CS 480: MOBILE NETWORKS

Lecture PowerPoints

WIRELESS DEVICES, ARCHITECTURES, OPERATING SYSTEMS AND APPLICATION PROGRAMMING INTERFACES

Topics Covered

- Wireless Devices
- Architecture and Configuration
- Wireless Web Implementation – Focusing on WAP
- WAP Implementations
- Web Clipping and PalmOS
- Operating Systems for Mobile Devices
- Application Programming Interface (API) for mobile devices – Development Frameworks
- Application Service Creation (APIs for Mobile Communications)
- Mobile Device Limitations

Learning Outcomes

The learning outcomes are the following

- Understand the advantage of using a layered architecture in mobile devices.
- Appreciate the capability and limitations of a range of current devices.

Wireless Devices

- Some examples of wireless devices, as discussed in lecture 1, are:
  - Sensors
  - Pagers
  - Mobile phones
  - Personal digital assistant
  - Pocket computers
  - Notebook/Laptop
- Refer to lecture 1 for details

Architecture and Configuration (1)

- In addition to the form factor for data devices, there are two broad usage models.
- In one model, the device manufacturer tries to pack as much functionality into a single device
  - One can have a PDA that has a headset for voice communications,
  - a detachable keyboard for some applications, perhaps an integrated disk drive for document storage,
  - and an MP3 player for music,
  - as well as a general purpose operating system for loading and running a variety of applications.
- A second model is where a mobile phone handles voice and data communication functions with the network

Architecture and Configuration (2)

  - With a personal area network, either 802.11 or Bluetooth, the user chooses the data or entertainment device that best suits the need at the time.
  - In simple voice communications, only a wireless headset is required.
  - On a business trip, the user may carry a laptop, and a small sales call, maybe just an organizer.
  - Each device uses the communications functions of the cell phone to access the network.
- As the wireless Internet gets more functional, the devices that access it will need to be more powerful, with more features.
- This means a more powerful operating system and application development platform.

Wireless Web Implementation – Focusing on WAP (1)

- When the WAP was first formulated in 1997,
  - the primary issue was to make the power of World Wide Web available to wireless handsets with their small screens and limited processors, memory and keyboards
  - A second goal was enabling the resulting architecture to run over a variety of wireless wide area data services available at the time
    - These included the GSM, CDMA, TDMA, and AMPS cellular voice networks, but also standalone packet networks like CDPD network.
- WAP is more than a browser;
  - it mirrors the Internet Protocol stack itself as a framework for providing a broad variety of wireless applications
- Although WAP is a completely different protocol from IP,
  - it is optimized for the wireless environment and it has increasing usage of IP standards to facilitate interoperability.

Wireless Web Implementation – Focusing on WAP (2)

- One of the most common functions of WAP
  - is to act as a gateway between the wired Internet and the variety of wireless networks and devices.
- The focus of WAP is to provide Web browsing on the small telephone screen.
  - Web browsers interpret commands of a markup language
  - HTML was the original mark-up language used to create the wired World Wide Web.
- The original Wireless Markup language (WML) used to create Web displays on cell phones is different from and incompatible with HTTP.
- WML models the cell phone screen as a card.
  - Each card contains text or graphics that appears on the cell phone screen

Wireless Web Implementation – Focusing on WAP (3)

- WAP uses many of the familiar tags available in HTML, but further restricts their use.
- Cards can be grouped together in a deck so they can be more efficiently sent down to the phone.
  - Users tend to quickly scroll through many cards in a deck
- WAP also has a scripting language, called WMLScript, similar to JavaScript language used on the wired web
  - These scripts are useful for performing small tasks on the terminal, such as validating input or displaying messages.
  - WMLScript is compiled before it is sent down to the phone. This makes the input processing much faster and responsive than if it is done on the server.

WAP Implementations (1)

- The original standards for implementing WAP were not based on the IP stack.
- The Transport, Session, and Datagram protocols were specifically conceived to handle the special characteristics of the wireless bearers.
  - These included highly variable end-to-end packet delay and dropout periods that can occur, e.g, when user entered a tunnel
- In order to integrate WAP functionality with corporate and ISP based Web servers, a WAP gateway architecture is implemented as shown in Figure 1.

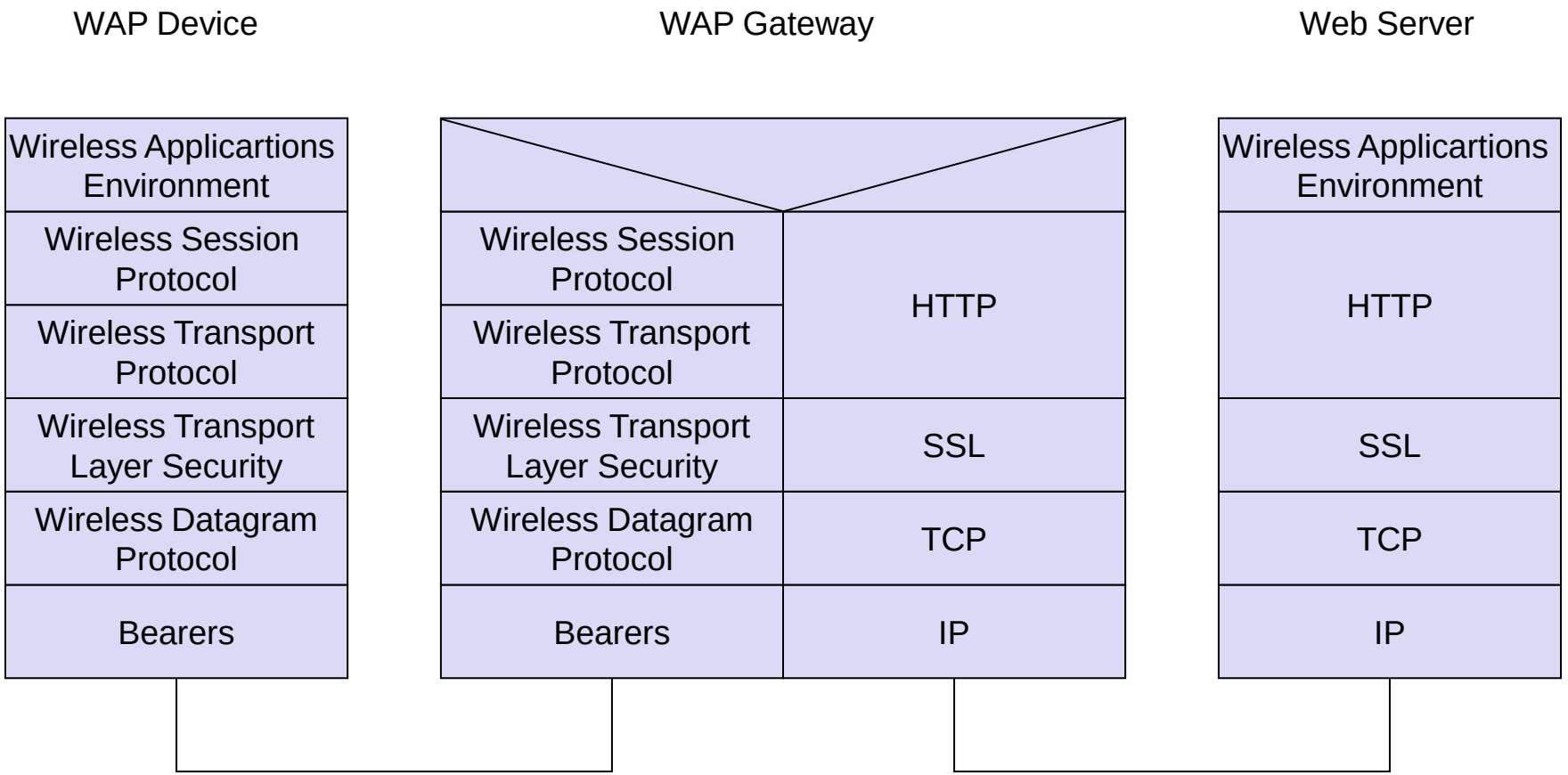

Figure 1: WAP Gateway architecture

WAP Implementations (2)

- In this environment, the user would use a mobile terminal with a WAP browser,
  - including the proprietary WAP networking protocols, on a supported wireless network.
- The user would connect to the WAP gateway, which would pull the WML decks from an IP-based Web server
- Example, consider, a company that wished to make corporate data available to their mobile sales office.
  - For a simple application that doesn't require a high throughput, the quickest way deploy the application is with Circuit Switched Data on a mobile phone with an integrated WAP browser
  - The user can dial up a WAP gateway on the corporate premises, which would handle the low level communication with the WAP phone, and then pull the WML decks from a web server in the corporate data center via the IP protocol.

WAP Implementations (3)

- Another example could be a service provider that is providing an operator-based circuit-switched data service using the IWF Wireless Data services
  - This operator would put a WAP gateway in the network and then pull WAP content from the Web servers in various content provider networks using the IP protocol
- In the wired Web, transactions like credit card validations or other private communications can use the services of the Secure Socket Layer (SSL).
  - A closed lock icon on the browsers indicate that SSL is used
- The WAP protocol has its own security protocol called WAP Transport Layer Security (WTLS)

WAP Implementations (4)

- Although WTLS is similar to SSL in many ways, it is not compatible.
- If the content provider's Web server wants to use encryption or to secure the communication, the Web browser on the mobile device would have to use WTLS encryption.
  - WTLS encryption would have to be decrypted on the WAP gateway and re-encrypted using SSL.
  - This so-called "WAP gap" creates some security concerns for some content providers and corporations.
- As WAP evolved into its second generation, WAP can use Transport layer Security, which is the follow-on standard for SSL adopted by the Web community.

WAP Implementations (5)

- It can use a newly standardised version of the TCP protocol called WP-TCP or Wireless Profiled TCP,
  - and the mark-up language is now more inline with the XHTML and XML of the wired world.
- These developments go a long way toward promoting more content for the wireless Internet,
  - and more standardised servers and development tools for service providers.

Web Clipping and PalmOS (1)

- PDAs have more screen area and a convenient pointing device,
  - but do not match the capabilities of a PC on the wired Internet
- Since the major PDA platform, the Palm Pilot, was released before the days of XHTML and XML,
  - the choice of the method to display Web-based information was between adapting WAP or developing a proprietary system
- To ensure that high quality content was available for the Palm, Palm Inc. choose a proprietary approach called Web Clipping.
- In this approach, a Palm Query Application is made up of images and Web pages that are stored locally on the PDA, and Web queries that it can send to a Web server.

Web Clipping and PalmOS (2)

- When a Web server sends back the requested content,
  - the Web Clipping Proxy server translates it into a subset of HTML commands that the applications can display.
- The PalmOS environment has been licensed to third parties, and Palm applications are available on a broad range of PDAs and phones including devices from Sony, Handspring, IBM and Symbol.
- Today PDAs can access all types of networks, and have an important role to play in the 2.5G and 3G networks.
- The larger screen area is ideal for many multimedia applications and its better input/output options
  - and processing power allow many new applications.

Web Clipping and PalmOS (3)

- The PalmOS environment has been moving more towards XHTML.
- The new PalmOS devices do not require Palm Query Application and can access XHTML over an IP stack.

Operating Systems for Mobile Devices

- Palm OS
- Microsoft Windows CE
- Symbian
- Linux
- iPhone OS (iOS)
- Blackberry OS
- Android

Palm OS (1)

- Palm OS supports a very specific hardware platform designed exclusively by Palm Computing.
- Because of the popularity of Palm Pilot devices, Palm OS has a very large community of developers and many freeware applications for end users.
- The Palm Pilot has a 16-bit memory allocation scheme, thus it can only address 64K at a time.
  - This requires software developers to address data in 64K chunks
- The basic versions of Palm OS (e.g., Version 4.0), include password-based security and Short Message Service (SMS) support.
- In addition, native Universal Serial Bus (USB) support and expanded alarm and notification capabilities are included.

Palm OS (2)

- Support for Address Book Dial Command and 65,000 colors is also provided.
- Newer versions, e.g., Palm OS release 5, support more powerful processors that allow new OS functionalities for new devices and new telephony features.

Microsoft Windows CE (1)

- Windows CE is Microsoft's embedded operating system for small-footprint and portable devices such as PDAs.
- Windows CE is a scaled-down version of Windows
  - to run on small handheld computers, PDAs, or wireless communication devices such as pagers and cellular phones.
- It is a portable and compact operating system requiring very little memory.
- Information appliances and consumer devices can use this operating system to share information with Windows-based PCs and to connect to the Internet.
- Stinger, Microsoft's platform for smart phones, is based on a scaled-down version of Windows CE 3.0.

Microsoft Windows CE (2)

- Windows CE uses the OEM Adaptation Layer (OAL) between the kernel and microprocessor to facilitate adaptation to a specific platform.
  - This architecture has made Windows CE portable to a large number of processors.
- In addition, CE support for the well known Win32 API has encouraged independent software vendor (ISV) development for CE applications.
- Various releases of Windows CE over the years have appeared with enhancements
  - to provide better embedded real-time capabilities ranging from car navigation systems to industrial controls.

Microsoft Windows CE (3)

- Enhancements have also included
  - multimedia capabilities,
  - integration with HTTP server and Internet Explorer for Web-based operations,
  - support for multiple languages, and increased component support for providing modularity for developer use.
- Security features have been added
  - by including SSL and Kerberos in CE for client-side authentication of connections from devices to enterprise networks
- Support for Bluetooth has also been added.

Symbian Platform (1)

- Symbian Platform, earlier known as EPOC, is developed and licensed by the Symbian consortium,
  - a joint venture between Ericsson, Nokia, and Motorola.
- It is an operating system platform for cell phones, smart phones, and communicators.
- The Symbian Platform is a 32-bit multitasking operating system, with communications support for remote dial-up and infrared.
  - This includes a telephony API that enables clients to initiate, control, and terminate data, fax, and voice calls using the same methods for any hardware.
- The Platform is designed for real-time capabilities for voice and data applications over wireless networks.

Symbian Platform (2)

- Like Windows CE and Palm OS,
  - it also supports voice recording and has its own synchronization code that lets smart phones exchange data with PC applications.
- Over the years, Symbian has introduced several product categories.
  - Some products are datacentric for data applications,
  - while others are more phone-centric (voice-first) design.
- These products have the capability to execute applications locally and thus perform tasks offline.
- The Symbian Platform also supports mobile phone technology with email, SMS, WAP, and HTML.
- It also supports general packet radio service (GPRS) networks, 3G wireless, and Bluetooth.

Linux

- As in the wired world, the open source movement and the Linux operating system has attractions to mobile equipment manufacturers and developers.
- The devices and protocols used on mobile devices now resemble a standard PC
  - so specialised software is no longer needed.
- Motorola built its basic and mid-range phones on Linux.
  - Linux was to become its dominant operating system with Java as the application run time.

iPhone OS (iOS)

- Strong user growth and data-hungry user base
- Application store creating a vibrant app ecosystem with great momentum
- Powerful technology enablers (e.g., multi-touch, GPS, accelerometer)
- Issues:
  - No Flash support
  - App approval process is largely a black-box to developers
  - Apps viewed as competitive to Apple are often shut down
  - Downloads highly dependent on “featured” or “top download” promotion in store
  - App store is the only authorized distribution channel
  - Apple / hardware dependent

Blackberry OS

- Developers not limited to single distribution channel
- Issues:
  - Developer momentum appears to be shifting to iPhone
  - Only 6% of market share.
  - Application distribution more difficult today vs. iPhone's app store
  - Users more email focused vs. web consuming iPhone users

Android

- Open source
  - could help accelerate pace of innovation
- Manufacturer-independent
  - could help accelerate consumer adoption
- Technology support
  - (e.g., touchscreen, GPS, accelerometer, video and still cameras)
- Issues:
  - Late to market relative to iPhone
  - At least initially, demand is expected to trail iPhone demand

API For Mobile Devices – Development Frameworks (1)

- You now know how WAP and Web browsing has been implemented on mobile phones.
- You also understand (assume) how technology had to do with the limitations of the mobile platform,
  - such as battery life, processor power, and memory
- As technology, miniaturisation and battery power have advanced, mobile platforms are now more advanced,
  - and can support other application paradigms than a Web-based server side model.
- Running more of the application logic on the client allows
  - better response time for games
  - faster access to cached information,
  - and better usage during disconnected periods.

API For Mobile Devices – Development Frameworks (2)

- Following are some of the application frameworks that are available to develop and deploy client applications.
- Similar to the issues discussed concerning the operating systems environments for cell phones and PDAs, application frameworks are also quite political,
  - and are in their early stages of development.

Java and J2ME

- Most corporate and enterprise applications are now designed to a three-tier development model
  - that separates the user interface, business logic, and data access into separate processes.
- The Java 2 Enterprise Edition (J2EE) has been the leading development environment for implementing these types of applications

J2ME Architecture (1)

- Using a framework like J2EE, a developer can focus on the actual application logic and not worry about the supporting infrastructure.
  - The framework handles all the security, transaction services, distribution and network support.
- In particular, logic supporting business functions are coded in modules called Enterprise Java Beans (EJBs)
  - that can be used and re-used to build applications in a modular fashion.
- EJBs are supported in a container that implements the supporting security and network functions.
  - As shown in Figure 2.

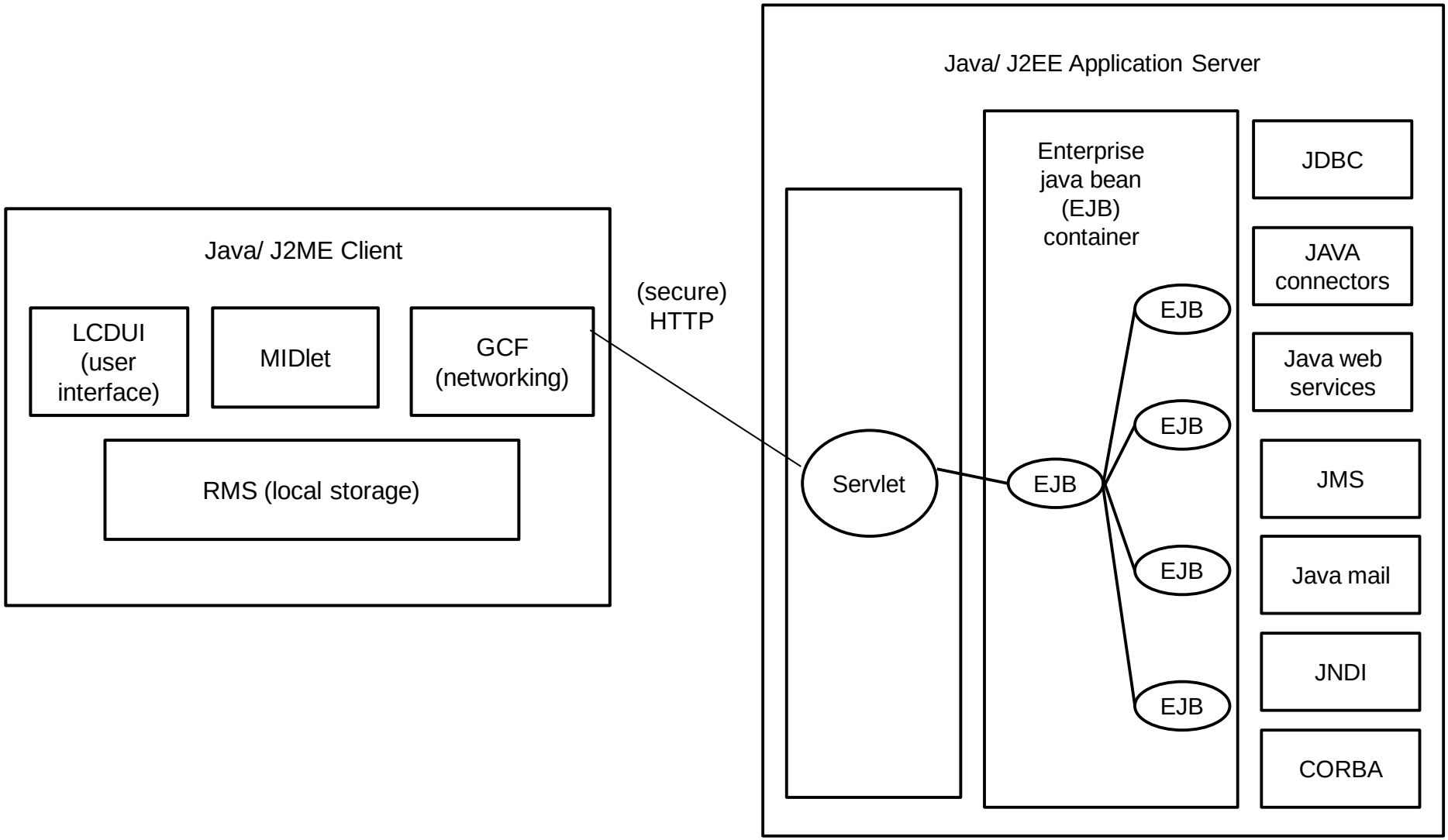

Figure 2: J2ME environment

J2ME Architecture (2)

- Sun has developed a version of the Java development environment called Java 2 Micro Edition, J2ME,
  - which is targeted at building applications for mobile devices.
- J2ME provided APIs and a runtime environment that allow developers to produce mobile applications called Mobile Information Device applets (MIDlets)
  - that correspond to the special requirements of mobile phones
- MIDlets interact with J2EE servers to provide access to important corporate applications on mobile devices,
  - or to develop new and exciting applications for mobile portals and content providers.
- Figure 2 shows an overview of the J2ME environment.

J2ME Architecture (3)

- The Connected Limited Device Configuration (CLDC) runtime environment that is built into the cell phone or PDA contains support for a user interface called the Limited Connected Device User Interface (LCDUI).
- The LCDUI creates a standardised look and feel for mobile applications and supports standard input/output methods.
- The Record Management Store (RMS) provides the capability to store persistent information
  - that allows the application to run efficiently even if the device is not connected to the network, or only connected for short periods.

J2ME Architecture (4)

- The Generic Connection Framework (GCF) handles communications between the client and server parts of the application
- GCF uses an HTTP model no matter what transport protocol is actually used, for example, the WAP Session Protocol (WSP).
  - In this way, J2ME hides the details of the cellular network or bearer service from the application.
- In the J2ME application environment, the mobile client connects to a standard J2EE server.
- In addition to supporting and encapsulating business logic through Enterprise Java Beans, the J2EE Application platform provides data and application integration via a number of mechanisms.

J2ME Architecture (5)

- The Java DataBase Connectivity module (JDBC) and Java connectors provide access to databases.
- The Java Naming and Directory Interfaces (JNDI) provide access to directory and naming services.
- The Java Messaging Services is a reliable messaging API for program-to-program communication.
- The Common Object Request Broker Architecture is an industry standard framework for building distributed applications.

J2ME Applications (1)

- Virtually, the current generation of smart phones includes the J2ME runtime environment.
- It is bundled into the Symbian Operating system, and runs on PalmOS, Windows and Linux.
- The J2ME runtime can now occupy the same position as WAP, in that its presence can be assumed for certain classes of mobile devices.
- This significant installed base drives a virtuous circle of application developers, implementing best practices, and mobile operator support.
  - As expected, Microsoft is not sitting still in this development and is heavily promoting its .NET architecture as an application framework for distributed wired applications.

J2ME Applications (2)

- Microsoft has extended the .NET environment to mobile phones with its Compact Framework CF.
- As Microsoft's .NET framework gets built out on wired PCs, this will translate over to Microsoft Pocket PC s and Windows CE devices.
- On high-end mobile platforms, both J2ME and .NET environments are available.
- On smaller more limited devices, the J2ME runtime is widely available on most phones, with WAP being the simplest and lowest common denominator.

Application Service Creation (APIs for Mobile Communications) (1)

- Various applications can be developed for the cellular networking environment.
  - This follows a number of mechanisms that deal with the different user interfaces on mobile phones and PDAs
  - as well as those that are used to structure client/server distributed computing in the mobile environment,
- One key aspect of developing applications in a cellular data network is interfacing with telecommunication services and applications that may be running in a mobile operator's network.
  - These may be billing or pre-paid servers, call control and routing, and user databases like the Home Location Register and Visitor Location Register.

Application Service Creation (APIs for Mobile Communications) (2)

- It is not easy for applications to be interfaced.
  - It requires expensive hardware and protocol stacks,
  - and operators are doubtful about putting untested applications on this mission critical network.
- The Parlay Group was an industry consortium formed in 1998 by British Telecom, Microsoft, Nortel Networks, Siemens, and Ulticom (a SS7 stack vendor)
  - to develop an open Application Programming Interface (API) that allow software developers to access these telecommunications services in a simple standardised matter.
- The 3GPP committee, as part of the UMTS architecture, formalised this API as part of its Open Service Architecture (OSA).

Application Service Creation (APIs for Mobile Communications) (3)

The Parlay/OSA API provides

- Framework APIs, which provide authentication, authorisation, logging, service discovery, service registration, and service subscription functions, etc.
- Service APIs, which are grouped into Service Capability Functions depending on the resources and protocols needed to perform the function. Examples include:
  - Call control (hang-up, retry, dial etc.)
  - User location
  - User status (on-line, powered-off, talking etc.)
  - Billing events
  - User interaction (playing messages, receiving input digits)
  - Messaging functions
  - Query terminal capabilities

Mobile Device Limitations (1)

- With the arrival of 2.5G systems that use IP-based packet switching technology and provide reasonable speed access to services, the wireless Internet is only just beginning.
- As this infrastructure is rolling out, the applications and the devices needed to harness this power are just coming into focus.
- Device manufacturers have produced a tremendous diversity of access devices.
- Wireless devices of the future will be more powerful, less heavy and comprise new interfaces to the user and to new networks.

Mobile Device Limitations (2)

- However, one big problem, which has not yet been solved, is the energy supply.
  - The more features that are built into a device, the more power it needs.
  - The higher the performance of a device, the faster it drains the batteries.
  - Furthermore, wireless data transmission consumes a lot of energy.
- Although the area of mobile computing and mobile communications is developing rapidly,
  - the devices typically used today still exhibit some major drawbacks compared to desktop systems in addition to the energy problem.

Mobile Device Limitations (3)

- Interfaces have to be small enough to make the device portable, so smaller keyboards are used.
  - This makes typing difficult due to their limited key size.
- Small displays are often useless for graphical display.
  - Higher resolution does not help, as the limiting factor is the resolution capacity of the human eye.
- These devices have to use new ways of interacting with a user, such as e.g., touch sensitive displays and voice recognition.