Course Overview

- Introduction
- Data in Wireless Cellular Systems
- Data in Wireless Local Area Networks
- Internet Protocols
- TCP over Wireless Link
- Ad-Hoc Networks, Sensor Networks
- Services and Service Discovery
- System Support for Mobile Applications

Challenges in Mobile Computing

- wireless communication is substantially different from wired communication:
  - frequent spurious disconnections (handoff, shadowed areas)
  - lower bandwidth (GSM, CDPD: less than 20 kbps)
  - high bandwidth variability (roaming outdoors vs. plugged into docking station)
  - heterogeneous networks (outdoors: cellular, indoors: infrared LAN)
  - security risks (everyone can listen in on wireless channel)
- mobility:
  - address migration (e.g., location management)
  - location dependent information (where is closest printer?)
  - migrating locality (optimal connection changes over time)

Challenges in Mobile Computing

- portability (of computing device):
  - limited power supply (battery)
  - data security (someone steals your laptop)
  - small user interface (look at screen size of a PDA)
  - limited storage capacity (disks, for example, are a “power liability”)
- solutions (or suggestions) exist to address all these issues, however, the solutions are often conflicting:
  - do not keep data on portable device to reduce security problem
  - do not send data if not necessary to save valuable energy

Solutions and Research Directions

- solutions to overcome hardware disparities:
  - sophisticated power management (sleep modes, disk mgmt, clock speed, prefer receiving data over sending: “ether as cache”)
  - delegate tasks to stationary computers (application partitioning, stationary computer has disk space, high bandwidth connection, powerful CPU)
  - trade increased data processing for reduced network bandwidth requirements (on-line compression, difference-based updates, filtering, less restrictive consistency semantics)
  - reduce network latency (data prefetching, batch multiple messages)

Solutions and Research Directions

- solutions addressing mobility:
  - flexible directory services to enable mobile to find location-dependent services, often based on "shopping" or negotiation:
    - scaleable, fault-tolerant architecture (exploit communication locality, replicate directory data, weakened consistency semantics: allow directory info to go stale as long as we can improve on suboptimal routes during connection)
    - brokerage service (e.g.: trading in ODP)
    - accounting service: mediate cost accounting between mobile client and servers
    - rebindable service interfaces (switch service providers "on-line")
  - ubiquitous security and authentication (IPv6 will go a long way towards providing this goal)

Solutions and Research Directions

- there seems to be general agreement that some sort of application partitioning is a good thing:
  - reduce bandwidth requirements
  - overcome the physical limitations of the mobile computer
- one open question: how do you partition and when?
  - static partitioning is easier (and we will see examples for WWW later)
  - however, computation environment can change radically during execution:
    - MH moves to area with better networking infrastructure (maybe gets back into his docking station)
    - PCMCIA cards all of a sudden add more memory
    - radio signal is dropped completely temporarily
  - implies growing interest in dynamic application partitioning

File Systems: Motivation

- Goal
  - efficient and transparent access to shared files within a mobile environment while maintaining data consistency
- Problems
  - limited resources of mobile computers (memory, CPU, ...)
  - low bandwidth, variable bandwidth, temporary disconnection
  - high heterogeneity of hardware and software components (no standard PC architecture)
  - wireless network resources and mobile computer are not very reliable
  - standard file systems (e.g., NFS, network file system) are very inefficient, almost unusable
- Solutions
  - replication of data (copying, cloning, caching)
  - data collection in advance (hoarding, pre-fetching)

File Systems: Consistency Problems

- THE big problem of distributed, loosely coupled systems
  - are all views on data the same?
  - how and when should changes be propagated to what users?
- Weak consistency
  - many algorithms offering strong consistency (e.g., via atomic updates) cannot be used in mobile environments
  - invalidation of data located in caches through a server is very problematic if the mobile computer is currently not connected to the network
  - occasional inconsistencies have to be tolerated, but conflict resolution strategies must be applied afterwards to reach consistency again
- Conflict detection
  - content independent: version numbering, time-stamps
  - content dependent: dependency graphs

File Systems: Limited Connectivity

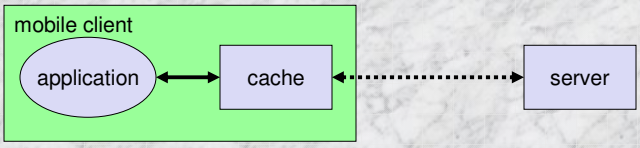
- Symmetry
  - Client/Server or Peer-to-Peer relations
  - support in the fixed network and/or mobile computers
  - one file system or several file systems
  - one namespace for files or several namespaces
- Transparency
  - hide the mobility support, applications on mobile computers should not notice the mobility
  - user should not notice additional mechanisms needed
- Consistency model
  - optimistic or pessimistic
- Caching and Pre-fetching
  - single files, directories, subtrees, partitions, ...
  - permanent or only at certain points in time

File Systems: Limited Connectivity

- Data management
  - management of buffered data and copies of data
  - request for updates, validity of data
  - detection of changes in data
- Conflict solving
  - application specific or general
  - errors
- Several experimental systems exist
  - Coda (Carnegie Mellon University), Little Work (University of Michigan), Ficus (UCLA) etc.
- Many systems use ideas from distributed file systems such as, e.g., AFS (Andrew File System)

File Systems: Coda

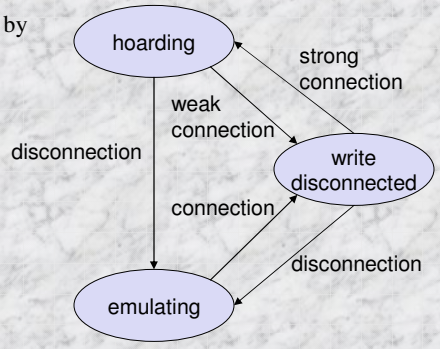
- Application transparent extensions of client and server
  - changes in the cache manager of a client
  - applications use cache replicates of files
  - extensive, transparent collection of data in advance for possible future use („Hoarding“)
- Consistency
  - system keeps a record of changes in files and compares files after reconnection
  - if different users have changed the same file a manual reintegration of the file into the system is necessary
  - optimistic approach, coarse grained (file size)



File Systems: Coda

- Hoarding
  - user can pre-determine a file list with priorities
  - contents of the cache determined by the list and LRU strategy (Last Recently Used)
  - explicit pre-fetching possible
  - periodic updating
- Comparison of files
  - asynchronous, background
  - system weighs speed of updating against minimization of network traffic
- Cache misses
  - modeling of user patience: how long can a user wait for data without an error message?
  - function of file size and bandwidth

- States of a client



File Systems: Other Examples

- Mazer/Tardo
  - file synchronization layer between application and local file system
  - caching of complete subdirectories from the server
  - "Redirector" responses to requests locally if necessary, via the network if possible
  - periodic consistency checks with bi-directional updating
- Ficus
  - not a client/server approach
  - optimistic approach based on replicates, detection of write conflicts, conflict resolution
  - use of „gossip" protocols: a mobile computer does not necessarily need to have direct connection to a server, with the help of other mobile computers updates can be propagated through the network
- MIo-NFS (Mobile Integration of NFS)
  - NFS extension, pessimistic approach, only token holder can write
  - connected/loosely connected/disconnected

World Wide Web and Mobility

- Protocol (HTTP, Hypertext Transfer Protocol) and language (HTML, Hypertext Markup Language) of the Web have not been designed for mobile applications and mobile devices, thus creating many problems!
- Typical transfer sizes
  - HTTP request: 100-350 byte
  - responses avg. <10 kbyte, header 160 byte, GIF 4.1kByte, JPEG 12.8 kbyte, HTML 5.6 kbyte
  - but also many large files that cannot be ignored
- The Web is no file system
  - Web pages are not simple files to download
  - static and dynamic content, interaction with servers via forms, content transformation, push technologies etc.
  - many hyperlinks, automatic loading and reloading, redirecting
  - a single click might have big consequences!

HTTP 1.0 and Mobility

- Characteristics
  - stateless, client/server, request/response
  - needs a connection oriented protocol (TCP), one connection per request (some enhancements in HTTP 1.1)
  - primitive caching and security
- Problems
  - designed for large bandwidth (compared to wireless access) and low delay
  - big and redundant protocol headers (readable for humans, stateless, therefore big headers in ASCII)
  - uncompressed content transfer
  - using TCP
    - huge overhead per request (3-way-handshake) compared with the content, e.g., of a GET request
    - slow-start problematic
  - DNS lookup by client causes additional traffic

HTTP 1.0 and Mobility

- Caching
  - quite often disabled by information providers to be able to create user profiles, usage statistics etc.
  - dynamic objects cannot be cached
    - numerous counters, time, date, personalization, ...
  - mobility quite often inhibits caches
  - security problems
    - how to use SSL/TLS together with proxies?
  - today: many user customized pages, dynamically generated on request via CGI, ASP, ...
- POSTing (i.e., sending to a server)
  - can typically not be buffered, very problematic if currently disconnected
- Many unsolved problems!

HTML and Mobile Devices

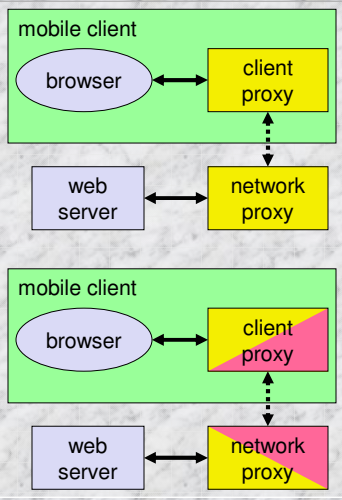
- HTML
  - designed for computers with “high” performance, color high-resolution display, mouse, hard disk
  - typically, web pages optimized for design, not for communication
- Mobile devices
  - often only small, low-resolution displays, very limited input interfaces (small touch-pads, soft-keyboards)
- Additional “features”
  - animated GIF, Java AWT, Frames, ActiveX Controls, Shockwave, movie clips, audio, ...
  - many web pages assume true color, multimedia support, high-resolution and many plug-ins
- Web pages ignore the heterogeneity of end-systems!
  - e.g., without additional mechanisms, large high-resolution pictures would be transferred to a mobile phone with a low-resolution display causing high costs

WWW for Mobile Devices

- Application gateways, enhanced servers
  - simple clients, pre-calculations in the fixed network
  - compression, filtering, content extraction
  - automatic adaptation to network characteristics
- Examples
  - picture scaling, color reduction, transformation of the document format (e.g., PS to TXT)
  - detail studies, clipping, zoom
  - headline extraction, automatic abstract generation
  - HDML (handheld device markup language): simple language similar to HTML requiring a special browser
  - HDTP (handheld device transport protocol): transport protocol for HDML, developed by Unwired Planet
- Problems
  - proprietary approaches, require special enhancements for browsers
  - heterogeneous devices make approaches more complicated

System Support for Mobile WWW

- Client and network proxy
  - combination of benefits plus simplified protocols
  - e.g., MobiScape, WebExpress
- Special network subsystem
  - adaptive content transformation for bad connections, pre-fetching, caching
  - e.g., Mowgli
- Additional many proprietary server extensions possible
  - "channels", content negotiation, ...



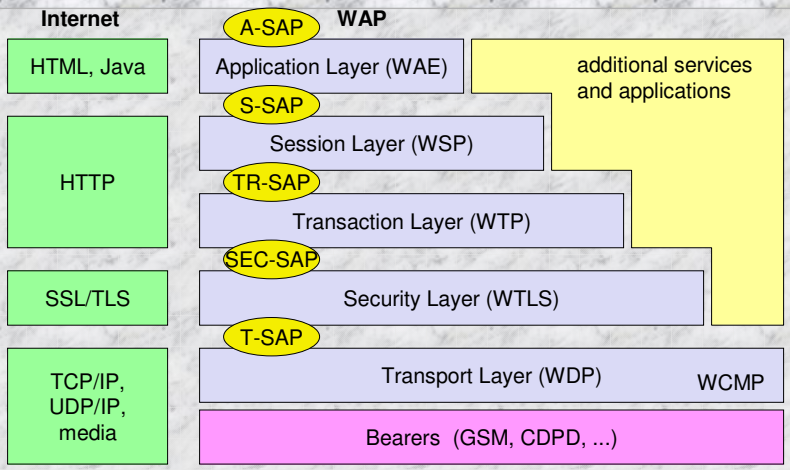
WAP - Wireless Application Protocol

- Goals
  - deliver Internet content and enhanced services to mobile devices and users (mobile phones, PDAs)
  - independence from wireless network standards
  - open for everyone to participate, protocol specifications will be proposed to standardization bodies
  - applications should scale well beyond current transport media and device types and should also be applicable to future developments
- Platforms
  - e.g., GSM (900, 1800, 1900), CDMA IS-95, TDMA IS-136, 3rd generation systems (IMT-2000, UMTS, W-CDMA, cdma2000 1x EV-DO, …)
- Forum
  - was: WAP Forum, co-founded by Ericsson, Motorola, Nokia, Unwired Planet, further information www.wapforum.org
  - now: **Open Mobile Alliance** www.openmobilealliance.org (Open Mobile Architecture + WAP Forum + SyncML + …)

WAP - Scope of Standardization

- Browser
  - "micro browser", similar to existing, well-known browsers in the Internet
- Script language
  - similar to Java script, adapted to the mobile environment
- WTA/WTAI
  - Wireless Telephony Application (Interface): access to all telephone functions
- Content formats
  - e.g., business cards (vCard), calendar events (vCalender)
- Protocol layers
  - transport layer, security layer, session layer etc.

WAP 1.x Reference Model and Protocols



WAE comprises WML (Wireless Markup Language), WML Script, WTAI etc.

WDP: Wireless Datagram Protocol

- Protocol of the transport layer within the WAP architecture
  - uses directly transports mechanisms of different network technologies
  - offers a common interface for higher layer protocols
  - allows for transparent communication using different transport technologies (GSM [SMS, CSD, USSD, GPRS, ...], IS-136, TETRA, DECT, PHS, IS-95, ...)
- Goals of WDP
  - create a worldwide interoperable transport system with the help of WDP adapted to the different underlying technologies
  - transmission services such as SMS, GPRS in GSM might change, new services can replace the old ones
- Additionally, WCMP (wireless Control Message Protocol) is used for control/error report (similar to ICMP in the TCP/IP protocol suite)

WTLS: Wireless Transport Layer Security

- Goals
  - data integrity
    - prevention of changes in data
  - privacy
    - prevention of tapping
  - authentication
    - creation of authenticated relations between a mobile device and a server
  - protection against denial-of-service attacks
    - protection against repetition of data and unverified data
- WTLS
  - is based on the TLS (Transport Layer Security) protocol (former SSL, Secure Sockets Layer)
  - optimized for low-bandwidth communication channels

WTP: Wireless Transaction Protocol

- Goals
  - different transaction services, offloads applications
    - application can select reliability, efficiency
  - support of different communication scenarios
    - *class 0:* unreliable message transfer
    - *class 1:* reliable message transfer without result message
    - *class 2:* reliable message transfer with exactly one reliable result message
  - supports peer-to-peer, client/server and multicast applications
  - low memory requirements, suited to simple devices (< 10kbyte)
  - efficient for wireless transmission
    - segmentation/reassembly
    - selective retransmission
    - header compression
    - optimized connection setup (setup with data transfer)

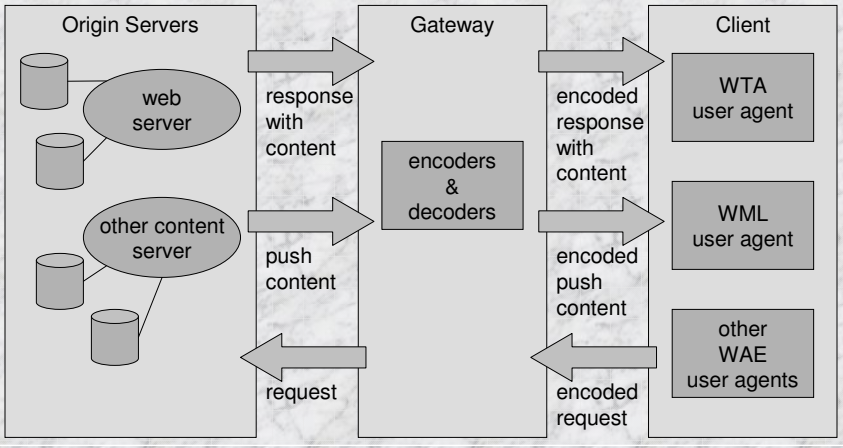
WSP: Wireless Session Protocol

- Goals
  - HTTP 1.1 functionality
    - Request/reply, content type negotiation, ...
  - support of client/server, transactions, push technology
  - key management, authentication, Internet security services
  - session management (interruption, resume,...)
- Open topics
  - QoS support
  - Group communication
  - Isochronous media objects
  - management

WAE: Wireless Application Environment

- Goals
  - network independent application environment for low-bandwidth, wireless devices
  - integrated Internet/WWW programming model with high interoperability
- Requirements
  - device and network independent, international support
  - manufacturers can determine look-and-feel, user interface
  - considerations of slow links, limited memory, low computing power, small display, simple user interface (compared to desktop computers)
- Components
  - architecture: application model, browser, gateway, server
  - WML: XML-Syntax, based on card stacks, variables, ...
  - WMLScript: procedural, loops, conditions, ... (similar to JavaScript)
  - WTA: telephone services, such as call control, text messages, phone book, ... (accessible from WML/WMLScript)
  - content formats: vCard, vCalendar, Wireless Bitmap, WML, ...

WAE Logical Model



Wireless Markup Language (WML)

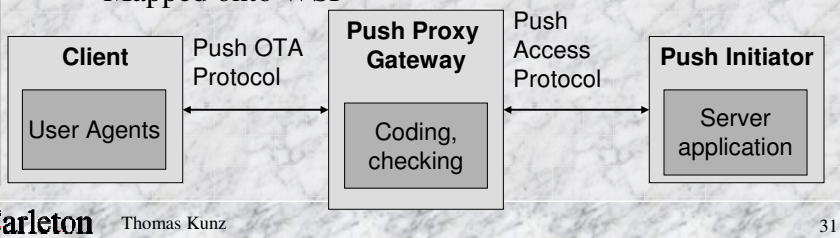
- WML follows deck and card metaphor
  - WML document consists of many cards, cards are grouped to decks
  - a deck is similar to an HTML page, unit of content transmission
  - WML describes only intent of interaction in an abstract manner
  - presentation depends on device capabilities
- Features
  - text and images
  - user interaction
  - navigation
  - context management

WMLScript

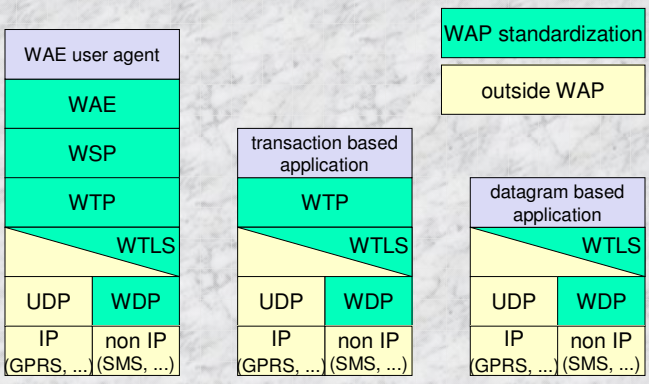
- Complement to WML
- Provides general scripting capabilities
- Features
  - validity check of user input
    - check input before sent to server
  - access to device facilities
    - hardware and software (phone call, address book etc.)
  - local user interaction
    - interaction without round-trip delay
  - extensions to the device software
    - configure device, download new functionality after deployment

WAP Push Architecture with Proxy Gateway

- Push Access Protocol
  - Content transmission between server and PPG
  - First version uses HTTP
- Push OTA (Over The Air) Protocol
  - Simple, optimized
  - Mapped onto WSP



Examples for WAP Protocol Stacks (WAP 1.x)



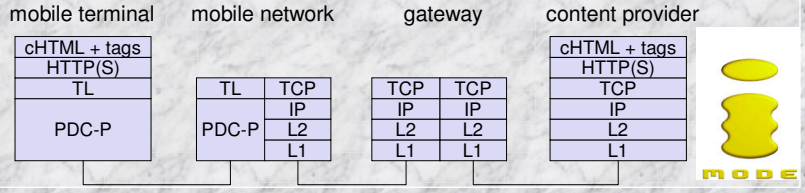
1. typical WAP application with complete protocol stack

2.

3. pure data application with/without additional security

i-mode: First of all a Business Model!

- Access to Internet services in Japan provided by NTT DoCoMo
  - Services
    - Email, short messages, web, picture exchange, horoscope, ...
  - Big success – more than 30 million users
    - Many use i-mode as PC replacement
    - For many this is the first Internet contact
    - Very simple to use, convenient
  - Technology
    - 9.6 kbit/s (enhancements with 28.8 kbit/s), packet oriented (PDC-P)
    - Compact HTML plus proprietary tags, special transport layer (Stop/go, ARQ, push, connection oriented)



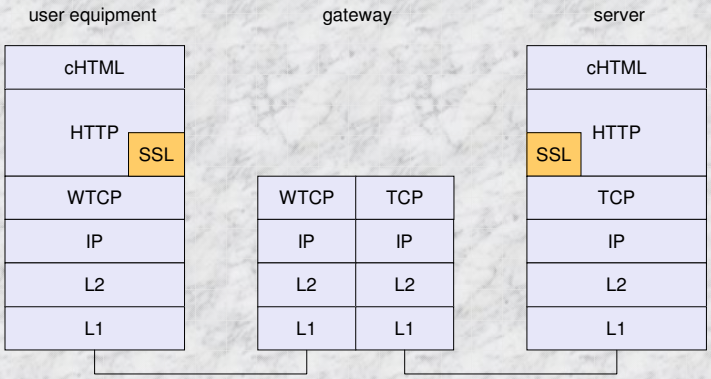
Email Example: i-mode Push with SMS

Operator sends an SMS containing a push message if a new email has arrived. If the user wants to read the email, an HTTP get follows with the email as response.

Popular misconception:
WAP was a failure, i-mode is different and a success – wrong from a technology point of view, right from a business point of view…

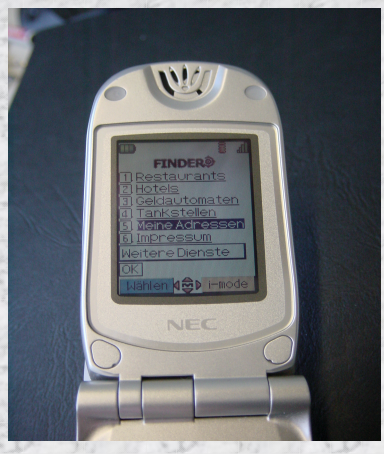
i-mode as a **business model**:
- content providers get >80% of the revenue.
- independent of technology (GSM/GPRS in Europe, PDC-P in Japan – but also UMTS!)

i-mode Protocol Stack based on WAP 2.0



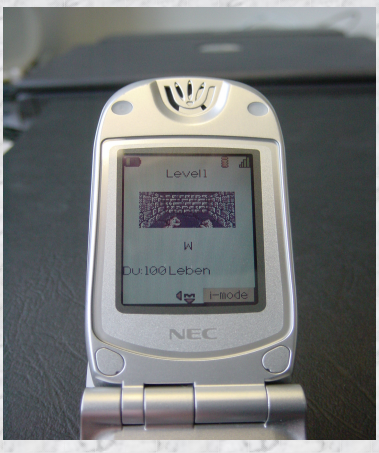
i-mode can use WAP 2.0/Internet protocols
(example: i-mode in Germany over GSM/GPRS)

i-mode Examples



i-mode Examples (cont)

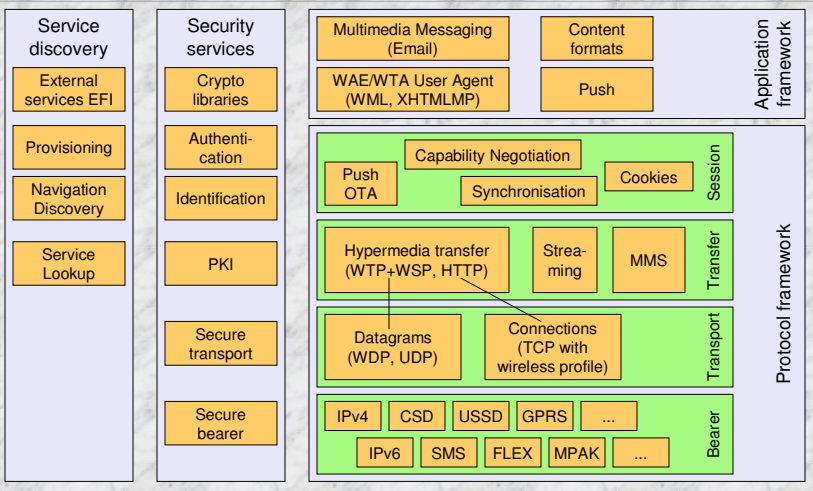
i-mode Examples (cont)



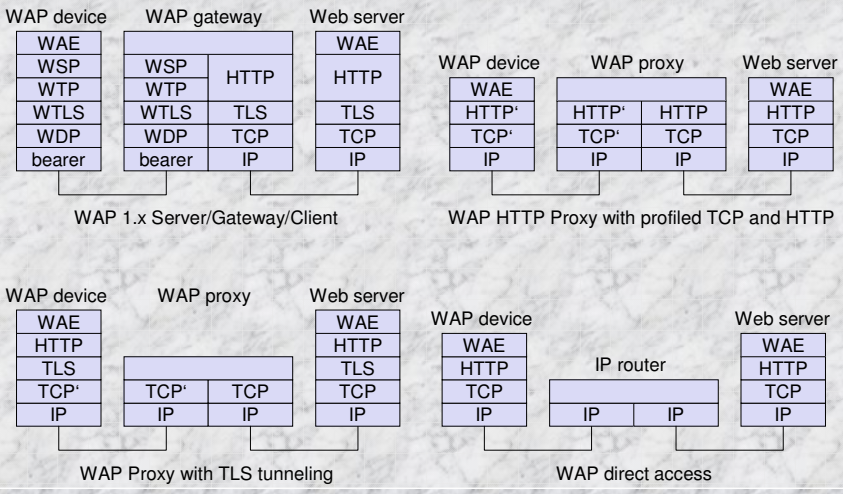
WAP 2.0 (July 2001)

- New for developers
  - XHTML
  - TCP with „Wireless Profile“
  - HTTP
- New applications
  - Color graphics
  - Animation
  - Large file download
  - Location based services
  - Synchronization with PIMs
  - Pop-up/context sensitive menus
- Goal: integration of WWW, Internet, WAP, i-mode

WAP 2.0 Architecture



WAP 2.0 Example Protocol Stacks



Java 2 Platform Micro Edition

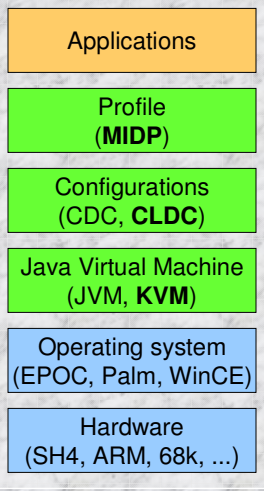
- „Java-Boom expected“ (?)
  - Desktop: over 90% standard PC architecture, Intel x86 compatible, typically MS Windows systems
  - Do really many people care about platform independent applications?
- BUT: Heterogeneous, “small“ devices
  - Internet appliances, cellular phones, embedded control, car radios, ...
  - Technical necessities (temperature range, form factor, power consumption, …) and economic reasons result in different hardware
- J2ME
  - Provides a uniform platform
  - Restricted functionality compared to standard java platform (JVM)

Applications of J2ME

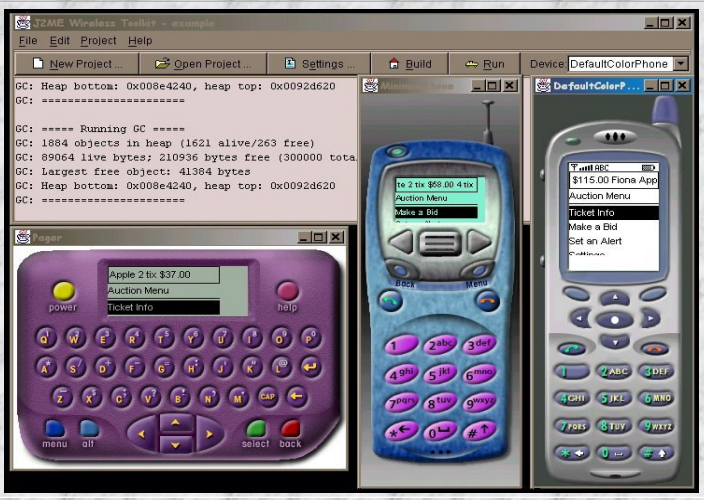
- Example cellular phones
  - NTT DoCoMo introduced iαppli
  - Applications on PDA, mobile phone, ...
  - Game download, multimedia applications, encryption, system updates
  - Load additional functionality with a push on a button (and pay for it)!
- Embedded control
  - Household devices, vehicles, surveillance systems, device control
  - System update is an important factor

Characteristics and Architecture

- Java Virtual Machine
  - Virtual Hardware (Processor)
  - KVM (K Virtual Machine)
    - Min. 128 kByte, typ. 256 kByte
    - Optimized for low performance devices
    - Might be a co-processor
- Configurations
  - Subset of standard Java libraries depending technical hardware parameters (memory, CPU)
  - CLDC (Connected Limited Device Configuration)
    - Basic libraries, input/output, security – describes Java support for mobile devices
- Profiles
  - Interoperability of heterogeneous devices belonging to the same category
  - MIDP (Mobile Information Device Profile)
    - Defines interfaces for GUIs, HTTP, application support, …



Hardware Independent Development



Summary J2ME

- Idea is more than WAP 1.x or i-mode
  - Full applications on mobile phones, not only a browser
  - Includes system updates, end-to-end encryption
- Platform independent via virtualization
  - As long as certain common interfaces are used
  - Not valid for hardware specific functions
- Limited functionality compared to JVM
  - Thus, maybe an intermediate solution only – until embedded systems, mobile phones are as powerful as today's desktop systems