

5

Evolution of Mobile Devices and Operating Systems

5.1 Introduction

Mobile devices with wireless network interfaces have gone through a tremendous evolution in recent years. From around 1992–2002, the main development goal was to make these devices smaller. While during that time the form factor of phones shrank considerably, voice telephony and SMS texting remained the main applications and overall functionality changed very little. By around 2002, technology had developed to a point where it became impractical to shrink phones any further from a usability point of view. The Panasonic GD55 is one of the smallest mobile phones ever produced, with a weight of just 65 g, and is smaller than a credit card [1]. To demonstrate the evolution that had taken place in only 10 years, Figure 5.1 shows one of the first GSM phones, the Siemens P1 of 1992.

Once devices could not shrink, any further development has concentrated on adding additional multimedia functionality to mobile devices. At first, black and white displays were replaced by color displays, and display resolutions quickly rose from 100×64 pixels over 640×360 pixels to very high resolutions such as 960×640 pixels on 3.5–4 in. screens. Pixels thus have become so small that individual pixels cannot be seen anymore at a normal viewing distance. High-resolution color displays are a prerequisite for all other functionalities that have been added to mobile phones since. These functionalities include cameras, multimedia mobile e-mail and web browsing, video streaming, and social network interaction, just to name a few.

High-resolution color displays, high processing power with low power consumption and an increase in available memory and storage space have given rise to a number of wireless mobile device categories, whose purpose and range of functionalities has extended and shifted over the years. Today, the most important ones are:

- **Smartphones** — a smartphone can be defined as a combination of a mobile phone, what was formerly referred to as a (non-connected) Personal Digital Assistant (PDA), and an extension of previously desktop-based social media web services to the mobile world. Smartphones now usually include a high-resolution camera for taking pictures and videos, GPS and compass functionality as well as motion sensors, and various

3G, 4G and Beyond—Bringing Networks, Devices and the Web Together, Second Edition. Martin Sauter.
© 2013 John Wiley & Sons, Ltd. Published 2013 by John Wiley & Sons, Ltd.

network interfaces such as high-speed cellular interfaces, Bluetooth, and Wi-Fi. These devices are usually shaped like mobile phones, but are slightly bigger to accommodate a larger screen and additional hardware.

- **Pads/tablets**—over many years, the computing industry has been trying to scale down desktop PCs and make them mobile and portable by adding a touch-sensitive display and adapting the operating system to make its operation touch friendly. This device category did not become successful; however, until screen resolution, power consumption, and battery capacity enhancements were combined with the idea of adapting the user interface (UI) of touch-based smartphones for this type of device rather than using a desktop UI. With sufficient processing power for full-screen web browsers with similar functionality as is found in desktop and notebook PCs and web services adapted for touch-based input, tablets now fulfill many functionalities that were formerly only used with either a smartphone while being underway or with a PC when being at a desk. With cellular and Wi-Fi connectivity, tablets have become an ideal tool for multimedia consumption and for staying connected with friends via email, instant messaging, and web-based social networks without the need to sit at a desk.
- **Netbooks**—formerly also referred to as ultramobile PCs, the idea behind this product category is to reduce the size of a typical notebook while keeping its main characteristics such as the use of a desktop operating system, near full size keyboard, and only a slightly reduced display resolution. Netbooks have a typical screen size of 10–11 in. and a power-efficient CPU, and the mainboard design enables long battery operation times at the expense of processing power. The first models were not very successful as storage and processor capacities were too small for the requirements of Microsoft's Windows operating system. Asus was the first company that developed devices for this category [2]. Instead of using Windows, Asus initially used a Linux-based operating system, which is less resource hungry. Later versions then became powerful enough to host Microsoft's Windows operating system. Even though those devices have been available for several years, the main means for connectivity is still the Wi-Fi interface. Built-in cellular connectivity can only be found in a few models. This is mostly because of the additional price of the cellular network card, which would significantly increase the typical sales price of €250–300 or less. It has thus become quite common to use netbooks and notebooks with a cellular modem via the Universal Serial Bus (USB) port, often referred to as a "3G dongle," or via Wi-Fi tethering to a mobile phone that acts as a Wi-Fi to cellular network bridge to the Internet. Today, netbooks compete with devices from the tablet category, and interest in them has diminished to some extent. While tablets are ideal for information and media consumption away from the desk, the strength of netbooks is their full keyboard integration and desktop operating system, which makes them preferable for many creative tasks that require text input. A significant part of this book, for example, was written on a netbook.
- **Ultrabooks**—devices in this category are usually slightly larger than netbooks with a typical screen size of 13 in. but are significantly thinner and still very light in weight without compromising on battery operation time and processing capacity. This comes at the expense of a significantly higher sales price than netbooks, usually in the order of €800–1200. As with netbooks, USB modems or Wi-Fi tethering is used to connect to the Internet when not at home or at the office.

- **Wireless computing equipment**—a well-established trend for home and office networks is to untether computer equipment such as printers and hard drives (or Network Attached Storage (NAS)) using Wi-Fi. With Wi-Fi chips having become a commodity, the additional price for consumers has dropped significantly and such devices are becoming more and more popular. This device category is different from those previously mentioned because the aim of equipping them with wireless interfaces is not mobility but to reduce the amount of cables in home and office environments. A further advantage of having wireless access to such devices is that it also makes them usable from the mobile devices mentioned above.

Most of the today's connected devices except netbooks and ultrabooks are based on a chip with a processor design from ARM [3]. Although many companies such as Texas Instruments, Marvell, ST-Ericsson, and Qualcomm design and manufacture chips for small devices, most are based on a CPU core licensed from ARM. On the desktop, Intel's x86 design dominates in a similar way. With both architectures now targeting sophisticated mobile devices, these two worlds are about to collide.

5.1.1 The ARM Architecture

The ARM design was initially targeted at ultralow-power embedded devices. As technology evolved so did ARM's processor design and it is estimated that an ARM processor



Figure 5.1 The Siemens P1, one of the first GSM telephones in 1992.

core is used in 95% of mid- to high-end mobile devices today [4]. The current ARM-Cortex A9 and A15 platforms used in high-end smartphones and tablets is the result of a bottom-up approach, as it has evolved from earlier platforms for simpler devices. Today, all mobile devices of mobile giants such as Sony, Nokia, LG, Samsung, and Apple are ARM powered. This shows the flexibility of the ARM architecture since requirements range from voice telephony with very low power requirements to multimedia devices that trade in a higher power consumption for higher processing capabilities.

Today, a lot of operating systems support the ARM architecture. Examples are fully embedded operating systems of low-end to mid-range mobile devices to operating systems for smartphones such as Symbian and Windows Phone. In addition, ARM processors are also used with operating systems that were initially developed for desktop computers such as Linux and Windows 8. Linux is a relatively new operating system for mobile devices as the first mass market device based on Linux was only shipped in 2008 as part of Google's Android operating system [5]. The advantage of using Linux as an operating system for mobile devices is that a significant amount of code can be shared between the desktop and the mobile version of the operating system, as relatively little code directly deals with the differences of the x86 architecture found in the PC world and the ARM architecture found on mobile devices.

It should also be noted at this point that unlike companies such as Intel or AMD, ARM does not produce processor chips themselves. Instead, ARM licenses its processor designs to other companies such as Qualcomm, Samsung, Mediatek, Marvell, Texas Instruments, and many others, which then include the processor designs in their own chip designs. Companies such as Intel, however, go one step further and also design the chips that include their processor designs.

ARM offers several types of licenses. The most basic license only offers the logic of a function block such as the CPU, a bus system, or the graphics chip for use with a chip design program such as Verilog. This is referred to as a soft-macro [6]. The licensees then use those function blocks with self-designed additional circuitry or function blocks bought from other companies and create a physical chip implementation, which is optimized for a certain production process, performance, power consumption, and die size. ARM also offers licenses that already include those steps. Such function blocks are then referred to as hard-macros. And finally, ARM also licenses the ARM architecture itself and allows companies to modify, design, and optimize their own CPU cores and other function blocks. ARM-compatible software still runs without modification on such modified processors but licensees have the opportunity to make enhancements to the architecture independently of ARM. Marvell and Qualcomm are companies that design their own ARM processors instead of buying a finished design.

5.1.2 The x86 Architecture for Mobile Devices

Intel is at the other end of the spectrum and is keen to play a major role in the mobile space with its x86 processor architecture. A few years ago Intel tried to get a foothold in the mobile space by licensing ARM technology and building a product line around that architecture. In the meantime, however, Intel has abandoned this approach and has been refining their x86 architecture for low power consumption and size for several years. In 2012, the size, processing speed, and power consumption of the chipset was for the

first time balanced enough for a smartphone-sized device. First prototypes were shown running an x86 version of Android on a form factor smartphone [7], and commercial products based on this design appeared shortly afterward on the market. This rather late competition to ARM's dominance in the mobile space is the result of Intel's approach that is directly the opposite of ARM's as they had to streamline a powerful desktop processor architecture for smaller devices.

Using an x86 platform for mobile devices has the advantage that even fewer adaptations are required for operating systems such as Linux and Windows to use them on mobile devices compared to the ARM approach described above. In the case of Android, most applications are executed in a virtual machine based in Java and only compiled to native code at runtime, so the same executable runs without modification or the need for recompilation on both CPU architectures.

At the time of publication, Intel and ARM have come quite close in terms of performance and power consumption and the two architectures are now competing for use in high-end mobile devices.

5.1.3 *Changing Worlds: Android on x86, Windows on ARM*

The significant advances in the mobile space in recent years and Intel's inability to establish themselves with their x86 architecture in the mobile domain over many years have also had consequences for traditional alliances formed in the PC space.

Over many years, Microsoft has only developed its Windows desktop operating system for x86-based architectures. With tablet devices having become attractive to end users starting from around 2011 and no company in sight to deliver power-efficient x86-based processors, Microsoft had to choose between extending their relatively novel Windows Phone operating system based on ARM to tablets or to scale down their Windows desktop operating system and adapt it to the ARM architecture. Microsoft chose to do the latter and has developed a new version of the Windows desktop operating system that can also be run on ARM-based devices such as tablets. Such a move would have been impossible only a few years earlier and demonstrates the significant increase in processing power that was achieved on the formerly low-cost low-processing power ARM architecture.

A new UI was developed based on the Windows Phone smartphone UI to complement the existing desktop UI in an attempt to integrate the mobile and desktop computing worlds in a single Windows operating system. This demonstrates how the rise in computing power in mobile devices also has an effect on desktop computing and how the industry is integrating the formerly disparate worlds of low-power mobile devices and high-power desktop computing into a single space.

New alliances are also formed on the x86 side with Android having been ported to the x86 architecture, as described in Section 5.1.2. This in effect enables Intel and other companies producing x86-based processors to move into the high-growth smartphone market, which further blurs the line between mobile and desktop computing.

A further positive effect of the competition between the ARM and the x86 architecture is likely to be further accelerated innovation and falling prices as mobile device manufacturers can now choose between two camps with each one trying to stay ahead of the other with further innovations in the areas of power consumption, processing speed, graphical capabilities, and integration of other components into a single chip.

5.1.4 From Hardware to Software

The following sections now take a look at how mobile device hardware has evolved over recent years and give an introduction to both hardware architectures mentioned above. Different parts of the world use different frequency ranges for wireless communication. This chapter therefore takes a look at the global situation and describes the impact on mobile hardware design and global usability of devices. Adding a Wi-Fi interface to mobile devices has been another important step in the evolution of wireless communication and this chapter will discuss the profound impacts of this step on networks and applications. Finally, this chapter takes a look at the Android operating system for mobile devices.

5.2 The System Architecture for Voice-Optimized Devices

In the entry level segment, mobile phones are sold today both in developed markets and emerging economies that are optimized for voice communication. While the functionality of such phones has not changed much in the past decade, prices have been on a steady decline due to much higher production volumes and reducing the number of required chips and electronic components. This is referred to in the industry as reducing the Bill of Materials (BOMs). Figure 5.2 shows a block diagram of a typical voice-optimized mobile phone computing platform which is offered by many companies. The example in this book is based on Freescale Semiconductor's GSM i.200-22 hardware platform [8], which is optimized for voice communication and even excludes functionalities such as basic General Packet Radio Service (GPRS).

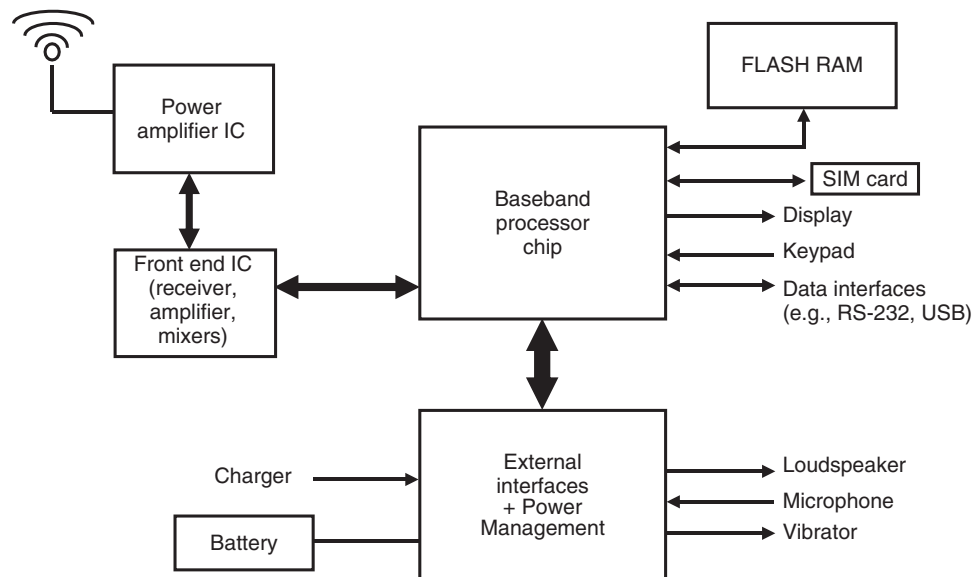
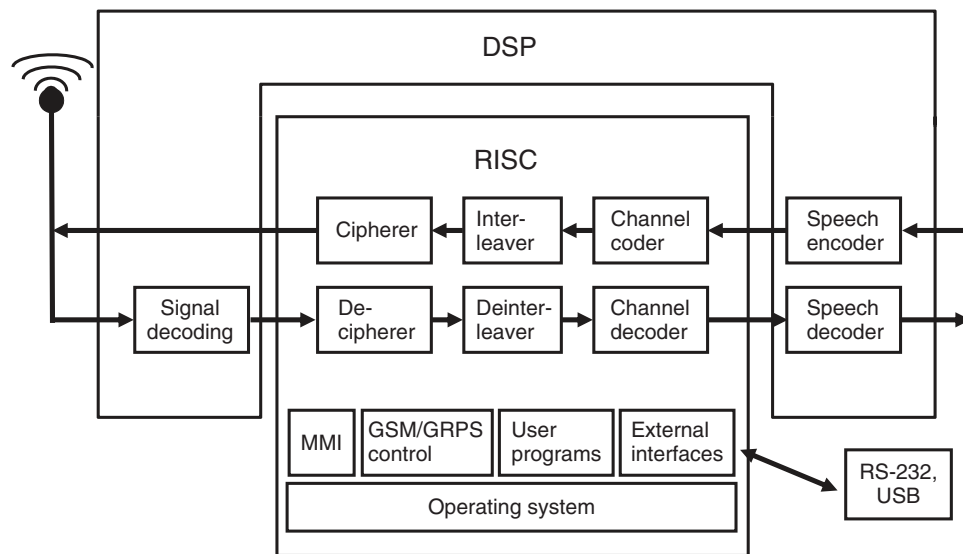


Figure 5.2 Block diagram of a voice-optimized mobile phone hardware platform. (Reproduced from *Communication Systems for the Mobile Information Society*, Martin Sauter, 2006, John Wiley and Sons, Ltd. Ref. [9].)

In addition, the baseband chip contains a Digital Signal Processor (DSP) of Motorola's 56x family, which is clocked at 130 MHz. DSP microprocessors are optimized for mathematical operations and run software which is usually designed for specific tasks. Figure 5.3 shows how the RISC CPU and the DSP are used in combination in a mobile phone. The DSP chip is responsible for decoding the received signal from the network and for encoding and decoding the voice signal. There are two main advantages of performing these tasks on the DSP and not on the main processor:



7

- A DSP has an optimized instruction set for mathematical operations required for dealing with codecs and decoding analog radio signals that have been digitized by an analog-to-digital converter.
- Encoding and decoding external signals is a continuous process and must not be interrupted by other activities such as reacting to user input or updating the display.

A typical voice call is treated by the baseband chip as follows:

- The analog input signal from the microphone is digitized and sent to the DSP chip.
- The DSP applies speech coding and forwards the data to the ARM RISC CPU.
- The ARM processor then packetizes the data stream, adds redundancy to the data (channel coding), changes the order of the bits so block errors can be more easily corrected on the other end (interleaving), encrypts the result and then sends the packet over the air interface.

In the reverse direction, the same actions are performed in the reverse order. In addition, the DSP performs signal decoding. This is a complicated task since the signal sent by the base station is usually distorted by interference. To counter these effects, packets contain training bits (in the case of GSM) that are set to predefined values [9]. These are used by the DSP to build a mathematical model of how the signal was distorted. The mathematical model is then applied to the user data around the training bits to decrease the transmission error rate.

In addition to the tasks above, the ARM CPU is responsible for interaction with the user (keyboard, display), to execute user programs such as Java applications, and to communicate with external devices (e.g., a computer) via interfaces such as USB. As all of these tasks have to run in parallel; a multitasking operating system is required that is able to give precedence to repetitive actions concerning communication with the network and assign the remaining time to less time critical tasks.

For executing programs, about 250 kb of RAM is typically available on the baseband processor chip. In addition, about 1.7 Mb of nonvolatile memory (ROM, Read Only Memory) is available. If more memory is required, the chipset offers an external memory interface that can be used to connect additional RAM and ROM (e.g., flash memory). A 225-pin multiarray ball grid array connects the baseband chip via a 13×13 mm connection field to the other components of the device (cf. Figure 5.2). Other important components of the baseband chip are the module to access the Subscriber Identity Module (SIM) card and a display module for a monochrome or color display.

In addition to the digital processing functionality of the baseband chip, other analog components such as power amplifiers, signal modulators, and functionalities to convert and control power for the device are required. These are implemented in separate chips as analog functionalities require a different manufacturing technology from the purely digital functions of the baseband chip.

5.3 The System Architecture for Multimedia Devices

The design intent for a voice centric mobile device chip set is to strip down the functionality to the bare minimum to reduce the price as much as possible. For high-end wireless

mobile multimedia devices, however, the aim is to include as many functions as possible in the chipset. At the same time the device must consume as little power as possible in idle mode in order to achieve acceptable standby times. The chipset has to find a balance between power efficiency and performance while the user interacts with the device.

There are three major building blocks of a high-end mobile chipset today. The first is the application processor unit, which usually consists of one or more CPU cores usually based on a 32-bit ARM architecture. Especially with the ARM Cortex CPU architecture introduced in 2005, ARM has increased performance by introducing a superscalar design that increases the number of execution units a machine instruction passes during its execution. This way, several machine instructions can be processed simultaneously as each can be in a different stage of execution. According to ARM, this increases the performance by a factor of 2–3 compared with the previous ARM processor generation at the same clock frequency. Performance gains for audio and video decoding are achieved with an extension referred to as NEON that allows application of the same operation with a single instruction to several variables simultaneously. This is used, for example, by Android's WebM library to decode this type of video format [11]. Another feature now prevalent in mobile CPUs is a floating point unit to perform non-integer calculations in hardware.

After many years of refinement, Intel presented a design to enter the mobile chipset domain with its x86 architecture as well. Its platform, which is referred to as "Medfield," seems for the first time be able to compete with the ARM design. A comparison between "Medfield" and current high-end ARM designs such as the ARM-Cortex-A9 and A15 can be found in [12]. In addition, Intel has ported the popular Android operating system to its x86 CPUs as well, which significantly helps to make future x86 platforms popular in the mobile domain if power consumption and processing speed develop along similar lines as those of the ARM architecture.

The second major building block of a chipset is the graphics processing unit (GPU). This processing unit is specifically designed to efficiently handle 2D and 3D graphical operations and effects. The calculations required for rendering of web pages and graphical effects such as scrolling a web page and zooming into specific parts, blending screens when changing from one application to another, and rotating the screen when the user changes the orientation of the device are mostly performed in the GPU. Unlike CPUs, which are optimized for sequential program streams, the GPU is optimized to perform many similar operations that are not dependent on each other in parallel. The general functioning of a mobile GPU is the same as that of a GPU in the PC world but its power and processing capabilities are scaled down to adapt to the limited power availability on mobile platforms as well as the limitations imposed by passive (i.e., fan-less) cooling. Today, there are several companies whose GPUs are commonly used in practice. ARM has designed its own GPU, which it has named "Mali." Nvidia, initially a graphics card manufacturer in the PC domain, has also developed a mobile GPU family, which it uses as part of its "Tegra" line of integrated chips for mobile devices. Imagination Technologies "PowerVR" is a division specifically focusing on mobile device GPUs, and its designs are also commonly found in mobile chipsets, with both ARM and Intel CPUs. And finally, Qualcomm also has its own GPU design referred to as "Adreno," which is integrated into their line of "Snapdragon" chipsets. While ARM CPUs from different manufacturers all share the same instruction set, this is not the case for GPUs. To make application programs

compatible with different GPUs, a standardized Application Programming Interface (API) is required. This is discussed in more detail in Section 5.4.

The third major building block in a mobile device is the cellular modem, sometimes also referred to as the “baseband” processor. It includes all digital components required to communicate with a cellular network. Analog parts such as power amplifiers, filters, and up- and downlink signal multiplexers are separate components on the motherboard as a different manufacturing process is required for such components. The baseband processor usually consists of an ARM-based processor and additional signal processing components such as a DSP, as described above for voice-optimized devices. While such low-end devices usually only include a GSM modem, baseband processors have significantly grown in complexity and processing power requirements as they now also include software and hardware to process much more complex radio signals than those of GSM, such as, for example, High-speed Packet Access (HSPA) and Long Term Evolution (LTE) (cf. Chapter 2). The baseband processor has its own operating system and communicates via a high-speed (HS) serial connection with the application processor unit on which operating systems such as Android, iOS, Symbian, Windows Phone, and so on are executed. The serial connection over which user data and modem commands are exchanged makes the baseband processor completely independent from the application processor block. When operating systems such as Android are used on the application processor, a modem driver software module simulates several serial connections, typically one for user data and one for modem control commands and feedback messages. Modem commands are, for example, the establishment of an Internet Protocol (IP) connection, and feedback messages contain information about signal strength and other frequently required network parameters. From a mobile operating system point of view, the cellular modem is thus used in a very similar way as a cellular USB dongle connected to a PC and allows the mobile operating system to be easily adapted to different baseband processor implementations as the cellular modem driver module is the only piece of software affected when using different modems. Baseband processors are developed by several companies such as Qualcomm, ST-Ericsson, Marvell, Renesas, and Nvidia.

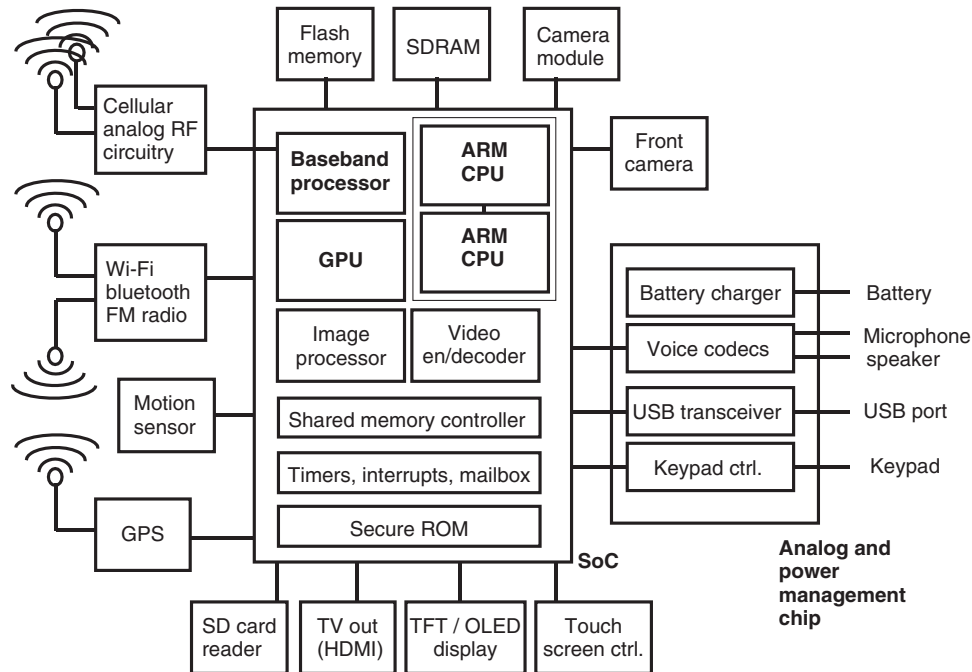
In the past, baseband processor, application processor, and graphics processor could often be found in dedicated chips. An example is the Nokia N8 that was released in 2010. It was built with a dedicated baseband processor chip of Texas Instruments, a dedicated application processor chip by Samsung based on an ARM11 core, a predecessor of the current ARM-Cortex platform, and a Broadcom graphics 3D processor chip. There is a strong trend, however, to combine all three components on a single System on a Chip (SoC) to save cost, reduce power consumption, and shrink the overall size of the circuit board. Table 5.1 shows examples of fully integrated systems on a chip and their manufacturers.

It is interesting to note that none of the companies listed in the table develop complete mobile devices themselves and also do not develop other components required in a device such as touch screens, displays, batteries, casings, the scratch-resistant glass, camera modules, and so on. In other words, a mobile device contains components from many different manufacturers and the company owning a device and whose logo appears on it is mainly acting as an integrator of the different hardware and software components. When operating systems such as Android and Windows Phone are used, the companies are not even the developer of the operating system software, as that is again done by different companies such as Google and Microsoft.

Table 5.1 Examples of all-in-one system on chip manufacturers

Manufacturer	Platform name	CPU	GPU	Baseband modem
Qualcomm [13]	Snapdragon	ARM, Qualcomm design, “Scorpion,” “Krait”	Qualcomm “Adreno”	Qualcomm
ST-Ericsson [14, 15]	NovaThor	ARM Cortex-A9	PowerVR and ARM “Mali”	ST-Ericsson
Renesas [16]	Renesas	ARM Cortex-A9	PowerVR	Renesas
Nvidia [17]	Tegra	ARM Cortex A9	ULP GeForce	Nvidia (Icera)
Mediatek [18]	Mediatek	ARM-Cortex-A9	PowerVR	Mediatek
Intel [12]	Atom	x86	PowerVR	Intel (Infineon)

While CPU, GPU, and baseband modem are the most important components of a mobile chipset, they are by no means the only ones. A modern SoC has a variety of other special hardware units, which are either integrated into the SoC itself or are placed on the circuit board in chips of their own. An example of a possible configuration is shown in Figure 5.4. The example SoC contains two application processor CPUs, typically driven at clock rates of around 1.5 GHz today, the GPU, and the baseband processor on the main chip as discussed earlier.

**Figure 5.4** Block diagram of a multimedia chipset for a mobile device.

In addition, the SoC in this example contains an image processor unit that is used for processing the input stream delivered from external camera modules. Camera sensors with resolutions of 12–16 MP are supported by high-end chipsets today but not necessarily used to their full capabilities. The image processor converts pictures to compressed jpeg formats on the fly. This significantly reduces the load on the CPU since this computationally intensive task is performed entirely in this dedicated unit.

The video encoder/decoder unit supports the CPU by encoding video streams delivered from the camera to MPEG4 formats. The unit is also used to decode and display videos that have been recorded earlier or downloaded to the device and stored on either internal flash memory or a memory card. Most high-end wireless devices have a main camera on the back of the device to take high-resolution pictures and videos and a small camera on the front for video calls. Consequently the baseband chip has two camera interfaces.

All processing units require access to the main memory to store data and to communicate with each other. A shared memory controller synchronizes the memory requests of the different units and delivers the data over a common bus. The chipset also contains dedicated hardware for mailboxes, which are used for communicating between different tasks, and interrupts, which are used to inform special program handlers of the operating system of external events (e.g., the user has pressed a button).

Finally, the SoC also contains a secure ROM to store confidential information and software, which cannot be accessed from outside the chip. This can be used, for example, to protect the unique equipment identity of the device (IMEI, International Mobile Equipment Identity) and to enforce a SIM lock limitation to bind the device to a specific network operator. Furthermore, the software loaded from the secure ROM when the device is reset can also be used to ensure that only a certified version of the operating system is loaded on the device [19].

In addition to the battery control and a USB transceiver, the analog and power management support chips, shown on the right in Figure 5.4, handle microphone, speaker and keypad input and output. Displays on the other hand are directly connected to the SoC, which in addition also features a TV out signal to connect the device to a television set. In recent years touch screen input has become another important feature and hence, SoCs now provide a direct input for touch panel sensors.

Memory for program execution and long-term storage is usually provided in separate chips and not directly as part of the SoC. Typical sizes of Random Access Memory (RAM) for program execution are 512 MB to 1 GB. For long-term storage of the operating system, applications, and user data, mobile devices usually include several gigabytes of flash memory, which can usually be extended via memory cards of up to 64 GB depending on the device.

The left side of Figure 5.4 shows how the SoC is connected to typical network interfaces required for high-end mobile multimedia devices. At the top, several antennas and analog RF circuitry are shown that connect to the internal baseband modem for GSM, CDMA (Code Division Multiple Access), UMTS (Universal Mobile Telecommunication System), and LTE. More than one antenna is shown as it is usually required to be able to communicate with different types of cellular networks using different frequency ranges. The impact of the multifrequency approach is further discussed in the following sections.

A Bluetooth network interface has become indispensable for most mobile devices today and Wi-Fi interfaces are also becoming more popular. Often, both functionalities are

included on the same chip and require only a single antenna as both technologies use the 2.4 GHz ISM (Industrial, Scientific, and Medical) frequency band [20]. In addition, such chips also often contain an FM radio receiver for which a dedicated antenna is required as FM radio is broadcasted between 87 and 108 MHz. As this requires a long antenna due to the longer wavelength of the signals, the headset cable is usually used for the purpose. This means that the FM radio application can only be used when a headset is plugged in.

GPS receivers and 3D motion sensors have also become a standard in today's smart-phones. These are either included in the SoC or as a separate chip.

5.4 Mobile Graphics Acceleration

In recent years, the GPU in mobile devices has become at least as important as the CPU itself because of the increasing screen sizes and display resolutions, direct touch screen interaction that requires fast reaction to input gestures, smooth scrolling through web pages, rich UI effects such as blending between screens of different applications, and rotation of the display content when the user turns the device and 3D gaming. Such operations are preferably calculated on a dedicated hardware unit, which is optimized for parallel floating point operations and graphics algorithms already included in the hardware.

As discussed in Section 5.3, there are a number of GPU manufacturers today with different approaches of how the graphical rendering process can be enhanced while at the same time using as little power as possible. As a consequence, each GPU requires the operating system to interact with it differently. A standardized interface is therefore required both for applications and the operating system running on the CPU to manipulate the screen content. As in the PC world, a GPU operating system driver allows to introduce a new GPU by only modifying the graphics driver, while the rest of the operating system remains unaltered.

5.4.1 2D Graphics

Mobile GPUs offer graphics acceleration in 2D and 3D modes. While 3D graphics are mostly used for gaming, 2D graphics is the default way of presenting information to the user in most applications. 2D graphics acceleration works with different 2D layers that are stacked on top of each other. In the web browser, for example, the background layer could contain the web page, while independent layers on top of the background contain other elements such as buttons. The web page on the background layer is usually much larger than the screen size; so, only a portion of it is shown at a time. When the user then scrolls through the page, the CPU simply instructs the GPU to show a different part of the already prepared background layer. This requires little processing power on the CPU and little communication with the GPU as the bitmap for the background layer is already present in the GPU's memory. The GPU can therefore create an appealing scrolling effect with little computational effort involved. The buttons on higher layers remain in their place on the screen even when the web page is scrolled. When the GPU renders a new screen view once every few milliseconds, it uses a different part of the background layer from its memory, which creates the scrolling effect and then just superimposes the graphical elements of higher layers, which do not require a modification of the background layer in memory.

The CPU, on the other hand, is responsible for constructing the full web page once out of the HTML code, JavaScript, and the pictures it receives from the web server. The result is then transferred to the GPU as the background layer. To speed up the process, the CPU usually renders the part of the web page that is visible to the user at first, transfers this part to the GPU so it can be displayed, and then continues to assemble the remaining parts of the page that can only be seen when the user scrolls through the page.

In most cases, the CPU has finished creating all parts of the web page before the user starts scrolling, which makes the delay invisible to the user. Once all parts of a web page are fully rendered and transferred to the GPU memory, the CPU then only reacts to user input by instructing the GPU to show a different part of the page. JavaScript code on the web page remains the responsibility of the CPU, and should the JavaScript code change the web page at runtime the CPU modifies the affected section of the web page and updates the background layer in the GPU's memory.

When displaying a web page that was originally designed for a large PC screen, it is usually presented to the user in a minimized form. The user can then zoom into specific parts of the web page. This part is then enlarged and if the original text would be cut on the right or left because of the magnification level, it is usually reformatted to fit the zoom level. Again, the work is split between CPU and GPU to create an appealing effect during the zooming operation. When the user starts the zoom, the CPU instructs the GPU to zoom into the background layer. The pictures and the text usually blur during this operation as the background layer was not optimized for a higher zoom level. When the user has finished setting the zoom level, the CPU reflows the text in this part of the web page, assembles a higher resolution background layer representing the web page in this zoom factor, and sends the result to the GPU. The GPU then displays the part of the web page the user has zoomed into and the CPU continues calculating the remaining web page; so it is already present in the GPU's memory when the user starts scrolling through the web page.

5.4.2 3D Graphics

For 3D games that project and animate a 3D world on the screen, a different approach is required than that for the 2D layering described above. For such purposes, the OpenGL-ES (Open Graphics Library for Embedded Systems) [21] API has become very popular on mobile operating systems such as Android, Symbian, and iOS. Originally developed and used in the desktop world, OpenGL-ES is a scaled-down version of OpenGL to take the reduced capabilities of mobile GPUs into account. Unlike in the 2D world where the web page is rendered on the CPU and then sent as a background layer to the GPU, 3D worlds are completely rendered on the GPU. The CPU is mainly responsible for defining where objects are located in a 3D space, what their properties are, from which location the objects are viewed by an observer, and how the objects move and change their shape over time. The following paragraphs now give a brief introduction of how this is done in practice. Further details can be found in [22].

To construct a 3D world, OpenGL-ES has standardized commands to define shapes, usually triangles, by defining where their corners (vertices) lie in a 3D x,y,z coordinate system. By combining the triangles, any shape can be constructed. The smaller the triangles, the more are required to form an object and the more realistic it will appear to the user. For defining very complex shapes, such as a racing car in a game, for example,

thousands of small triangles are defined in a floating point number array, which is then given via an OpenGL-ES command to the GPU. The GPU is then responsible to render an image based on the coordinates of each triangle's vertices. Triangles can be colored by assigning an RGB value to the area of each triangle. Bitmaps can be used as textures for triangles. Again, it is the graphics processor that does the image processing. The CPU just transfers the textures and the information of how to apply them to the different triangles to the GPU, which will then do all the computational work required to apply the texture to the shapes defined in the x,y,z coordinate system.

Two more things are required for the GPU to render a 3D scene. First, the GPU needs to know the location of light sources that illuminate the objects, the type of light, its direction, and several other parameters. These are then used to realistically illuminate a scene including the shading created by objects. And finally, the GPU needs to know from which point of view the scene should be shown, in which direction the viewer is looking, and the parameters for his field of view. If, for example, a car is rendered, the viewer could see the car from the front, from the back, from the left, from above, from below, and so on. From each point of view, the rendered image looks completely different. Also, the viewer could be very close to the car and would hence only see a part of it or he could be quite far away. In this case, he would see the car and perhaps also further objects or parts of them beyond or to the sides of the car. Objects more distant appear smaller than objects closer to the viewer. All of these properties can be described by a viewer that is a certain distance away from a rectangle he looks through. The distance to the rectangle determines which part of the 3D world he can see. If he is close to the rectangle, he has a wide angle of view. If he is distant to the rectangle, his angle of view is very narrow. This is shown in Figure 5.5. And finally, the view is limited by a plane in the background. All objects behind the plane are invisible. The viewer, the viewing rectangle, and the limiting plane in the background form a pyramid, with the user being at the peak of the pyramid and the viewing window and the plane in the background forming a trunked pyramid,

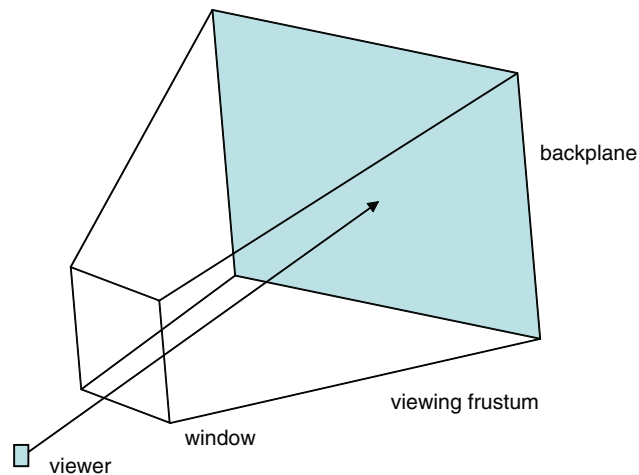


Figure 5.5 The viewing frustum.

which in mathematics is referred to as a frustum. Hence, the definition of the viewing frustum in computer graphics refers to the definition of the view onto a 3D space that the GPU then maps on the 2D display.

As long as the objects in the scene do not change or move, the CPU can instruct the GPU to render the 3D world from a different point of view by only changing the frustum parameters with a simple instruction. The GPU then autonomously recreates the 3D world as seen from the different point. Interesting effects can thus be created, such as flying through the world without the CPU being required to redefine the scene as the user flies through the world. This greatly reduces the work required from the CPU for 3D animations and it can thus concentrate on the calculation of object movements in the 3D worlds and updation of the descriptions of the shapes. In many games, many objects move at the same time. Especially in such situations, several CPU cores are very beneficial, as this means that the movement of several objects can be calculated in parallel instead of sequentially on a single CPU.

5.5 Hardware Evolution

From an end user's point of view, little has changed in the PC world in the past five years. Processors have become faster, graphics capabilities continue to evolve, but usage scenarios and applications are still the same. Mobile computing, however, has evolved significantly since the first edition of the book was published and an inflection point has been reached as mobile Internet access is now widely used. While only few had used Internet-based services on mobile devices only five years ago, over half of the devices sold in 2012 were smartphones in some countries [23]. Table 5.2 compares the typical

Table 5.2 Smartphone hardware comparison over five years

	2007	2012
Typical device	Nokia N95	Samsung Galaxy S-II, iPhone 4G
CPU clock speed	332 MHz	1.2 GHz
RAM	128 MB	512 MB to 1 GB
Screen resolution	240 × 320 pixels	800 × 480, 960 × 640 pixels
Flash	8 GB	32 GB
Camera	5 MP	8–12 MP, 41 MP (Nokia 808)
Video recording	640 × 480 (VGA)	1280 × 720 (Full HD)
Cellular connectivity	3.6 Mbit/s down, 384 kbit/s up	>20 Mbit/s downlink, >5–15 Mbit/s uplink
Infrared data exchange	Common	No longer included
Wi-Fi	New concept, few devices	All smartphones
GPS	New concept, few devices	All smartphones
Motion sensor/compass	Not included	All smartphones
Touchscreen	Few models, slow reaction, single touch	All models, multi-touch
Weight	120 g	120 g

hardware features of a high-end device sold in 2007 when the first edition of the book was written with those of a typical smartphone in 2012. It is likely that this evolution will continue at a similar pace in the years to come and the differences shown in the table give an idea of the potential capabilities devices will have in five years in the future. The following sections now give an overview over the current and future developments of a number of mobile device components.

It should be noted at this point that ever more sophisticated hardware and also software do not only benefit devices in the smartphone category but may also bring about new device types, new applications and usage scenarios, and new types of users. Two recent examples of this are the rising popularity of tablets and e-book readers. While such devices have existed already many years ago, they had only become appealing to a wider public once display resolution, size, and weight reached acceptable values and touch input became refined enough for easy use with little to no perceived delay between an action of the user and a response of the device.

5.5.1 Chipset

In the case of ARM-based chipsets such as those described in Section 5.3, the next few years will see further performance enhancement of all subcomponents. In 2008, most high-end devices using ARM and OMAP products were based on the ARM 11 processor family with processor speeds between 330 and 400 MHz. The modem, the CPU, and the GPU were often implemented in different chips. Current platforms based on the Cortex-A9 architecture exceed the 2008 performance level several times because of the four times higher clock speeds of 1.5 GHz and more, dual- or quadcore technology, and an enhanced processor core. Taken together, it is reasonable to assume a 10 times higher performance compared to 2008. Also, there is a clear trend to integrate the modem, the CPU, and the GPU units in a single chip as described in Section 5.3 and even further components that are now part of every smartphone such as Wi-Fi and Bluetooth support [24].

When looking at the PC sector, it can be noticed that processor clock rates have remained at around 2 GHz for several years now because of the high-power requirements and excessive heat generation at higher clock rates. Instead, processing speeds have increased by enhanced architectures such as hyperthreading, that is, simulating two virtual CPU cores in a single physical CPU and by adding additional CPUs to the die (multi-core operation). Another way of increasing overall processing speed is by using more transistors on the chip to form more complex processing units to reduce the number of clock cycles required to execute a command and by increasing the on-chip memory cache sizes to reduce the occasions the processor has to wait for data to be delivered from external and slow RAM.

While using more than one CPU in a device and increasing the clock rate undoubtedly increase the performance, it also leads to a higher power consumption not only when the CPUs are running at full speed but also when the CPUs are mostly idle. A number of different approaches exist to mitigate this problem. In Nvidia's 2012 Tegra design [25], four HS CPU cores are used to run four independent tasks in parallel. In practice, four cores are only required in few situations and therefore the design allows to deactivate and reactivate individual cores at run time to reduce power consumption. In addition, Nvidia uses a fifth core, which is referred to as a "companion core" that takes over when

only little processing power is needed, for example, while the display is off and only low-intensity background tasks have to be served. This is transparent to the operating system; so no software adaptations are required for the switchover. The reason why a fifth low-speed CPU is required instead of just lowering the clock speed of one of the fast CPUs is that the average power consumption is governed by two factors, leakage power and dynamic power. When processors are run at high clock speeds, the chip has to be optimized for low voltage and fast switching operations as the power requirement increases linearly with the clock frequency but in square with the voltage. Unfortunately, optimizing the chip for low voltage and fast switching operation increases the leakage power, that is, the power consumption when voltage is applied to a transistor even when its state is not changed for a long time. It is this leakage power that becomes the dominant power consumption source when the CPU is idle, that is, when the screen switched off, when only background tasks are running, and so on, and hence the processor clock speed is reduced. It is during such times that the main CPUs are switched off and the companion CPU is used. A different manufacturing process is used for the location where the companion CPU is located, which is less optimized for HSs but more optimized for lower leakage power. The companion CPU can thus only be run at clock speeds up to 500 MHz but has the advantage of having a lower leakage power when only slower clock frequencies are required. Switching back and forth between the companion CPU and the four standard cores is seamless to the operating system and can be done in around 2 ms.

Qualcomm has used a different approach in their Krait CPU architecture to conserve power. Instead of running all cores at the same clock speed, which has so far been the default option, each core can be run at a different speed depending on how much workload the operating system assigns to each of the cores. Rather than optimizing one processor for leakage power consumption, this approach conserves power by reducing the clock speed of individual processors when less processing power is required [13].

Other components such as flash memory and RAM for program execution are also likely to continue getting cheaper. In 2007, high-end smartphones featured 128 MB of memory and the design goal was to minimize the RAM in a device for cost and form factor reasons by as much as possible. In 2012, this has changed significantly and high-end smartphones were equipped with 512 MB to 1 GB of RAM. Even netbooks today running Windows 7 do not have more than 1 GB of memory, mostly because of a marketing restriction put in place by Intel to distinguish netbooks from higher priced notebooks. Both smartphones and low-cost netbooks are likely to have their memory “only” doubled in the next few years as memory for program execution is no longer a limiting factor in high-end user devices. In such devices, the limit is rather set by power consumption, battery capacity, weight, size, processing power, and graphics capabilities.

Storage memory in mobile devices has also increased significantly over the past years. While in 2008, 8 GB of built-in flash memory was state of the art in high-end devices, 2012 models include 16 or even 32 GB. In addition, memory card slots allow the extension by another 32 GB. As a result, local storage has almost become limitless for most applications such as the number of applications that can be installed, storage of pictures taken with the camera (1–2 MB per image), and storing data locally such as email file attachments, ebooks, and so on. Only few applications such as storing a complete music library on a device or taking high-definition (HD) videos with the built-in cameras, which require

around 100 MB (0.1 GB) of storage space per minute [26], have become applications that can fill up local storage.

After many years of competing formats, microSD has become the de-facto standard for flash memory extension cards for mobile devices. In 2012, the Secure Digital High Capacity (SDHC) version is commonly supported by mobile devices and allows a maximum flash memory size of 32 GB. One of the limiting factors is the FAT16 file system commonly used, which only supports 32 GB volumes. The next version of the microSD standard is referred to as Secure Digital Extended Capacity (SDXC) and enables storage capacities of up to 2 TB. In 2012, storage capacities of up to 64 GB were available on microSD cards. Also, the physical interface itself has been enhanced to allow data transfer rates beyond those of USB 2.0, if supported by the host device and the card.

5.5.2 Process Shrinking

Decreasing the power consumption and increasing the clock speed and the number of components and subsystems on a chip require transistors on the chip to become smaller. Most microprocessors, static RAM (memory), and image sensors are based on Complementary Metal–Oxide Semiconductor (CMOS) technology today. In 2007, chips using ARM11 cores were manufactured with a 90 nm CMOS process. The 90 nm length refers to the average half-pitch size of a memory cell. In 2012, high-end smartphone chipsets are manufactured in 45 nm technology and the next step to 28 nm technology is already in sight for next generation chipsets [27]. Traditionally this value has considerably shrunk over time. In 1972, the Intel 4004 CPU was manufactured in a 10 μm CMOS process, which is 10 000 nm. The CPU speed at the time was 108 kHz or around 2000–3000 times slower than the current clock rates between 2 and 3 GHz for high-end desktop and notebook processors. Since then transistor sizes have been shrinking around 70% every two to three years.

Early CMOS transistors only required power to switch from one state to another while otherwise drawing almost no current. As a consequence the power requirement of a processor could be reduced during idle times by lowering the clock speed. Owing to continuing size reductions, however, the layers between the different parts of transistors have reached a thickness of less than 1.5 nm; that is, only a few atoms separate the different parts of the transistor. This leads to increasing leakage currents while a transistor is idle. This effect is undesired since the leakage is independent of the clock speed. This means that the leakage power, also referred to as static power, remains the same even if the system clock speed is reduced to 1 Hz. In theory, reducing the voltage also reduces the leakage power. In practice, however, chip voltage has already reached a very low level and a further decrease would result in unwanted interference from external components such as conducting paths between different chips. For the moment leakage power can still be controlled by new manufacturing methods which, however, slow down the possible switching speeds of transistors and hence reduce the clock speed improvements that would otherwise result from reducing the size of transistors. Still, Globalfoundries reports that the scaling from a 40 nm process to a 28 nm process results in a 60% higher performance at comparable leakage with a 50% lower energy requirement per switch and 50% lower static power requirement [28]. Nevertheless, industry experts expect that at some point in the future leakage prevention techniques will eat up any performance gains in terms of

increased clock speeds that a reduction in transistor size would normally bring about. So while for the moment reducing the size of components still results in faster processors, lower power consumption, and smaller chips, this trend is unlikely to continue with current technologies. The only benefit from reducing transistor sizes in the future is thus a reduction of the size of the chip unless new methods are found to prevent leakage that do not interfere with the speed a transistor can switch from one state to another.

5.5.3 Displays

Despite the limitations imposed by the size of mobile devices, mobile displays have undergone an interesting evolution in recent years. While 2.5–3 in. screen sizes were considered as large, most smartphones in 2012 had screen sizes between 3.5 and 4.5 in. as less space was required for keys because of touch screen-based input. Some smartphones even boasted a screen size of over 5 in. to fill the gap between smartphones and larger tablet devices. Also, display technology has advanced significantly. While a resolution of 320×240 pixels was state of the art in 2007, this has increased to 840×480 pixels and even further. While LCD displays are still used in many devices, Organic Light-Emitting Diodes (OLEDs) have also become very popular because the power consumption is less, lively colors are available, and there is no need for background illumination. With OLED displays, basic information such as time, date, missed calls, an SMS indicator, and so on can be displayed even when the device is locked without significantly increasing the standby power consumption.

While in 2007, the tablet category did not exist, this device category has become a huge success in 2012 and screen sizes between 7 and 10 in. have become standard. Typical screen resolutions range between 1024×600 and in some cases even 2048×1536 pixels, a resolution at such a screen size at which the human eye is no longer able to distinguish individual pixels.

Only indirectly related to the display technology is the material used on top of the displays. While plastic was mostly used in 2007 in most phones, it was quickly replaced by glass once display and touch-based input became more popular. At first, the glass was prone to scratches in a similar way as the plastic that was used before. In recent years, however, display glass has become scratch resistant and not even coins, keys, and other objects with sharp edges in pockets are able to damage the surface. As the glass surface adds significantly to the overall weight and dimensions of the device, one area of evolution is to reduce the thickness and thus the weight of the glass without compromising the scratch resistance and touch sensitivity [29].

A different type of display technology that has become very popular especially for e-book reading is electronic paper. Unlike LED (Light-Emitting Diode) or OLED displays, e-paper displays do not emit light themselves but only reflect light from their surface in a similar way as ordinary paper. As a consequence, this display technology is very power efficient and while the content does not change, no power is required to sustain the image. Devices with e-paper displays are thus very power efficient and require only small and lightweight batteries. The overall device can thus be very thin and lightweight, which makes holding such devices, even with large screens and over a longer time, much more enjoyable than tablets with active displays with a high power consumption and a significant battery weight. The downside of e-paper displays is, however, that in 2012,

only black and white displays are available and that refreshing the content of the screen is too slow for dynamic content. While this is not an issue for traditional books, which are mostly black and white as well, it limits the use of the device to only a few applications. With the rising popularity of multipurpose tablet devices and multimedia e-book content with animations, and so on, it is difficult to predict if e-paper-display-based devices will remain popular unless their two major limitations, that is, no color and slow display refresh times, are mitigated. It should be noted, however, that e-paper-based e-book readers have become very inexpensive with prices of around €100 in 2012. If prices were to fall further, their popularity and use could remain high as many consumers would not have to choose between an e-paper-based device for e-book consumption or a more expensive multipurpose tablet device on which e-book reading is just one of the many applications.

5.5.4 Batteries

As power requirements of high-end mobile multimedia devices are unlikely to decline in the future, another research focus is better batteries or a different kind of energy storage for mobile devices. Current lithium ion technology is unlikely to get significantly more efficient in the future as physical limits of the technology are almost reached. A possible solution for the future could be fuel cells producing electrical energy from methanol, water, and air [30]. Research has been ongoing for many years now and it is expected that fuel cells can store 10 times the amount of energy compared with current battery technology. However, no major breakthroughs have been reported so far and it seems unlikely that fuel cells will be used in mobile devices during the next few years. It should be noted at this point that new battery or power cell technology would mostly be used to extend the operating time of a mobile device and not to supply more energy to the device itself. Unlike notebooks and PCs, which tend to get warm and require active cooling, fans are not acceptable for mobile devices.

5.5.5 Camera and Optics

In 2007, high-end mobile phones in Europe were equipped with 5 megapixel camera sensors and a typical sensor size of $1/2.5$ in [31]. Most mobile devices at that time, however, had a far lower resolution and a smaller sensor. As a consequence, picture quality was significantly inferior to digital compact cameras at the time. In 2012, most smartphones and tablets are equipped with camera components producing at least equal results as those in high-end devices only five years earlier. Typical sensor resolutions in such devices have reached 8 megapixels, and better optics now help to capture images, which come close to what is possible with dedicated entry level point and shoot cameras as long as no optical zoom is involved.

One major shortcoming of many camera phones back in 2007 was the use of fixed-focus lenses. Together with the low-resolution sensors, this often resulted in mediocre pictures. In the meantime, most mid- to high-end smartphones are equipped with auto-focus lenses, which allow wider apertures and thus a higher light sensitivity, or extended depth of field (EDOF) lenses and software for postprocessing to generate a sharp image. While producing similar results in most situations, the major drawback of EDOF is its

inability to produce sharp images of objects closer than 1 m, which makes such lenses unsuitable for photographing documents at close distance [32].

Video recording has also improved significantly over the past five years. Owing to dedicated video encoding units in the SoC as described earlier, smartphones are now capable of recording videos in 720p HD resolution at a rate of 30 frames per second. First devices are also capable of 1080p full-HD video recording. Only little compression is applied, and 720p video recording requires about 100 MB of storage space per minute. Uploading a video directly from a mobile device to a video portal such as Youtube is therefore time and data intensive. Once on the server, services such as Youtube then compress the 100 MB file of a 1 min 720p HD stream to about 46 MB while keeping the 720p resolution to 15 MB for the default 480p resolution typically used when videos are viewed with a PC and to less than 6 MB for the 360p version, which is used for streaming to mobile devices [33]. It is interesting to note that this transcoding process is quite time intensive. This might be an indication of the significant processing power required for the transcoding, which is perhaps the reason why this is not already done on the mobile device. In this regard, LTE networks can help as, under good reception conditions, uplink speed transmission is faster compared to that at HSPA (cf. Chapter 2).

Some specialized smartphones trade overall thickness of the device to include enhanced sensors and optics such as, for example, the Nokia N8. A large 1/1.83 in. camera sensor, a resolution of 12 MP, and specially crafted optics produce results with fine grained details without a need for postprocessing such as sharpening, which produces undesired side effects. Even in low light conditions, the sensor produces sharp images and a xenon flash similar to those used in compact cameras, enabling indoor photography with good results.

The largest and highest resolution camera sensor used in a smartphone at the time of publication of the second edition is the 41 MP 1/1.2 in. sensor in the Nokia 808 Pureview. While the size of each pixel on the sensor is the same as in high-end 2012 smartphones such as the iPhone 4S or the Samsung Galaxy S-II, the very high resolution of the sensor is used to combine the information of 8 pixels into one output pixel. This results in a 5 megapixel image of very high quality, especially in low-light situations. Also, the 41 megapixels can be used to act as a zoom without mechanical components, another first in the mobile camera domain. While point and shoot cameras use a mechanical system to move the lenses, which requires significant space and thus makes cameras much thicker than mobile phones, zooming with the 41 megapixel sensor is achieved by only using a part of the sensor and saving the result as a 5 megapixel image. This of course reduces the sensitivity as fewer pixels are used per output pixel, but the size of each physical pixel is still the same as in average smartphones. While being much more compact than mechanical zoom lenses in point and shoot cameras, the camera sensor and optics requires more space and thus the device is significantly thicker than average smartphones with smaller and lower resolution sensors. While this increased size is unlikely acceptable to many users, photography enthusiasts might find such phones appealing. Also, further miniaturization might enable manufacturers to further reduce space requirements for such camera sensors in the future.

With the processing speed available in smartphones in 2012, it has also become possible to analyze the data stream received from the camera in real time, which has enabled functionalities such as face detection for proper focus and smile detection to help the user



Figure 5.6 A 2D barcode on an oversized advertisement poster.

to take the picture at the right moment. Also, smartphones have become powerful enough to assemble a panorama from several pictures in near real time.

Another interesting use of mobile phones and their built-in cameras is barcode scanning. While 2D bar codes have become popular in Japan many years ago, a similar trend took much longer to emerge in other parts of the world. Different bar code formats were developed and deployed, which made it difficult for the user in practice as applications could often only decode some of the different formats. In the meantime, however, a number of bar code uses have become quite popular, for example, on advertisement posters to get more information as shown in Figure 5.6 and also when shopping to compare prices of a product found in one shop to see if the product is cheaper elsewhere.

On the basis of the evolution of camera module hardware in the past, it is likely that there will be a further reduction in size and cost of camera modules while keeping camera sensors big enough for high-quality imaging. Image quality is likely to further improve, partly also because of intelligent hardware and software combinations that have enabled cheap EDOF lenses to produce good results except in the very near field.

5.5.6 Global Positioning, Compass, 3D Orientation

In 2007, one of the major new features in high-end mobile phones was the inclusion of a GPS receiver. The Nokia N95 released in early 2007, for example, was Nokia's first GPS-enabled smartphone [34]. By 2012, GPS has become a standard feature in all smartphones, and first chipsets such as the Qualcomm Snapdragon platform have included a GPS receiver in the main chip and thus a specialized GPS chip is no longer necessary.

Advances have also been made in obtaining the first position after activating the GPS component in a device (Time To First Fix, TTFF). Calculating a location requires the reception of the signals of at least four GPS satellites. In addition, the receiver must know the exact orbit of each of these satellites, which is referred to as the “ephemeris” data.

The ephemeris data is transmitted by the GPS satellites themselves. At a transmission rate of 50 bits per second, however, reception of the complete information can take several minutes, especially when several iterations of the data have to be received because of reception errors due to a weak signal or user movement, which can easily interrupt the data stream. Mobile devices thus usually get the ephemeris data from a network-based server over a cellular or Wi-Fi connection. Initial signal acquisition is also sped up by knowing the approximate location of the device. In combination with the ephemeris data, the device can quickly calculate which satellites should be received at the approximate location. The GPS receiver can then try to receive those signals first. Getting the approximate location is usually done by collecting information about nearby Wi-fi networks and cellular network base stations (Wi-Fi SSIDs (Service Set IDs) and network/base station identifications). These are also sent to a network-based server with a database of network identifications and their locations. Using the approximate location and ephemeris data received via the Internet, the TTFF is usually below 5–10 s. This is the case even if the GPS receiver is activated at a totally new location, for example, after an intercontinental flight. In this case, the previous location data still stored in the device that may otherwise also be used to improve the acquisition speed is invalid.

It is interesting to note at this point that the location information obtained by querying a database in the network for a coarse location is also useful to load initial maps data and show an approximate location to the user before a more precise location is available via GPS. As the range of Wi-Fi networks is quite short, location information obtained this way is usually accurate within a few tens of meters, which is sufficient for many applications such as street navigation. While specialized companies were needed initially to obtain the location of Wi-Fi access points by driving through countries with measurement equipment picking up Wi-Fi signals and storing this data with location information, it is the navigation and mapping applications on the user devices today that provide this information to databases in the network by reporting Wi-Fi SSIDs and MAC addresses together with GPS information. This helps companies such as Google, Nokia, Microsoft, and others to keep their location databases up to date.

In addition to GPS, other navigation systems have also become operational. The Russian GLONASS system, for example, is now also available worldwide at the time of writing, and the Chinese Beidou navigation system covers parts of Asia. And finally, the European Galileo positioning system is also set to become operational in this decade. Some GPS chips now also support GLONASS and use both systems to obtain location information [35]. This increases the number of visible satellites in challenging situations, for example, when the user is located between high rising buildings or in narrow streets where only a few degrees open sky is visible. Having more satellites thus reduces the times in which no valid positioning data is available and thus increases the usability of street-side navigation.

When location information is used for applications such as route navigation in cars, the direction the user is moving toward can be easily obtained from the directional movement. The map displayed on the screen can then be rotated accordingly. For street-side navigation, however, this is often not possible as the user is often immobile while deciding the direction in which to walk. In 2012, electronic compass components have also become a default feature in smartphones and offer an easy way to rotate maps according to the direction the user is looking. This significantly helps street-side navigation as the user is instantly aware of the direction in which he is looking. Knowing not only the

location of the user but also the direction he is facing has enabled other applications that use the camera, location information, and the direction from the compass to overlay a video stream with additional information, for example, Wikipedia entries for locations and objects in the area a user is currently located in. Examples of such applications are Wikitude and Layar [36].

Motion sensors, also referred to as accelerometers, are also common in mobile devices now. These sensors can detect 3D motions, which can be used, for example, by astronomy programs to show stars and constellations the user points the device at. For this application, it is not only necessary to know the approximate location of the user and his viewing direction but also the angle above the ground the device is held at to display stars and other objects at the right location. Examples of astronomy applications making use of motion sensors are Google Sky Maps and Astroller. Other uses of accelerometers include pedometer apps to count steps, sleep phase alarm clock apps that detect motions while the user is sleeping and adapt the wake up call accordingly, and also motion detection in games for controlling 2D and 3D movements (e.g., a virtual steering wheel). These are just a few examples that demonstrate the many uses for location, direction, and motion information.

5.5.7 *Wi-Fi*

It was only in 2006 that first mobile phones included Wi-Fi functionality. Early examples are the Nokia N80 [37] and N95 [38]. At that time, it was by far not certain if this would be a continuing trend as not all market participants were happy that Wi-Fi was included in a mobile device as a means to access the Internet. In the meantime, however, Wi-Fi has become a standard feature even in cheap smartphones and other mobile devices, and some versions of mobile devices such as tablets and ebook readers are available with Wi-Fi connectivity but without a cellular interface. Mobile network operators also benefit from the Wi-Fi interface as many users prefer to connect to their fixed line Internet connection at home via Wi-Fi because of mobile data costs and therefore reduce the amount of traffic transmitted via the cellular network. Some mobile network operators have also deployed Wi-Fi hotspots in hotels, train stations, and airports and offer combined Wi-Fi and cellular plans to their customers.

As described in Chapter 2, most mobile devices now include a Wi-Fi interface according to the 802.11n standard. While the standard offers many options to increase the data rates, most mobile devices only use a single antenna, that is, no Multiple Input Multiple Output (MIMO) operation, and are designed for single 20 MHz channel operation. Also, most devices only use the 2.4 GHz band and do not operate in the 5 GHz band. In 2011, highly integrated Wi-Fi front-end-modules optimized for smartphones have become available that can operate in the 2.4 and 5 GHz bands. Together with the emerging 802.11ac standard (cf. Chapter 2), the support of the 5 GHz band might thus become more common in mobile devices in the next few years.

As the Wi-Fi interface of a device does not only offer Internet connectivity but also enables local communication in the home network to other connected devices, device manufacturers now include software to interact with connected TVs to show pictures and videos stored on the device or to connect to Hi-Fi equipment to stream music stored on the device. In the reverse direction, it has also become possible to use a mobile device to listen to music stored on a server in the home network via streaming over

Wi-Fi. Such functionality is, for example, enabled by the Digital Living Network Alliance (DLNA) standard and while still not a mass market application in 2012, there is a rising number of devices supporting it. For exchanging any kind of files in the home network between PCs, NAS drives, and mobile devices, protocols such as Microsoft's Server Message Block (SMB) can be used, as this protocol is supported by all desktop operating systems and also by mobile operating systems such as Android in combination with third-party file manager applications.

As Wi-Fi chips can be used in client mode and access point mode, many smartphone operating systems now include this functionality and a smartphone thus becomes a bridge to the Internet for other Wi-Fi devices such as notebooks. Some network operators restrict the use of this functionality in devices sold over their sales channel but most smartphones and tablets sold via independent sales channels (e.g., via Amazon) have this functionality activated.

In an ordinary Wi-Fi network, a data packet from one device to another must always pass through an access point. This in effect cuts the available data rate into half as each packet has to be transmitted twice. As the use of an access point complicates many usage scenarios such as the ad hoc transmission of a picture or file to another device not connected to the same network, the Wi-Fi Alliance has created the "Wi-Fi direct" specification [39]. As the name of the specification implies, Wi-Fi direct allows ad hoc communication between two Wi-Fi devices for any kind of data transfer. This is achieved by one device becoming a Wi-Fi direct access point for the time of the transfer to which the other device can connect to. The Wi-Fi Protected Setup (WPS) procedure (also specified by the Wi-Fi Alliance) is used to establish a secure and encrypted connection with minimal user interaction. In essence, Wi-Fi direct thus works in a similar way as file transfers or data exchanges over Bluetooth. In 2012, only few devices included this functionality but since Google's Android operating system has started to support the functionality in version 4 for use cases such as file transfers and the exchange of address book and calendar entries, the number of supporting devices is growing.

Some devices have also started to support Wi-Fi as a bearer for Bluetooth connections. Some Symbian-based devices make use of this for establishing a Bluetooth connection to transfer calendar and address book entries. Once the devices detect that they not only support Bluetooth but also Wi-Fi, the bulk of the data is transferred over the much faster Wi-Fi channel. In effect, this functionality is similar to that offered by the Wi-Fi direct specification and might thus be superseded by it.

Another interesting aspect of including a Wi-Fi interface is to use it for positioning purposes. As Wi-Fi access points are usually always at the same location and emit a unique name (SSID) and MAC hardware address, this information can be used in combination with a location database located in the network as an additional means to quickly pinpoint the user's approximate location indoors when no GPS signal is available or while the GPS receiver is still decoding the information for an initial set of coordinates.

One important topic that is often overlooked when discussing Wi-Fi on mobile devices is security. While home and office Wi-Fi networks that are using WPA (Wireless Protected Access) and WPA2 authentication and encryption are well secured against external attacks, there are significant security issues when using public Wi-Fi hotspots. Here, encryption is usually not used, which means that applications themselves have to encrypt the data they transmit. This, however, is still often not the case and tools such as Firesheep

and Droidsheep make it easy for attackers to spy on the data traffic [40]. Companies therefore often protect their employees with Virtual Private Network (VPN) products that offer strong authentication and encrypt all incoming and outgoing data. Private users, however, usually do not use VPN products because of the overhead, complexity, and cost involved. If the public Wi-Fi hotspot is managed by a mobile network operator, another possible mitigation would be to use authentication and encryption based on the Extensible Authentication Protocol—Subscriber Identity Module (EAP-SIM) protocol, which uses the credentials stored on the SIM card. While this standard has existed for years, there has been little take up so far and it is not certain if there will be more interest from the operator community in the future. As a consequence, care should be taken when public Wi-Fi hotspots are used. Applications transmitting private data should be disabled unless it is certain that the application offers adequate application-level authentication and encryption. As a general rule, 3G or 4G cellular connectivity should be preferred in public places whenever possible.

5.5.8 Bluetooth

A wireless technology that has been part of mobile devices for many years is Bluetooth. Over the years, the specification has been continuously enhanced and evolved. Most applications that Bluetooth is used for today are still the same as when they were first developed:

- Wireless headsets for bidirectional voice telephony,
- Wireless headsets for high-quality music streaming,
- Exchange of data between devices such as files, pictures, calendar, and address book entries,
- Wireless keyboards and mice for notebooks, smartphones, and tablets (note that wireless mice and headsets in the PC sector are often not using Bluetooth but a proprietary protocol).

Some applications where Bluetooth was used in early years such as offering Internet connectivity to PCs and notebooks via a mobile device have mostly disappeared in the meantime because of the integration of Wi-Fi access point functionality in smartphones and data speeds in mobile networks exceeding the maximum Bluetooth data rates of about 2–3 Mbit/s. Also, ideas such as Bluetooth marketing, that is, sending messages to mobile phones of customers in shopping malls and other places have not proved to be very successful. On the other hand, there are a number of applications that make use of Bluetooth, which have become quite successful such as, for example, wirelessly connecting heart rate sensors for sports purposes [41].

Most of the Bluetooth functionality used today is part of the Bluetooth 2.1 specification, which was released in 2007. Many devices today are declared as Bluetooth 3.0 compliant, as they are using a number of evolutionary enhancements to existing functionalities contained in this version. In addition to those enhancements, version 3.0 of the specification offers an alternate MAC/PHY layer specification, which enables devices to use the Bluetooth protocol stack for establishing an initial connection and then using a Wi-Fi bearer for transmitting the actual user data. To inform users that this optional feature is supported by a device, the +HS designation was created by the Bluetooth Special Interest Group

(SIG). Devices supporting combined Bluetooth and HS Wi-Fi transmissions are thus to be labeled as Bluetooth 3.0 + HS compliant. In practice, however, there are currently only few Bluetooth 3.0 devices supporting this feature.

The latest Bluetooth version at the time of publication of the second edition is Bluetooth 4.0, incorporating a very low energy physical layer implementation, which it has inherited from the WiBree project. This very low energy physical layer has little in common with the traditional Bluetooth standard except for the use of the license exempt 2.4 GHz band that it also shares with Wi-Fi. Its very low power requirements make it ideal for very small sensors with a very limited power supply, for example, via coin cells. Mobile devices could act as UI for such sensors but so far, only few implementations have appeared on the market. With competing technologies such as Wi-Fi and the emerging Near Field Communication (NFC) standard now more and more to be found in mobile devices, it is likely that Bluetooth will mainly be refined in the coming years for the purposes it is already used today rather than being used for revolutionary new applications.

5.5.9 NFC, RFID, and Mobile Payment

Another area of ongoing evolution is embedding NFC chips in mobile devices for local exchange of information and mobile payment functionality. In Japan, Sony's FeliCa NFC system is already widely used as a payment system for public transportation, convenience stores, and vending machines. Embedding the chip in a mobile phone and making it accessible to applications running on the mobile phone extends the use of the payment system for online ticketing for flights and sporting events via a mobile device. Tickets can then be printed at the airport, at the stadium, or the embedded FeliCa chip used directly to gain entry.

As the FeliCa system is designed as a micropayment system, linking the system to the Internet via a mobile device extends the payment system beyond direct interaction with an NFC reader and it is likely that many online services beyond the two examples given above will make use of such a system in the future.

In addition, FeliCa-equipped mobile phones can be used for non-payment purposes as well, for example, as door keys [42]. If the door lock is connected to the Internet, users can even check remotely with the mobile phone if the door is locked.

In other parts of the world, it was and still is more difficult to introduce a standardized NFC-based local information exchange and mobile payment system because of the number of different market players and the uncountable number of public transport companies, each using their own ticketing system. Thus, it will take much longer for a single system to reach a critical mass. Having a universally adopted standard, however, is critical for mobile device manufacturers to consider including NFC technology in their mobile devices, which are often produced for a global market without national hardware variants. Some progress has been made in this regard with companies such as Google, Nokia, and RIM backing the NFC-Forum [43] based standard for local information exchange as a basis for mobile payment services. In 2011, first Android- and Symbian-based mobile devices with NFC chips appeared on the market and many others have followed since.

Payment and non-payment NFC services rely on the same NFC protocol stack and only differ in how this stack is made use of by the applications mentioned here. The following

paragraphs therefore first describe the basic operation of the NFC protocol stack and how it is used for different purposes.

On the physical layer, NFC uses a carrier frequency of 13.56 MHz to transmit data between two NFC devices via inductive coupling as specified in ISO/IEC 14443 [44]. Two transmission modes are specified. The passive mode is used in combination with NFC RFID (Radio Frequency ID) tags that do not have their own power source. In this mode, the originator's RF field is used as an energy source by the passive RFID tag to return a message. The message is usually short in nature and can, for example, be an address of a web page (URL) or a business card entry in vCard [45] format.

It is also possible for two active NFC devices to communicate with each other such as two mobile devices or a mobile device and a point of sales (POS) terminal. In active mode, both devices are equal and transmit and receive alternatively using their own power source. The practical working distance between the two NFC devices is 4 cm, but much longer distances can be achieved with high-gain antennas. As a consequence, sensitive data exchanges must be encrypted and applications using the communication channel for such information need to authenticate each other to prevent over-the-air attacks such as eavesdropping, information theft, and man-in-the-middle attacks. Supported data rates over the NFC air interface are 106, 202, and 424 kbit/s.

The next higher level of the NFC protocol stack defines different message formats. A popular format is the NFC Data Exchange Format (NDEF) [46] defined by the NFC forum. The format defines a general message structure and allows devices to identify what kind of information is contained in the message it has received. The following list gives some examples of message types that can be encoded in an NDEF message:

- Universal Resource Identifier (URI) (web address and email address);
- Plain text;
- Smart poster = text + uri;
- Bluetooth and Wi-Fi parameters;
- Business card (vCard format);
- Signature.

Once an NDEF message has been received, a content-dispatching system forwards the content of the message to an application that has registered itself for a certain content type. This way, different applications can register themselves for different message types and the dispatching system automatically forwards the message to the application that can handle the particular message type. When several applications register themselves for the same message type, a dialog box is usually presented to the user to choose which application should receive the message. When a web address has been received, for example, it could be sent to the web browser while an NDEF message containing a vCard would trigger the address book application that has previously registered itself as a receiver for this kind of content.

Applications on a mobile device can not only receive NFC messages but also transmit them. The address book application, for example, can be extended to send NFC messages, for example, an address book entry to the next NFC-enabled device that comes in range.

Third-party applications can also register themselves as recipients of NDEF messages with a certain content type and can also trigger the transmission of messages themselves.

An example of such an application is the Foursquare mobile application [47]. The aim of this application is to let users “check-in” at certain locations and share their current location with their friends by various means. The traditional “check-in” procedure uses the GPS receiver in a mobile device to pinpoint the user’s location. The location of the user is then sent to a database in the network to retrieve all places near this location that are registered with Foursquare. The user can then select the place he is visiting. With NFC tags, this process is significantly simplified as the “check-in” is now performed by the user holding his device close to a Foursquare RFID tag. The type of content of the tag is then forwarded to the Foursquare application on the device, which then registers the user in the network at the current location. NFC functionality can also be used to exchange information between two Foursquare apps on devices of different users to make friend requests and exchange location lists. This is done by launching the Foursquare applications on both devices and then holding the two devices close together.

For mobile payment purposes, the same NFC hardware and software is used as described above with one major addition. As the payment software (e.g., from Visa, Mastercard, American Express, a local transportation company, etc.) and the data identifying the user for the payment process are highly sensitive, it must not be accessible from outside to prevent unauthorized monetary transactions or theft of personal information. Consequently, the payment software and the user data have to be stored in a secured area in the mobile device, the “secure element.” One approach is to use the SIM card for this purpose as it is already used as a secured storage element for the user’s mobile network subscriber data. Another approach is to include an independent secure element chip in the mobile phone. The banking industry, for example, requires a secure element that is compliant to the Global platform specifications [48].

When the user places his mobile device on a POS terminal, the terminal then sends a message identifying the payment applications it is compatible with. The mobile device or the user may then select one of the payment applications stored in the secure element to perform the transaction. For performing a payment, the secure element requires two interfaces, one to the NFC chip to be able to exchange messages with the POS terminal and another to interact with the user. In other words, there must be an application on the mobile device that can be reached by the secure-element-based payment software; so information such as, for example, the amount to be paid or a PIN input screen can be shown to the user.

Several companies are involved in this value chain. On the one end, there is the bank that issues the payment software and the user identification that is to be stored in the secure element. On the other end is the issuer of the secure element, which is the mobile network operator in case the SIM is used as the secure element, or the device manufacturer or the user himself in case a dedicated chip is used. A tricky aspect in this regard is how the software and information can be securely transferred between the issuer (e.g., the bank) and the secure element, which again depends on who is in control of the secure element. Some types of payments such as, for example, NFC-based entry systems to public transportation systems do not require an interaction with the user at the time the device is brought into contact with the NFC-based entry system. This is the case, for example, when the user buys a weekly or monthly ticket in advance or pre-pays a certain amount of money that is then automatically deducted when the device is held over an NFC reader at an entry or exit gate. In such cases, however, an application might be

supplied by the public transportation provider that interacts with the software application on the secure element so the user can later on access details of the purchases made, that is, the trips he has made and the amount of money that was deducted from his account.

It should be noted at this point that there is no standardized process for mobile payments; so different banks/card issuers will have their own software stored in the secure element. Also, the payment procedures may be different in different parts of the world; so it is not certain that the NFC payment processes used in Europe will be compatible with those used in North America. Further details on NFC and mobile payment processes can be found in [49].

5.5.10 *Physical Keyboards*

One of the main limitations of mobile devices because of their size and primary use as an information consumption device rather than an information creation device is the missing or miniaturized keyboard. This makes text input difficult, slow, and error-prone, which significantly inconveniences activities beyond information consumption such as writing e-mails or longer documents. A solution is external keyboards that are connected to devices via wireless technologies such as Bluetooth. Even though the keys are sometimes slightly smaller than on a full notebook or desktop keyboard, typing with 10 fingers is nevertheless possible. This makes text input for e-mail, mobile blogging, and many other activities very simple and convenient while preserving mobility. Alternatives to foldable or fixed add-on keyboards presented over the years are rollable keyboards and laser-projected keyboards. Over the years, however, such products did not have much success.

With the increasing processing capabilities especially of tablet devices, add-on keyboards further blur the line to traditional computing devices such as notebooks and netbooks with full keyboards that are part of the overall product design. In 2012, processing power and battery capacities already came close to the performance of early single CPU-based netbooks introduced only a few years earlier. The biggest difference between the two categories is thus the operating system and its optimization. Tablet operating systems are tailored for touch input, the use of only a single application at a time, and the functionality of applications is often limited to one purpose without many options. In contrast, netbooks, notebooks, and other “traditional” devices use full desktop operating systems that are optimized for keyboard and mouse input to facilitate fast switching between applications and focus on information creation rather than consumption. Also, programs tend to be more complex to be more versatile.

A trend can be observed to make desktop operating systems more touch friendly for use in the tablet space as well, and it remains to be seen how far such an adaptation can go before usability on traditional devices is negatively affected. Physical keyboards being used as an extension to tablets and other mobile devices will surely influence the changes in mobile operating systems as they extend tablet usage scenarios while at the same time offering a traditional and effective way of interacting with a computing device. This in turn reduces the number of changes required for operating systems to be usable on tablets and also on more desktop-oriented devices.

5.5.11 *TV Receivers*

Many new hardware functionalities were added to mobile devices over time but not all of them have been successful. One interesting example is mobile TV receivers. In the first edition of this book, DMB, DVB-H, MediaFlo, and MBMS have been looked at from a critical point of view because of the strong competition from Internet-based audio and video streaming applications and downloading content for later use. These services are in many cases better suited for mobile use as content can be consumed at exactly the time the user has time to spend (e.g., while waiting for the bus) rather than only at predefined times. In addition, content that has previously been retrieved can be consumed at places where no or only sporadic network coverage does not allow receiving a continuous data stream. The main advantage of mobile TV was thus broadcasting of live events such as football games, Formula 1 races, and so on. Over the years, this prediction has held and only few devices appeared in Europe and the USA including a digital television receiver. Falling prices for mobile Internet use and Wi-Fi having become a standard feature, mobile devices have made streaming and downloading of content easy and affordable. For the remaining real time services where mobile TV broadcasting has an advantage, pricing and marketing strategies for mobile TV have not found widespread user acceptance. As a consequence, mobile TV services launched in a number of countries were closed again after a few years despite significant investments in standardization, development, and build-out of broadcasting infrastructure.

A few mobile devices appeared on the market that included a standard DVB-T receiver [50] had the advantage of being able to receive the free-of-charge standard terrestrial digital television signal and not the handheld adaptation DVB-H, for which a monthly service fee had to be paid. Nevertheless, these devices had only limited success and no successor models were produced. Since then no further large-scale attempts have been made in practice to include live television broadcast reception in connected devices such as smartphones and tablets.

It should be noted at this point that a number of mobile network operators offer TV streaming services of select content such as soccer games. These offers, however, are based on individual user data streams (IP unicast) rather than broadcasting a single signal to all users. This has the advantage that no special hardware is required in the device for the reception of the stream. Instead, an application on the device receives the TV stream over a mobile Internet connection. The downside of this approach is the amount of data that has to be streamed from a central place, which increases with the number of users and the limit of the number of concurrent user per cell.

5.5.12 *TV-Out, Mobile Projectors, and DLNA*

With increasing multimedia capabilities and storage capacities, mobile devices are commonly used today for storing pictures and videos and for presenting them to other people. The biggest disadvantage of smartphones compared with prints or to using a tablet or notebook for this purpose is the size of the screen, which is limited by the portability and mobility of the device. One current solution is to include a TV out port so the device can be connected to a standard television set (cf. Figure 5.4). While early devices had an analog TV-out port, current devices feature a mini-HDMI output port and can thus

be connected to computer LCD screens and modern television sets. The disadvantage of this solution is that the smartphone needs to be close to the TV set or computer screen as the HDMI signal requires short cables. Also, carrying an HDMI cable and a mini-HDMI converter is often inconvenient.

A potential alternative are miniaturized projectors, which are small enough to fit into ultramobile devices such as smartphones and tablets. First working samples have been shown on occasion [51], and standalone devices are available on the market today. Smartphones equipped with small projectors have, however, so far not gone into mass production. With devices becoming slimmer and slimmer, it is not certain if combined smartphone/projector products would become a success, especially since in the meantime software alternatives to show pictures and videos on other devices such as, for example, the DLNA [52] protocol, which allows streaming data over Wi-Fi to TVs and other connected devices, have become available.

5.6 Multimode, Multifrequency Terminals

While in the past, only a few frequency bands were used for cellular wireless systems, their number has risen rapidly in recent years. Table 5.3 shows some of the bands currently assigned and used in different parts of the world for 3GPP GSM/UMTS/HSPA [53] and LTE [54] networks.

While it is unlikely that further bands will be actively used for UMTS and LTE in Europe in the next few years beyond those listed in the table, it is likely that new bands will be added in the North American market for the use of LTE. This is due to the different ways the bands are defined. While in Europe, large spectrum allocations are put into a single band, North American regulators have split up the available frequency bands

Table 5.3 Popular frequency bands for 3GPP UMTS and LTE

Band	Uplink bands (MHz)	Associated downlink bands (MHz)	Locations used (not exhaustive)
<i>UMTS</i>			
I	1920–1980	2110–2170	Europe, Asia, Australia
VIII	880–915	925–960	Europe, Asia, Australia (GSM refarming)
II	1850–1910	1930–1990	North America
IV	1710–1755	2110–2155	North America
V	824–849	869–894	North America, Australia
<i>LTE</i>			
1	1920–1980	2110–2170	Europe, Asia, Australia (UMTS refarming)
3	1710–1785	1805–1880	Europe, Asia, Australia (GSM refarming)
7	2500–2570	2620–2690	Europe, Asia, Australia
20	832–862	791–821	Europe, digital dividend
4	1710–1755	2110–2155	North America
13	776–787	746–757	North America
17	704–716	734–746	North America

especially in the 700 MHz band in a way to only allow relatively small pieces of spectrum (e.g., 2×10 MHz) to form a single band. For spectrum in the 2600 MHz band, the use of Frequency Division Duplexing (FDD) or Time Division Duplexing (TDD) has been left open, which will further add to the number of used bands being put into service over time.

In practice, the growing number of bands has a number of undesired implications for both users and network operators. In 2007, typical mobile devices supported only one or two 3G frequency bands like, for example, the 2100 MHz band for European models or 850/1900 MHz for US models. In addition, such high-end phones usually also supported four bands for 2.5G GSM/GPRS, that is, 900 and 1800 MHz for Europe and 850 and 1900 MHz for the USA. In 2012, band support has risen to typically four UMTS bands, for example, 2100 and 900 MHz for Europe and 850 and 1900 MHz for North America. In some devices, penta-band radios are used that also include the 1700/2100 MHz band combination, which is the third band used in North America for UMTS services. The Nokia N8 was one of the first smartphones that supported five UMTS frequency bands [55].

While connectivity with such devices is currently guaranteed in most places in the world, the support of 5 UMTS bands is far from the 16 bands currently defined in 3GPP for UMTS (excluding the Chinese TDD bands) and 37 bands currently defined for LTE (including the TDD bands). In the future, mobile phones thus have to support additional bands in order to ensure that travelers can use their wireless equipment globally.

For device manufacturers and network operators, the increase in the number of bands is equally disadvantageous. T-Mobile, for example, who have bought spectrum in the USA in band IV (1710–1755 MHz and 2110–2155) [56], have struggled over the years to get a wide variety of devices for its UMTS services in this band as it is specific to the USA and Canada. This means that the volume of devices using this band is very limited compared with global sales of billions of devices for more popular frequency bands. Furthermore, incentives are small to include this local band, only used by a few operators, in mainstream devices since the support of each new frequency band adds to the production cost per device. Since the market grows only insignificantly for a device, if this band is included this reduces sales margins and will thus inhibit production of such mobile devices for global sales. In addition, operators using local bands cannot profit from users roaming into the country from abroad who have at best a quad-band 2.5G or quad-band UMTS device. Already today this is felt by network operators not using GSM or UMTS, since they cannot profit from visiting GSM and UMTS roamers [57].

Adding support for an additional frequency band in a mobile device has no impact on the digital components of the phone since processing of signals once they are digitized are independent of the band. The software for the DSP and the radio protocol stack software of the baseband chip, however, have to be adapted to the additional frequency bands. Examples of such changes are the support of the additional channel numbers of a new frequency band, scanning of all available frequency bands for networks at power-up or while in idle mode and changing messages, and parameters to inform the network of the additional capabilities of the device.

The analog part of the mobile consists of antennas, front-end filters, and RF chips. Here, each additional frequency requires additional hardware components. To limit the number of additional components, hardware designs are usually multiplexing certain components for use with more than one frequency band. As multiplexing and switching components

reduce sensitivity, each additionally supported frequency band further decreases the reception sensitivity of the device. As technology improves, this is usually compensated for by improved hardware designs.

It is estimated in [58] that the analog part of a mobile device is responsible for 7–10% of the cost of a mobile device, independent of whether it is a low-, mid-, or high-end handset. This is due to the fact that, while low-end devices are cheaper, they do not support as many radio interfaces and frequency bands as high-end devices. The report also estimates that the hardware cost per supported frequency band is around US\$2. For GSM devices this includes the frond-end switching and routing functionality and an additional duplex filter. For an additional UMTS or LTE frequency, additional duplex filtering is required. In addition, a separate antenna is required if the band is too far away from other bands. Furthermore, high-end devices require a special chip and board design to prevent unwanted inter-modulation effects between the cellular, Wi-Fi, Bluetooth, and GPS radio units. The white paper further estimates that, in addition to the hardware costs, the engineering cost for a new band is in the order of US\$ 6 million. This includes development costs, type approval, and testing. With annual global GSM 900 MHz phone sales of several hundred million devices, the development costs per device are only a few cents. For national frequencies for which only a few million devices are sold per year, the development costs for supporting the frequency band per device can easily exceed the price of the hardware.

From a development point of view, many manufacturers thus prefer to focus their design resources to decrease the price of their next-generation designs and to improve sensitivity rather than to add exotic frequency bands. In other words mainstream bands attract much more engineering effort resulting in less expensive hardware with higher sensitivity.

In the future, analog hardware in mobile devices is likely to get more expensive since technologies such as HSPA+ and LTE require at least two antennas and receiver chains for the MIMO transmission (cf. Chapter 2). Standards even include data transmission modes for 4×4 MIMO, which requires four separate antennas and receiver chains in mobile devices. As the hardware components cannot be shared between receiver chains, this increases the number of required components and thus increases costs. If in addition to Wi-Fi, Bluetooth, and GPS, several frequency bands for cellular B3G networks are supported, physical limits will limit either the number of MIMO antennas per band or the number of supported bands per device.

With the launch of LTE networks, two issues are further complicating a generic cellular modem design for mobile devices that can be sold globally. The first is the different evolution paths to LTE, which are incompatible to each other. On the one hand, the 3GPP GSM and UMTS standards have evolved to LTE. Devices sold globally should thus incorporate these three radio technologies. On the other hand, the CDMA standard, mostly popular in North America because of its use by Verizon, Sprint, and a few others, also uses LTE as an evolution path. As a consequence, there must be devices for these network operators that support CDMA and LTE. While in the past, LTE devices either contained GSM/UMTS or CDMA for backwards compatibility, first chipset designs now incorporate all radio technologies [59]. This makes it easier in practice to reuse most of the design of mobile devices for different parts of the world. This might thus perhaps lead to devices that are capable of operating in GSM/UMTS, CDMA, and LTE networks. At the time of publication, however, no such products were yet announced.

The second issue that complicates a generic cellular modem design is the intentionally missing circuit-switched voice capability of LTE (cf Chapter 2). While this is not a problem for data centric devices such as tablets and netbooks, smartphones and other devices which are also used for voice calling require a telephony integration that not only works over an Internet connection when an LTE network is available but also over circuit-switched channels provided by CDMA, GSM, or UMTS networks. As discussed in Chapter 4, there are different approaches to this, which are summarized again from a cellular modem point of view, as hardware support is required for each:

- Concurrent use of CDMA and LTE: Early mobile devices required two wireless radio chips; so CDMA voice and LTE data could be used simultaneously [59]. Newer chipset designs now include CDMA circuit-switched and LTE data in a single modem chip. This is also referred to as “Simultaneous Voice and LTE Data” (SVLTE).
- GSM/UMTS network operators have chosen to use a mechanism referred to CS-fallback. Incoming voice calls are signaled over LTE to the mobile device, which then requests from the network to transfer to a GSM or UMTS overlay network to receive the call over a circuit-switched bearer. The advantage to the dual-mode approach mentioned above is that the device needs to observe only a single network at a time. The downside is that the fallback to another radio technology requires several seconds, which thus significantly increases call setup times.
- Single Radio Voice Call Continuity (SR-VCC): This method is foreseen to be used with IP Multimedia Subsystem (IMS) based IP telephony in LTE networks (cf. Chapter 4). When the user leaves the LTE coverage area, an ongoing voice call can be handed over to a GSM or UMTS circuit-switched channel. At the time of publication, no network operator has deployed this method in practice.

All methods are thought to be temporary in the standardization bodies as the final goal is to have nationwide LTE networks with a coverage area that is equal to that of already installed networks. When such a level might be achieved is difficult to predict. The majority of mobile network operators are unlikely to make this step in the next five years. Until then, companies offering chipsets and devices that have integrated as many solutions as discussed above will have a significant advantage in terms of volume shipments of a single hardware version of a device.

5.7 Wireless Notebook Connectivity

One of the most important factors of the success of Wi-Fi has been Intel’s push for an embedded Wi-Fi chip in all notebooks with its mobile Centrino chipset. For some time, Intel was considering repeating this approach with a combined Wi-Fi/HSPA wireless chipset, but they gave up on these plans when their alternative WiMAX strategy emerged. This approach has not been successful, however, and Intel has since given up support on integrated WiMAX solutions.

Instead, several other possibilities have emerged to connect notebooks to cellular networks as shown in the list below. These have gained significant popularity among travelers and users who are using a wireless broadband Internet connection as an alternative to fixed line connectivity via Digital Subscriber Line (DSL) or cable:

- A built-in wireless broadband network card via a mini PCI expansion slot. Such notebooks are sold, for example, by mobile network operators in combination with a 12 or 24 month contract and a monthly fee. While an interesting concept, it has remained a niche market as the majority of people have continued buying their notebooks from other sources.
- A USB stick. This sort of wireless broadband network adapter is completely external and has the advantage that it can be placed in a convenient position and orientation, which is especially important when the signal strength to the wireless broadband network is weak (cf. Chapter 3). Sizes of external USB sticks have shrunk considerably in recent years. Also, prices have reduced significantly and entry-level devices are available in many countries with prepaid SIM cards for Internet access for less than €30–40.
- In some countries, the concept of a wireless broadband router has also had reasonable success. The dedicated router device would connect to notebooks and other devices over Wi-Fi by acting as an access point, and the Internet connection is established over a wireless broadband network such as UMTS or LTE.
- Another connectivity option for netbooks, notebooks, and other devices that enjoys rising popularity is the use of a smartphone or tablet as a Wi-Fi access point in a similar way as the router device above while outside the home or office. This is also referred to as “wireless tethering.” While already envisaged in 2006 [60], it took until 2010 when Google introduced this feature in version 2.2 of their Android operating system [61]. The downside of this approach is the limited operation time of a smartphone as the constant operation of cellular radio and Wi-Fi quickly depletes the battery. The use of a power adapter is therefore required for longer sessions. Another unintended consequence is that while routers and wireless broadband USB sticks usually have an advanced antenna design (e.g., a diversity antenna) to improve the performance, this is usually not the case in smartphones. Here, the space for an antenna is much more limited, which results in a performance degradation.

In recent years being able to access the Internet with productivity devices such as notebooks while on the go has become almost ubiquitous in many countries. This has been made possible by falling prices and cheap USB wireless broadband dongles. With monthly charges even on prepaid SIM cards between €10 and 20 for moderate use, this has triggered a significant adoption of cellular network connectivity with notebooks. In Austria, for example, about 30% of broadband subscriptions were already wireless (mostly over HSPA networks) in 2008 instead of DSL and cable [62].

5.8 Impact of Hardware Evolution on Future Data Traffic

In 2003, one of the most sophisticated connected mobile devices available in Europe was the Siemens S55 GSM/GPRS phone. It was one of the first mass market devices with a Bluetooth interface and a stable GPRS protocol stack already in the first version of the device’s software. The S55 was also the first device in this product line with a real color display with a resolution of 101×180 pixels and 256 colors. Memory card slots in mobile phones were not yet available, so file storage space in the built-in 1 Mb flash memory was limited to a few hundred kilobytes. Even if the device would have had a multimegabit

B3G network interface, it would not have been possible to take full advantage of such capabilities. The simple Web browser was only suitable to load and display small Web pages designed for mobile use with a size of only a few kilobytes.

Five years later in 2008, the mobile device landscape had changed completely. Cellular data speeds had evolved from 45 kbit/s to several megabits per second, onboard memory has grown from a few hundred kilobytes to several gigabytes and memory slots allowed an expansion of the storage capacity to tens of gigabytes. Device sizes on the other hand have only increased slightly, mostly to accommodate bigger and higher-resolution displays with resolutions exceeding 320×480 pixels and 16 million colors. At the same time, the sales price had remained the same. Web browsers had evolved to handle standard Web pages composed of hundreds of kilobytes of information. This is more than the total available flash memory storage of the S55 from 2003.

From around 2008, Internet applications have started to enjoy mainstream success and required a high amount of data to be transferred over the network. Such applications were and still are today podcasts, videocasts and music downloads. Until a few years ago, users have mostly used cable and DSL access networks to first download such files to the computer and from there transfer them to a mobile device. This process is also referred to as sideloading. With devices now incorporating Wi-Fi and broadband cellular network interfaces, it is no longer required from a technical standpoint to place a computer between the source in the Web and the mobile device. Widespread direct downloading of music files, podcasts, and video via Wi-Fi and cellular networks while having only been envisaged back in 2008 is now a reality and has reached the mass market.

This trend is further strengthened by players such as Google and Intel entering the market who do not see voice telephony as the primary application for a mobile device and a cellular wireless network but rather Internet connectivity itself. With iOS- and Android-based smartphones and tablets having become main stream devices in 2012, over half the mobile devices sold in 2012 in countries such as Germany are Internet-connected devices and hence, are usually connected to the Internet around the clock [63]. Without connectivity, these devices become almost useless. The ideas behind such devices are very different from those of previous mobile phone design as they no longer focus on local applications such as calendars and address books that run well even without network connectivity. On mobile Internet devices, applications such as Web browsing, feed reading, Web radio, on-demand videos, Voice over Internet Protocol (VoIP) telephony, video chatting, messaging, and gaming are prevalent. Calendars and address books are only playing a secondary role and even these applications are tightly integrated with an Internet-based backend; so users synchronize them between different mobile devices and PCs. Over the past five years, this trend has seen data volumes in wireless networks increase every year between 50% and 100% as more and more people have bought a connected mobile device and because of the rising number of connected applications. It should be noted however, that it has been reported by the end of 2011 that this year on year growth has slowed to some extent in some networks. Potential reasons cited for this are more stringent data caps and take-up of mobile Internet-based devices by users who, unlike those who were taking-up these devices earlier, are not using their mobile devices as often or for as many services [64].

Applications that require a significant amount of bandwidth are music and video streaming. Streaming a 30 s video from platforms such as Youtube generates around 2.7 MB of

data, that is, around 5.4 MB per minute [33]. Another data-intensive application is radio streaming. Streaming a radio station 8 h a day with a data rate of 128 kbit/s, for example, results in a data volume of over 460 MB per day or 13 GB per month. As users mostly watch video clips and listen to radio at home or in the office, the Wi-Fi interface in mobile devices is an ideal way to offload this type of traffic from the cellular network to DSL and cable connections.

With the examples above it becomes clear that mobile devices today and in the future, especially those dedicated for being used mainly with Internet-based applications will continue to drive the amount of data in mobile networks. Also, first attempts are made to build devices where all applications are executed in a web browser such as, for example, the Google Chrome book [65]. The applications and the data they use are downloaded from a server in the network and only cached locally. While still in its infancy in 2012, this approach might become more interesting to users with broadband networks becoming more and more ubiquitous. It is thus likely that network operators will continue to increase the overall capacity and reach of their cellular networks. This will likely change the overall network design over time as discussed in the next section.

Another future challenge will be the management of several devices per user. Requiring a separate contract or prepaid SIM subscription for each device will become impractical once users are connected with their notebook, their private phone, their business phone, their dedicated MP3 player, an Internet tablet, their car, and so on.

5.9 Power Consumption and User Interface as the Dividing Line in Mobile Device Evolution

Tablets, also referred to as pads, are an interesting new device category since Apple has launched the first iPad back in April 2010. Many manufacturers have followed since with their own devices, mostly based on the Android operating system or Amazon, with their pads primarily designed for reading e-books. Microsoft had been a bit behind other players in this area and has thus decided to follow a different approach than that of Apple and Google.

The software used by Apple and Google for pads initially came from the low end of computing, that is, from smartphones. Smartphones are optimized for low power consumption, they have a low-power CPU, low-power components, and the operating systems were designed around low power consumption as well. Previous mobile operating systems might have been more power optimized but iOS and Android were specifically built from scratch to adapt to this environment. No legacy requirements and application had to be catered for, even though these operating systems use kernels once designed for the desktop PC, for example, Linux in the case of Android. But those kernels were shrunk, unnecessary parts were removed, and the graphical UI was designed from scratch. It was these operating systems that were then subsequently used as the basis for the new tablet device category.

From a user's point of view, the UI on smartphones and pads running the same operating system look very similar, and most applications running on smartphones can adapt to the higher screen resolution of tablets without modification. Pads might have the screen size of netbooks or even small notebooks but they have to be light, which limits battery capacity.

Consequently, the processors used in tablets are the same as those in smartphones. This can be noticed when the processor is asked to do complex tasks such as rendering graphics-intensive web pages including flash content. Such web pages are rendered more slowly than on a PC and scrolling is not as smooth. It would of course be possible to design a pad with a faster processor but this would come at the expense of how long the device will run on a single battery charge. And, when power consumption rises so does the heat generated, which is immediately noticed by the user. The speed compromise works well in most cases but power consumption can be seen as the dividing line to netbooks and notebooks and their operating systems.

Microsoft, however, is approaching tablets from a different point of view in an attempt to do things in a different way compared to their already established competitors. Microsoft has decided to scale down their Windows desktop operating system to run on the ARM platform. In addition, they have also ported their office suite and other desktop programs to run on this platform as well. In theory, only an additional Bluetooth keyboard and perhaps a mouse are required and the tablet might become a replacement for a notebook. Whether this will work in practice remains to be seen for a number of reasons: First, there is the power divide described above. While complex office suites runs smoothly on high-power Intel platforms, it is not yet certain how they will perform on a platform that has only a fraction of the processing power by design to conserve energy and limit the operating temperature. Second, success will also depend on the UI: On a tablet, big buttons and other UI elements are required that can be touched reliably with a finger. Also, current tablet devices usually use the complete display for a single application, which, despite multitasking capabilities, significantly limits the use of a tablet for complex tasks that require the use of several applications simultaneously. Simultaneous use of multiple applications can be made much simpler with a UI that allows instantaneous task switching with the help of a keyboard and a mouse. In other words, the tablet UI today is much more suited for information consumption than for interactive creation of content. A taskbar, for example, is ideal to switch between many applications instantly. A taskbar, however, is usually not part of a tablet UI, where holding a menu button for a second is the typical way to open the task manager.

Both the UI and power consumption are two areas where significant progress is made in each device generation. It is thus likely that both will evolve in a way to enable devices that are light in weight, powerful, and yet have similarly low power consumption as today's devices. This will shift the dividing line and the type of use in the future that today still separate ultramobile devices such as smartphones and tablets on the one hand and nomadic devices such as netbooks on the other.

5.10 Feature Phone Operating Systems

The middleware between the mobile device hardware and applications is the operating system. It decides how applications can access device properties and network resources directly, how the UI looks like, and how the user can interact with the device. Operating systems thus have a significant impact on the usefulness, popularity, and market success of a mobile device. The following sections now describe the most popular mobile operating systems, APIs, and business models and discuss their potential evolution in the future.

For low-end mobile phones, also referred to as feature phones, most device manufacturers are using proprietary, or closed, operating systems today. Their main purpose is voice telephony and text messaging but some of these devices also offer basic Internet capabilities. The use of such phones in many parts of the world is on a steep decline. Nevertheless, it is likely that this device category will still be produced and used in countries with very low general income for quite some time to come, despite smartphones becoming cheaper with some basic smartphones having reached a price level of around €100 [66].

Feature phones typically have no or only limited multitasking support for user applications. Multitasking capabilities are usually restricted to running a single program in the background such as a music player application. Third-party developers have no access to the operating system and can only extend the functionality of such devices with Java programs that are executed in a Java Virtual Machine (JVM) of the Java Platform Micro Edition (ME) environment, originally developed by Sun Microsystems.

5.10.1 *Java Platform Micro Edition*

The advantage of the Java Platform Micro Edition, also known as Java 2 Micro Edition (J2ME), is that a program does not run on only one device or operating system but across a broad range of different devices and operating systems. Since JVM implementations and available Java packages differ slightly between devices, some adaptations are required for support of a broad range of devices. The downside of a JVM is that applications can only get a generalized access to the operating systems and have only limited capabilities to access data from other applications such as calendar and address book entries. Also, other applications cannot exchange data with Java applications, which makes it difficult, for example, to open a Web page from a link embedded in an application in a Java-based Web browser. This limitation, imposed by the Java sandbox concept, significantly reduces the usability and interaction between different applications on the device. From a security point of view, however, the sandbox ensures that the application cannot gain access to network functions such as sending an SMS or initiating a phone call without the consent of the user. This is of particular importance on mobile devices since, unlike on PCs, programs can accidentally or intentionally cause costs by accessing the network. The Java environment is open to developers and most developers choose to distribute their applications themselves, directly to the users. Users can then download the application file to the mobile phone via the cellular network, via Bluetooth from a PC or by transferring the application file to the device from a PC via a cable.

5.10.2 *BREW*

Another cross platform application runtime environment is Binary Runtime Environment for Wireless (BREW), developed by Qualcomm. It is mostly used in CDMA-based mobile devices in the USA and Japan. BREW developers have the choice between several programming languages, C, C++, and Java. Similar to the Java ME environment discussed above, BREW offers a cross-platform programming environment but without the restrictions imposed by the Java sandbox approach. To reduce potential security problems and to ensure the quality of applications, BREW applications have to pass rigorous tests in

a certification laboratory before they can be distributed. This increases the time to market and reduces the number of developers since certification is not free. Furthermore, most CDMA network operators do not allow customers to install BREW applications themselves. Therefore, developers depend on network operators to distribute their applications. This makes developers dependent on network operators and requires negotiations with many network operators to reach a large customer base. Therefore, the BREW environment and ecosystem, which is controlled by the network operator, only attracts few developers compared with the Java ME environment, which is fully open.

5.11 Smartphone Operating Systems

At the publication of the first edition of this book in 2008, the most popular smartphone operating systems were Symbian and Windows Mobile and the use of feature phones that were mainly used for voice telephony and SMS was prevalent. Accessing services and data on the Internet on mobile devices was only used by few people at the time. In only five years, this has changed significantly. Today, smartphones, tablet devices, and use of Internet-based services on those devices are well established in all parts of society and connected mobile devices are used by people of all ages. This has been enabled by more powerful hardware in mobile devices and new players entering the market with new operating systems, new ideas, new business models, and fewer limitations imposed by maintaining backwards compatibility. In only a few years, new players such as Apple and Google have completely reshaped the mobile operating system landscape. While the first edition of this book gave an in-depth description and reasoning of why these new operating systems might become popular in the future, this has become reality at the time of publication of the second edition. As a consequence, this section has been completely rewritten to describe the current state of the art and how this part of the industry might evolve in the future.

5.11.1 *Apple iOS*

Since the launch of the first Apple iPhone in 2007, this device and its successors have enjoyed tremendous success. The operating system, referred to as iOS, has also been developed by Apple and only runs on devices developed by the company. The limited number of device variants, the release of new phones only about once every 18 months, and full control over the development of the operating system have allowed the company to create a simple to use but powerful mobile operating system that works in the same way across different device generations.

Another important element of the iOS ecosystem is a centralized store to download third-party applications. Apple was not the first company to launch a centralized app store for users to download mobile applications to their device that are either available for free or paid for. Owing to their tremendous end user popularity and the resulting negotiating power with other players in the ecosystem, it was the first company that succeeded in doing so without network operator-imposed restrictions, local adaptations (except for languages), or revenue sharing with network operators. This was difficult and impossible for other companies to achieve, which were already established in the industry because of their dependency on network operator sales channels.

iOS is a closed and proprietary operating system and the source code is not publicly available. Apple also keeps tight control over third-party applications that can only be downloaded via the company's app store. To place an application in the app store, the developer has to adhere to Apple's developer guidelines and agree to a revenue-sharing model with Apple. This model applies for payments at the time of initial purchase via the app store and also later on for in-application purchases. The latter is a popular method for e-book and e-magazine publishers to monetize their content.

On the one hand, this strict control over applications allows the company to weed out malicious applications and thus to protect users by not approving apps in the first place or to remove them quickly from the store and on devices if deemed necessary later on. This tight control over the way applications have to behave and their central download place on the other hand make users completely dependent on the interests, moral viewpoints, and technical ideas of a single company. This has given rise to a number of solutions that offer ways to lift some of the restrictions imposed such as application downloads from outside the app store and the removal of "carrier locks," that is, the use of the device with only a SIM card of one network operator. Such solutions are often referred to as "jailbreak" toolkits.

In summary, because of its success in the marketplace, it seems unlikely that Apple will change its business model of tight control of every aspect of its iOS ecosystem in the next few years. With their focus on the ease of use rather than offering many options, it is likely that their platform will continue to be successful in the foreseeable future.

5.11.2 *Google Android*

A completely different business model for mobile devices to that of Apple is used by Google with their Android operating system that was launched with a first device in 2008. Except for a few proprietary hardware drivers, every aspect of the system is available as open source. Built on a mainstream Linux kernel that is also used for Linux-based desktop and server distributions such as Ubuntu, Red Hat, Debian, Suse, and so on, most of the additional software such as the mobile optimized UI and the Java-based Dalvik third-party application execution environment has been written by Google.

For third-party applications, Google maintains the "Google Play" app-store. Although applications that are deemed malicious can be removed, there are no restrictions on developers on what kind of applications can be put in the store. Also, there is no review process before applications can be made available. Developers are also free to publish their apps in other application stores or offer them for download on their own web pages. This makes the Google third-party ecosystem fully open in a similar way as programs for desktop PCs are available today. Access rights to different parts of Android such as the address book, calendar, network, GPS, and so on have to be requested by the application at installation time from the user. The user can then either grant all requested rights together or reject them in which case the application is not installed.

Like other Linux-based operating systems, Android applications are executed in user mode and thus do not have direct access to sensitive parts of the operating system to protect the system's integrity. An application, for example, has only access to a private directory and the external memory card, while all other parts of the file system are not accessible. While this is no obstacle and even desired for the majority of applications,

there are applications that require more rights to directly access settings and protected parts of the system. An example of such an application is a network monitor program that can inspect and store data, which is exchanged with the network. Such rights are usually only granted to the administrator account of a system, which is referred to as “root” in the Linux world. Android per default does not have a mechanism for an application to request root privileges. There are, however, third-party applications that have found ways to enable the request of root privileges by applications. This is also referred to as “rooting” an Android device. Rooting a device does not give applications root rights by default. When an application requests root rights, there is still an interaction with the user required to allow or deny the request. It should also be mentioned at this point that there is a significant difference between Android device rooting and Apple jail breaking. While the former is a way to grant applications additional rights, the latter is intended to break out of Apple’s walled garden business model. Details can be found in [67].

An important part of Google’s business model is for Android to be closely linked with many of Google’s online services such as the app store, the browser, search, maps, mail, talk (instant messaging), and so on. Although it is convenient, the downside is that users give very detailed personal and usage information to a single company. Owing to the openness of the system, however, many third-party alternatives to those programs are available that are not tied to the Google ecosystem.

Most parts of Android are available as open source and are governed by the Berkeley Software Distribution (BSD) and Apache software licenses (for details see Chapter 6), which allow any company to reuse the code for their purposes. This has given rise to modified versions (“mods”) of Android for popular devices of third-party companies such as HTC and Samsung. An example is CyanogenMod [68] that is available for many different phone models. “Mods” have become popular among advanced users who would like to update to newer versions of Android more quickly than update cycles of the device manufacturers permit. Furthermore, such users often do not like the modifications made by the device manufacturers of the original Android UI and thus prefer “mods,” which use the original Google UI. In addition, mods are usually already rooted, that is, applications can request more rights to access otherwise protected parts of the operating systems and storage space.

Another aspect of Android’s license model is that any company can take the source code and produce an Android-based device outside the default Google development branch. This is referred to as “forking.” There are a number of different Android forks such as Amazon’s Kindle Fire, a tablet focusing on e-book reading [69].

As discussed earlier in this chapter, support of multitasking, that is, executing several applications simultaneously, is one of the big differences between mobile and desktop devices. Although both the iOS and Android kernels are capable of multitasking, iOS significantly restricts this capability for third-party applications. In effect, iOS only permits one third-party application to run at a time and the application is forced to exit when the user returns to the main screen. This is in contrast to Symbian, for example, that has had full multitasking capabilities for many years without compromising system stability or higher power consumption that are often quoted as being the reason for restricting multitasking on mobile devices. Android’s multitasking capabilities are somewhere in between, with applications being permitted to run in the background if programed in a certain way. However, there are no guarantees that applications will continue to run in the background

and it can be observed in practice that applications are often terminated after some time of inactivity in the background or when memory becomes scarce. This is quite different to Symbian and desktop operating systems, which can swap memory to the file system when more memory is required for active applications. As RAM in mobile devices continues to increase, this is likely to become less of a problem in the future and it is likely that Android's existing multitasking capabilities for third-party applications will continue to evolve over time to reach similar levels as Symbian and desktop based operating systems.

5.11.3 *Android, Open Source, and its Positive Influence on Innovation*

When Java was implemented on mobile devices many years ago, third-party companies for the first time had the possibility to write software that runs on many different mobile phones. Many other programming environments have followed over the years and the most popular ones are currently the native programming environment for the iPhone, Android, and Symbian. While those programming environments are very flexible, they nevertheless deliberately set one limitation: All third-party programs are shielded from the operating system. Applications can use the API offered to them but they are restricted from directly accessing any hardware or to interact directly with other parts of the operating system. This has partly changed, however, as the Android operating system is open source.

As Android does not only publish an API for applications running in the Dalvik virtual machine but the complete source code of the operating system, companies interested in offering functionality that requires interaction with the hardware or that directly extends the functionality of the operating system can do so relatively easily. This can be demonstrated with the following two examples:

- **GAN for Android:** Orange UK and T-Mobile US are selling Android-based phones with Kineto's GAN stack that tunnel GSM voice calls over Wi-Fi. Details can be found in [70, 71]. Developed many years ago, GAN depended on Nokia, Samsung, and others to integrate the GAN protocol stack into mobile devices as the programming environments of their phones did not allow third parties to dwell deep enough in their operating system.
- **NFC functionality:** NXP and G&D have worked on integrating NFC and SIM Secure Element functionality into the Android OS [72]. This is also clearly something that would have previously only been possible for device- or operating system manufacturers. With Android being open source, this can now be done by third parties to enable their services.

Modifying or enhancing the operating system, however, has one disadvantage from a software distribution point: an application code running outside the environment provided by the API cannot be installed by users on non-rooted devices. In other words, such solutions have to be delivered as part of the Android firmware image of a particular manufacturer unless the code is included by Google in the default Android release cycle.

5.11.4 *Other Smartphone Operating Systems*

Apart from iOS and Android described above, there are a number of other mobile operating systems in use today. At the publication of the previous edition of this book, Symbian

was used by a number of different companies for their smartphones. This has significantly changed with the arrival of iOS and Android for a number of reasons. Like Linux, the Symbian kernel and the various user interfaces such as S60, UIQ, and MOAP have a long history dating back to the first PSION PDAs. In later years, Symbian was developed in collaboration by established device manufacturers among which Nokia took the most active role. After the launch of iOS and Android however, Symbian's popularity saw a quick decline as the companies owning the OS had difficulties adapting the UI for touch input quickly enough. This is perhaps in part because of device manufacturers not being software driven and Internet based compared to Google and Apple, which resulted in slower innovation cycles. Also, their competition was free from legacy requirements and could thus develop ideas more quickly than the incumbent device manufacturers. In 2012, Symbian had evolved to a UI comparable to Android and iOS but by this time it had lost a lot of support in the community. A further contributing factor in this was Nokia's decision to hire a CEO who preferred Windows Phone as the new operating system for the company and who abolished the previous succession path from Symbian to Meego. Meego, a successor of Nokia's Memo and Intel's Moblin is based on a Linux kernel and got high praises by industry observers when launched on the N9 in a few countries. Despite its success, however, it was decided not to develop the system further until today. The future of Symbian and Meego is therefore uncertain and unless Nokia's course taken in 2011 is reversed, these operating systems are likely to disappear from the market.

Microsoft's Windows Mobile saw an equal drop in popularity at the end of the 2000s and Microsoft took the drastic step of designing a completely new mobile operating system, which became Windows Phone 7. With no backwards compatibility to Windows Mobile, Microsoft was free to innovate on various aspects of the operating system, especially on the UI, which is based on tiles unlike any other mobile operating system today. From a business model point of view, Microsoft has chosen a similar walled garden model as Apple with iOS, which means a closed-source operating system, an app store controlled by Microsoft without the possibility of loading third-party application from other sources. Also, multitasking capabilities are severely limited compared to Symbian and Android. Unlike Apple, however, Microsoft had initially chosen not to develop the hardware itself but to license the operating system to third-party companies. In 2012, companies such as HTC, Samsung, and LG produce Windows Phone 7 devices but only few models have appeared so far, perhaps because of the fact that Microsoft gives strict guidance on the hardware to be used for the devices and does not allow the manufacturer companies to modify the operating system. This approach has not led to an initial success and it remains to be seen how the OS will fare in the future. One possibility might be that the cooperation between Microsoft and Nokia may be intensified in the future to give Microsoft more control over the hardware itself. With Microsoft's decision to use a derivate of the Windows 8 desktop operating system for its new tablets, it is also not clear how the Windows Phone 7 operating system will evolve in the future as the evolution toward a combined smartphone and tablet operating system is thus in effect blocked. With smartphone hardware becoming more and more powerful and the desktop Windows operating system becoming more adapted to the tablet domain, there are speculations that at some point the desktop OS might also be used for smartphones.

5.11.5 *Fracturization*

Compared with the desktop computing world with its three main operating systems, Windows, Mac OS, and Linux, with relatively stable market shares, the mobile device landscape has been much more diverse and fluctuant over time. In addition to the operating systems and APIs discussed in this chapter, there are many other proprietary operating systems used in mid-tier and low-end mobile phones. This makes it difficult for developers to design mobile applications across a wide range of different devices. In recent years, two dominant platforms have emerged with iOS and Android, while other previously dominant platforms such as Symbian are being phased out.

This concentration on two or perhaps three major mobile operating systems in the future has made it easier for developers, and the situation is now similar to the PC world. iOS only runs on few device versions and thus is the easiest of all platforms to develop for from a software point of view. Android is used on very different devices in terms of screen resolution, processor, and graphics capabilities, and several versions of the operating system are in active use. Although most programs can be written in a generic way to accommodate for these differences, this environment is especially difficult for applications that are hardware dependent. 3D games are a typical example. Despite having a common API for 2D and 3D graphics programming as described above, the difference in processing speed of both the application processor and the graphics unit has to be taken into account.

On the positive side, diversity helps reduce the effect of malware. Although today the threat from malware such as viruses on mobile devices is still small, it is likely that this area will get more attention in the future as the number of users and devices increases. The more different operating systems and device combinations are on the market, the more difficult it is for a virus or other harmful program to propagate from one system to another.

It is interesting to note at this point that the market share of the major mobile operating systems is the reverse compared to their desktop counterparts. Microsoft Windows enjoys an overwhelming desktop market share and is followed at a distance by Apple. Linux only has a miniscule percentage of the overall desktop market share. In the mobile world, however, Linux-based Android is now the most popular mobile operating system. It is followed by Apple's iOS, which is in turn followed in the distance by Microsoft's Windows Phone operating system.

With services and applications now used on both mobile and desktop-like devices, it will be interesting to observe how this will influence market shares on both sides. Microsoft is hoping to replicate their success in the desktop market by developing versions of their desktop operating system for tablets and perhaps in the future for smartphone-like devices, thereby hoping to change the current market share situation in the mobile space to their advantage. Google tries to increase the take-up of web services that will run on any underlying operating system, thus securing their cloud-based business model. Apple's approach to control the ecosystem from end to end continues to attract many new users to their mobile devices and this popularity also had a positive effect on their desktop market share in the past few years. It is thus very likely that unlike in the PC world where Microsoft dominated the operating system market for many years, there will be at

least two or three thriving ecosystems in the mobile space in the foreseeable future with the potential to change the balance of power in the desktop world as well.

5.12 Operating System Tasks

Today, operating systems for mobile devices have reached a level of functionality and complexity equal to operating systems for PCs and notebooks. With the introduction of Linux as an operating system for mobile devices, there is no longer even a difference from a practical point of view. The following section now takes a look at the basic building blocks and functionalities of a high-end mobile operating system such as Android, iOS, Symbian, and Windows Phone.

5.12.1 *Multitasking*

Operating systems of all mobile devices, from entry level to high-end devices, must be capable of multitasking as there are a number of tasks that need to be performed quasi simultaneously. The most important task of a connected mobile device, even while not communicating with the network, is to monitor periodic transmissions of broadcast information from the network. This is important to stay synchronized and to receive paging messages for incoming calls and SMS messages. In addition, the device also needs to react to user input and to execute the code for the required action, like, for example, updating the display as the user moves between applications or from one menu level to another. While, on simple devices, multitasking is limited and the execution of several user applications is not possible, smartphone operating systems such as Android and Symbian offer third-party application multitasking support, which includes the execution of several user applications in addition to all tasks required for staying connected with the network and dealing with all other external interfaces such as Wi-Fi, Bluetooth, USB, and so on.

While multitasking on the application layer is usually not time-critical, monitoring the network and making decisions about moving to another cell while in idle mode is a time-critical process and the processor has to be available at specific times to analyze incoming information. In smartphones, these tasks are usually managed by a dedicated processor. One processor thus runs a proprietary operating system and all software required to communicate with the cellular network. It is often referred to as the baseband processor, while the operating system visible to the user runs on one or more application processors.

Linux and other operating systems use pre-emptive multitasking. This means that a task cannot block other tasks from running as each is interrupted by the processor when its allocated time slice has been used up and the application has not yet returned control to the operating system.

5.12.2 *Memory Management*

Programs, also referred to as tasks, running in a multitasking environment do not only share the processor but also the available main memory (RAM), where programs and data are loaded from flash memory, sometimes also referred to as the flash disk, before

they can be executed. Management of the main memory is therefore another important function of the operating system.

The operating system must ensure that programs can be executed no matter where they were loaded in memory. This is required since the order in which programs are started and stopped is not known to the operating system in advance. To make an unpredictable place in memory predictable for a program, virtual memory addresses are used in high-end mobile operating systems in the same way that they are used in PC operating systems. When a program is prepared to run by the operating system for its timeslice, the micro-processor's memory management unit is instructed by the operating system how to map the virtual memory addresses known to the program to real memory addresses. While the program is executed the memory management unit of the processor transparently translates the virtual memory addresses used by the program into physical addresses for each command. This mapping has the additional benefit that a program cannot access the memory of another task since the memory management unit would never map a virtual memory address to a physical memory address belonging to another task. For additional security, the memory management unit can only be configured by the operating system, as code running outside the operating system's scheduler does not have the permission to access the unit.

As RAM is expensive and thus a scarce resource, most high-end operating systems have the ability to use a part of the flash disk as a swap space. If the operating system detects it is running out of memory, it starts removing parts of programs and data to the swap space, which cannot directly be accessed by the processor. If a program requires access to data that has previously been swapped out, the operating system interrupts the program and retrieves the data from the swap space. Afterwards, the program is allowed to continue. In practice this works quite well in a multitasking system because only a few applications are actively running, although many applications might be loaded into the main memory. Most applications are usually in a dormant state while the user does not interact with them. A practical example of this is a Web browser that the user starts once and then leaves running while using other applications like the calendar, the notes application or the photo application to take a picture. While the Web browser application is still in memory, it is not scheduled for execution by the operating system as it does not interact with the network or the user (assuming a static Web page without Java Script or flash content, both of which can run in the background). Thus, the application is completely dormant. If in such a situation the main memory is almost fully used, the operating system can start swapping out parts of the memory required by the Web browser to the flash disk and use it for another program. For this purpose the main memory is divided into pages. The operating system is aware when each page was last used and it can decide which pages to swap out to the flash disk once memory gets scarce. As memory is organized in pages, a program does not have to be fully dormant before the operating system can swap out some of its pages. Even if active in the background, there is usually always some program code or data that has not been used recently and can thus be swapped out in the hope that not being used recently also means that this part of memory will also not be used again soon.

It is interesting to note that while Symbian can make use of memory swapping to flash memory, this is not part of Android so far. Instead, it was decided to terminate background programs if memory was required.

5.12.3 *File Systems and Storage*

In the past few years the amount of internal storage space on mobile devices for applications and data such as pictures, videos, and music files has skyrocketed. Today, device internal memory for storage has reached 16–32 GB in high-end devices and this trend is likely to continue in the future. For a short time, hard disks were used in mobile devices to reach high capacities. With falling flash memory prices, the popularity of hard disks has decreased as flash memory is much smaller, requires less energy and is more robust against vibrations and shocks. Most devices also have an external memory slot for removable flash memory cards. As these cards can be used to exchange data with other devices including PCs and notebooks, the FAT (File Allocation Table) or FAT32 file systems are used on such cards, which were originally developed by Microsoft many years ago. While due to its age it is not the most sophisticated file system standard, it is supported by all major operating systems today including Microsoft Windows, Unix/Linux and Apple Mac OS. It is thus the best choice for use in mobile devices.

5.12.4 *Input and Output*

Like any other operating system, mobile operating systems are abstracting devices attached to the system for applications. The display is the first example that comes to mind as it is the main output device to interact with the user. The display is connected to the graphics device which in turn is connected to the processor via a bus system. The operating system then abstracts the function of the graphics card into an API that can be used by applications. These APIs offer a wide variety of functions to applications ranging from simple primitives of drawing lines and shapes, to printing text at a certain location on the screen to generating graphical menus and buttons. The keyboard is a typical input device, again abstracted for programs by the operating system. A keyboard driver receives keyboard input (a key was pressed) and the operating system is then responsible for passing that information to the currently active program.

Other external input and output devices in mobile devices are, for example, Wi-Fi and Bluetooth interfaces, GPS receivers, FM radios, cameras, TV video output, touch screens, and so on, as shown in Figure 5.4. To connect these external devices with the chipset, a number of different I/O bus systems are used.

Only a few years ago, a Universal Asynchronous Receiver/Transmitter (UART) interface was also used to connect mobile devices via a cable to the serial interface of a PC or notebook for exchanging address book entries and to establish data calls to the Internet, to another computer or to a FAX machine via the mobile phone. On the PC, the UART interface is limited to speeds of around 110 kbit/s. While sufficient for many applications, such a standard serial interface is no longer capable of transporting data exchanged via broadband wireless networks as data rates now exceed several megabits per second. In recent years, serial interfaces have been replaced by USB, which is also a serial bus system but capable of much higher speeds. Until recently, most mobile devices were equipped with a USB 1.1 interface with speeds of up to 11 Mbit/s. While almost increasing transfer rates by a factor of 10 and being sufficient for using the mobile device as a broadband wireless network interface for a PC or notebook, USB 1.1 quickly became a bottleneck for applications such as transferring music files, videos, pictures and maps

between a mobile device and a PC. Consequently, high-end mobile devices now use USB 2.0 (also referred to as USB Hi-Speed) as an external interface. With data rates of up to 480 Mbit/s, the bottleneck has now moved from the transfer capabilities of the interface to how fast the processor and operating system can send and receive data and how fast that data can be written to the flash disk.

While USB 2.0 will be sufficient for connecting mobile devices to PCs for the next few years, USB 3.0 is now shipped as part of desktop and notebook computers with a peak data rate of 4 Gbit/s.

5.12.5 Network Support

As one of the main purposes of a mobile Internet device is to connect the user with other devices and people via a wireless network, support of different network types and interfaces and their abstraction for applications is another important task of mobile operating systems. Applications are usually not aware of the type of a network interface and instead request the creation of a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) IP connection to another device from the operating system. If no network connection is established at the time of the request, the operating system either decides on its own to connect to a network via one of the wireless interfaces or opens a dialog box to allow the user to select an appropriate network and configuration. Once a connection to a network is established, the operating system processes the application's TCP or UDP connection request and program execution continues. For many years now, the Internet community has been trying to migrate the current version of the IPv4 to IPv6 to counter the diminishing number of available IP addresses. This has proven to be a difficult process mainly on the network side and is due, in part, to a lack of IPv6-capable applications. Nevertheless, mobile operating systems such as Android and Symbian already support IPv6 and are shipped with IPv6 capable web browsers and other applications.

5.12.6 Security

In the days when connected mobile devices were only used for phone calls, there was little danger from external attackers gaining access to the mobile phone and the data inside. The reason for this was that the network itself isolated the devices from each other via a switching center and all commands exchanged were originated, terminated or filtered on the switching node. In addition, hardware and software of such devices were simple (cf. Figures 5.1 and 5.3) and thus offered few if any opportunities for external attacks. Today's sophisticated mobile devices are connected to the Internet, however, and are thus more and more exposed to the same kind of security threats as PCs and notebooks. It will therefore become increasingly important in the future for the operating system to defend the device against attacks or exploits. In practice there are many ways for malicious programs to gain access to a system:

- **Malicious programs**—a program should only be installed on a mobile device if its origin is known and trusted. It is therefore important that users realize that installing programs is a potential security risk and should only be done if the source is trusted.

Once a program is installed, operating systems like Linux protect the integrity of the system by executing the program in user mode, which prevents programs from making changes to the system configuration. A malicious program, however, still has access to the user's data and thus could potentially destroy or modify data without the consent of the user. Spyware, sometimes, also referred to as a Trojan horse, goes one step further and send private data it has found to a remote server on the Internet. The Symbian operating system uses a slightly different approach to application security. As described in Section 5.10.2, application developers have to get a certification for their program from an independent body before they can be distributed. Programs which do not need direct access to drivers and other lower layer components of the operating system can be distributed without being certified. The user is then informed during the installation process that the program has not been certified, which actions it wants to perform (e.g., access to the network, access to the file system, etc.) and that this presents a certain security risk. The user can then choose to abort the installation or to proceed. As most programs do not require access to lower-layer operating system services, this is the most common distribution method. Noncertified Symbian applications have similar capabilities as described for Linux applications in user mode and can thus also potentially corrupt or steal user data.

- **Peripheral software stack attacks**—another angle of attack is trying to break the software stack of network peripherals. Several well-known attacks on the Bluetooth protocol stack used malformed Bluetooth packets. In this way it was possible to access the calendar and address book on some devices without the consent or knowledge of the user. While such an attack is still possible today, fewer reports about successful attacks have been published recently and it appears that most mobile device manufacturers have done their homework. If a new vulnerability is found it is important that the manufacturer can and does react quickly and provides a patch, as is done in the PC world today.
- **Web browser attacks**—such attacks, which are quite common in the desktop PC world, try to exploit vulnerabilities of the browser software, for example, of the JavaScript implementation, to break out of the browser environment to execute system commands. Other attacks aim at potential vulnerabilities of plugins such as PDF or Flash, which can be tricked into executing infiltrated code or launch system commands with malformed documents or video files.
- **Attacks over the IP network**—in the PC world, attacks aimed at server programs waiting for incoming connections are also widespread. Like in the Web browser example above, such attacks exploit an operating system and processor weakness known as stack overflow, sometimes also referred to as buffer overflow. When one function in a program calls another, the return address is stored in a part of the memory referred to as the stack. Once the function has performed all its tasks the program returns to the previous function via the memory address stored on the stack. In addition to storing the return address, the stack is also used to store temporary data used by the called function. If the function does not ensure that the amount of memory is sufficient for incoming data, for example, from the network, then the return address and other variables can be overwritten by the incoming data. Under normal circumstances, this would result in a program fault as the program can no longer return to the previous function and the application would be terminated. This weakness, if not properly handled by the

program, can be used by malicious exploits to send a specific stream of data that overwrites the return address with a value that points to the data that was sent over the network. Instead of returning to the original function, control is given to the code which was contained in the data sent over the network. This code can then exploit further weaknesses in the operating system to gain higher operating system privileges to load further program code and to install itself in the system. While such exploits have mainly hit Microsoft's Windows operating system due to its widespread use, other operating systems such as Linux and Mac OS are by no means immune to this issue. One of the few operating systems immune to this kind of attack is Symbian as it uses descriptors that prevent buffer overflow attacks.

For the moment, there have only been a few reports about widespread or planned attacks on connected mobile devices. One reason for this is that their number compared with PCs and notebooks on the Internet is still small. Therefore, programs tailored to attack a specific type of mobile device would not find many targets yet. This also limits the spread of a virus from one mobile device to another as the virus would not work if it attacked a device such as a PC that runs a different operating system. As the number of mobile devices grows, however, there are no guarantees that mobile devices will not come under attack from viruses and other exploits in the future. Quick reaction from manufacturers to provide patches and systems automatically updating themselves is thus likely to become equally important on mobile devices as in the PC world today.

References

1. GSM Arena (2008) Panasonic GD 55—Full Phone Specifications, http://www.gsmarena.com/panasonic_gd55-372.php (accessed 2012).
2. Asus eeePC Product Homepage, <http://web.archive.org/web/20080622074016/http://usa.asus.com/products.aspx?l1=24&l2=164> (accessed 30 April 2012).
3. ARM (2008) <http://www.arm.com/> (accessed 2012).
4. Krazit, T. (2007) ARM says it's Ready for the iPhone. Cnet Newsmaker, June 22, 2007, http://www.news.com/ARM-says-its-ready-for-the-iPhone/2008-1006_3-6192601.html (accessed 2012).
5. HTC (2007) T-Mobile Unveils the T-Mobile G1—the First Phone Powered by Android, <http://web.archive.org/web/20110712230204/http://www.htc.com/www/press.aspx?id=66338&lang=1033> (accessed 2012).
6. The Digital Electronics Blog (2008) Soft Macro Vs Hard macro? September 2008, <http://digitalelectronics.blogspot.de/2008/01/soft-macro-vs-hard-macro.html> (accessed 2012).
7. Smith, M. (2012) Orange Announces First Intel-Powered Android Phone for Europe, Codenamed Santa Clara, February 26, 2012, <http://www.engadget.com/2012/02/26/orange-santa-clara-intel-medfield/> (accessed 2012).
8. Freescale Semiconductor (2006) i2xx Platform Family Product Brief, document number I2XXPB, Revision 3, June 2006.
9. Sauter, M. (2006) *Communication Systems for the Mobile Information Society*, Chapter 1.7.3, John Wiley & Sons, Ltd, Chichester.
10. ARM (2008) ARM7TDMI, <http://www.arm.com/products/CPUs/ARM7TDMI.html> (accessed 2012).
11. ARM Google Android & Chrome OS Support, <http://www.arm.com/community/software-enablement/google/index.php> (accessed 30 April 2012).
12. Schimpi, A. (2012) Intel's Medfield & Atom Z2460 Arrive for Smartphones: It's Finally Here, January 10, 2012, <http://www.anandtech.com/show/5365/intels-medfield-atom-z2460-arrive-for-smartphones> (accessed 2012).
13. Qualcomm (2011) Snapdragon S4 Processors: System on Chip Solutions for a New Mobile Age, October 7, 2011, <https://developer.qualcomm.com/download/qusnapdragons4whitepaperfnlrev6.pdf> (accessed 2012).

14. ST Ericsson Novathor L9540, <http://www.stericsson.com/products/L9540-novathor.jsp> (accessed 30 April 2012).
15. ST Ericsson Novathor U8500, <http://www.stericsson.com/products/u8500-novathor.jsp> (accessed 30 April 2012).
16. Renesas Renesas Mobile Introduces First Integrated LTE Triple-Mode Platform Optimised for Full-featured, High Volume Smartphones, <http://renesasmobile.com/news-events/news/news-20120215-LTE-Triple-Mode-Platform-Optimised-For-Full-Featured-High-Volume-Smartphones.html> (accessed 30 April 2012).
17. Nvidia Tegra 2, <http://www.nvidia.com/object/tegra-2.html> (accessed 30 April 2012).
18. Mediatek MediaTek Launches MT6575 Android Platform, http://www.mtk.com.tw/en/News/news_content.php?sn=1052 (accessed 30 April 2012).
19. ARM (date unknown) Achieving Stronger SIM-lock and IMEI Implementations on Open Terminals Using ARM Trustzone Technology, http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf (accessed 2012).
20. Texas Instruments (2007) WiLink 6.0 Single-Chip WLAN, Bluetooth and FM Solutions, http://focus.ti.com/pdfs/wtbtu/ti_wilink_6.pdf (accessed 2012).
21. Khronos Group OpenGL ES—The Standard for Embedded Accelerated 3D Graphics, <http://www.khronos.org/opengles/> (accessed 30 April 2012).
22. Jacobs, D. (2010) iPhone Application Programming CS 193P Lecture 19 OpenGL ES, March 9, 2010, http://www.youtube.com/watch?v=_WcMe4Yj0NM (accessed 2012).
23. Becker, L. (2012) Marktforscher: Apple größter Smartphone-Hersteller 2011, February 15, 2012, <http://www.heise.de/newsticker/meldung/Marktforscher-Apple-groesster-Smartphone-Hersteller-2011-1434905.html> (accessed 2012).
24. Klug, B. and Shimpi, A. (2011) Qualcomm's New Snapdragon S4: MSM8960 & Krait Architecture Explored, July 10, 2011, <http://www.anandtech.com/show/4940/qualcomm-new-snapdragon-s4-msm8960-krait-architecture/4> (accessed 2012).
25. Nvidia (2011) Variable SMP—A Multi-Core CPU Architecture for Low Power and High Performance, http://www.nvidia.com/content/PDF/tegra_white_papers/Variable-SMP-A-Multi-Core-CPU-Architecture-for-Low-Power-and-High-Performance-v1.1.pdf (accessed 2012).
26. Sauter, M. (2011) Video Stream Data Rates, December 1, 2011, http://mobilesociety.typepad.com/mobile_life/2011/12/video-stream-data-rates.html (accessed 2012).
27. Qualcomm Snapdragon Processors, <http://www.qualcomm.eu/products/snapdragon> (accessed 30 April 2012).
28. Globalfoundries 28 nm, <http://globalfoundries.com/technology/28nm.aspx> (accessed 30 April 2012).
29. Corning (2012) Corning Delivering Corning® Gorilla® Glass 2 for Next Generation Consumer Electronic Devices, February 27, 2012, http://www.corning.com/news_center/news_releases/2012/2012022701.aspx (accessed 2012).
30. Anscombe, N. (2000) Methanol Fuel Cells Seen as Mobile Phone Power Source. EETimes (Aug 2000), <http://www.eetimes.com/story/OEG20000208S0036> (accessed 2012).
31. Litchfield, S. (2012) For Those Who Were Wondering about the Apple iPhone 4S's Camera vs the N8, October 5, 2012, http://www.allaboutsymbian.com/news/item/13334_For_those_who_were_wondering_a.php (accessed 2012).
32. Litchfield, S. (2010) Camera Nitty Gritty: Supplemental — EdoF, April 15, 2010, http://www.allabout symbian.com/features/item/Camera_Nitty_Gritty_Supplemental-EDoF.php (accessed 2012).
33. Sauter, M. (2012) Youtube Data Rates to Smartphones, February 17, 2012, http://mobilesociety.typepad.com/mobile_life/2012/02/youtube-data-rates-to-mobiles.html (accessed 2012).
34. Niccolai, J. (2007) Nokia Ships Its First GPS Phone, March 22, 2007, http://www.pcworld.com/article/130064/nokia_ships_its_first_gps_phone.html (accessed 2012).
35. Qualcomm (2011) GPS and GLONASS: "Dual-core" Location for Your Phone, December 14, 2011, http://www.youtube.com/watch?feature=player_embedded&v=gB9pXInwh3k (accessed 2012).
36. Layar Wikipedia. <http://www.layar.com/layers/goweb3dwiki/> (accessed 30 April 2012).
37. Nokia Nokia N80 Support Pages, <http://europe.nokia.com/find-products/devices/nokia-n80> (accessed 30 April 2012).
38. Nokia Nokia N95 Support Pages, <http://europe.nokia.com/support/product-support/nokia-n95> (accessed 30 April 2012).

39. Wi-Fi Alliance Wi-Fi Direct, <http://www.wi-fi.org/discover-and-learn/wi-fi-direct> (accessed 30 April 2012).
40. Sauter, M. (2012) Droidsheep: Firesheep Moves to Android, January 4, 2012, http://mobilesociety.typepad.com/mobile_life/2012/01/firesheep-moves-to-android.html (accessed 2012).
41. Sports-Tracker Turn Your Mobile Into a Social Sports Computer, <http://www.sports-tracker.com/> (accessed 30 April 2012).
42. FeliCA Networks KESAKA e-Key Service, <http://www.felicanetworks.co.jp/en/cases/others.html> (accessed 30 April 2012).
43. The NFC Forum FAQ, <http://www.nfc-forum.org/resources/faqs> (accessed 30 April 2012).
44. WG8 (2010) ISO/IEC 14443. *Proximity Cards (PICCs)*, October 25, 2010, <http://wg8.de/sd1.html#14443> (accessed 2012).
45. Dawson, F. and Howes, T. (1998) vCard MIME Directory Profile, September 1998, <http://tools.ietf.org/html/rfc2426> (accessed 2012).
46. NXP Semiconductors (2009) NFC Forum Type Tags, http://www.nfc-forum.org/resources/white_papers/NXP_BV_Type_Tags_White_Paper-Apr_09.pdf (accessed 2012).
47. Perez, S. (2012) Foursquare Adds NFC Support to its Android App, February 10, 2012, <http://techcrunch.com/2012/02/10/foursquare-adds-nfc-support-to-its-android-app/> (accessed 2012).
48. Clark, S. (2012) EMVCo Picks GlobalPlatform Secure Element Compliance Programme, February 21, 2012, <http://www.nfcworld.com/2012/02/21/313555/emvco-picks-globalplatform-secure-element-compliance-programme> (accessed 2012).
49. Vétillard, E. (2012) NFC Payments 101, February 16, 2012, <http://javacard.velilles.com/2012/02/16/nfc-payments-101/> (accessed 2012).
50. Mobile Gazette (2008) LG HB620T with DVB-T, May 1, 2008, <http://www.mobilegazette.com/lg-hb620t-08x05x01.htm> (accessed 2012).
51. Blass, E. (2007) Hands-on with Texas Instruments' Cellphone Projector, Engadget, September 2007, <http://www.engadget.com/2007/09/20/hands-on-with-texas-instruments-cellphone-projector/> (accessed 2012).
52. DLNA *dlna Connect and Enjoy*, <http://www.dlna.org>. (accessed 30 April 2012).
53. 3GPP (2012) User Equipment (UE) Radio Transmission and Reception (FDD). TS 25.101, Table 5.0, version 11.1.0.
54. 3GPP (2012) Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception. TS 36.101.
55. Sauter, M. (2010) The Nokia N8—Pentaband UMTS, July 1, 2010, http://mobilesociety.typepad.com/mobile_life/2010/07/the-nokia-n8-pentaband-umts.html (accessed 2012).
56. Ziegler, C. (2006) T-Mobile Details 3G Plans, Engadget, October 2006, <http://www.engadgetmobile.com/2006/10/06/t-mobile-details-3g-plans/> (accessed 2012).
57. Sorensen, C. (2008) Telus Considers Dumping its 'Betamax' of Wireless Networks, The Toronto Star, January 2008, <http://www.thestar.com/Business/article/293353> (accessed 2012).
58. Varrall, G. (2007) RF Cost Economics for Handsets, RTT, May 2007, <http://www.rtonline.com/Research/RF%20Cost%20economics-Handsets-white%20paper.pdf> (accessed 2012).
59. Klug, B. (2011) HTC Thunderbolt Review: The First Verizon 4G LTE Smartphone, April 27, 2011, <http://www.anandtech.com/show/4240/htc-thunderbolt-review-first-verizon-4g-lte-smartphone> (accessed 2012).
60. Sauter, M. (2006) A Nokia N80 as WLAN Access Point—A Double Blow for Mobile Operators? August 2, 2006, http://mobilesociety.typepad.com/mobile_life/2006/08/a_nokia_n80_as_.html (accessed 2012).
61. Sauter, M. (2011) Android Wi-Fi Tethering—Great But Watch The Battery, April 5, 2011, http://mobilesociety.typepad.com/mobile_life/2011/04/android-wi-fi-tethering-great-but-watch-the-battery.html (accessed 2012).
62. Sauter, M. (2008) Wireless now Accounts for a Third of Austria's Broadband Connections, February 2008, http://mobilesociety.typepad.com/mobile_life/2008/02/wireless-now-ac.html (accessed 2012).
63. Bitkom (2012) Zeitenwende auf dem Handy-Markt, February 15, 2012, http://www.bitkom.org/de/presse/8477_71243.aspx (accessed 2012).
64. Bubley, D. (2011) Has Mobile Data Growth Flattened Off? Are Caps & Tiers Working Too Well? November 11, 2011, <http://disruptivewireless.blogspot.fr/2011/11/had-mobile-data-growth-flattened-off.html> (accessed 2012).

65. Google Introducing Chromebooks, <http://www.google.com/intl/en/chrome/devices/> (accessed 30 April 2012).
66. Catanzariti, R. (2010) Huawei IDEOS (U8150) Android Smartphone, November 11, 2010, http://www.pcworld.idg.com.au/review/mobile_phones/huawei/ideos_u8150/363402 (accessed 2012).
67. Sauter, M. (2011) Rooting Does Not Equal Jailbreaking, June 1, 2011, http://mobilesociety.typepad.com/mobile_life/2011/06/rooting-does-not-equal-jailbreaking.html (accessed 2012).
68. CyanogenMod, <http://www.cyanogenmod.com/> (accessed 30 April 2012).
69. Spurbeck, J. (2011) Three Android Forks that Exist Today, September 17, 2011, <http://news.yahoo.com/three-android-forks-exist-today-135000744.html> (accessed 2012).
70. Campbell, S. (2010) T-Mobile Offering WiFi Calling on Android Phones with UMA, October 7, 2010, <http://fixed-mobile-convergence.tmcnet.com/topics/mobile-communications/articles/107057-t-mobile-offering-wifi-calling-android-phones-with.htm> (accessed 2012).
71. Beren, D. (2010) Kineto Wireless Brings UMA to Android, Will T-Mobile Adopt? September 21, 2010, <http://www.tmonews.com/2010/09/kineto-wireless-brings-uma-to-android-will-t-mobile-adopt/> (accessed 2012).
72. Clark, S. (2011) NXP and G&D Confirm Joint Android NFC Secure Element Interface, February 15, 2011, <http://www.nfcworld.com/2011/02/15/36027/nxp-and-gd-confirm-joint-android-nfc-secure-element-interface/> (accessed 2012).