and how to encrypt all communications. For the developers the problems emerge when all these standards and APIs need to be put together. Some vendor specific APIs specify how to access the encrypting hardware, some standards specify specific ways to access the authentication information, and so on. It can be tedious for the developer to create applications for these platforms, not to mention the problems in porting the applications to other platforms.

Another way to look at the mobile phone security is to focus on the applications running on the platform. The problem is then how to constrain applications so that they can only access the platform features that they need and that they are restricted to the specific space where they are running.

This application specific approach is typical when developers are creating applications that have to process sensitive information. There are some coding methods and techniques that developers can use in order to secure their applications, but they are still highly dependent on the security of the underlying platform.

Security will be a key concern for the designers of next generation mobile terminals. The main reasons are twofold:

- UMTS terminals will allow users to update their applications, or to download, install and run new applications after the terminal has been bought. There is a risk that malicious code could be illegally installed, even without the user's knowledge or consent, and create an economic damage to the terminal owner and / or to the network.

- Various m-commerce applications will require the mobile terminal to actually become a tool that enables economic transactions. Security measures will have to make it impossible an illegal use of such an economic tool either by its owner or by somebody else who could steal the terminal and then use it to damage its owner.

Security features will have to permeate the whole electronics in the terminal. If in traditional voice-oriented terminals a SIM card connected to the telephone's modem device is sufficient to securely manage the various aspects of identification, authentication, and charging, in UMTS terminals new architectural solutions will have to be devised so to guarantee security for a number of new applications, most of which are not even known at the time of fabrication of the terminal, but will be installed later as software programs.
UMTS terminals will need to be resilient to software attacks such as, for example:
A malicious code gets hold of the display and make it show to the user a price for a good that is being bought that is different than the one that is being authorized (trusted display). Similar consideration for the keyboard

In the more distant future not only will software download be targeted, but also dynamic reconfiguration of the hardware will be addressed, so that the result is that best suited to the user requirements. The consequences of moving to this type of system are enormous. The security checking procedures in the downloaded software, and in the supporting software system and environment, will have to answer the following questions:
It would be hard to derive a 'solution to security' that could be deemed all-encompassing at the outset therefore:

- If possible, separate the application downloadable software area from the air interface area.
- Incorporate robust, less 'power hungry' air interface encryption techniques.
- Adopt a 'hand-shaking' man–machine interface (question and answer type procedure verifying that download content and source are trusted by the user).
- Networks should continue to adopt a policy of minimum transfer of non-encrypted personal information (for example, International Mobile Subscriber Identity and equivalents thereof should not be transmitted unnecessarily).
- Avoid the use of static ID where possible (for example, use dynamically assigned IP addresses for Internet sessions).
- Provide extremely secure minimum back-up (fixed) configuration software with good default security settings (to ensure that complete system lock-out cannot occur).
- Provide (updateable) virus capture software.
- Provide content encryption of stored data and all removable memory (this will also ensure integrity if mobile devices are lost/stolen).
- Use secure user authentication (PIN numbers/pass phrases etc.).
- Encourage security consciousness. Provide clear/ simple guidelines for using terminals in a secure manner.

There is much to research in this area and some of the scenarios currently proposed look large and complex for mobile devices. Everybody relies on mobile phones, and their reliability cannot be compromised (simply for the sake of technical flexibility), especially if prices to the consumer are likely to increase. A balance will therefore have to be struck.

- Watermarking

Digital watermarking is a way to embed copyright or other information in the data of a media stream. Techniques for watermarking vary widely from one solution to the next. The simplest form of watermarking is simply adding an identifying image to each frame of a video stream. Television stations frequently use this technique to insert a semi-transparent logo in a bottom corner of their broadcast. More sophisticated forms of digital watermarking are imperceptible to the user watching or listening to the content.

The watermark information survives a variety of image processing processes, good quality photocopying and cropping too, hence the watermark, a geometric pattern encoding the secret bits, is holographic. Inspired by this property of the geometric watermark pattern, the question to clarify was how to represent images in ways that enable their reconstruction from arbitrary portions/packets of their representation. This could be very useful for dealing with images in distributed environments that cannot ensure that all packets of information about the image will indeed be made available to the user, and the order in which the packets are made available is arbitrary. The holographic representations enable image recovery with progressive refinement insensitive to the order of packet arrival. Along with Digital Rights management (DRM) systems promise a new approach to information management and guarding of content will be implement and has an impact on the Device.

DRM's protect digital content from unauthorized distribution and copying, and are therefore an important catalyst in the dissemination of digital content. This also valid also for:

- Multimedia services will have compelling content only if properly protected by DRM.
- Digital Rights Management systems (DRM) provide an alternative and superior means to ensure fair compensation to right holders for the use of their works.
- DRM systems are also key for the Internet-based distribution of content and the offering of attractive broadband services.

3.7 MEMS

Micro-electromechanical systems (MEMS) may become key component of radio frequency devices, particularly in the mobile market place. In filters, they may lower radio size and power consumption while increasing sensitivity. In switches, they could herald the construction of cheaper, electronically steer able antennas for radar and communications applications.

MEMS are electromechanical devices with tiny moving parts. They can be built using IC-compatible materials, such as poly-silicon allowing their integration on a silicon chip side by side with semiconductor circuits.

Mobile wireless devices come up hard against three problems: power consumption, sensitivity and physical size. Improving power and sensitivity gives devices longer battery life and extends their range. Better sensitivity is vital to working in an increasingly crowed radio spectrum. Reducing the size opens up room for larger batteries and the ability to offer more features in one Device.

A prime example of a space-saving device is the MEMS resonator filter, which uses mechanical vibrations to filter RF signals. MEMS resonators can be made at the micrometer scale and could someday replace surface acoustic wave (SAW) filters, which are several millimetres in size.

Many MEMS switches and filters could be integrated on a single chip to create a parallel array of resonator filters, all with different pass frequencies. A MEMS switch would precede each filter, and channel selection would be achieved by opening all switches except the one connected to the appropriate filter. The array would filter an RF signal before it reached a low-noise amplifier (LNA). The advantage of this architecture is that it obviates the need for tuned amplifiers, which makes circuit design much simpler. It also ensures that the LNA does not waste energy by amplifying out-of- channel signals. The architecture could reduce interference from adjacent channels and simplify the design of multimode handsets that operate across a range of frequencies.

Although MEMS have their advantages, several problems must be overcome before these devices reach the low-cost, mass production stage. The challenges include minimizing trapped charges in dielectric materials, which interfere with the electromechanical operation of the devices. Another problem is “stiction”, where a thin layer of condensation can interfere with the mechanical action of a device. Also developing appropriate chip-packing technology.

Prototype radio-frequency micro-electromechanical system (RF-MEMS) switches promise high reliability, low power consumption, low actuation voltage and compatibility with system-on-chip (SoC) fabrication techniques. The reliability of the prototype was demonstrated by carrying out more than 10^9 switching cycles without degradation. The RF characterizations yield insertion loss of just 0.18 dB and isolation of 57dB at 2GHz, which is in the frequency region of mobile-phone operation. The cost is currently a major issue because they require more than ten levels of masks and a complex process flow. Optimisation is on its way and RF-MEMS should be on the market by 2005/ 2006. These could also support various frequencies with compact and flexible structures. The next step would be the intelligent support of SDR.

3.8 Robustness

There is a Critical need of Robustness at the “terminal” level and some of the attributes of a robust system will be that it:

- Works correctly
- Copes under stress
- Continues to be usable under all conditions

It is well understood that devices would have a large display as possible and eschew voice commands and pen-based computing for a simple GKOS QWERTY Keyboard. It should nevertheless fit into the consumers jacket pockets or ladies purses.

3.9 Memory

Memory (on-board cache, synchronous DRAMs (SDRAMs)) within the phone is never enough and therefore, there will be a need for external memory. Flash memory is poised to make high volume entry into the cell phone industry. The new generation of multimedia phones will have memory expansion slots and, as a result of the strict design parameters flash vendors have introduced, a new generation of mobile flash storage formats. These formats include Memory Stick Duo, miniSD and reduced-sized MultiMediaCard ('RS-MMC). Today, Memory Sticks and miniSD (SanDisk) are available with 256Mbytes but it will not be long before low cost 1Gbyte memory will be on the market.

4 THE CONSTRAINTS

4.1 Power Consumption

- Power management

Power management is important, to allow the device to function at peak performance when mobile and maintain extended service lives. Power consumption increases with complexity of simultaneous sessions of a variety of applications and puts an increasing demand on rechargeable-battery technology. In order to avoid this, a Battersmart, which is a series of hardware and software technologies, are incorporated on all device platforms. The main goal here is to reduce power consumption to the lowest possible level, for a given handheld device application environment.

Subscriber battery-life expectations will not be reduced with increased handset functionality. As a result, applications that traditionally run on the client device may be best designed to intelligently balance data-storage and processing loads between the client and an intermediate server.

4.2 Antenna-Diversity processing cuts power consumption

If a terminal is within 10-15% area of the cell that support conventional high-speed data transmission, the diversity-processing circuitry is switched into power-saving mode for maximum efficiency. Elsewhere in the cell the terminal would operate at the highest data-transmission rate using diversity processing. The result is a terminal that can download data so quickly that it expends just 20% of the energy required by a conventional handset.

4.3 Batteries

Cell phones are useful devices only as long as they have enough energy to make calls. Battery life and size are therefore, important issues and needs to be calculated according to the application.

Current 3G Batteries offer 180 hours continuous dynamic stand-by time and 250 hours static-standby time. These are considered way to low for the potential of 3G.

Most digital cellular phone offer two to three hours of talk time and up two to three days of standby time on a single charge from a fully charged lithium-ion battery. But what if the battery is nearly spent? Then people rush to recharge, connecting the phone to a wall socket with an ac adapter or to a car's 12-V socket (cigarette lighter) with a dc adapter. If no socket is handy, there is now another option before finding a pay phone: an Instant Power call from Electric Fuel.

Both the battery and re-charger cartridges are based on zinc-air fuel cell technology. Strictly speaking, though, neither is a fuel cell since the electrode material (zinc) is consumed and, moreover, not replaced. What is replaced is the entire re-charger cartridge, whereas in a true fuel cell only the zinc anode would be replaced. Many manufacturers of such devices have already announced a replaceable-cartridge re-charger for personal digital assistants.

The need to increase talk time in cellular terminals is the challenge for the Industry. Especially in the context of simultaneous sessions of data in multimedia terminals such as picture messing, video conferencing and internet surfing, high-resolution colour displays, cameras, music players are power-hungry and consumers will not accept the fact that application such as these could drain their mobile-phone battery too quickly.

The users want terminals with batteries that have to be charged less frequently. Lower power consumption means that battery size-and ultimately handset size-can be reduced, which also pleases consumers. Data-intensive 3G services need a new approach to power management because traditional low-power design principles are intended for voice communication and only focus on three areas. The first area involves the reduction of active power consumption using innovative circuit designs and advanced semi-conductor production techniques. The second involves reducing standby power by slowing down clock frequencies or shutting down entire blocks of the circuit when they are not in use. Third is the structure and way the software is written. Also, an increased download time means higher energy consumption. The designer, this represents a classic engineering trade-off required to optimise performance and battery life.

Batteries will not be able to provide the sort of energy densities required. Faced with this problem, the industry is turning to alcohol. Fuel cells that generate electricity from methanol have emerged as the power source of the future for mobile terminals. A fuel cell is an electrochemical device that generates electricity from the combination of hydrogen and oxygen to produce water-the reverse of electrolysis. In the past two years developers have demonstrated direct methanol fuel cell (DMFC) prototypes that produce hydrogen directly from methanol with much greater energy densities than current rechargeable batteries.

5 UMTS COMPONENTS OPEN EMBEDDED ARCHITECTURE

5.1 Definition

In the mobile communication technology community the word “open architecture” or “open platform” has become somewhat of a buzzword lately. However, most definitions for openness share similar characteristics, such as fair and transparent multilateral governance model, and feasible licensing terms of essential Intellectual Property Rights. Moreover, in open architecture, there is a complete disclosure of the technical specifications, and any possible extensions are contributed to standards.
This chapter thus tries to define first a point of reference and then elaborate on different aspects of an open architecture, including both technical and strategically.

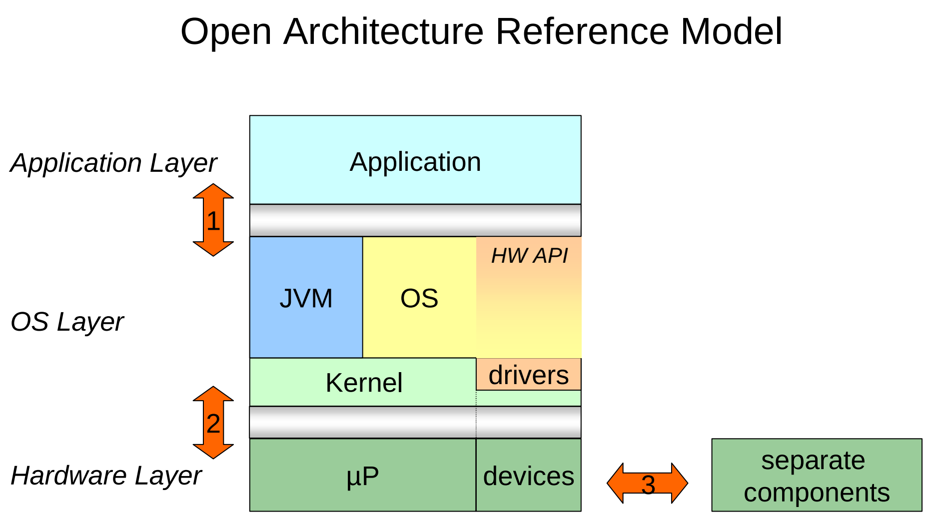

Figure 3 – Open Architecture Reference Model

Note: At the moment there is no agreed Open Architecture Reference model which is approved by the industry. This an example of Open Architecture for future discussion. The picture does not include middleware which is essential part for 3G services and applications driving the UMTS deployment. In addition regarding HW API in the OS layer. It is the job of a competent OS to abstract the hardware and provide hardware-independent APIs. The OS should have plug-ins for hardware adaptation (including but not limited to device drivers) and a few other points might be included.

The reference system used – on a high abstraction level – for defining the term “open architecture” in the context of embedded components can be seen in **Figure** . The lowest layer represents the hardware components of a mobile terminal, which are grouped around a central SoC (System-On-Chip) that controls the terminal’s functionality and its peripherals.

On top of the hardware-layer resides the software that is needed for controlling the hardware, specifically the operating system's kernel and the device-drivers. Three columns represent the higher levels of the OS-layer: the left one represents a Java Virtual Machine that provides application programmers with an identical runtime-environment for Java programs on different underlying platforms. The column in the middle illustrates the terminal's operating system API (Application Programming Interface – according to www.acronyms.ch: "a set of calling conventions defining how a service is invoked though a software package") that provides application programmers with access to the routines and services that the operating system offers. The third column named Hardware API represents the interface through which application programmers can access the mobile terminal's peripherals by using the hardware drivers provided for by the chipmaker. More details on the reference system and its interfaces can be found in chapter 5.3.1.

In an open architecture, according to a definition by Carnegie Mellon's School of Computer Science, the interface specifications of its components are "fully defined, available to the public and maintained according to group consensus". In this context the three most important interfaces in the context of 3G mobile terminals are represented as double-headed arrows in Figure . Being the interface that programmers will use while developing application-software, structure and availability of interface 1 are of utmost importance. In the context of embedded components however, also interfaces 2 and 3 (i.e. low-level hardware access and inter-component communication) play an important role. By rendering public all information on these two, each semiconductor company has the possibility to develop systems that, by complying with the standard, can be incorporated in any mobile terminal. Differentiation would then no longer be based on interface-structure but solely on "smart design, a strong brand and enhanced services"(Nokia Press Backgrounder on its Open Mobile Architecture initiative).

5.2 Market perspective

Open Industry Platform is important to ensure interoperability and faster take off of new services based on UMTS wireless networks.

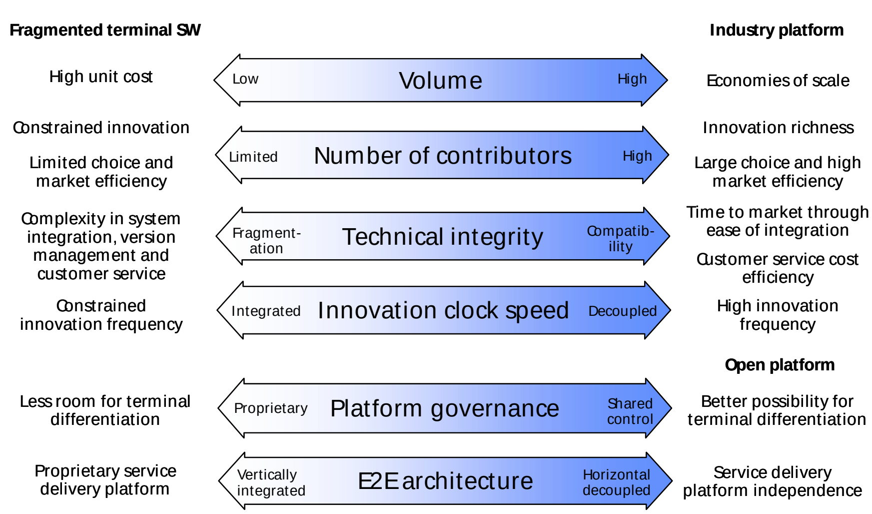

Figure 5: Why Open Industry Platform

Operators are in key position to implement the service Delivery Platform. By having a Industry Platform it provides operators several business benefits:

Economies of scale
Lower unit cost/functionality as Platform Functionality accumulates into the Industry Platform

Innovation Richness
Large volume creates and better business case attracts more developers and partners

Large choice and high market efficiency
Operators have freedom to select the best price/quality devices and applications through fair competition. There will be many devices from multiple manufacturers and competing applications from multiple developers.

Time to market through ease of integration
Fast introduction of new services across device categories through interoperability through open standards based on one open backend

Customer service cost efficiency

> **Consistent device provisioning and management**
> Same OTA and retail channel provisioning process across devices
>
> **Reduce customer service costs**
> Troubleshooting service issues across vendors is much more straightforward
>
> **Significant time reduction in service development**
> Common applications and optimized content adaptation across devices
>
> **Reduce end user education and marketing costs**
> Support marketing evolution from device marketing to service marketing
>
> **Significant reduction in device testing time**
> Once one device is tested on one handset, testing competence starts to accumulate

High innovation frequency
Fast introduction of new differentiating services as there will be applications from large developer pool

Open Industry Platform provides two main benefits for the operators:

1. Better possibility for terminal differentiation

Larger accessible market segments because source code licensing enables availability of differentiated compatible terminals from multiple manufacturers

2. Service delivery platform independence
Many services on one compatible platform. Availability of applications from large developer pool and innovation speed is decoupled from platform software release time schedules.

Seen from a market perspective, the advantages of an open architecture for embedded components are obvious: by publicizing all interface-information necessary to produce embedded microchips for mobile terminals (i.e. defining a standard interface 2), in theory each semiconductor company, technologically capable of designing the requested systems, gets the chance to become the supplier of any device manufacturer. At the same time interface-standardization would make it easier for the device manufacturers to change their component-suppliers (in other terms: their switching costs would decrease), since they would not have to worry about compatibility issues between the new hardware and the old software they are using. This would ultimately boost competition amongst chipmakers. The so-realized introduction of free market mechanisms at this stage of the value-chain would eventually result in better systems as competition would rise and chipmakers would have to differentiate themselves by performance rather than by using obscure interfaces as an insurance of their customers' loyalty or as defence mechanisms against rivals. In other terms: by establishing an open architecture for embedded components no party would hold disproportionate power and the development of market and technology would be subject to market-forces, and thus – in theory – converge to an optimum.

In the contrary case of a closed system, two developments are likely (Figure 3): either market-fragmentation increases with the result of several relatively weak participants that go more or less their own way, using their proprietary systems which are incompatible with each other. Or one proprietary hardware/software platform will emerge with its producer(s) ultimately dominating the device manufacturers, absorbing most of the power to decide on where the market and the technology are heading. A similar domination-scenario is of course also possible in an open system – the type of domination differs however. Domination in an open system would be largely based on performance and not (at least not exclusively) on strategic decisions. In theory, a company that offers a product, which is superior to all the competitors' ones, could also dominate in an open system.

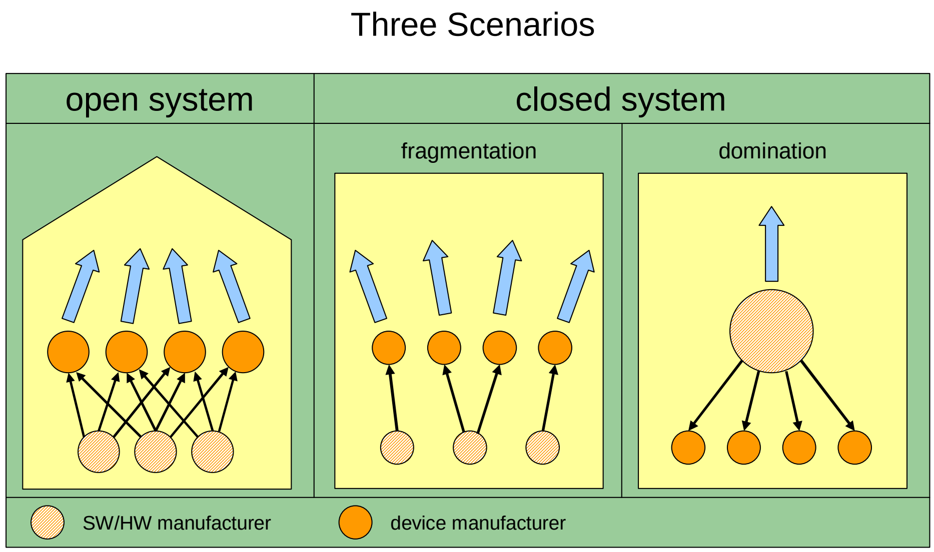

Figure 3 – Three market scenarios

Another important argument in favour of an open architecture is the solution of the compatibility issue. For software producers and device manufacturers it is important to have the security that future changes in hardware-components don't render oblivious today's investments in application-development. This type of compatibility is called forward-compatibility as opposed to the equally important backward-compatibility that allows end-users to run software, which had originally been produced for older systems, on any new device they buy. Last but not least, open component architecture could result in inter-terminal compatibility and eventually in a less-fragmented market if also the interfaces of higher system-levels were standardized. In fact, it is interface 1 which is subject to most of the standardization initiatives in the mobile communication business.

From a technological point of view, this overall compatibility gets accomplished by defining comprehensive interfaces that ultimately decouple the different layers of the reference model. For software suppliers a standardized interface 1 would represent a well-defined common playground from which they can start developing optimised software without being limited to a certain platform. Since this playground would be equal for all, the community of software-supplying companies for the same platform would grow, as would competition, ultimately leading to better, more innovative products and a greater market.

Standardizing interface 2 would decouple hardware from software. This would not only result in the above described strategic market advantages but also in a shorter time-to-market span of new devices as soft- and hardware development could take place in parallel.

The end user knows little if at all anything of the technological details of such an open architecture at the hardware-level. In the end however it is he who profits from the devices' higher quality and the broadened range of products and systems from which to choose from.

5.3 Interfaces

5.3.1 The Reference Model Revisited

Communication between the logical blocks that make up the reference model as it is shown in **Figure** gets regulated and formalized by the definition of their interfaces. Hence, it is important to define and understand the functionality of these blocks before identifying the peculiarities of the interfaces.

At the heart of a mobile terminal a microprocessor manages different hardware devices and various types of memory. In this context, a device is a dedicated piece of hardware that can either be incorporated on the same silicon chip as the microprocessor (the ensemble is then called SoC – System On Chip) or reside on a physically separated semiconductor component. It either provides specialized functionality (e.g. a hardware mp3-codec or a DSP-core) or acts itself as an interface for other components. Examples for these so-called "I/O cores" include a UART component for serial I/O ports or a Keyboard Interface.

On top of the hardware layer resides the operating system's kernel. This piece of software is highly dependent on the microprocessor architecture incorporated in the underlying SoC and is usually written in a mix of C and assembler. The kernel offers basic services like scheduling, task-, thread-, process- and memory-management, interrupt- and exception-handling, timers and counters, etc. to the higher OS-layers.

In order to render the above discussed devices visible for the operating system, and thus ultimately for the application programmers, the hardware developer has to provide for pieces of software called drivers. Also known as Low-Level API, a driver defines the interface of a set of functions that can be performed by the device and takes care of their hardware implementation. This is done either by translating driver-functions in commands for dedicated hardware-components or – in the case of Software IP – by using the microprocessor core and its instruction set to "simulate" a Hardware IP block. Driver-structure depends on the operating system vendor's specification and guidelines, as does its connection with the kernel. It is however common to all platforms that device-drivers are connected very tightly with the kernel as they often make use of kernel-services, especially scheduling and interrupt handling.

Apart from the kernel and diverse drivers, the operating system offers various enhanced services to the application programmer. These include a file system, multimedia support (video, audio-routines), telephony and communication infrastructure (GPRS-, UMTS-stacks etc., TCP/IP, etc.), security functionality, messaging support and many others. Besides these services, some operating systems also offer pre-fabricated modules that can be incorporated in third-party products (e.g.: Symbian's organizer-engine) or even complete applications like browsers or email-clients as part of the system.

Apart from the operating system's proprietary functions, a Java Virtual Machine provides an alternative way for applications to access the device's hardware. The JVM is often incorporated into the operating system and generally uses the same kernel and drivers, as does the OS.

The top-layer of the reference model belongs to the applications, i.e. programs written by third-party software developers. They either reside directly on the operating system or are divided between the client (i.e. the mobile terminal) and a server. Examples of applications include office-programs, video-conferencing software as well as games.

5.3.2 OS vs. Applications

Interface one, as it is illustrated in the reference model, defines the mobile terminal's appearance to the user-side, namely application programmers and end Users. For the latter, this interface is mainly defined by the GUI (Graphical User Interface), which determines the look-and-feel of the terminal. The GUI can either be directly incorporated in the operating system or be added as a module that complies to the standards set by the OS vendor, thus giving the mobile terminal manufacturer the flexibility to distinguish his end-product by using a proprietary look-and-feel.

Application programmers are especially interested in the mobile terminal's high-level API, rather than in the GUI which generally does not influence the functionality of applications. As can be seen in the reference model, this interface actually consists of three parts, giving the programmer access to the three underlying blocks.

The Java Virtual Machine's API is basically the same on each mobile terminal (depending only on the installed Java-version like J2ME or JavaPersonal) since the very idea of Java is platform-independency. How the JVM is implemented depends for the most part on the operating system. In some cases it is based directly on the kernel and the device drivers, in other cases it is implemented as an extension of the OS, making use of its advanced functions. The programmer however does not see the difference as long as the performance of the JVM is the same.

Access to the Operating System via its API is very platform specific and not standardized at all. There is however a big consensus among all parties involved in the 3G communication business that market fragmentation and system incompatibility at the device-level represent a great danger for the economic success of UMTS and other 3G standards. Many companies have thus joined various initiatives (such as OpenML or OMA) that aim at standardizing the High-Level API, in order to provide end-users with true interoperability and full compatibility of different terminals.

The third part of Interface One is called Hardware API and provides the programmer with access to specific components of the mobile terminal's hardware. Like the above-described OS-interface, also the HW API is very platform dependent. Some Operating Systems completely cover the device drivers written by the semiconductor manufacturers and include the HW API in their proprietary OS API. In these cases, the semiconductor manufacturer has to work closely together with the OS vendor in order to render specific chip-components (such as a hardware mp3-codec) visible for the programmer. Other Operating Systems allow for direct access to device drivers by application programmers, imposing only some rules and guidelines on the driver's structure (more on device drivers in chapter 5.3.3).

5.3.3 HW Components vs. OS

In the context of this chapter "components" are to be understood as elements of the silicon chip-set at the core of the mobile terminal. More specifically, besides chips that implement specialized functionality, there is one central SoC that incorporates a microprocessor and several dedicated IP blocks (see Figure 5). All terminal functionality gets controlled and accessed through this central chip, regardless if the hardware is incorporated on the same piece of silicon or on a separate chip.

As described in chapter 5.3.1, the kernel is the central part of the Low-Level API. It implements basic software-concepts like tasks or threads that the microprocessor itself does not know. The kernel also allows for direct access to the microprocessor's registers and special functions that depend on the processor type. Also the drivers rely on the kernel as they make use of the thread-model or interrupt-delivery functions. The kernel is optimised for and highly dependent on the microprocessor type. The drivers on the other hand are dependent on the kernel's scheduler and it's interrupt handling, which is why the two have to be treated as one functional block when identifying logical interfaces in the reference model.

Interface Two, as it is highlighted in the reference model, is – as of today – not standardized at all. When developing a new SoC, chipmakers have to choose a microprocessor core and the target-operating system. These two choices are interdependent because the microprocessor developer has to have reached an agreement with the operating system vendor in order to be able to offer an optimised kernel for its processor core. It is then the SoC-maker's task to adhere to the operating system vendor's guidelines and interfaces when programming the drivers for the incorporated devices. These driver-programming rules differ considerably between operating systems and practically represent a tight bond that lets SoC and operating system appear as an entity, a platform.

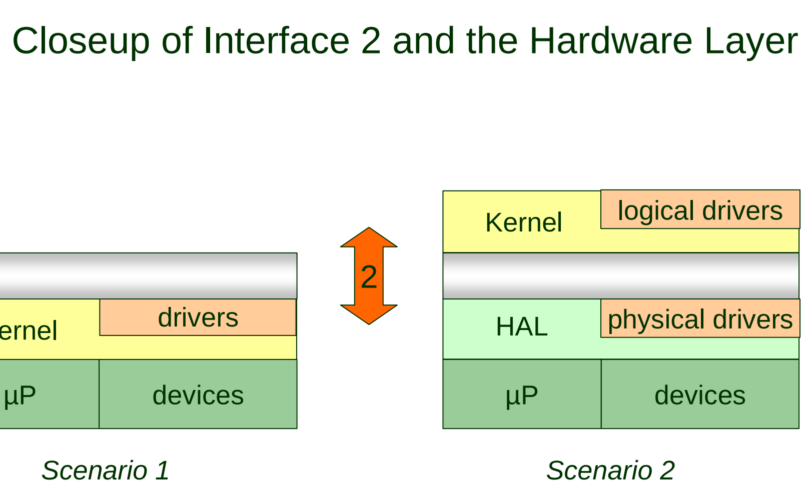

Figure 4 – Close-up of Interface 2 and the Hardware Layer

The idea behind a standardization of the Low-Level API (or Interface Two – not to be confused with device-drivers, which are also sometimes called this way) is to break up this entity and render the SoC independent from the operating system (and vice-versa). As Figure 4 shows, there are two ways to do this: Scenario 1 attributes the kernel to the hardware layer whereas Scenario 2 inserts a further layer called HAL (Hardware Abstraction Layer) in the stack. This is done largely because it seems unlikely that OS vendors are willing to base their systems on a generic kernel, as it would be the case in Scenario 1. The more realistic way to realize the separation of hardware and software is the introduction of the HAL, a piece of hardware-specific software that abstracts the microprocessor and to some extent the underlying hardware-platform in general. The HAL does in fact nothing else than extracting basic functionality like interrupt delivery, context switching or platform-start-up from the kernel. The kernel's remaining tasks include thread scheduling, exception- and interrupt-handling, and thread synchronization.

Having defined a standard HAL-interface remains the task of standardizing device driver-interfaces in order to decouple the hardware layer from the software layers. Since device-drivers rely heavily on kernel-functionality such as thread scheduling and interrupt handling, they cannot be treated as a separate entity but must be tightly coupled to the kernel. The situation as it is today is depicted in the reference model (**Figure**): drivers are completely dependent on the OS-kernel and their interfaces, being subject to the OS-vendors design decisions, are different for every OS. In the case of an open architecture, the monolithic driver-block has to be broken up into a logical part, which defines the generic interfaces of different driver-types and is tightly coupled with the kernel, and a physical part, which takes care of the driver's actual hardware implementation.

The idea behind the open embedded architecture now is to define the Low Level API that makes up interface 2. Given the interface-specifications, chipmakers could provide their customers with a HAL and the physical part of the system's device-drivers, without having to adapt their products to certain operating systems. Device-manufacturers would be free to choose any operating system that complies with the standard and any hardware components that provide for the functionality, on which the OS's kernel and the logical drivers are based.

Defining a set of generic device-drivers for all kinds of mobile terminals is a delicate issue, since the terminal's functionalities vary a lot. Chipmakers also have to be able to differentiate their products by adding dedicated pieces of hardware, while application programmers have to be offered an interface that does not change whether there is, e.g., an MPEG-4 accelerator on the SoC or not. Moreover, such a driver-interface is designed to last for some device-generations and is thus also to be designed in a way that does not hinder future innovation.

Unix and its descendant Linux have found a very elegant way to solve the driver interface-issue by implementing a file-based device-concept. Each device under Linux is seen as a file on which a limited number of operations can be evoked. The programmer sees for example a device called "mpeg4" to which he can write encoded data and from which he can read the decoded output. It is the chipmaker that takes care of the actual device-implementation either by using the driver of a dedicated component like a hardware mpeg4-accelerator or by delegating the task to the system's microprocessor (the routines for en-/decoding mpeg4-data could be provided for by the microprocessor producer or a third party). Both sides (programmer and chipmaker) are extremely flexible as the interfaces remain the same across different hardware platforms and the only agreement to be made between the parties is the list of available devices.

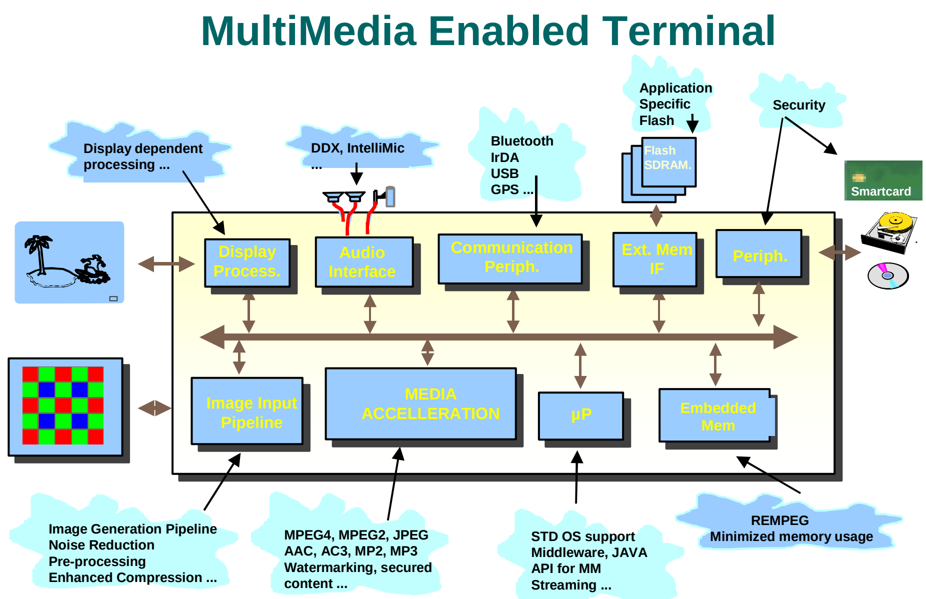

Figure 5 – Example of a Multimedia SoC *(Source: STMicroelectronics)*

5.3.4 Component vs. Component

Often strategical reasons – rather than technological ones – influence the SoC-maker's partitioning decision, not to incorporate certain kinds of hardware in the SoC (for more information on this partitioning issue, see also Chapter 5.4.2). By not including components like a telephony-baseband processor or an LCD controller in the SoC, chipmakers give terminal-producers the flexibility to choose their components from different suppliers. Another important reason for not including certain functionality into a SoC is the ability to respond flexibly to the diverse technological evolution of various components. A complete SoC would be rendered oblivious if only one of the integrated components changed as a new standard emerges. Obviously the inter-component interface has to be standardized in some way in order to grant component-interoperability.

Chips generally communicate via parallel or serial interfaces. As of today a number of different industry standards for these interfaces has emerged, resulting in the need for SoC-makers to incorporate redundant I/O modules that comply with different standards but practically perform the same task.

In inter-chip communication, serial interfaces are most often used, since their data-transfer rates have risen significantly over the last few years (up to 2.5 GBit/s according to the 2001 ITRS) and the chip's pins are a valuable resource for its designers. Parallel ports can be found more often in communication interfaces with external devices.

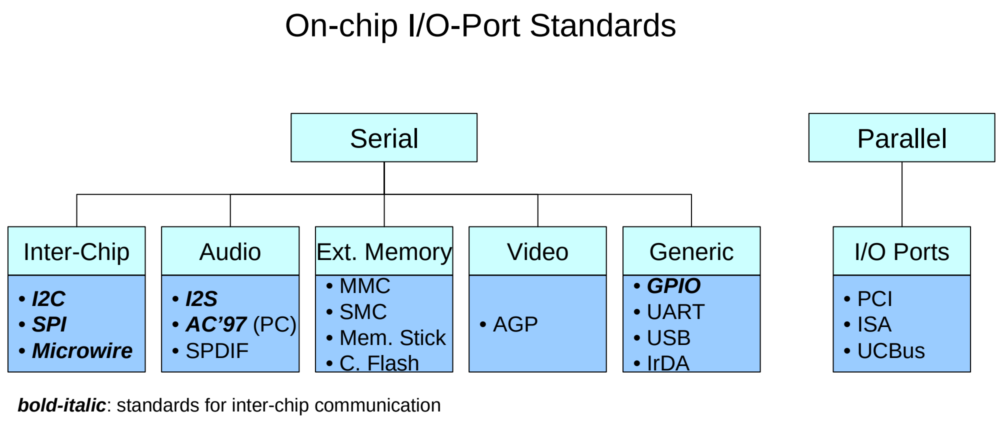

Figure 6 – Overview of various on-chip I/O-standards

There exists a plethora of different standards for serial hardware communication. Many of them are – as shown in Figure 6 – application driven or regard communication with external devices. A typical standard defines a number of pins (usually there are at least two of them: TxD for transmitting and RxD for receiving data/commands; if the standard is asynchronous there is also a third pin called CLK for the clock signal,) and the way they are used. This includes signalling schemes (signal timing and signal interpretation) as well as system diagrams, explaining for example the generation of the bus' clock signal. Also the data- and command-frame-format the communicating devices have to adhere to be defined in the standard.

The decision which interface standard to support depends largely on the context the SoC is designed to be used. One possibility to save pins and still provide for high compatibility is to use a multiplexer to assign pins to the logic modules that implement similar standards such as I^2C, SPI or Microwire. Another feature often found in SoC's is a general-purpose port (GPIO) that can be programmed flexibly to satisfy system-specific needs.

Often chipmakers, especially in the PC-market, perceive slow bus-speeds as a bottleneck, which limits their products' performance. A CPU can only process data as fast as the system-bus can deliver or fetch it. This has caused many chipmakers to drive interface-development by founding or joining diverse initiatives, which are trying to establish new I/O standards. Busses like USB or PCI have been established thanks to initiatives supported by Intel. Right now, there are three initiatives (3GIO (Intel), Hyper Transport (AMD), RapidIO) that are proposing three different solutions for a universal high-speed communication interface. The general direction in which these initiatives are pointing is however the same: they all aim for a universal standard for inter-chip and inter-board communication, which is packet-switched and scalable, provides for legacy-compatibility (namely with PCI-bus drivers) and error-correction. Bus speed will be up to 10 Gbps per direction.

In the context of 3G mobile terminals the number of physically separate chips is limited as a result of the well-known constraints on form-factor and power-efficiency. Hence, only some well-chosen functionalities such as telephony baseband-processing or the RF-Transceiver might be incorporated outside the SoC, thus limiting the requirements posed to the interface and the protocol. The interface between the RF-part and the mixed-signal SoC is special, since it is an analogue interface. Instead of defining bit-patterns and clock-frequencies, an analogue interface-standard has to define the intermediate frequency of the signal that leaves the baseband-chip and is to be mixed with the high-frequent carrier-signal in the transceiver (see also Chapter 5.4.1). By defining the different components that can make up a mobile terminal and their communication-requirements, a common open inter-chip communication interface could be agreed on, thus eliminating the need for multiple I/O standards to be incorporated in the SoCs.

5.4 Partitioning

The reference model that has been introduced in chapter 5.1 gives a very abstract overview of a mobile terminal's system architecture. The apparently solid blocks that make up the reference model consist – seen in the context of a systemic architecture description – of a number of smaller blocks, which themselves contain even smaller ones and so on. Modelling the system in more detail becomes more difficult, since it is no longer clear how the small blocks, that describe the details of a greater block's functionality, have to be grouped.

The issue that system-designers face after having specified the functionality of the terminal they want to develop is how to distribute its functionality among hardware- and software-components. The key-question, they have to answer, is the following: "What functionality should be realized by which component?" In real-world cases, this translates to the following examples: should the (de)modulator and the Transceiver block be integrated into one component? Is the system's MPEG-4 accelerator realized in hardware or software? Is it integrated in the Operating System or has it rather to be seen as an application? All these questions concern the partitioning issue that is to be addressed in the course of this chapter.

This partitioning issue is very difficult to resolve, since the trade-off that has to be made among the manifold and complexly interrelated variables cannot offer an ideal solution. Whole teams of engineers tackle this issue in the early phases of a project in order to find an acceptable solution. This report does not intend to offer any solution but aims at analysing and highlighting some aspects of the issue.

5.4.1 How a mobile UMTS terminal works

The model in Figure 7 illustrates, on an abstract level, the dataflow between the different logical blocks that make up a mobile UMTS terminal. The segmentation of the system is based on functionality and does not imply any hints on the physical realization. The scope of this paragraph is to explain the logical steps that are necessary in order to send and receive data on a mobile UMTS terminal. This knowledge is necessary in order to understand and explain the hardware partitioning issue that engineers face when developing a mobile terminal.

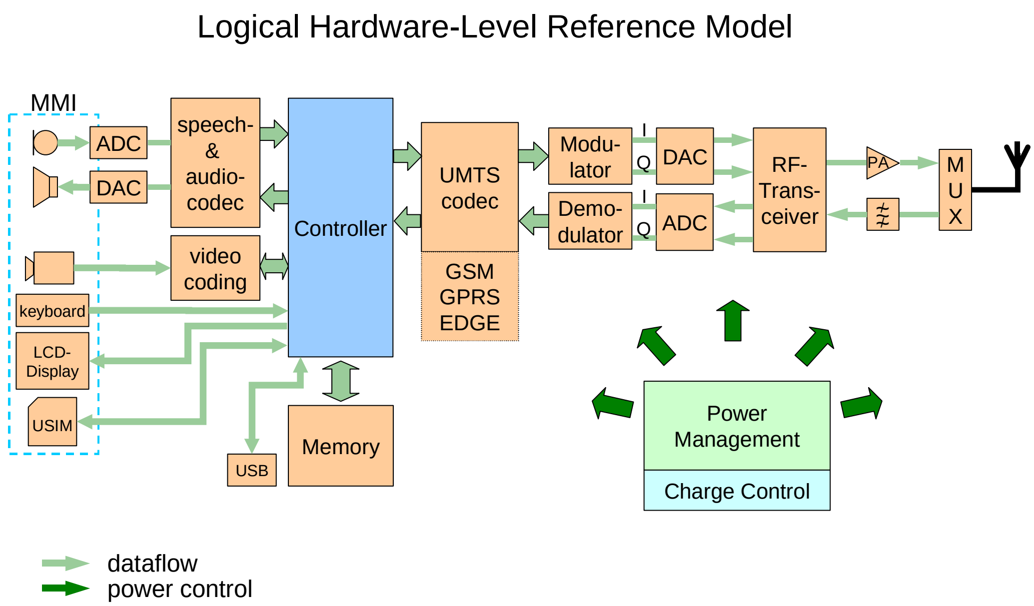

Figure 7 – Logical Hardware Reference Model of a UMTS Terminal

There are three ways through which data can enter the mobile terminal: the Man-Machine-Interface (MMI), the antenna or other input-ports like USB or IrDA. As can be seen in the reference model, any data that enters the terminal has to pass an ADC (Analogue to Digital Converter) and thus be digitalized prior to any processing step. The following paragraphs will explain the first two scenarios (MMI and antenna), neglecting the special case of already digitalized data entering the terminal through generic I/O ports since this way does not provide any new insight.

5.4.1.1 Data entering through the MMI

Any analogue data that enters the mobile terminal gets digitised prior to any further treatment. In the case of a UMTS terminal, analogue data is usually a speech signal that enters the terminal through a microphone. This signal gets sampled and quantized, resulting in a bit-stream which is however far too large to be transmitted or stored directly in an efficient way. In order to reduce the amount of data, while still keeping its quality on an acceptable level, the raw data gets coded. The type of code that gets applied depends on the data-type and its characteristics (speech or music for example), the quality expectations of the decoded signal and the capacity of the transmit-channel (or the memory-space available for storing the data). Besides compressing it, a code can also protect data, which is to be transmitted, from errors that might occur during transmission due to bad channel-conditions.

In the specific case of a UMTS terminal, the speech-codec (which is defined in the standard) is the Adaptive Multirate Codec (AMR) with a data-rate of 4.75-14 kbps. Besides the speech codec, necessary for voice-communication, future UMTS-terminals are likely to incorporate also other functionality that requests diverse codecs. Music files could for example be stored in the mp3-format, and video-streams, originating from the device's digital camera, could be encoded in the MPEG-4 format.

Any data that is to be transmitted through the "air interface" (whether it is coded speech, IP-packages or any other type) gets directed by the terminal's control unit (typically a microprocessor) to the UMTS-codec, traditionally called baseband processor. In early UMTS terminals there will also be a GSM-, GPRS- or EDGE-codec, in order to provide backward compatibility with existing networks, as it is demanded in the UMTS-outlines. The UMTS baseband processor performs a series of coding steps, which are defined in the standard, like channelisation, rate matching, interleaving, and scrambling. The coded data is then modulated, using a digital modulation technique, which is defined by the standard (QPSK for UMTS, GPSK for GSM-based networks). At the modulator's exit, two digital signals, representing the modulation's I/Q-value, enter a DAC, which "translates" the digital version of the modulated signal into an analogue wave-signal that consists of the two orthogonal I/Q-waves. In a process called up-conversion, the modulated signal is then mixed with its high-frequent carrier-signal (UMTS: 1920-1980 MHz). The resulting signal then gets amplified and transmitted via the antenna.

5.4.1.2 Data entering through the antenna

The UMTS downlink-signal is transmitted within a frequency band of 2110-2170 MHz. Before any other operation, the signal that enters the mobile device through the antenna gets filtered and amplified. The RF-Transceiver then extracts the carrier-signal and passes the remaining signal's I/Q components on to the ADC and ultimately to the Demodulator. This block translates the I/Q coordinates into symbols of two bits (QPSK), which are then handed over to the UMTS-decoder. This block recovers the transmitted information bits by performing the same operations as its coding counterpart, only in the opposite order.

It is the terminal's control block that decides on what happens with the received and decoded data. If it originates from a voice conversation, the received data-blocks get handed over to the speech-codec who, by applying the standard's decoding-algorithm, recovers the bits that describe the speech-signal that has been produced by the calling party. These bits then get converted to an analogue signal, which exits the terminal through a loudspeaker.

If the received data is of any other type (e.g. an MPEG-4 video stream or a Java-applet), the control chip either delegates the processing-task to specialized blocks or takes care of it itself.

5.4.1.3 Non-data-processing related blocks

Besides the blocks that have to be passed by data in order to be transmitted via the air-interface, there are a few more blocks, needed to operate the system. The more important ones among these are: the (already mentioned) controller, different types of memory, a power-management component and a block that controls the terminal's battery charging.

At the core of any UMTS terminal will be a microprocessor, which not only controls the interplay of the other hardware components, but which also runs the operating system and thus acts as the hardware's interface to the upper software layers. Although there is little doubt that a microprocessor will be part of UMTS terminals, the extent of its responsibility and functionality (with respect to other computing hardware such as a DSP) is subject to the partitioning issue.

Tightly connected to the terminal's control unit is the memory block. Far from being a homogeneous structure, as is suggested by the model, it can consist of many different types of memory, serving for different purposes like video-memory, memory-bank for inter-chip communication or storing-space for MP3-files.

With power consumption being the most critical constraint a mobile terminal faces, the power management block assumes great importance. By not supplying energy to the parts of the hardware that are currently not used, an intelligent power-management can contribute a lot to the terminal's power efficiency.

5.4.2 Hardware vs. Hardware

At the base of the hardware-partitioning issue, i.e. the question of how to distribute functionality among different hardware components, six drivers can be individuated:

- **Cost** – The target area of the overall system-cost has a great impact, not only on the terminal's performance-characteristics, but also on the structure of the underlying hardware. In a high-end terminal, for example, a low-power, high-performance DSP could be implemented to handle modem-functions, whereas in a low-cost device, this job would be delegated to dedicated hardware blocks. The more expensive DSP-variant would allow for flexible soft-updates, once a standard gets enhanced, and would thus increase the terminal's life expectancy. It is the system designer who has to make the decision whether this feature justifies the increase in cost. The cost-driver is the most universal one, since it is connected with all other drivers, i.e. enters in any partitioning-consideration.
- **Form-factor** – Ever smaller form-factors impose very rigid constraints on component-count and -size. If the goal were, for example, to produce an ultra-small phone-terminal, this driver would clearly favour an integrated solution with a SoC containing most elements.
- **Power consumption** – There is a large degree of consensus in the wireless communication business, that this driver is the most important one. End users want to use their terminal without having to worry about the battery status every other day. However, the evolution of energy density in batteries does not keep pace with the evolution of the chips' computational capabilities (see Figure 8). This is the reason why power consumption has a crucial influence on hardware partitioning – future mobile terminals have to offer more functionality while consuming less (or at least an equal amount of) energy.

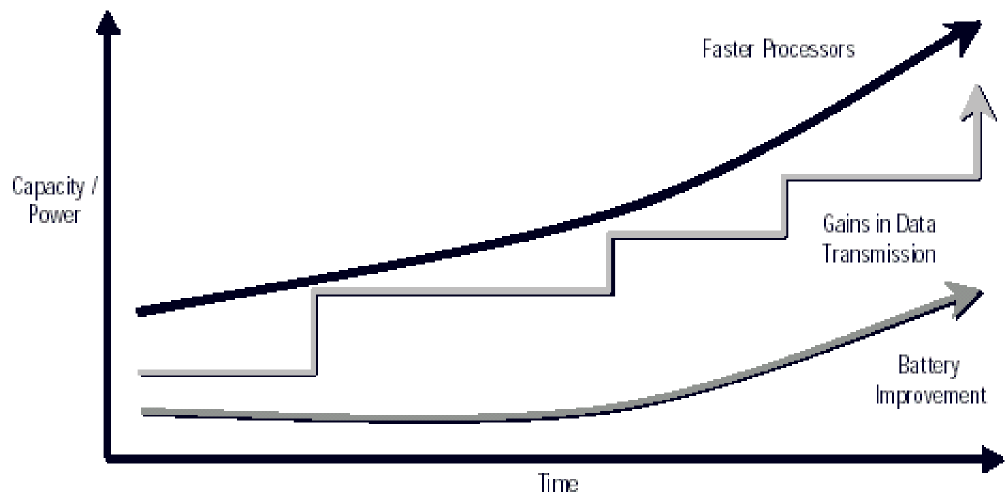

Figure 8 – Battery Power a Limiting Factor (*Source: Salomon Smith Barney)*

The impacts of this driver on system-partitioning are two-fold: on the one hand, it favours large scale integration, because the overhead that's connected with each physically separated component, e.g. a chip's pins, consumes power (pin dissipation). On the other hand: separation of functionality among different components allows for a simpler power-management as the single components can be switched on and off independently. While being technologically more complicated, it is however also possible to individually activate and deactivate the blocks which make up a SoC.

- **Computing power** – The QoS-classes (Quality of Service), defined in the UMTS standard, influence not only the characteristics of the communication-link but also the requirements of the terminal's computing capacity. Speech-processing during a phone-conversation, for example, has to take place within a certain time limit in order not to be noticed by the communicating parties. Thus, a certain amount of computing power that cannot be fallen short of is required. In general, dedicated pieces of hardware (chips or hardware-IP blocks) are computationally faster than generic components like DSP's, albeit their dedication to one task renders them inflexible and difficult to update or exchange.

- **Technological constraints** – Besides the rather technical drivers, seen so far, there are some technological limits that significantly reduce the design-engineer's freedom in partitioning the system. One example for such a constraint is the integration of functional elements, which require diverse production processes (such as the RF-Transceiver and the Baseband-Processor), into one chip. In order to deliver such a multi-process component, a chipmaker needs to possess a distinct technological know-how that can only be built over time. The economical implications of the decision to deliver an integrated multi-process SoC are noteworthy as well: the chipmaker needs to be financially potent when he decides to develop such a component, since its production implies huge upfront NRE (Non-Recurrent Engineering) costs that have to be amortized over time by high-volume sales.

Another technological effect, that has some influence on hardware partitioning, is the electronic noise, generated by digital circuits. The steep edges of a digital signal result in high-frequent waves that might distort and "pollute" radio signals like, e.g., the RF-signal received via the antenna. As a result, the ADC/DAC-blocks are often integrated in the mixed-signal baseband-processor in order to keep the RF-component fully analogical and thus not worsen the quality of the radio-signal.

- **Strategic marketing constraints** – For a chipmaker, which is designing a mobile terminal chipset, the partitioning-issue poses also strategic, marketing-related questions. If his company, for example, is very competent in designing the baseband-processor and also offers an RF-section which, however, is inferior to a competitor's one, a decision has to be made on how tight these two components should be coupled. Integrating the two in one SoC or defining a proprietary interface between them, would effectively force clients to buy both or none. Such a strategy could help boost sales for the weaker RF-component or jeopardize the success of the base band processor.

  A further strategic factor that has some impact on the hardware-partitioning decision results from the different roadmaps a terminal-manufacturer might follow for the different components he includes in his product. In general, it is undesirable for a manufacturer to redesign every single component when developing a new product-version. A terminal-manufacturer, who is pleased with the RF-Transceiver chip he uses in his current model, might want to change only the base band processor for the devices' next version. An integrated SoC – though technologically attractive and maybe even cheaper than the two-component solution – would not satisfy his needs.

Although, from a technological point of view, hardware partitioning is a crucial system-design aspect, it is of little importance for market-players other than the chip- and the device makers. As long as the underlying system meets the latter's performance expectations and offers the interfaces that have been agreed on in the standard, any hardware-partition is fine. It is here that chipmakers have the chance to diversify their products in the open architecture environment, by offering the best trade-off between the above named drivers.

5.4.3 Hardware vs. Software

Closely related with the hardware-partitioning issue is the question of which functionality to realize in hardware and which one in software. With the exponentially growing computing power of microprocessors, software IP-blocks that handle, for example, speech-processing have become feasible and represent a true alternative for hardware components.

Codec-functionalities (audio, video, UMTS) have traditionally been realized using a DSP, since they are based on signal-processing routines. In the context of a mobile terminal with its strict constraints on power consumption and system cost, however, also software IP solutions get proposed. These consist of optimised software drivers that are based directly on the underlying microprocessor. Besides the trade-off they make between performance and cost (monetary as well as in terms of energy consumption), an advantage of these solutions is the great flexibility they offer to the terminal manufacturers, as those can adapt (i.e. reprogram) the mobile terminal rapidly to new standards that are emerging. In the near future, according to the ITRS

2001 System Drivers chapter, this re-programmable-feature will however be limited to high-end terminals.

One traditional domain of hardware-developers so far, has been the analogue part of a mobile terminal. For a tri-band cellular, for example, three parallel RF-mixers had to be integrated on a chip. Recent research by the Computer Science groups of MIT and the Georgia Institute of Technology indicates however that this is about to change (IEEE Spectrum, May 2002). Software Defined Radios (SDRs) allow switching between different radio interface standards such as cdma2000, UMTS, EDGE, and GPRS without needing to replace any hardware component. “Analogue hardware functions such as frequency tuning, filtering, modulation, and demodulation is replaced by software that implements those functions digitally.”

A technology’s life cycle represents another factor that plays an important role in the decision on whether to implement certain functionalities in hardware or software. An example for such a technology is the EMS (Enhanced Messaging Standard). With the advent of more potent UMTS-phones already looming at the horizon, this improvement of the traditional SMS-technology, which allows to include simple sound-files and black and white images in a message, is bound to have a short life cycle and make way soon for the more advanced MMS (Multimedia Message Standard) which allows for inclusion of JPEG images and MPEG-4 files. A hardware-bound implementation of the messaging standard would thus not be advisable since it would shorten the terminal’s life cycle by coupling it to the technology’s one.

Besides the two extremes of a hardware- and a software-implementation, there also exists the possibility to combine the two solutions and support certain functionalities by adding “architectural extensions” to the terminal’s microprocessor. Examples for all three approaches can be found in the context of different chipsets’ Java-support. Software implementations consist of optimised Java Virtual Machines (JVMs) or just-in-time (JIT) compilers, whereas hardware solutions are based on dedicated Java processors or Java co-processors. An example for the third approach – the architectural extension – is ARM’s Jazelle-technology, which seeks to make a trade-off between the performance constraints of the software-solutions and the complexity-issues of the hardware-solutions. Jazelle consists of a specialized processor instruction-set and a software-module, called Jazelle Support Code, which “manages the interface between the core hardware, the virtual machine and the OS.” (ARM brochure on its PrimeXSys platform)

5.4.4 Software vs. Software

At the beginning of this chapter, the partitioning-issue was summarized in one sentence: “What functionality should be realized by which component?” While the logical structure of the terminal’s hardware-functionality could easily be described by a simplified figure, things are not that straightforward anymore when it comes to software partitioning. The logical blocks that make up a mobile terminal’s software layer are heavily interdependent and cannot be easily modelled by straight signal-flow diagrams. It is however necessary to define what functionality the terminal’s software is supposed to provide for, in order to explain the software-partitioning issue.

The functionality, a mobile terminal’s software has to provide, can be summarized in four groups:

- **Access to Hardware** – The kernel, a piece of software which is the first one to be loaded when the system boots and which stays in the memory as long as the terminal is running, serves as an interface to the terminal's microprocessor. It offers basic services like thread-, process- and memory-management, power-management, scheduling, context switching and exception-/interrupt-handling to the higher software-layers and the drivers. The latter, also known as kernel-extensions, serve as interfaces to the other hardware-components. At this level of the software stack, there is little discussion about partitioning, because kernel and drivers – while being very important – are in general regarded as commodities.

- **Read & Write Data** – This functionality-group contains data-management services like a file-system, needed to store and retrieve data from different memory-types, as well as access-routines to I/O-ports (like USB or IrDA) or communication-channels. The latter comprise different communication-protocol-layers (speaking in terms of the ISO/OSI reference model), of standards like GSM, GPRS, UMTS and TCP/IP, as well as services like HTTP or WAP.

  Another important functionality of this group is represented by the routines that allow software applications to directly access (i.e. read data from and write data to) the devices that constitute the Man-Machine-Interface (MMI). These devices include the terminal's display, its keyboard, the speaker and the microphone etc.

- **Framework for Applications** – Applications need a framework to run in, which has to be defined by the terminal's operating system. The so-called High-Level APIs, which act as the interface between the system and the application, are the central part of this framework. Also called execution environment, it can either be made for *native* applications and thus be specific for the underlying operating system, or it can be made for *contained* applications and thus be portable across platforms (e.g. JVM). Other important components are the Graphical User Interface (GUI), which defines the terminal's look-and-feel, and the security-system, which enables data confidentiality, integrity and authentication by providing underlying support for secure communication-protocols.

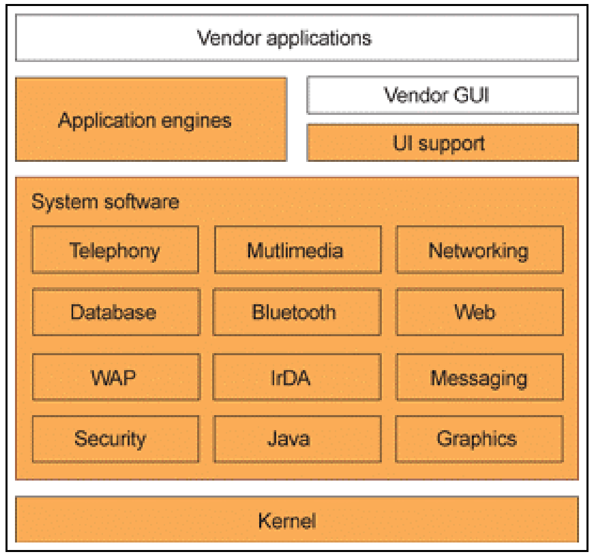

Figure 9 – System Software Layers of the Symbian OS *(Source: Symbian)*

A special case is the Java Virtual Machine. As the name implies, this piece of software simulates a machine that runs a platform-independent framework for (contained) applications, written in the Java programming language. The exact implementation of the JVM is an important detail of the partitioning issue and will be discussed later on in this chapter.

- **Offers Functionality to the User** – The most important part of the whole terminal is the one that makes the terminal's functionalities accessible for the user. This is, generally speaking, done by software applications, whose level of sophistication differs however a lot. Examples include the software that provides the user with telephony services (already a commodity) as well as a phonebook application, a calendar, games, a notepad, a calculator, etc.

Generally undisputed is the raw software-structure as it can be seen in the reference model (**Figure**). The kernel and the device-drivers are located at the base of the operating-system layer, which itself builds the basis for the applications. The motive behind the software-partitioning process, i.e. the more detailed description of the reference model's blocks, are twofold: for the most part, partitioning decisions are based on strategic marketing considerations, but also technical arguments and constraints have their influence.

An important technical constraint in the context of mobile terminals is for example the necessity of an operating system with a "small footprint", i.e. a system whose kernel consumes little memory. In order to adapt the operating system to the needs of the terminal, some operating system vendors allow not only the configuration of higher OS-layers but also the configuration of the kernel. However, as new terminals will be equipped with more memory, the importance of this constraint will diminish and the operating system's capabilities in general will not be limited strongly by technological issues.

Rather than being technical, the main arguments influencing software partitioning are strategically motivated. The important decisions that have to be made regard three aspects:

1. **Graphical User Interface (GUI) –** Should the operating system provide for its own interface (as does for example Microsoft Windows CE.NET) or should the terminal-maker be given the possibility to define its own look-and-feel concept by adding a custom UI-modules, like Series 60 or UIQ, to the OS (as is the case with Symbian)? This question implies some fundamental strategic concepts, since a mobile terminal's GUI defines a very important aspect of the whole terminal's appearance to the user. Companies like Nokia consider their proprietary user interface, an early version of which is implemented in the GSM phones, as a very important element of the "Nokia experience". From their point of view, a possible drawback of equipping their terminals with the OS-specific GUI could be that the end user would associate the terminal rather with a certain type of operating system than with the manufacturer's brand. By letting OS-vendors alone decide on the GUI, Terminal manufacturers would be deprived of a lever for differentiating their products and would risk becoming mere commodity suppliers, as has been the case with many PC makers.

2. **Java** – With an estimated 2.5 million developers worldwide, the Java language with its "Write Once, Run Everywhere" concept has become an industry standard, which is constantly evolving and offers with its J2ME (Java 2 Micro Edition) a special version for wireless terminals. It is thus not so much the question if, but rather how the mobile terminal should support Java.

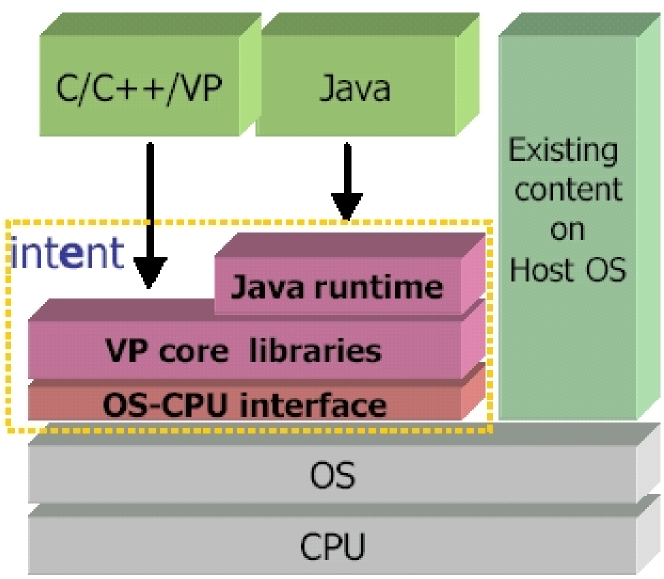

Figure 10 – The system proposed by OCPA *(Source: OCPA)*

Mobile Industry is moving towards open operating systems in the future. Open mobile operating system has following characteristics: APIs are fully defined and available to the public and licensees get source code and the right to modify/replace certain modules.

Four levels at which openness creates value

1. **Contributions by partners to device creation, facilitating differentiation**
   - … Hardware adaptations
   - … Technology point solutions
   - … Applications
   -
2. **Contributions by partners to customisation and service creation**
   - … UI customisation & branding
   - … Service and application frameworks
   - … End-to-end applications
   -
3. **Contributions by partners to enterprise customisation**
   - … Service and application frameworks
   -
4. **Contributions by partners to end-user customisation**
   - … Applications
   - … Services & Content (e.g. Games)

The different solutions, that have been proposed so far, vary greatly in the way; the Java Virtual Machine (JVM) is implemented and reflects the different strategic attitudes of their proponents. SavaJe for example, a small Massachusetts based company, has developed a operating system that offers only Java APIs and thus runs exclusively Java applications.

Symbian has incorporated a JVM in its Symbian7 OS, thus offering the Java environment and its native C++ OS application framework in parallel. Microsoft has not included a JVM in its PocketPC- and Smart phone-platforms – a consequence of the company's 2001 lawsuit with Sun Microsystems. It is however possible to install a plugging that provides for Java support on Windows CE.NET driven machines.

The Open Content Platform Association, a mainly Asian industry initiative, is proposing an OS- AND CPU-INDEPENDENT Java-solution. Among its executive members are companies like Kyocera, Sharp, JVC and Tao Group, whose intent-product, a JVM that uses Virtual Processor core (VP core) libraries, builds the core of the proposed system.

1. **Applications** – Speaking in terms of the reference model (**Figure**), the third important partitioning issue deals with the height of interface one in the software stack. The question is, where the operating system ends and where applications start, or in other terms, what applications should be part of the operating system and thus be available on all terminals, that incorporate the OS. The borderline between strategical arguments and technological necessities cannot be drawn clearly, as can be seen in the current anti-trust lawsuit US v. Microsoft. Microsoft argues that its Internet Explorer browser has to be integrated into its Windows operating system (for PCs) for technical reasons, the US government believes that the reasons are purely strategical ones in order to drive competitors out of business.

   GSM phone users are used to the fact that all the software on their phone comes from the phone's manufacturer. As long as the terminal's technical capabilities are rather limited, this approach makes sense and is the only feasible one. With the cell-phone's evolution towards a data-centric terminal however, the applications become more sophisticated and OS-vendors realize the importance of enabling their main-product (the operating system) with complementary products, such as a browser, games, an agenda-application, etc. These applications can either originate from independent software developers, who might be stimulated by a developer program (organized by the OS-vendor), or be directly implemented in the operating system. The decision to incorporate these applications in the operating system reflects an understanding of the OS as a software-platform rather than just a framework for third-party applications. A hybrid solution has been chosen by Symbian – the company's Symbian7 OS incorporates so-called Application Engines, which provide for certain functionalities like an email-client, an agenda-engine or a chart engine. These engines can then be incorporated in third-party applications.

Advanced mobile operating systems like Symbian OS have good separation between system data and application data to manage error and any stability issues potentially caused by third-party applications, running on a mobile terminal. If an underlaying operating system can not handle error situations, a cell-phone may crash, due to bad software, which has to be rebooted or cannot accept calls during that time, i.e. it does not offer the very service it was designed for. Terminal-manufacturers and operators have recognised the potential issue with unstable third-party applications as it might have a negative impact on the overall-system impression the user perceives. It is unlikely that the user will differentiate between an application and the phone in the case of a crash. He will most likely associate the terminal's instability with the device or its brand – a horror-scenario for terminal manufacturers. This issue is being addressed in operating systems like in Symbian OS with a number of measures including application signing, platform security and data caging. In addition, all big handset manufacturers have created certification centres that test third-party software and issue compatibility-certificates.

5.4.5 UICC vs. Processor

Figure 11 shows a logical model of the so-called USIM-card. As this subchapter's title suggest, this is however not the technically correct name of the smart card, which should rather be called UICC. 3GPP defines the two terms as follows (TS 21.111 V. 4.0):

UICC – A removable Integrated Circuit Card (ICC) containing a USIM.

USIM – Universal Subscriber Identity Module; a 3G application on an ICC.

For compatibility reasons, 3G terminals will also accept traditional SIM-cards and offer the user the possibility to integrate a SIM as an application, either in native code or as a Java-applet, next to the USIM on his UICC. It will also be possible to store two USIM's on one card. A third type of identity module is the WIM (WAP Identity Module), which can also be run as an application on the UICC.

With the USAT (USIM Application Toolkit), a successor of the STK (SIM Toolkit), 3GPP has defined a functional framework for applications that run on the USIM. These programs, written either in native code for the processor-processor or in Java, can use the USAT-mechanisms to interact and operate with the mobile terminal. These interactions include proactive ones like sending a short message as well as call-control or menu selection mechanisms. The USAT is defined in the TS 31.111 document by 3GPP.

As mentioned above, modern smart cards also support JavaCard-applets (aka Cardlets) besides processor-specific applications, since many smartcard-producers include a JVM (Java Virtual Machine) in their products. In fact, it seems likely that in the future – as a result of the “write once run everywhere” characteristic and the uncomplicated usage of Java – the only kind of applications, running on smart cards, will be Java-applets.

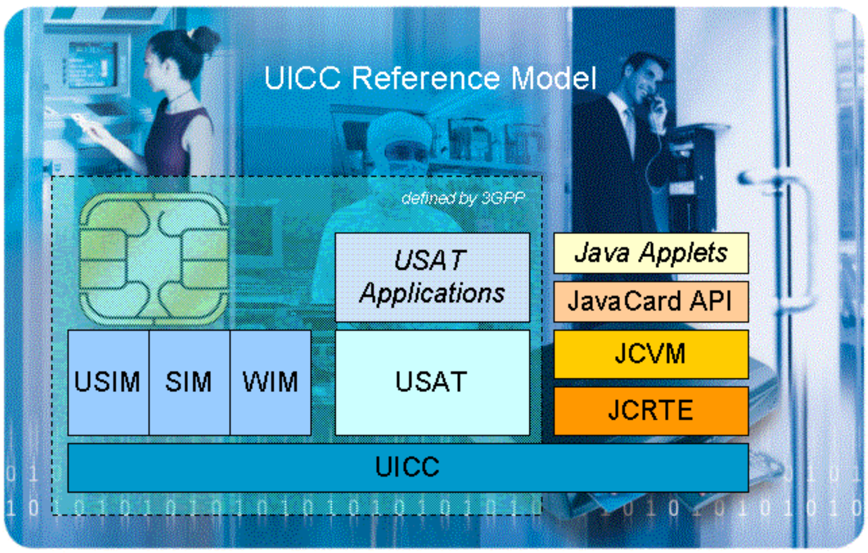

Figure 11 – UICC Reference Model

The blocks in the reference model show the different stack-layers needed to implement JavaCard, a version of Sun’s Java programming language, which is optimized for the use in “smart cards and related small-memory embedded devices” (Sun). On top of the JavaCard Runtime Environment (JCRTE), which abstracts the underlying hardware, resides the JavaCard Virtual Machine (JCVM), which offers the specific JavaCard API to the programmer.

Although the UICC-system is far less powerful in terms of computing-performance than the terminal’s application processor, there are some factors that make it special and justify its existence. From a technical point of view, these factors are security-related (tamper-proof, integrated cryptography algorithms, secure storage) and form-related: the UICC can easily be removed from one terminal and be used with any other terminal.

Besides the technical aspects, there is also an important economical one, namely the fact that the smart card and its content are products sold by the wireless operator to its clients. Operators can thus differentiate their products by including certain

applications on the UICC they sell their customers, even though the terminal's application processor might technologically be more adequate.
From the terminal-designer's point of view there is little need to consider the UICC when partitioning the system. Since the smart card's processing power is inferior to the terminal-processor, the terminal could consider the smart card rather as an element of the system's memory structure than a co-processor. The two systems are likely to coexist without influencing or relying on each other strongly.

5.5 Middleware

Middleware was originally the abstraction of the middle layers of end systems in distributed computing systems, which isolates applications from the details of the data distribution mechanisms.

In Handsets, Middleware is a set of software components that permit scaling of applications. It is made available to application developers through a consistent set of interfaces, or APIs, for the purpose of reducing the complexity of application development and integration.

The application services that are accessible via middleware are what make middleware relevant.

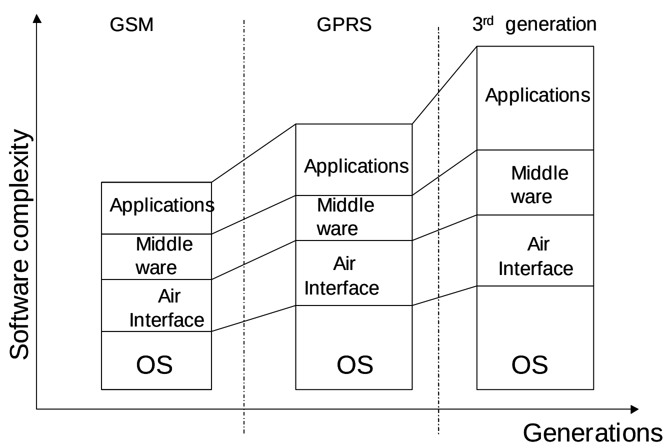

Figure 14 – Middleware Reference Model

Note:

Applications and middleware role will increase more than the OS in relative terms when evolving towards 3G. The question is of course, where the operating system ends and where applications start, or in other terms, what applications should be part of the operating system and thus be available on all terminals, that incorporate the OS.

6 LONG TERM EVOLUTION OF UMTS DEVICES

While the Industry is struggling to meet the qualities of devices (first models) for the 125 UMTS/3G licensees, one can expect already in the second generation models a number of improvements. Only after the commercial launch of the 3G networks, there will be new applications and services available for the users. They will call for new and better supportive devices.

However, various technological advances have conspired to sustain the validity of Moore's Law, and together these advances have been behind the successes of the computer industry and, in a large way, the communications industry.

A Software-defined radio, SDR, is a radio in which the receive digitization is performed at some stage downstream from the antenna, typically after wideband filtering, low-noise amplification, and down conversion to a lower frequency in subsequent stages, with a reverse process occurring for the transmit digitization (source www.sdrforum.org).

SDR will be implemented gradually to terminals and equipment.. As a software-defined platform technology, it can support multiple protocols simultaneously and it promises to enable rapid deployment of 3G wireless services and applications. SDR Chips sets will be available within the next two to four years.

Bluetooth, which has often been described as a solution looking for a problem will make its break through. The IT industry has dreamt up myriad applications for the short-range wireless technology, but none has achieved success. This will change with the strong interest car manufacturers are showing for Bluetooth. Only two per cent of the phones sold last year had Bluetooth but this will change rapidly.
By 2007, around 20% of all new cars will have Bluetooth technology. This will allow to setup a wireless connection between the Bluetooth equipped mobile phone to the car's built-in speaker and microphone, allowing drivers to take and make calls without taking their hands off the wheel-"hands-free". Many cars are equipped with voice dialing and the number is displayed on the car's dashboard screen. Using Bluetooth, the in-car system automatically "syncs" to the phone book stored in the mobile phone. Even if the phone is in a bag inside the car, it can still be used. When the car stops, the driver can continue the conversation outside the car because the system automatically "hands over" the call to the mobile phone.

The incredible shrinking computer will support mobile applications that are virtually limitless once the Device can be dynamically programmed with USIM cards. But, many operators are requesting for "Naked Phones" that they can tailor to the services they offer to their Users. This again requires simple but accurate Toolkits.

To an increasing extent, the 'mobile Internet' will have programmers write code where they refer to remote software objects residing in mobile handsets, servers and other nodes of mobile IP-based networks. Hence, they will have a growing need for an intelligently planned, uniform naming structure within the mobile realm to ensure that objects are named so that they can be easily found. This is closely linked with the needs of the operators in the areas of service roaming.

Software applications for mobile devices will evolve rapidly as various standards emerge and will call for mobile TLD to provide a consistent structure for mobile content identity and navigation.

The consumer/users will need convenient user interfaces to access the advanced mobile services.

Annex A Abbreviations and Glossary

Abbreviation	Meaning	Explanation
2G	Second Generation	Generic name for second-generation networks, for example GSM.
2G+	Second Generation enhanced	Name given to 2G networks enhanced with GPRS or EDGE.
3G	Third Generation	Generic name for third generation mobile networks.
3GPP	Third Generation Partnership Project	A co-operation between regional standards bodies to ensure global inter-working.
AAA	Authentication Authorisation Accounting	
B2B	Business to Business	Term used to identify a business-to-business transaction.
B2C	Business to Consumer	Term used to identify a business to consumer transaction.
Bluetooth™	Wireless industry standard	Short-range radio link standard. Uses licence-exempt spectrum @ 2.45 GHz to provide 1 Mbit/s.
BTS	Base Transceiver System	
CAMEL	Customised Applications for Mobil networks Enhanced Logic	CAMEL specifies how features normally associated with Intelligent Networks can be integrated into a GSM network. The greatest benefit CAMEL provides is to allow information on the caller's location to be passed from the network to an Internet web site.
CD	Compact Disc	Initially used to store music, now used for data and increasingly films/movies.
CN	Core Network	Physical infrastructure linking wireless base stations. Predominantly circuit-switched, core networks will increasingly become packet-switched.
DAB	Digital Audio Broadcasting	A digital radio technology used for radio broadcasting in a number of countries.
DECT	Digital Enhanced Cordless Telecommunications	A wireless technology used for short-range communications, for example cordless telephones.
DHCP	Dynamic Host Configuration Protocol	An IP-based protocol that allows the automatic configuration of an IP address to a host.

Abbreviation	Meaning	Explanation
DVB	Digital Video Broadcasting	A digital radio technology used for television broadcasting in a number of countries.
EBPP	Electronic Bill Presentation and Payment	With EBPP, the process of creating and delivering the bill and the process of paying the bill are connected through integrated systems and common databases.
EC	European Commission	
e-commerce	Electronic Commerce	Term used to describe transactions that take place on-line where the buyer and seller are remote from each other.
EDGE	Enhanced Data rates for Global Evolution	A further enhancement to TDMA systems which allows for data speeds up to 384 kbit/s.
EDI	Electronic Data Interchange	The exchange of standardised document forms between computer systems for business use. EDI is part of electronic commerce.
EDIFACT	Electronic Data Interchange For Administration, Commerce and Transport	EDIFACT or UN/EDIFACT (full acronym) comprises a set of internationally agreed syntax standards, directories and guidelines for the structuring and exchange between independent computer systems of data that can be generated in character format. The UN/EDIFACT rules are published in the United Nations Trade Data Interchange Directory (UNTDID).
EMI	Electro Magnetic Interference	
E-OTD	Enhanced-Observed Time Difference	
ETSI	European Telecommunications Standards Institute	One of the standards body for Europe.
EU	European Union	A community of 15 European nations comprising the European Economic Community.
FDD	Frequency Division Duplex	One technique used for wireless communications where the up link and down link are at different frequencies.
FER	Frame Error Rate	

Abbreviation	Meaning	Explanation
G5 Messaging		G5 Messaging is designed with a fallback to Group 3 fax and Internet e-mail as core capabilities. With a single keystroke, a message may be sent to multiple recipients using any mix of Group 3 fax, Internet e-mail and full G5 Messaging.
GIF	Graphics Interchange Format	
GIS	Geographic Information Systems	GIS is a computer system capable of assembling, storing, manipulating, and displaying geographically referenced information, i.e. data identified according to their locations.
GPRS	General Packet Radio Service	Technique used to upgrade current TDMA mobile networks. Allows a subscriber to gain up to eight 14.4 kbit/s channels. Also introduces packet switching.
GPS	Global Positioning System	A US satellite-based positioning system.
Group 3 fax		The most recent standard for fax devices which accounts for about 99% of all fax machines built since 1980. Group 3 actually refers to two recommendations defined by the ITU known as T.4 and T.30.
GSM	Global System for Mobile communications	The most popular standard for 2G mobile networks.
H.323		An umbrella recommendation from the ITU that sets standards for multimedia communications over Local Area Networks that does not provide a guaranteed QoS. These networks dominate today’s corporate desktops and include packet-switched TCP/IP and IPX over Ethernet, Fast Ethernet and Token Ring network technologies.
HiFi	High Fidelity	
HIPERLAN/2	High Performance Radio LAN Type 2	Wireless LAN (specify by ETSI/BRAN) in the 5 GHz IMS Band with a bandwidth up to 50 Mbit/s.
IANA	Internet Assigned Numbers Authority	
ICT Group	Information and Communication Technology Group	A working group in the UMTS Forum.
ID	Identification	

Abbreviation	Meaning	Explanation
IETF	Internet Engineering Task Force	An engineering and protocol standards body that develops and specifies protocols and Internet standards, generally in the network layer and above.
i-mode		Proprietary HTML-based mobile information service offered by NTT DoCoMo in Japan. The i-mode service is similar to WAP.
IMSI	International Mobile Subscriber Identity	
IMT-2000	International Mobile Telecommunications	ITU initiative for a global standardised 3G wireless network.
IP	Internet Protocol	The dominant network layer protocol used with the TCP/IP protocol suite.
IPDR	Internet Protocol Detail Record Organisation	
IPv4	Internet Protocol version 4	The version of IP in common use today.
IPv6	Internet Protocol version 6	The emerging standard, which rectifies some of the problems in IPv4, not only the address space. It offers new features and security and improves the network management control.
ISDN	Integrated Services Digital Network	A telephone service that offers high-speed digital services for devices connected to a telecommunications network.
ISM	Industrial Scientific and Medical	
ISP	Internet Service Provider	A company or organisation that provides access to the Internet to users.
ITU	International Telecommunication Union	
JPEG	Joint Picture Expert Group	Standard for the compression of still pictures.
MAC	Media Access Control	Part of the physical layer of a network that identifies the actual physical links between nodes.
m-commerce	Mobile Commerce	Similar to e-commerce but the term is usually applied to the emerging transaction activity in mobile networks.
MIME	Multipurpose Internet Mail Extensions	A specification for the transfer of non-text files with regular Internet e-mail.

Abbreviation	Meaning	Explanation
MMAC	Multimedia Mobile Access Communication systems	MMAC systems will provide high-speed, high-quality mobile communications via seamless connections to fibre optic networks, enabling the use of multimedia services anywhere and at anytime. MMAC services are targeted for launch around 2002 in Japan.
MMI	Man Machine Interface	A term used to describe the environment that encompasses the activities surrounding a user and their interaction with a device.
MP3	Music Player	The term has become synonymous with the MP3 player, which delivers CD quality music; It is the MPEG-1/2 audio layer 3.
MPEG	Moving Picture Expert Group	Standard for compression of moving pictures and sound. MPEG-1, 2, 4 are used.
NAI	Network Access Identifier	
PC	Personal Computer	Common term to describe the personal computer, usually based on a common architecture.
PDA	Personal Digital Assistant	
PSTN	Public Switched Telephone Network	The network, or groups of networks, consisting of switches and transmission that provide the bulk of switched services to the general public.
QoS	Quality of Service	Subjective and objective metric sets that quantify the performance of a network and its suitability for use with some applications and services.
RADIUS protocol	Remote Access Dial-In User Service protocol	An access server authentication and accounting protocol. The RADIUS authentication protocol is documented separately from the accounting protocol, but the two can be used together for a comprehensive solution.
Release 2000	Release from 3GPP	Term applied to the group of specifications due to be released in early 2001, which will concentrate on the core network. Also known as Version 5.
Release 99	Release from 3GPP	Term applied to the group of specifications forming the first phase of release specifications by 3GPP mainly concentrating on the radio access network.

Abbreviation	Meaning	Explanation
RF	Radio Frequency	
RSVP	Resource ReSerVation Protocol	The RSVP protocol is part of a larger effort to enhance the current Internet architecture with support for QoS flows. The RSVP protocol is also used by a host to request specific QoS from the network for particular application data streams or flows. RSVP is also used by routers to deliver QoS requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service.
SA	Selective Availability	An artificial error introduced into satellite data by the US DoD (Department of Defence) to reduce the possible accuracy of a position to 100 metres for commercial users.
SIP	Session Initiation Protocol	A signalling protocol for Internet conferencing and telephony. SIP was developed within the IETF MMUSIC (Multiparty Multimedia Session Control) working group, with work proceeding in the IETF SIP working group.
SLA	Service Level Agreement	
SLP	Service Location Protocol	An emerging Internet standard for automatic resource discovery on IP networks.
SME	Small to Medium Enterprise	Term used to describe a company that has less than 500 employees.
SMS	Short Message Service	The service that enables the sending and receiving of short text messages of up to 160 characters.
SS7	Signalling System No. 7	
SWAP	Shared Wireless Access Protocol	SWAP is the name given by the Home RF Working Group to its specification for data and voice wireless communication. SWAP is a combined protocol from the TDMA-based DECT protocol for voice communication and CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) based IEEE 802.11 protocol for data communication.

Abbreviation	Meaning	Explanation
TAP3	Transferred Account Procedure version 3	TAP is the process that allows a visited network operator (VPLMN) to send billing records of roaming subscribers to their respective home network operator (HPLMN). TAP3 is the latest version of the standard and will enable billing for a host of new services that networks intend to offer their customers.
TCP	Transmission Control Protocol	A transport layer protocol that offers connection-oriented, reliable stream services between two hosts. This is the primary transport protocol used by TCP/IP applications.
TDD	Time Division Duplex	One technique used for wireless communication where the up link and down link use the same frequencies.
TIFF	Tag Image File Format	A widely used format for storing image data.
TOA	Time of Arrival	
TV	Television	General term used to describe the broadcasting and reception of video and audio.
UDP	User Datagram Protocol	
UMTS	Universal Mobile Telecommunications System	UMTS is a modular system that incorporates several technologies that realise the convergence of existing and future mobile and fixed networks, including the Internet. The UMTS concept embraces also all applications and services that can be offered to the end-user. UMTS is a member of the IMT-2000 family of systems. .
UMTS Forum	Cross industry body	Non-profit, independent forum that gives guidance to standards and other bodies in terms of market requirements and issues to be solved to allow for a smooth deployment of UMTS. UMTS Forum's “Extended Vision” embraces all elements of the value chain beyond the standards (specified by 3GPP/ETSI) for 3G mobile networks.
USIM	Universal Subscriber Identity Module	The module that identifies, and is unique to, the mobile subscriber.

Abbreviation	Meaning	Explanation
UTRA	Universal Terrestrial Radio Access	
VHE	Virtual Home Environment	The term used to describe the concept of offering a subscriber the same services and facilities that he experiences on his home mobile network.
VoIP	Voice over IP	The generic term used to describe the techniques used to carry voice traffic over IP.
W3C	Worldwide Web Consortium	
WAN	Wide Area Network	
WAP	Wireless Application Protocol	Used to allow the transmission of simple web pages in 2G networks. Consists of a protocol stack that covers layers 4 to 7 of the OSI model. Uses IP but replaces TCP and HTTP with UDP. Web pages are written in WML.
WLAN	Wireless Local Area Network	
WML	Wireless Mark-up Language	
xDSL	Digital Subscriber Line	A group of technologies that allow higher speed access over standard wired lines to a telecommunications network, for example ADSL, which offers up to 512 kbit/s in one direction and up to 8 Mbits/s in the other (A= Asymmetric).
XML	eXtensible Mark-up Language	An open standard for describing data from the W3C. It is used for defining data elements on a web page and business-to-business documents. By providing a common method for identifying data, XML supports business-to-business transactions is expected to become the dominant format for electronic data interchange.

Table 1: Abbreviations and Glossary

Annex B Bibliography

[1] UMTS Forum Report No. 1: "A Regulatory Framework for UMTS", October 1998.

[2] UMTS Forum Report No. 2: "The Path towards UMTS Technologies for the Information Society", October 1998.

[3] UMTS Forum Report No. 3: "The impact of licence cost levels on the UMTS business case", October 1998.

[4] UMTS Forum Report No. 4: "Considerations of Licensing Conditions for UMTS Network Operations", October 1998.

[5] UMTS Forum Report No. 5: "Minimum spectrum demand per public terrestrial UMTS operator in the initial phase", September 1998.

[6] UMTS Forum Report No. 6: "UMTS/IMT-2000 Spectrum", December 1998.

[7] UMTS Forum Report No. 7: "Report on Candidate Extension Bands for UMTS/IMT-2000 Terrestrial Component", March 1999.

[8] "UMTS Market Forecast Study", Final Report for EC DG XIII, Analysys/Intercai.

[9] Including Annex A-B, February 1997.

[10] "Global Circulation of IMT-2000 Terminals", ERC Report 60, September 1998.

[11] UMTS Forum Report No. 8: "The Future Mobile Market", March 1999.

[12] UMTS Forum Report No. 9: "The UMTS Third Generation Market – Structuring the Service Revenue Opportunities", October 2000.

[13] UMTS Forum Report No. 10: "Shaping the Mobile Multimedia Future", October 2000.

[14] UMTS Forum Report No. 11: "The UMTS Third Generation Market – Structuring the Service Revenue Opportunities", October 2000.

[15] UMTS Forum Report No. 12: "Naming,Addressing and Indentification Issues for UMTS", February 2001.

[16] UMTS Forum Report No. 13: "The UMTS Third Generation Market –Phase II", April 2001.

[17] UMTS Forum Report No. 14: "Support of Third Generation Services using UMTS in a Converging Network Environment", February 2002.

[18] UMTS Forum Report No. 15: "Key Components for 3G Devices", January 2002.

[19] UMTS Forum Report No. 16: "3G Portal Study", November 2001.

[20] UMTS Forum Report No. 17: "The UMTS Third Generation Market Study Update", December 2001.

[21] UMTS Forum Report No. 18: "UMTS 3G Market Forecasts", February 2002.

[22] UMTS Forum Report No. 19: "Benefits and Drawbacks of Introducing a Dedicated Top Level Domain within the UMTS Environment", March 2002.

[23] UMTS Forum Report No. 20: "IMS Service Vision for 3G Markets", May 2002.

[24] UMTS Forum Report No. 21: “Charging, Billing and Payments Views on 3G Business Models”, July 2002.

[25] UMTS Forum Report No. 22: “Impact and Opportunity: Public Wireless LANs and 3G Business Revenues”, July 2002.

[26] UMTS Forum Report No. 23: “A Harmonized Frequency Solution for Early Implementation of UMTS/IMT-2000 in Central and South American Countries, July 2002

[27] UMTS Forum Report No. 25: “WLAN Spectrum Report”, updated May 2003

[28] UMTS Forum Report No. 26: “Social Shaping of UMTS- Preparing the 3G Customer”, January 2003

[29] UMTS Forum Report No. 27: “Strategic Considerations for IMS-the 3G Evolution”, January 2003

[30] UMTS Forum Report No. 28: “Relative Assessment of UMTS TDD and WLAN technologies”, March 2003