



# Selected Topics in Computer Engineering ENC-360

**Dr. Feda AlShahwan**

Computer Section

Electronic Engineering Department

College of Technological Studies

Public Authority for Applied Education and Training





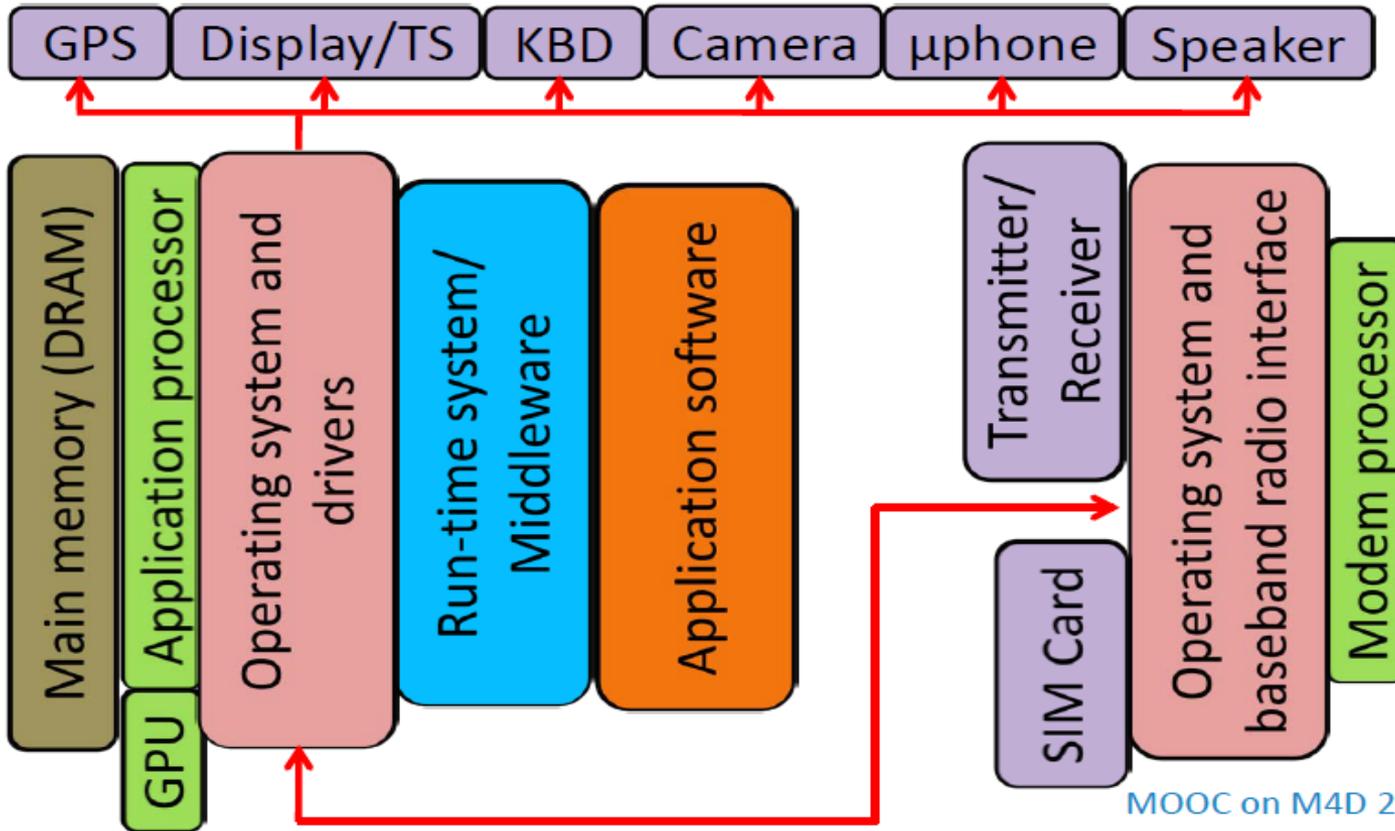
# Chapter 3

- What is the hardware architecture of a smartphone?
- How does communication take place in a smartphone?
- Where does a user application execute in a smartphone?
- What are the important peripheral devices in a smartphone?
- Which processors are commonly used in a smartphone?
- What is ARM TrustZone?



# Chapter 3

## Smartphone hardware architecture



MOOC on M4D 2013



# Chapter 3

## Reference hardware

### Smartphone hardware architecture

- A System-on-a-Chip (SoC) architecture
  - a combination chipset that features things such as the actual processor cores, the graphics chipset, the RAM and possibly ROM as well, interface controllers for things such as USB and wireless tech, voltage regulators and more.
  - SoC consists of three primary components
    - An application processor, Graphical Processor and memory executing the end-user's application software with assistance from the middleware and operating system (OS).
    - A modem or baseband processor with its own operating system components responding to the baseband radio activities (transmission and reception of audio, video, and other data contents).
    - A number of peripheral devices for interacting with the end-user.



# Chapter 3

## Reference hardware

### I) Peripherals

- These are the I/O devices through which the end-user interacts with the handheld
  - The OS needs to have the driver software installed for each such device
  - Typical peripheral devices
    - LCD and touchscreen
    - Keyboard
    - Camera
    - GPS
    - Speaker and audio output for headset/earphone
    - Microphone
    - Bluetooth and Wifi
    - HDTV



# Chapter 3

## Reference hardware

### II) Modem Processor

- Modem processor is either a separate ARM core or a **Digital Signal Processing (DSP)** extension of the application processor ARM core
  - What is DSP : Techniques for improving the accuracy and reliability of digital communications.
- Some architectures use a modem accelerator along with the application processor core

### How does communication take place in a smartphone?

- Reception
  - The receiver hardware (part of the modem) senses incoming signals and generates interrupts for the radio interface logic of the operating system.
    - The radio interface and the operating system software run on a baseband or modem processor
  - Once the reception begins (after a physical layer handshake), the incoming audio, video, and other data are processed by the modem processor
  - The radio OS components talk to the peripheral device drivers to present the incoming data to the user through appropriate devices (display, speaker, etc.)



# Chapter 3

## Communication Mechanism

- **Transmission:**
  - The data to be transmitted are collected by the radio OS components from memory regions populated by the device drivers:
    - For example, audio data captured by the microphone driver or an image or a video captured by the camera or a position information captured by the GPS device
- These data can be further processed by the modem processor to suite the transmission protocol
- A transmission is initiated by the radio interface logic through the modem transmitter hardware
- The subscriber identification module (SIM) plays an important role in reception and transmission



# Chapter 3

## Reference hardware

### III) Application Processor

- Executes the user applications and the related OS services
- Applications include audio/video codec and players, games, image processing, speech processing, internet browsing, text editing, etc.
- Application processor takes help from graphics accelerators as and when needed.
- Most handheld applications are graphics-intensive
- Handhelds come with reasonably large amount of storage in the form of volatile SDRAM (1-2 GB) as well as non-volatile compact storage (10+ GB)
- The OS is mostly a traditional one, stripped down and optimized to cater to smartphone applications
- There are two types of application processors that differ in the performance, power consumption, and cost : ARM-based processor and snapdragon.



# Chapter 3

## Reference hardware

- Need to balance performance, power consumption, and cost
- ARM-based processors are very common
  - ARM Holdings plc is the British-based company that, since 1983, has developed the ARM microprocessor instruction architecture which is used inside their ARM processor cores.
  - Optimized for battery life as well as performance
  - Remarkably low area and transistor count
  - Important for small form factors and low energy drain
- Snapdragon-based processors is another common processor for use in smartphones, tablets, and ebook devices.
  - It has many features similar to those of the ARM Cortex-A8 core and it is based on the ARMv7 instruction set
  - But theoretically has much higher performance for multimedia-related (Single Instruction Multiple Data)SIMD operations
  - Contains the circuitry to decode High-Definition video (HD) Snapdragon SoCs have been unique in that they have had the modem for cellular communication on-die. That is, they do not require a separate external cellular modem on the SoC.
  - A multi-core processor is an integrated circuit (IC) to which two or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks (see parallel processing). A dual core set-up is somewhat comparable to having multiple, separate processors installed in the same computer, but because the two processors are actually plugged into the same socket, the connection between them is faster.



# Chapter 3

## Reference hardware

### Some Common handheld processors:

- Modern handhelds include multiple application processor cores (two, four, or eight)
  - Apple iPhone 5
    - 1.3 GHz dual-core Swift (ARMv7-based) + PowerVR GPU
  - Nokia Lumia 920T
    - 1.7 GHz dual-core Qualcomm Krait + Adreno GPU
  - Samsung Galaxy S4 i9500 comes in two possible configurations depending on the region and network
    - 1.9 GHz quad-core ARM Krait + Qualcomm's Adreno GPU
    - 1.6 GHz quad-core ARM Cortex-A15 + 1.2 GHz quad-core ARM Cortex-A7 + Imagination's PowerVR GPU (only four cores out of the eight app. cores can be active at a time)
  - Samsung Nexus 10
    - 1.7 GHz dual-core ARM Cortex-A15 + ARM Mali-T604 GPU
  - Asus Nexus 7
    - Nvidia Tegra 3 platform with 1.2 GHz quad-core ARM Cortex-A9 + ultra-low-power Nvidia GeForce GPU
- These multi-core app. processors can deliver excellent performance although low-power remains a primary goal for longer battery life



# Chapter 3

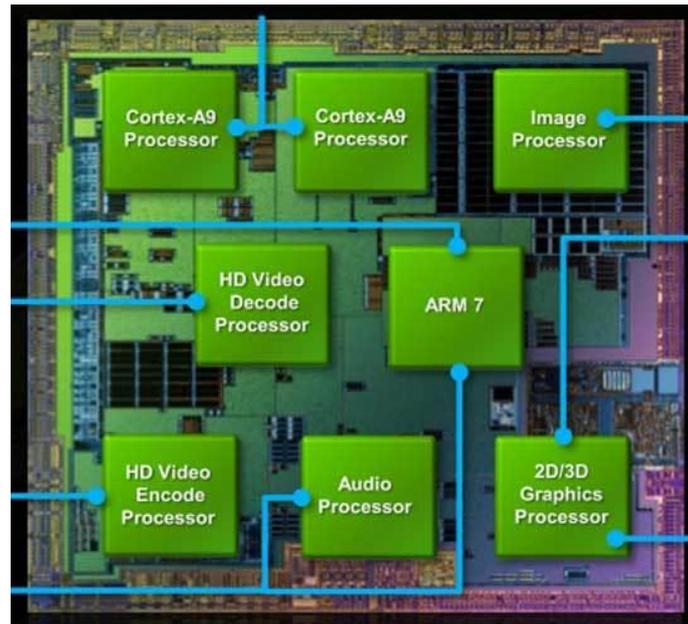
## Reference hardware

### GPU

Graphics Processing Unit (GPU) is very close to the processor

Why?

- First, smartphones and tablets do not have a huge amount of internal space to work with, and so having critical components packaged together allows the device's mainboard to be small and the battery to be large. Secondly, packaging the two units as one reduces the heat output of the device as it's more localized and can save power through tightly integrating the two. Finally, it saves manufacturing costs to produce one chip instead of two.



The GPU is the "2D/3D Graphics Processor" part of the Tegra 2 SoC



# Chapter 3

## Reference hardware

### GPU Functionality:

#### The use of the GPU depends on several factors:

- Structure of the system-on-a-chip
- The operating system used on the device.
- if the SoC does not have a dedicated media decoding chip then the GPU is used to handle high-resolution videos. There is also the possibility that compatible power intensive tasks are offloaded to the GPU.
- When it comes to the operating system things are a lot more complex. First and foremost the GPU is used entirely for all 3D rendering in games and applications. The Cortex processing cores are simply not designed to handle these sorts of tasks and in all operating systems the GPU will take over from the CPU to handle the rendering more efficiently. The CPU will help out for certain calculations while rendering 3D models on screen (especially for games), but the main grunt will be done by the graphics chip.
- Most graphics cores also support 2D rendering in certain areas: things such as interface animations and image zooming are two good examples. The processor can also usually handle these tasks so whether the GPU is used is usually up to the operating system used on the device.



# Chapter 3

## Reference hardware

### Types of GPU

#### Qualcomm Adreno GPUs

- The Adreno graphics processing unit is the proprietary graphics chipset used in Qualcomm SoCs. Adreno GPUs used to be called Imageon and they were manufactured by ATI until Qualcomm bought the division from AMD and renamed the products to Adreno. The old Adreno 1xx series were used in old Qualcomm 7xxx SoCs, with the newer Adreno 2xx series being used inside the Snapdragon series.
- In the current range of Snapdragon SoCs you see three Adreno 2xx series GPUs used: the Adreno 200 (for S1), 205 (for S2) and 220 (for S3).

#### Imagination Tech PowerVR GPUs

- The second major producer of smartphone graphics chipsets is Imagination Technologies, which makes the PowerVR line of mobile GPUs.
- There have been many series of PowerVR GPUs, though current devices use products from either the PowerVR SGX 5 or 5XT series.

#### ARM Mali GPUs

#### NVIDIA ULP GeForce

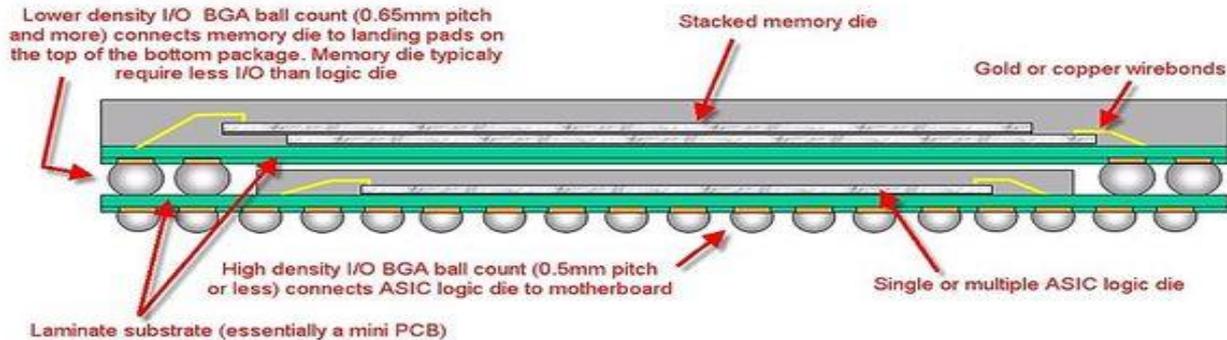


# Chapter 3

## Reference hardware

### Memory and Storage

- RAM that is used in smartphones is technically DRAM, with the D standing for dynamic. The structure of DRAM is such that each capacitor on the RAM board stores a bit, and the capacitors leak charge and require constant “refreshing”; thus the “dynamic” nature of the RAM. It also means that the contents of the DRAM module can be changed quickly and easily to store different files.
- RAM usually can be found on top of the SoC in what is known as a package-on-package (PoP) set-up. This allows the SoC direct access to the RAM and the close proximity means less heat output and power consumption.



Package-on-package set-up. The lower die is SoC and the upper is the RAM



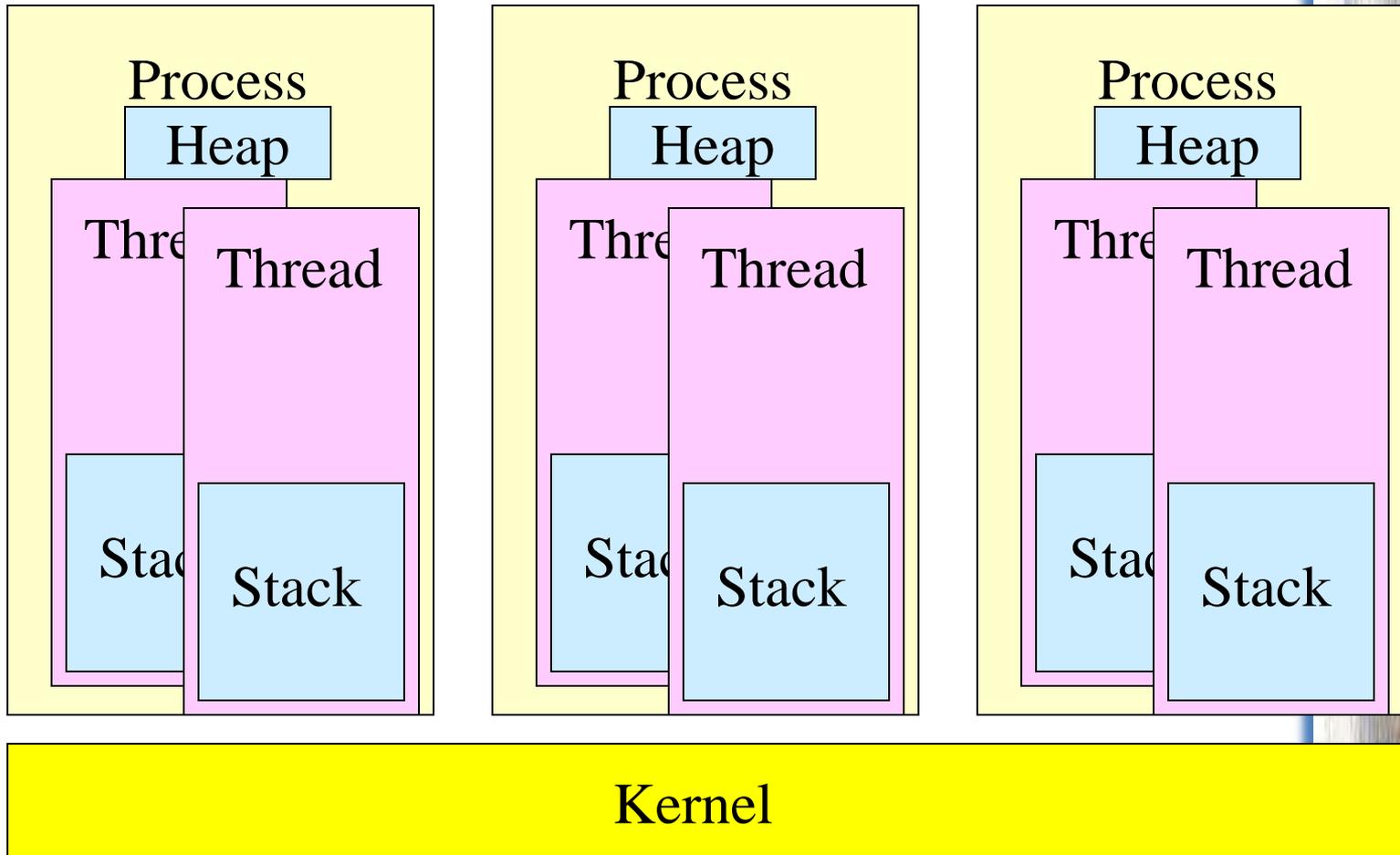
# Chapter 3

## ARM Trust-Zone

- ARM TrustZone is a hardware-software solution for security in handhelds
  - Security: a property of the system which ensures that resources of value cannot be copied, damaged, or made unavailable to genuine users.
  - Important pieces of information such as various encryption keys must be protected
  - TrustZone hardware allows the application processor to execute in one of the three modes: normal, monitor, and secure
    - Normal to secure transition happens through monitor
  - TrustZone software offers a set of secure OS services and the capability to add new user-defined secure services through TrustZone APIs such as SIM-locking



# Reference software



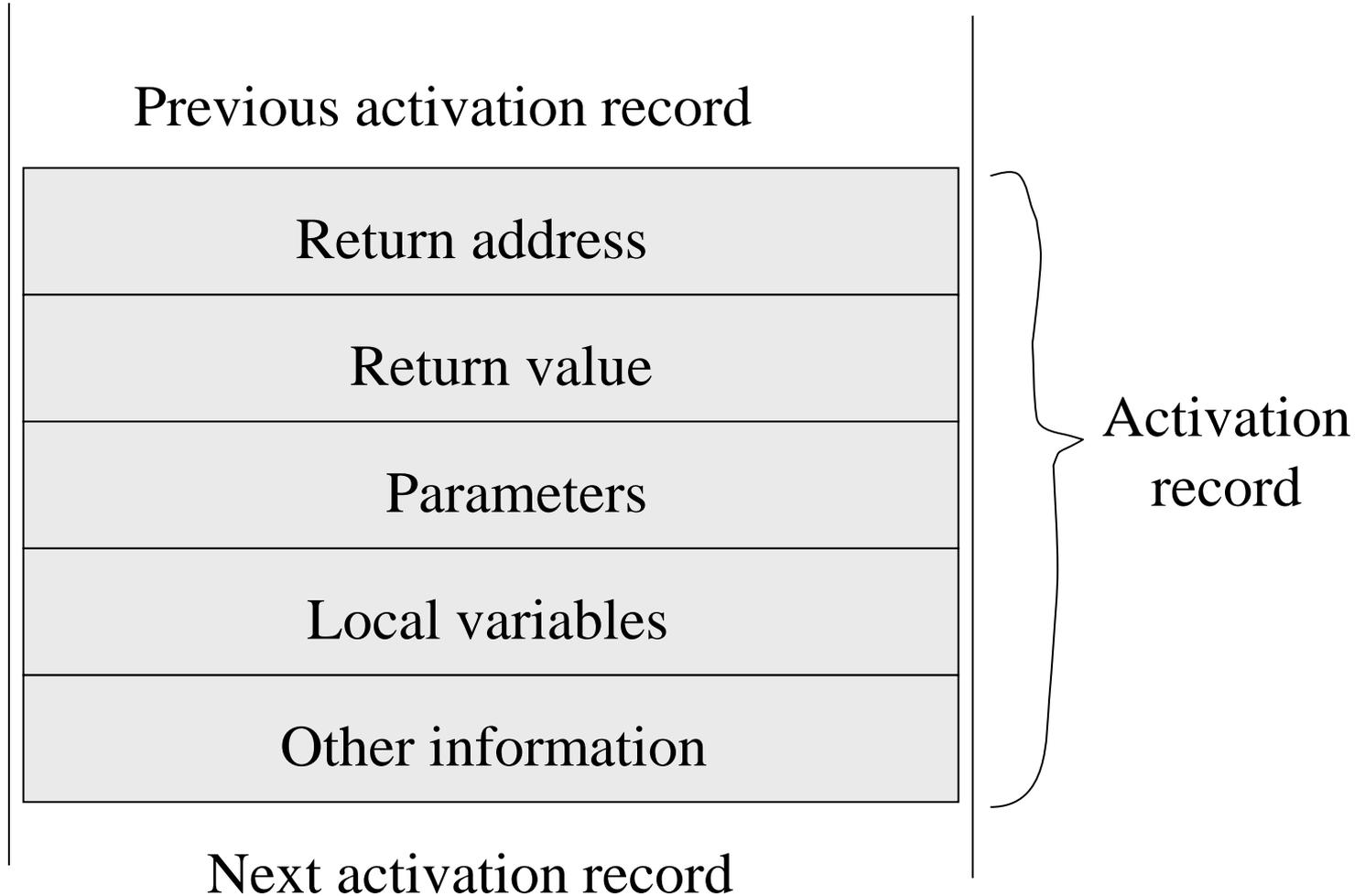


# Chapter 3

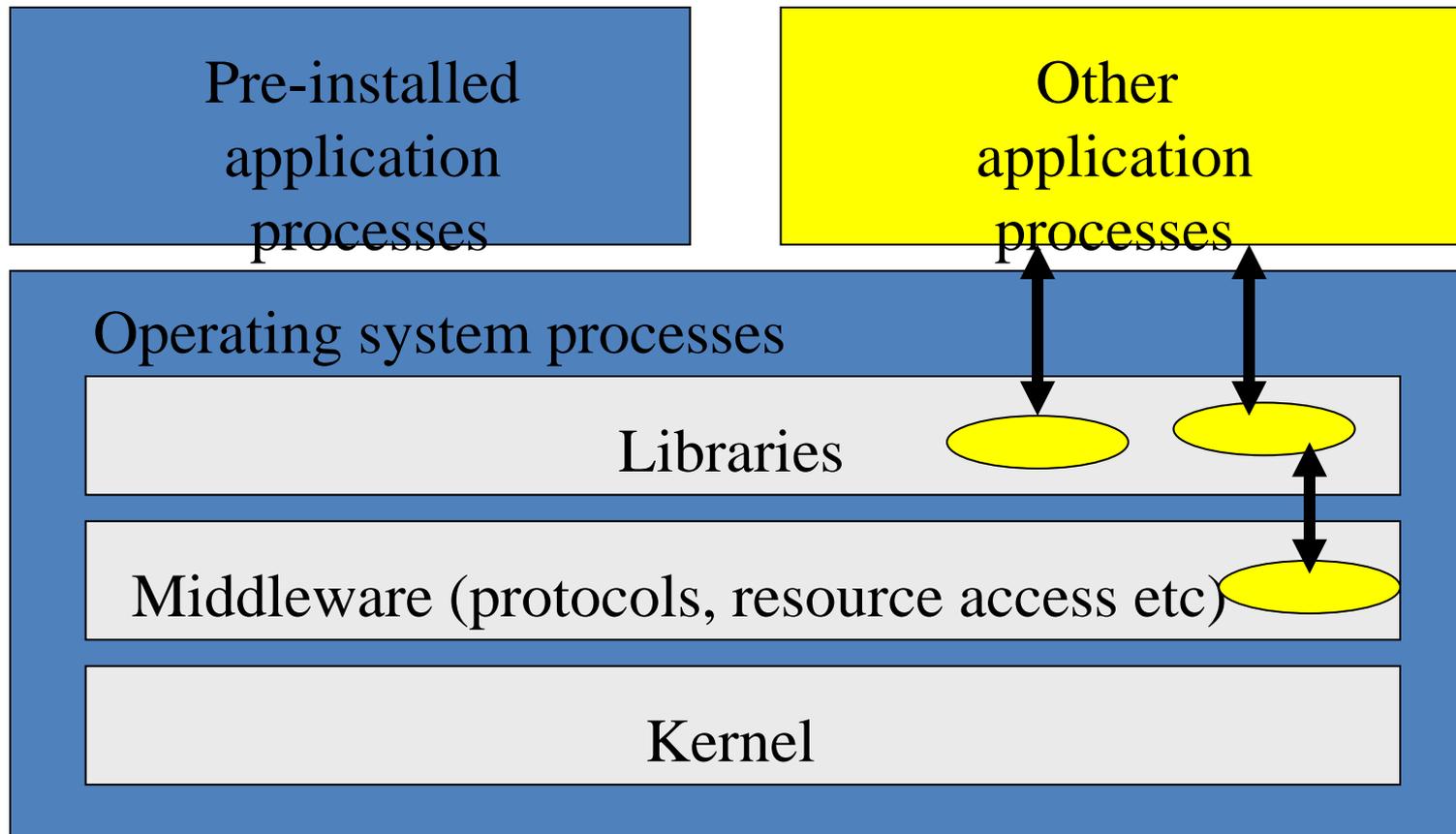
## Reference Software

- **Kernel:** In computing, the **kernel** is a computer program that manages input/output requests from software, and translates them into data processing instructions for the central processing unit and other electronic components of a computer. The kernel is a fundamental part of a modern computer's operating system.
- When a computer program (in this context called a *process*) makes requests of the kernel, the request is called a system call. Various kernel designs differ in how they manage system calls and resources. For example, a monolithic kernel executes all the operating system instructions in the same address space in order to improve the performance of the system. A microkernel runs most of the operating system's background processes in user space, to make the operating system more modular and, therefore, easier to maintain.
- Because of its critical nature, the kernel code is usually loaded into a *protected area* of memory, which prevents it from being overwritten by other, less frequently used parts of the operating system or by application programs. The kernel performs its tasks, such as executing processes and handling interrupts, in *kernel space*, whereas everything a user normally does, such as writing text in a text editor or running programs in a GUI (graphical user interface), is done in *user space*. This separation is made in order to prevent user data and kernel data from interfering with each other and thereby diminishing performance or causing the system to become unstable (and possibly crashing).
- Features of Kernel
  - Context-switching machine
  - Pre-emptive vs. Non-pre-emptive: In computing, preemption is the act of temporarily interrupting a task being carried out by a computer system, without requiring its cooperation, and with the intention of resuming the task at a later time. Such a change is known as a context switch. It is normally carried out by a privileged task or part of the system known as a preemptive scheduler, which has the power to preempt, or interrupt, and later resume, other tasks in the system.
- **Process**
  - Unit of resource management (memory, files, etc)
- **Thread**
  - Unit of execution
- **Stack:** What is the stack? It's a special region of your computer's memory that stores temporary variables created by each function (including the main() function). The stack is a "FILO" (first in, last out) data structure, that is managed and optimized by the CPU quite closely. Every time a function declares a new variable, it is "pushed" onto the stack. Then every time a function exits, all of the variables pushed onto the stack by that function, are freed (that is to say, they are deleted). Once a stack variable is freed, that region of memory becomes available for other stack variables.
  - Memory needed for execution, system handles automatically
- **Heap:** The heap is a region of your computer's memory that is not managed automatically for you, and is not as tightly managed by the CPU. It is a more free-floating region of memory (and is larger).
  - Memory under programmer's responsibility

# Stack content



# Infrastructure





# Chapter 3

## Program Binary

- The compiler compiles the source units and the linker links them together. The result is called the program binary. When declaring static and constant variables, their data is included as part of the program binary. Such variables have to be initialized in the program code. Since they are constants, their data cannot be altered during the program run, so they can be passed as part of the program binary.
- The program can be run directly from ROM or it can be loaded to read-only RAM and run from there. Variables in both memory areas are protected from improper code that tries to modify them. This is why constant static data, like literals, are stored to the program binary.
- Variables declared to the program binary exist for the lifetime of the program runtime. They do not require any memory management care from the programmer; when the program ends, the RAM allocated for the program binary is freed by the operating system. In the case of ROM, nothing is done for the program binary.



# Chapter 3

## Global Data

- When a variable is declared as static but not constant, it is considered global data. In Symbian OS, only the .exe type of executables can have global data. If global data is declared for .dll, the build process stops with an error when building against the ARM platform.
- When .exe is started as a process and it has global data, an own memory area is reserved for global data.



# Chapter 3

## Stack

- Each thread in Symbian OS has its own stack. Automatic variables are allocated to the stack. When calling a method, its parameters are also allocated to the stack. Since stack size is quite limited (especially in Symbian OS), only a small amount of data should be allocated to the stack. In function calls, pointers or references to objects should be used to decrease the need to allocate copies of the objects to the stack.
- The lifetime of the variables in the stack is specified so that when the program block ends, all automatic variables declared in that block are deleted automatically. If the variable was a class that had a destructor, it is called before removing the object from the stack.



# Chapter 3

## Heap and Memory Leaks

- Each thread in Symbian OS has a heap. This is a memory area where all dynamically allocated objects are stored. Objects are allocated to the heap with the C++ keyword `new`. This returns a pointer to an allocated object (or `NULL` if there is no more memory available in the heap for the object).
- When an object in the heap is no longer needed, the memory it has taken has to be freed with the keyword `delete`. If the object had a destructor, it is called before the memory is freed. Leaving an object to the heap without deletion is called as a memory leak. In the long run, memory leaks cause the heap to run out of free memory. If a process runs a long time and has memory leaks, it will exhaust the heap memory and the program will not be able to continue. It may also prevent other processes from running because there is no memory remaining for them. In the worst case, the entire machine will have to be rebooted (or the leaking program killed).
- Heap size is typically a lot bigger than stack size. All bigger objects should always be reserved from the heap rather than from the stack. The only thing to remember is to keep pointers to allocated objects and delete objects when no longer needed. That is required since the heap, unlike the stack, is not automatically cleaned up.
- Symbian OS provides `CleanupStack`, whose methods can be used to push pointers to the heap object in a safe place. When heap objects are no longer needed, the pointers are popped out from the cleanup stack and the pointed objects are deleted.



# Chapter 3

## Mobile Operating Systems

### Operating System

- An operating system (OS) is software that manages computer hardware and software resources and provides common services for computer programs. The operating system is an essential component of the system software in a computer system. Application programs usually require an operating system to function.

### Mobile Operating Systems

- It is an operating system for smartphones, tablets, PDAs, or other mobile devices.
- Although laptops are mobile, however, the operating systems usually used on them are not considered mobile ones as they were originally designed for bigger stationary desktop computers that historically did not have or need specific "mobile" features.
- Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use; usually including, and most of the following considered essential in modern mobile systems; a touchscreen, cellular, Bluetooth, Wi-Fi, GPS mobile navigation, camera, video camera, speech recognition, voice recorder, music player, near field communication and infrared blaster.



# Chapter 3

## Mobile Operating Systems

- Operating systems for mobile devices generally are not as fully featured as those made for desktop or laptop computers, and they are not able to run all of the same software. However, you can still do a lot of things with them, like watch movies, browse the Web, manage your calendar, and play games
- Mobile devices with mobile communications capabilities (smartphones) contain two mobile operating systems - the main user-facing software platform is supplemented by a second low-level proprietary real-time operating system which operates the radio and other hardware.
- Currently, main mobile device software comes in one of two forms: open-source, which is effectively free to download and modify; and closed-source, otherwise known as *vendor-specific*, which cannot be modified without express permission and licensing. There are benefits and drawbacks to each type of system. Because you will see both in the field, you should know each one equally. Let us go over these two systems.



# Chapter 3

## Mobile Operating Systems

### 1. Open-Source: Android

- Android is an example of open source software. It is a Linux-based operating system used mostly on smartphones and tablet computers and is developed by the Open Handset Alliance, a group directed by Google. Google releases the Android OS code as open-source, allowing developers to modify it, and freely create applications for it. Google also commissioned the Android Open-Source Project (AOSP); its mission is to maintain and further develop Android. You'll know when you are dealing with the Android open-source OS and related applications when you see the little robot caricature, usually in green.
- Android OS versions are dubbed with names such as Cupcake, Gingerbread, and the two latest: Honeycomb (version 3) and Ice Cream Sandwich (version 4). To find out the version you are currently running, start at the Home screen; this is the main screen that boots up by default. Then tap the Menu button, and then tap Settings. (Settings is used often in this chapter as a starting point, so remember how to get there!) Scroll to the bottom and tap the About Phone (or just About) option. Then tap Software Information or similar option. This displays the version of Android. Figure 17.1 shows a smartphone using Android version 2.3.3 (Gingerbread)
- Say a company wanted to create a custom version of the Android OS for a handheld computer that it was developing. According to the license, the company would be allowed to do this and customize the OS to its specific hardware. Some companies opt to use Android for this purpose, whereas others use Windows CE or Windows Mobile (for a fee), both designed for handheld computers.
- Android programming is very similar to Java, but with Android-specific APIs and frameworks

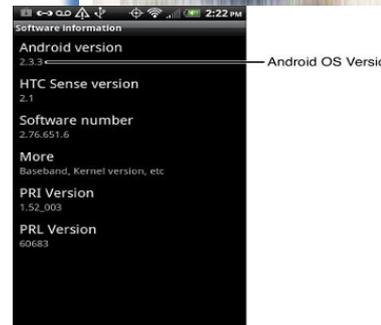


Figure 17.1. Typical smartphone using Android version 2.3.3



# Chapter 3

## Mobile Operating Systems

### 2. Closed-Source:

#### a. iOS

- Apple's iOS is an example of closed-source software. It is found on iPhones and iPads as well. It is based off Mac OS X (used on Mac desktops and laptops) and is Unix-based.
- To find out the version of iOS you are running go to the Home screen, and then tap Settings. Tap General and then tap About. You see the Version number. For example, Figure 17.2 shows an iPad2 running Version 5.0 (9A334). 9A334 is the build number; this was the public release of version 5.0.
- Unlike Android, iOS is not open-source, and is not available for download to developers. Only Apple hardware uses this operating system. This is an example of vendor-specific software. However, if a developer wants to create an application for iOS, they can download the iOS software development kit (SDK). Apple license fees are required when a developer is ready to go live with the application.
- Objective-C and Cocoa Touch are used as programming language.



Figure 17.2. iPad2 using version 5.0 of iOS

@Feda AlShahwan



# Chapter 3

## Mobile Operating Systems

### b. Symbian OS

- Symbian OS is officially the property of Nokia. It means that any other company will have to take permission from Nokia before using this operating system.
- Nokia has remained a giant in low-end mobile market, so after Java symbian was the most used in the mobile phones till a couple of years ago. Still Symbian is widely used in **low-end phones** but the demand rate has ben continuously decreasing. By upgrading Symbian mobile OS, Nokia has made it capable to run smartphones efficiently.
- **Symbian ANNA and BELLE** are the two latest updates which are currently used in Nokia's smartphones. Overall, the Symbian OS is excellently designed and is very user-friendly.
- Unfortunately, Symbian OS graph is going downwards nowadays due to the immense popularity of Android and iOS.
- Some of the phones currently running on Symbian OS are Nokia C6-01, Nokia 603, Nokia 700, Nokia 808 Pure View, Nokia E6 (ANNA) and Nokia 701 (BELLE). Symbian is a popular choice among nokia dual sim mobile phones as well



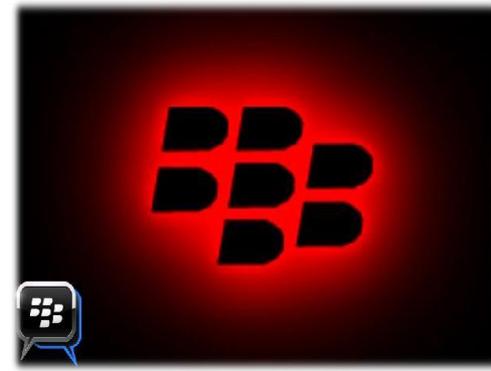


# Chapter 3

## Mobile Operating Systems

### c. BlackBerry OS

- Blackberry OS is the property of RIM (Research In Motion) and was first released in 1999. RIM has developed this operating system for its Blackberry line of smartphones. Blackberry is much different from other operating systems. The interface style as well as the smart phone design is also different having a trackball for moving on the menu and a qwerty keyboard.
- Like Apple, Blackberry OS is a close source OS and is not available for any other manufacturer. Currently the latest release of this operating system is Blackberry OS 7.1 which was introduced in May 2011 and is used in **Blackberry Bold 9930**. It is a very reliable OS and is immune to almost all the viruses.
- Some of the smartphones operating on Blackberry OS are Blackberry Bold, Blackberry Curve, Blackberry Torch and Blackberry 8520.





# Chapter 3

## Mobile Operating Systems

### d. Windows OS

- All of you will be familiar with Windows OS because it is used in computers all over the world. Windows OS has been also been used in mobile phones but normal mobile phone users find it a bit difficult to operate it but at the same time it was very popular among people who were used to it.
- This was the case until Nokia and Microsoft joined hands to work together. The latest Windows release by Microsoft is known as Windows 7 which has gained immense popularity among all kind of users. With its colourful and user friendly interface it has given Windows OS a new life and is currently in demand all over the world. Another reason behind its success is that this latest OS is used in very powerful devices made by Nokia. The computer like look has totally vanished from the windows phones with the release of Windows 7. Samsung and HTC also released some Windows based phones but they could not much place in the market.
- Nokia Lumia series is completely windows based. Some of the latest Windows Phones are Nokia Lumia 800, Nokia Lumia 900, Samsung Focus and HTC Titan 2.
- Microsoft's Windows Phone 7 uses Silverlight, that relies on C# .NET,





# Chapter 3

## Mobile Middleware

### Mobile middleware (platform):

- Computer software that provides services to mobile apps in addition to those provided by O.S.
  - Software Glue
  - Layer that lies between O.S. and application.
- Types of mobile platform (middleware):
    - 1- Symbian :
      - Nokia & Sony Ericsson
      - C++
      - Runs on ARM
      - ARM: family of instruction set based on Reduced Instruction computing set. That has the following features:
        - 1) fewer number of transistors.
        - 2) less heat.
        - 3) less power consumption

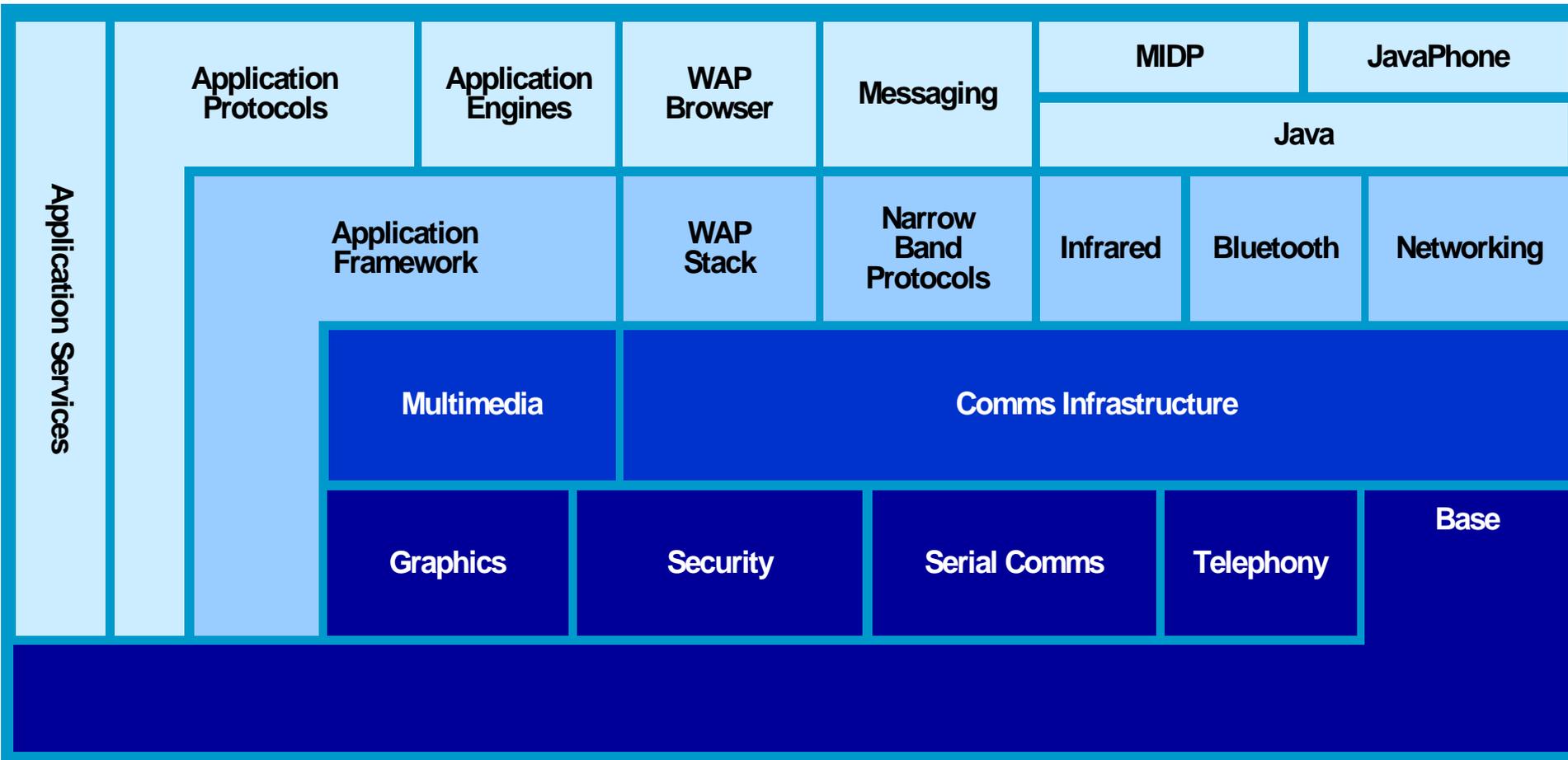


# Chapter 3

## Mobile Middleware

- Object oriented design.
  - Model for programming based on objects that have state and behavior rather than actions and functions. Which has the following feature:
- Well established architecture principles.
- Easy mobile system development.
- Event driven , micro kernel design.
  - Event driven : flow of program determined by event.
    - Such as : user actions (key press, mouse click, etc.), sensor output, message from others (program , etc.)
  - Micro kernel design: minimum amount of software that can provide the mechanisms needed to build an O.S.
    - Mechanisms : low level of address management, low level of thread management, low level of inter process communication
- Preemptive multitasking
  - Allows simultaneous execution of tasks (parallel execution at same time).
- Good power management
- Runs from ROM.
- Rich middleware ; it has many features.

# Symbian overview





# Chapter 2

## Platform

### Java:

- Interpreted byte code
- Virtual machine, Configuration (CLDC 1.0->CLDC 1.1), Profile (MIDP 1.0->2.0)
- Intended as an add-on
- Enables sandbox security, MIDP 2.0 also introducing digital signatures and rights categories

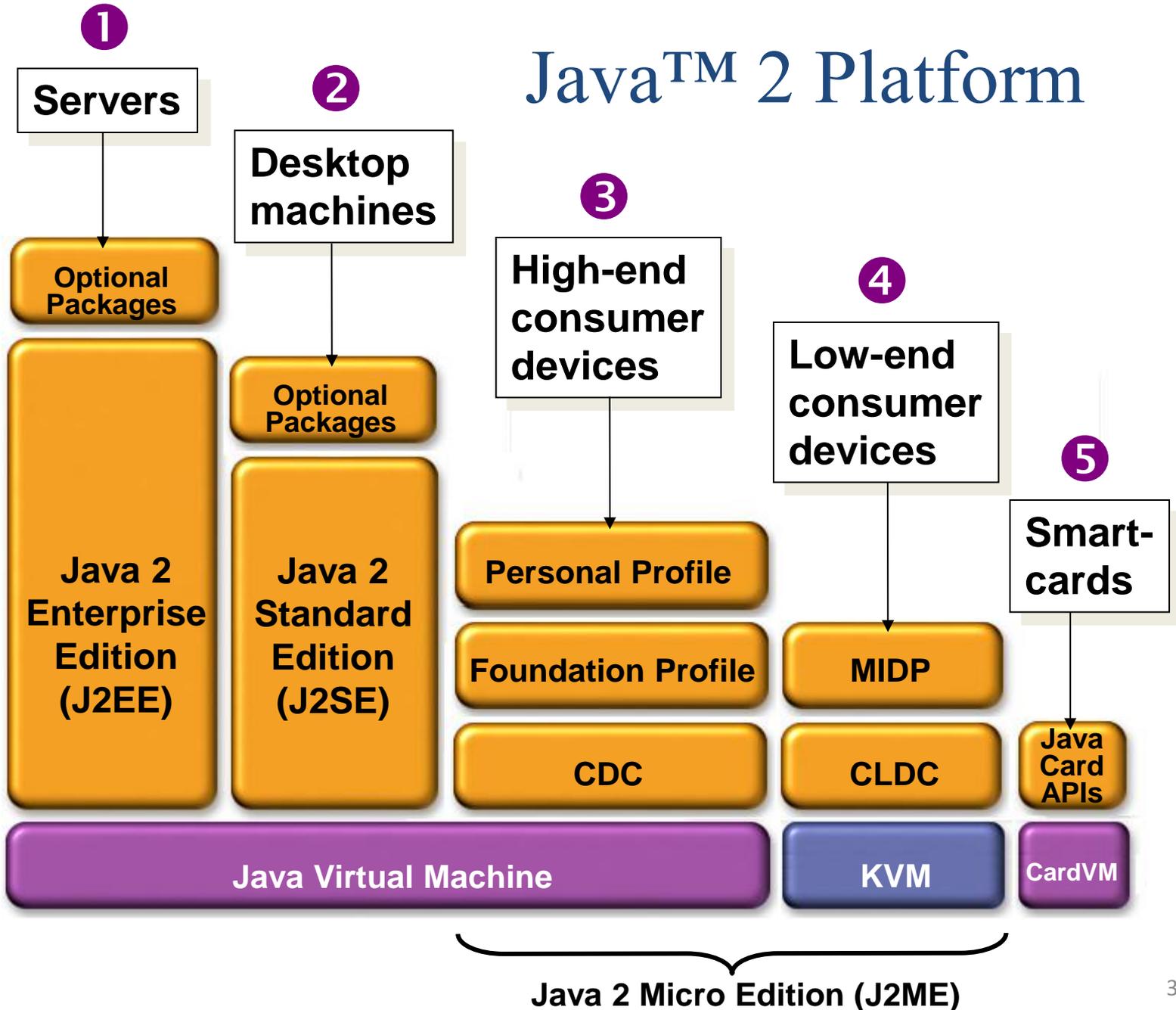


# Chapter 2

## Platform

- Small size, fits in low-end mobile phones
- An important add-on to many systems that have other main programming support
- SavaJe – an all-Java mobile platform
- Further info: [www.sun.com/java](http://www.sun.com/java)

# Java™ 2 Platform





# Chapter 2

## Introduction

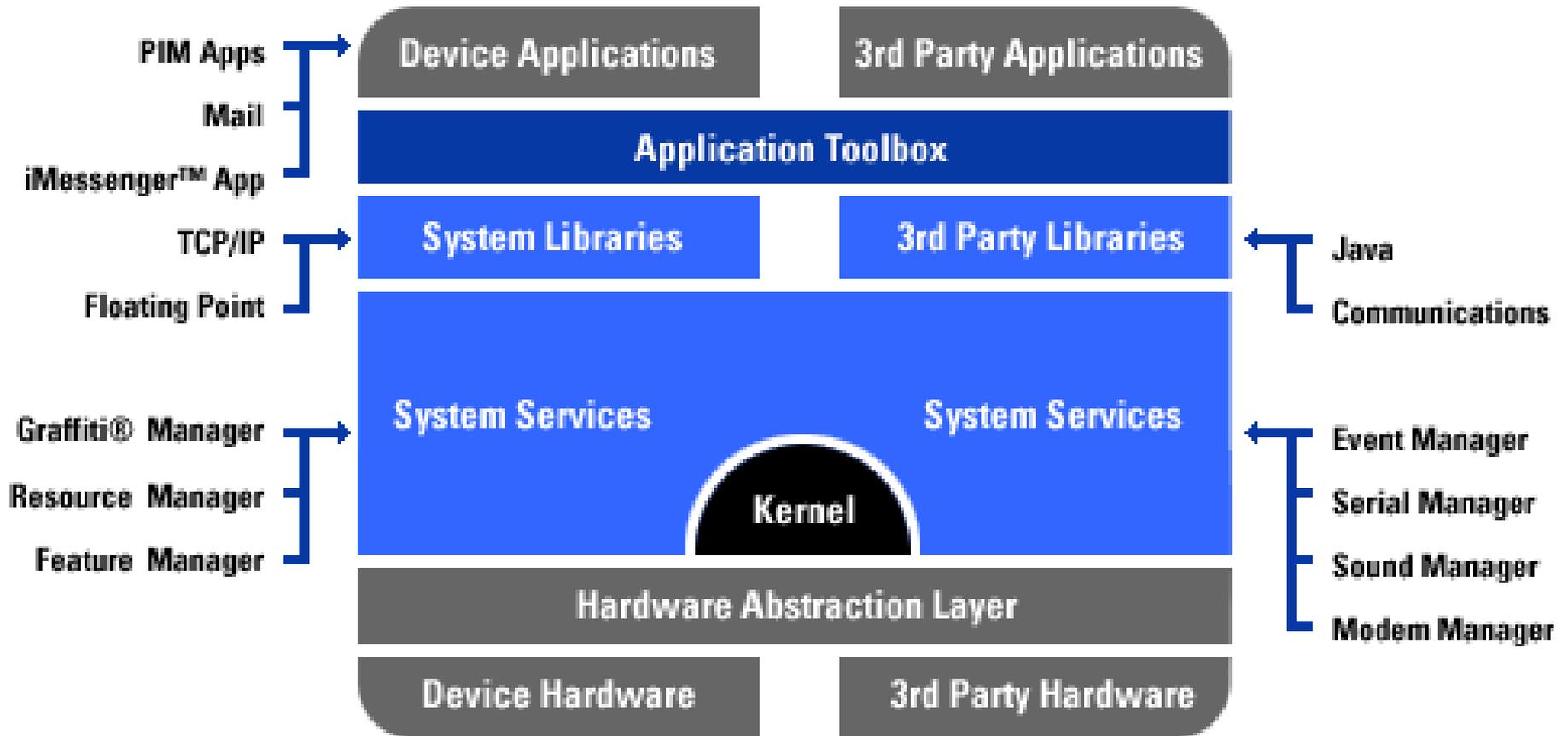
### Palm:

- Palm 4.1 based on AMX kernel
  - Programs are basically event handling loops
  - Single threaded
  - Motorola DragonBall (68000 series)
- Palm 5.0 true multitasking
  - ARM
- C, C++, Java
- Open and modular architecture
  - Introduction of 3<sup>rd</sup> party SW
  - Introduction of 3<sup>rd</sup> party HW accessed by related SW -> Easy development of add-on modules
- Further info: [www.palm.com](http://www.palm.com)

|



# Palm Reference Architecture





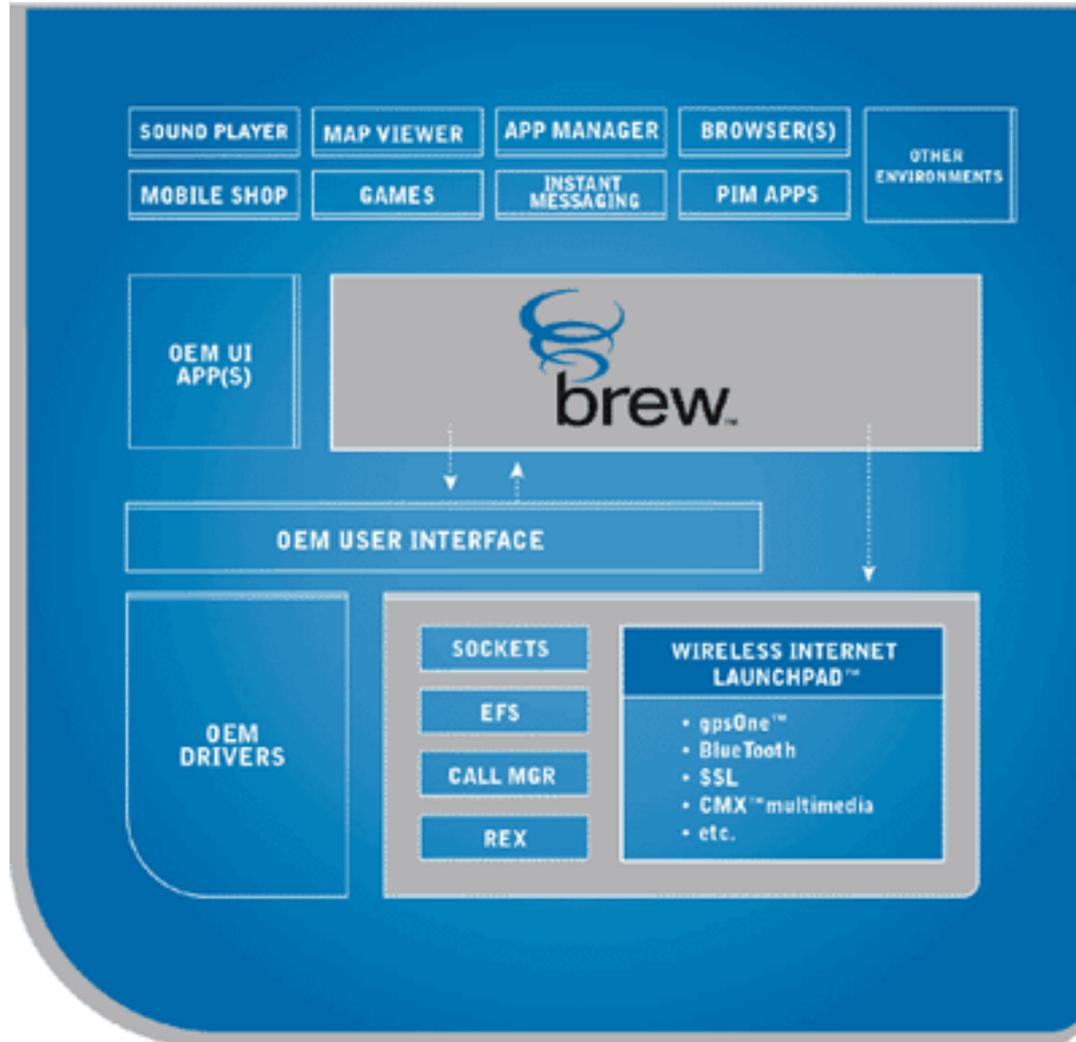
# Chapter 2

## Introduction

### Brew:

- Aims to be the mobile platform for CDMA phones
  - ... but claims to be extendable to GSM as well
  - Marginal (if any) interest in Europe
- Controlled distribution: “Before your application can be certified for distribution, you must provide certain information that will allow you to become an authenticated BREW application developer”
  - Qualcomm and its clients (e.g. Operators) may find this convenient, whereas developers may not like vendor-enforced restrictions
- C, C++, Java
- Further info: [www.qualcomm.com/brew](http://www.qualcomm.com/brew)

# Brew Reference Architecture





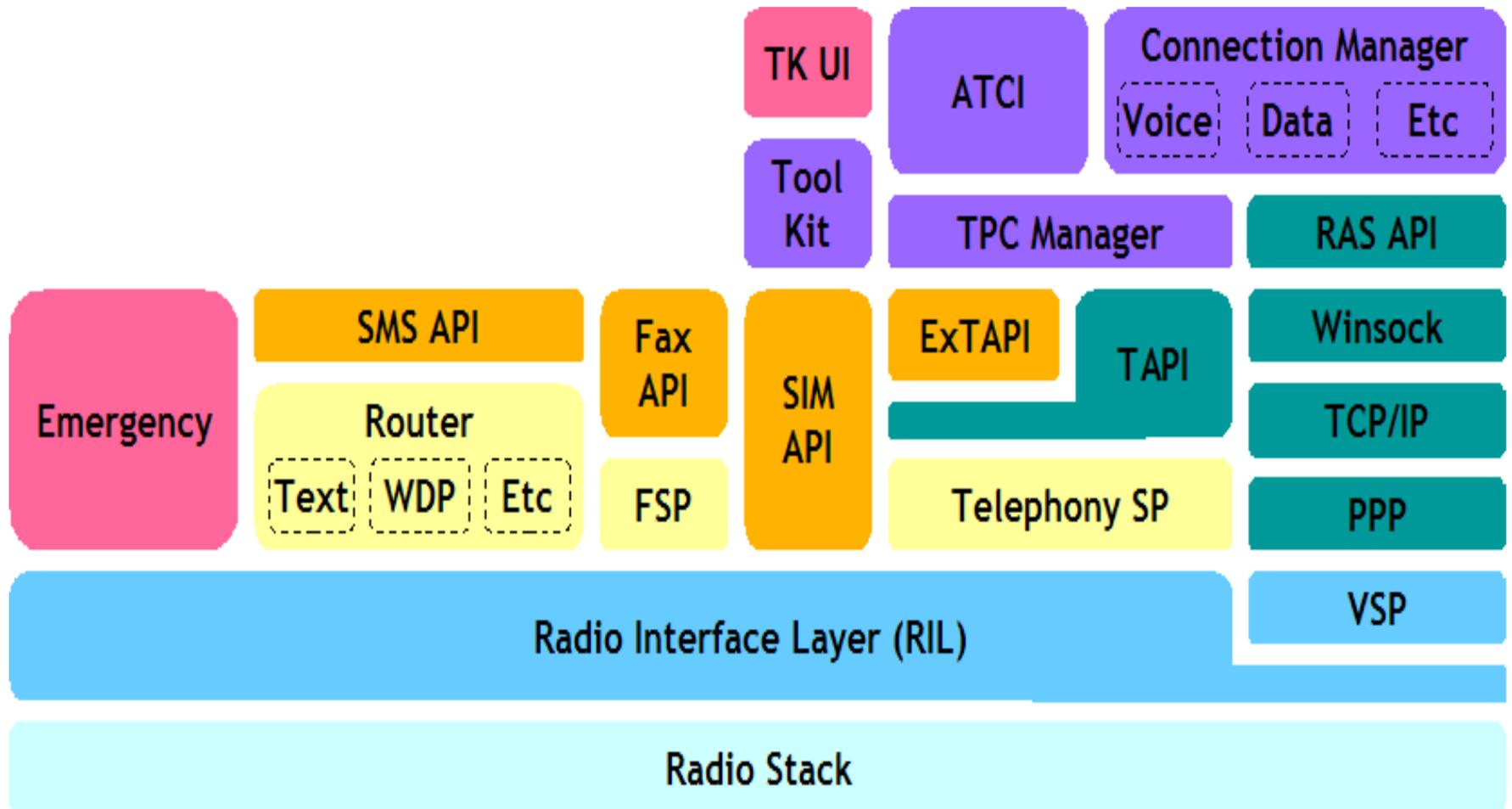
# Chapter 2

## Introduction

### Microsoft CE:

- On and off interest in mobile systems
- Market leader in PCs, not dominant in PDA/handheld computers
- Windows CE can be installed to different reference hardware
  - Pocket PC 2002
  - SmartPhone 2002
- Compatibility to PC world is an obvious advantage
- C, C++, Windows-like APIs
- Further info: [www.microsoft.com/mobile](http://www.microsoft.com/mobile)

# Microsoft Smartphone 2002





# Chapter 2

## Platform

### Linux:

- Three levels of systems
  - Minimal system environment
    - One process, deeply embedded
  - Intermediate system environment
    - Mass memory
    - I/O
    - Dynamic linking
    - File system interfaces
  - Full system environment
    - "Almost" full Linux
- Released systems from e.g. Motorola
- Monta Vista Software/Consumer Electronics Edition for core, application framework also needed
- More info: [www.embedded-linux.org](http://www.embedded-linux.org)



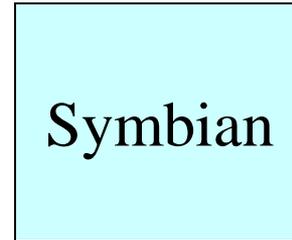
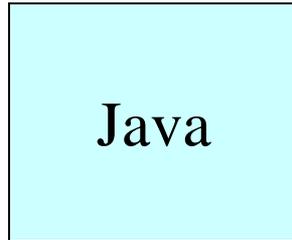
# Chapter 2

## Platform

- Do it Your Self
- Own operating system
- Well-defined interfaces to system resources
- Dynamic and reliable memory and resource management
- Tool chain availability
- Key questions:
  - Who maintains what?
  - Who invests and for what motivation?

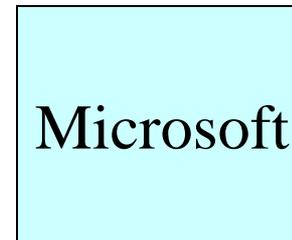
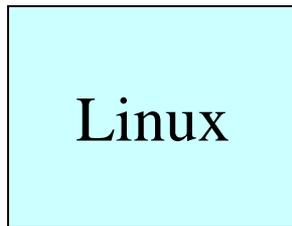
# Overview

Guiding consortia



Open development  
and standards

No 'own'  
development needed



Single company  
development effort



# Chapter 3

## Software Resources (S60

### Platform SDKs for Symbian OS, for C++)

- S60 Platform SDK for Symbian OS, for C++ allows C++ developers to quickly and efficiently run and test applications for devices that are compatible with the S60 Platform. Using a C++ for Symbian OS language integrated development environment (IDE), development with the SDK is hosted on a PC. The SDK delivers all the tools required to build C++ for Symbian OS applications. The tool's package contains the S60 device emulator, API implementations, documentation, and sample applications. The S60 Platform device emulator allows applications to be run and tested without a device.
- The Web browser of the S60 Platform device emulator allows web applications to be run and tested without a device.
- Try to Download **2nd Edition, FP 3** from <http://www.forum.nokia.com> or [http://www.forum.nokia.com/main/resources/getting\\_started/index.html](http://www.forum.nokia.com/main/resources/getting_started/index.html)
- [http://www.forum.nokia.com/devices/matrix\\_all\\_1.html](http://www.forum.nokia.com/devices/matrix_all_1.html)



# Chapter 3

## Software Resources (IDE)

- [Carbide.c++ Express v1.1](#) (now V2.0 available)
- Carbide.c++ Express is the first in a family of Eclipse-based development environments targeting Symbian OS C++ development. Carbide.c++ Express is a free tool for non-commercial mobile application development. Previously, the only free development environment available to non-commercial developers has been the command line tools. With the introduction of Carbide.c++ Express, hobbyist, students etc will be able to use a modern and powerful IDE for Symbian OS development.
- [http://www.forum.nokia.com/main/resources/tools\\_and\\_sdks/carbide\\_cpp/index.html](http://www.forum.nokia.com/main/resources/tools_and_sdks/carbide_cpp/index.html)



# Chapter 3

## Design

### Design Considerations:

- Minimal overhead (copying, context switching, memory use, etc)
- Flexibility regarding add-on software
- Robustness
- Often optimized for price -> Limited memory etc.



# Chapter 3 Summary

- Architecture of a mobile system
- Mobile Operating Systems
  - Android
  - IOS
- Software Resources



# References

- *Series 60 Platform Basics*” Provides basic information on the Series 60 platform.
- *“Introduction to Designing C++ Applications”* This manual deals also with the *Hello World Basic* application, but provides more in-depth descriptions and instructions concerning the application, its structure and implementation
- *“Designing C++ Applications”* Provides detailed instructions on designing C++ applications for Series 60. It is recommended that you read the *Introduction to Designing C++ Applications* before this manual

