18MCA33C Mobile Computing
UNIT III
Mobile Network Infrastructure

FACULTY

Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

Government Arts College (Autonomous),

Coimbatore-641018.

GSM

- The GSM network can be broadly divided into:
- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS)
- Given below is a simple pictorial view of the GSM architecture.
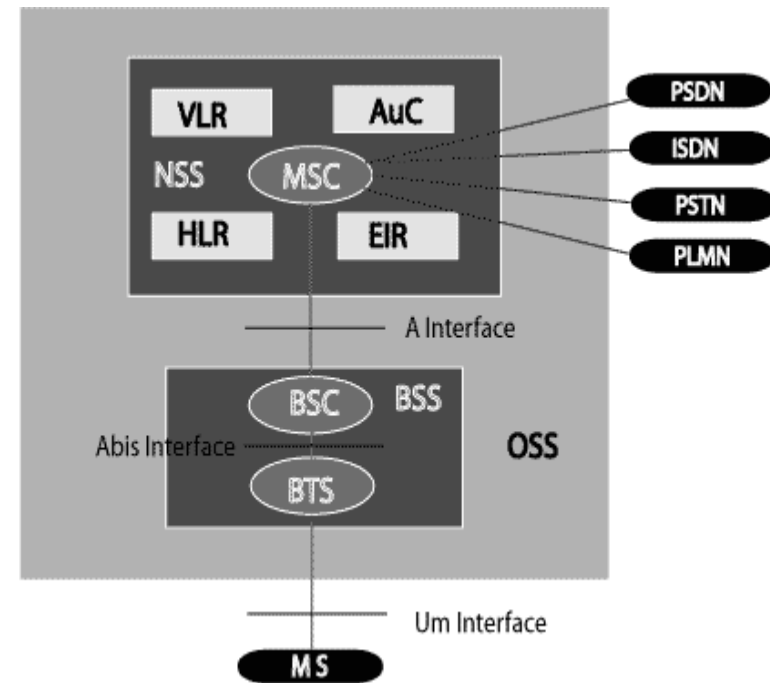
GSM….

The additional components of the GSM architecture comprise of databases and messaging systems functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AuC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU)
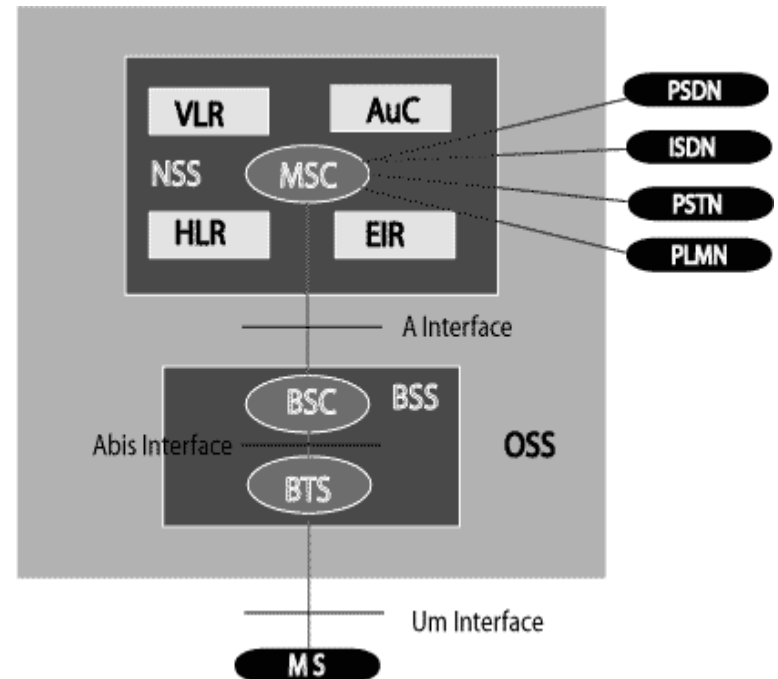
GSM...

The following diagram shows the GSM network along with the added elements:
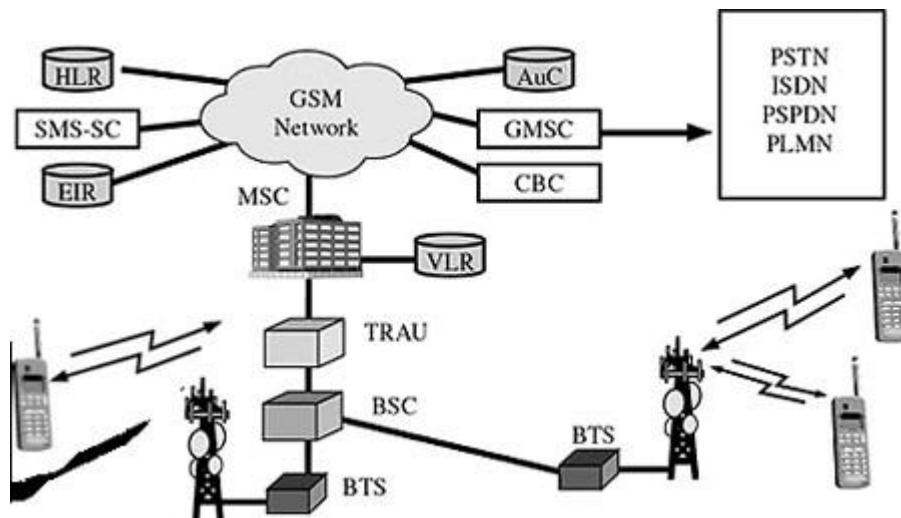
The MS and the BSS communicate across the Um interface. It is also known as the *air interface* or the *radio link*. The BSS communicates with the Network Service Switching (NSS) center across the *A* interface.

GSM network areas

- In a GSM network, the following areas are defined:
- **Cell** : Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.
- **Location Area** : A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs.
- **MSC/VLR Service Area** : The area covered by one MSC is called the MSC/VLR service area.
- **PLMN** : The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

The Mobile Station (MS)

- The MS consists of the physical equipment, such as the radio transceiver, display and digital signal processors, and the SIM card. It provides the air interface to the user in GSM networks. As such, other services are also provided, which include:

  - Voice teleservices
  - Data bearer services
  - The features' supplementary services

GSM Phone

GSM SIM Card

The MS Functions

- The MS also provides the receptor for SMS messages, enabling the user to toggle between the voice and data use. Moreover, the mobile facilitates access to voice messaging systems. The MS also provides access to the various data services available in a GSM network. These data services include:
  - X.25 packet switching through a synchronous or asynchronous dial-up connection to the PAD at speeds typically at 9.6 Kbps.
  - General Packet Radio Services (GPRSs) using either an X.25 or IP based data transfer method at speeds up to 115 Kbps.
  - High speed, circuit switched data at speeds up to 64 Kbps.

What is SIM?

- The SIM provides personal mobility so that the user can have access to all subscribed services irrespective of both the location of the terminal and the use of a specific terminal.

- You need to insert the SIM card into another GSM cellular phone to receive calls at that phone, make calls from that phone, or receive other subscribed services.

BTS and BSC

GSM BSS

- The BSS is composed of two parts:
  - The Base Transceiver Station (BTS)
  - The Base Station Controller (BSC)
- The BTS and the BSC communicate across the specified Abis interface, enabling operations between components that are made by different suppliers. The radio components of a BSS may consist of four to seven or nine cells.
- A BSS may have one or more base stations. The BSS uses the Abis interface between the BTS and the BSC.
- A separate high-speed line (T1 or E1) is then connected from the BSS to the Mobile MSC.
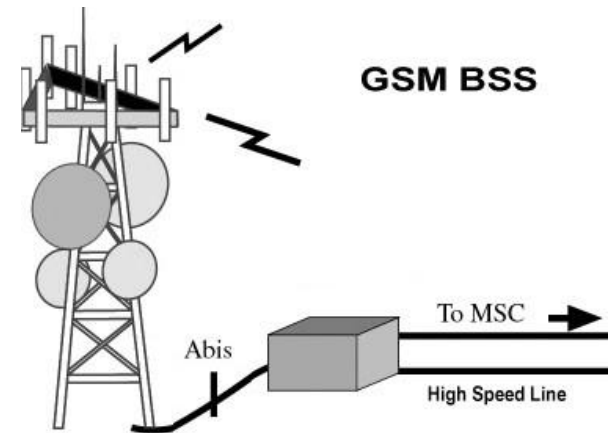
BTS…

- The Base Transceiver Station (BTS)
  - The BTS houses the radio transceivers that define a cell and handles the radio link protocols with the MS. In a large urban area, a large number of BTSs may be deployed.
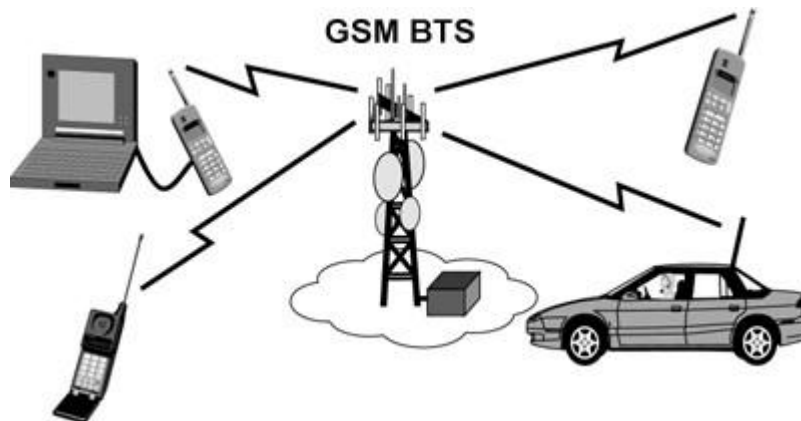
BTS….

- The BTS corresponds to the transceivers and antennas used in each cell of the network. A BTS is usually placed in the center of a cell. Its transmitting power defines the size of a cell. Each BTS has between 1 and 16 transceivers, depending on the density of users in the cell. Each BTS serves as a single cell. It also includes the following functions:

- Encoding, encrypting, multiplexing, modulating, and feeding the RF signals to the antenna

- Transcoding and rate adaptation

- Time and frequency synchronizing

- Voice through full- or half-rate services

- Decoding, decrypting, and equalizing received signals

- Random access detection

- Timing advances

- Uplink channel measurements

The Base Station Controller (BSC)

- The Base Station Controller (BSC)
  - The BSC manages the radio resources for one or more BTSs. It handles radio channel setup, frequency hopping, and handovers.
  - The BSC is the connection between the mobile and the MSC.
  - The BSC also translates the 13 Kbps voice channel used over the radio link to the standard 64 Kbps channel used by the Public Switched Telephone Network (PSDN) or ISDN.
  - It assigns and releases frequencies and time slots for the MS. The BSC also handles intercell handover.
  - It controls the power transmission of the BSS and MS in its area.
  - The function of the BSC is to allocate the necessary time slots between the BTS and the MSC.
  - It is a switching device that handles the radio resources.

BSC….

- Additional functions include:
  - Control of frequency hopping
  - Performing traffic concentration to reduce the number of lines from the MSC
  - Providing an interface to the Operations and Maintenance Center for the BSS
  - Reallocation of frequencies among BTSs
  - Time and frequency synchronization
  - Power management
  - Time-delay measurements of received signals from the MS

GPRS - Overview

- **General Packet Radio System** is also known as **GPRS** is a third-generation step toward internet access. GPRS is also known as GSM-IP that is a Global-System Mobile Communications Internet Protocol as it keeps the users of this system online, allows to make voice calls, and access internet on-the-go. Even Time-Division Multiple Access (TDMA) users benefit from this system as it provides packet radio access.

- GPRS also permits the network operators to execute an Internet Protocol (IP) based core architecture for integrated voice and data applications that will continue to be used and expanded for 3G services

Key Features

- Following three key features describe wireless packet data
- **The always online feature -** Removes the dial-up process, making applications only one click away.
- **An upgrade to existing systems -** Operators do not have to replace their equipment; rather, GPRS is added on top of the existing infrastructure.
- **An integral part of future 3G systems -** GPRS is the packet data core network for 3G systems EDGE and WCDMA.
- Goals of GPRS

Goals of GPRS

- GPRS is the first step toward an end-to-end wireless infrastructure and has the following goals:
- Open architecture
- Consistent IP services
- Same infrastructure for different air interfaces
- Integrated telephony and Internet infrastructure
- Leverage industry investment in IP
- Service innovation independent of infrastructure

Benefits of GPRS

- **Higher Data Rate**
- GPRS benefits the users in many ways, one of which is higher data rates in turn of shorter access times. In the typical GSM mobile, setup alone is a lengthy process and equally, rates for data permission are restrained to 9.6 kbit/s. The session establishment time offered while GPRS is in practice is lower than one second and ISDN-line data rates are up to many 10 kbit/s.
- **Easy Billing**
- GPRS packet transmission offers a more user-friendly billing than that offered by circuit switched services. In circuit switched services, billing is based on the duration of the connection. This is unsuitable for applications with bursty traffic. The user must pay for the entire airtime, even for idle periods when no packets are sent (e.g., when the user reads a Web page).

GPRS - Architecture

- GPRS architecture works on the same procedure like GSM network, but, has additional entities that allow packet data transmission.

- This data network overlaps a second-generation GSM network providing packet data transport at the rates from 9.6 to 171 kbps.

- Along with the packet data transport the GSM network accommodates multiple users to share the same air interface resources concurrently.

- Following is the GPRS Architecture diagram:

GPRS - Architecture
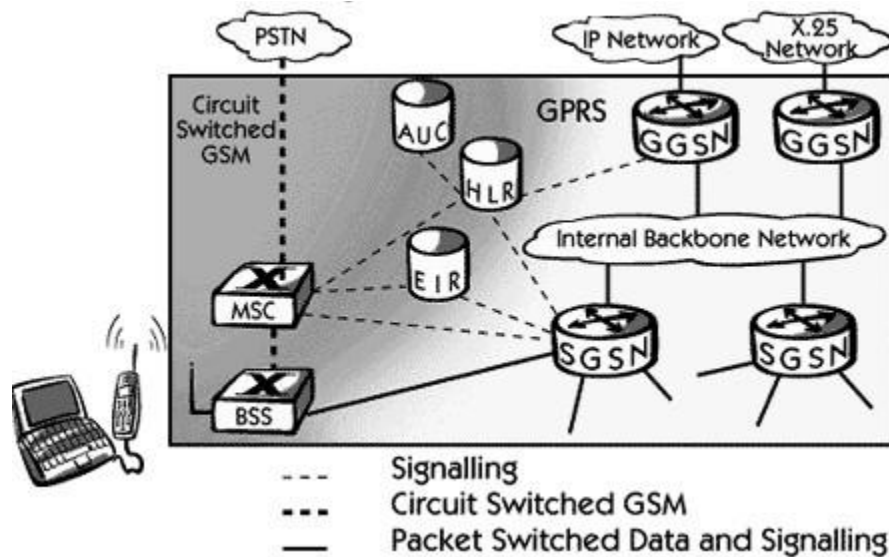
GPRS attempts to reuse the existing GSM network elements as much as possible, but toeffectively build a packet-based mobile cellular network, some new network elements, interfaces, and protocols for handling packet traffic are required. Therefore, GPRS requires modifications to numerous GSM network elements as summarized below:

GSM Network Element	Modification or Upgrade Required for GPRS.
Mobile Station (MS)	New Mobile Station is required to access GPRS services. These new terminals will be backward compatible with GSM for voice calls.
BTS	A software upgrade is required in the existing Base Transceiver Station(BTS).
BSC	The Base Station Controller (BSC) requires a software upgrade and the installation of new hardware called the packet control unit (PCU). The PCU directs the data traffic to the GPRS network and can be a separate hardware element associated with the BSC.
GPRS Support Nodes (GSNs)	The deployment of GPRS requires the installation of new core network elements called the serving GPRS support node (SGSN) and gateway GPRS support node (GGSN).
Databases (HLR, VLR, etc.)	All the databases involved in the network will require software upgrades to handle the new call models and functions introduced by GPRS.

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

Internet WAP

HTML, Java	A-SAP	
	Application Layer (WAE)	additional services and applications
	S-SAP	
HTTP	Session Layer (WSP)	
	TR-SAP	
	Transaction Layer (WTP)	
	SEC-SAP	
SSL/TLS	Security Layer (WTLS)	
	T-SAP	
TCP/IP, UDP/IP, media	Transport Layer (WDP)	WCMP
	Bearers (GSM, CDPD, ...)	
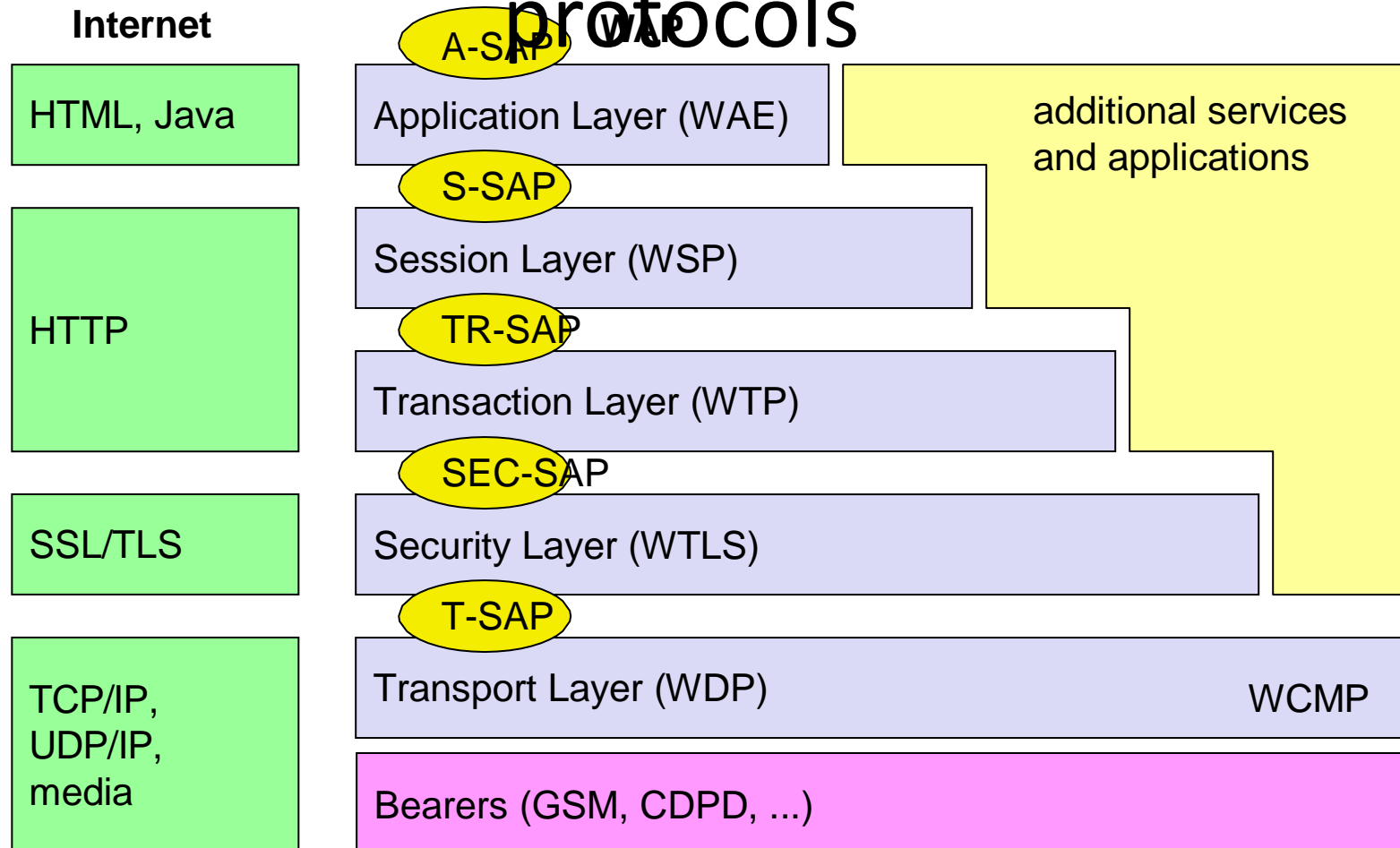
WAE comprises WML (Wireless Markup Language), WML Script, WTAI etc.

WAP - network elements

fixed network

wireless network

Internet

HTML

filter

WML

WAP
proxy

Binary WML

HTML

HTML

WML

web
server

HTML

filter/
WAP
proxy

Binary WML

PSTN

WTA
server
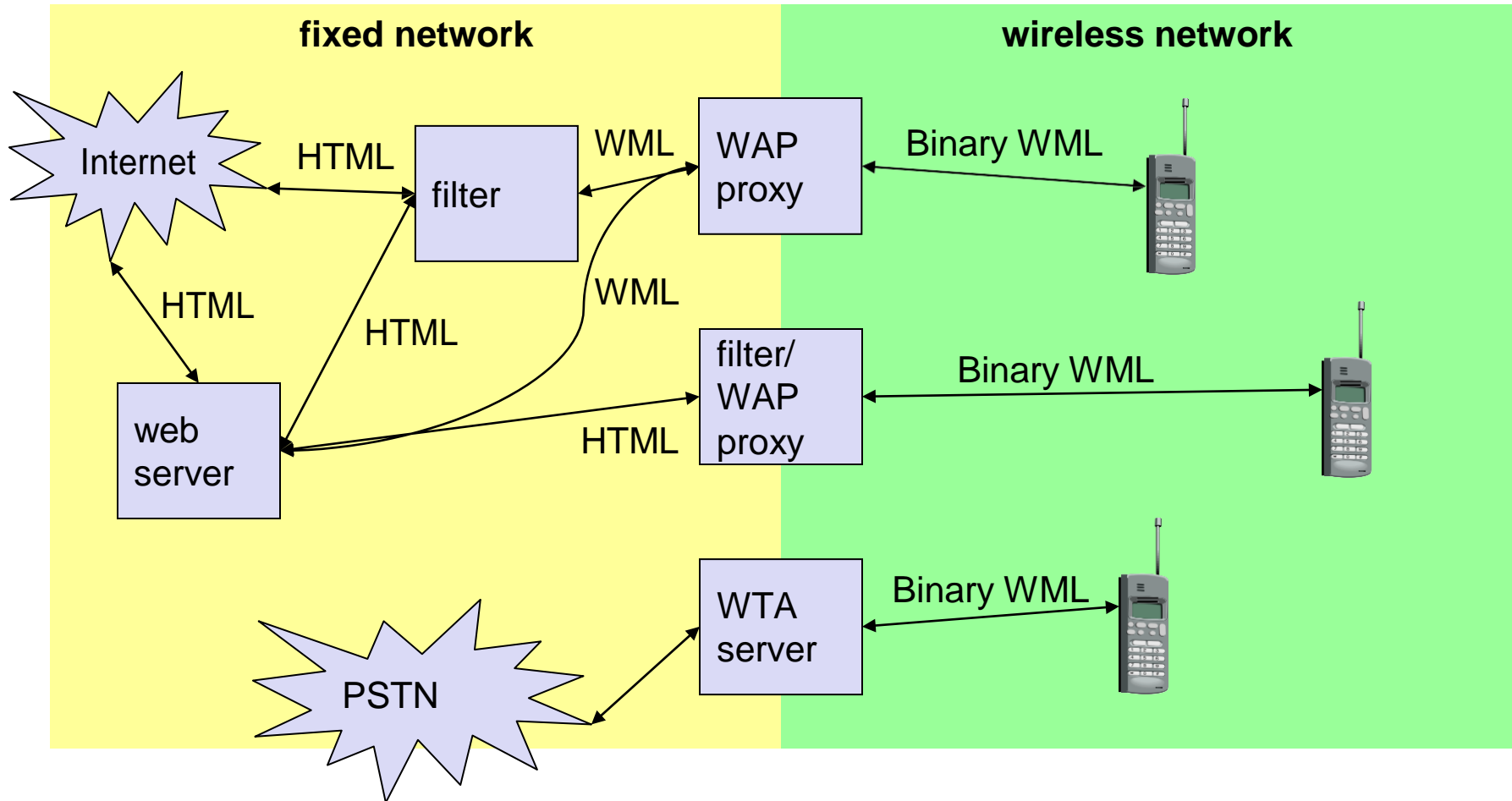
Binary WML

Binary WML: binary file format for clients

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)

WDP - Service Primitives

Usage of WDP

GSM-SMS

Wireless Data Gateway

| WTLS |
| WDP & Adaptation |
| SMS |

| SMS | Tunnel |
| | Subnetwork |

| WTLS |
| WDP & Adaptation |
| Tunnel |
| Subnetwork |

GSM-CSD

WAP Proxy

| WTLS |
| UDP |
| IP |
| PPP |
| CSD-RF |

Interworking Function

| CSD-RF | PSTN Circuit |

Internet Service Provider Remote Access Service

IP	
PPP	Subnetwork
PSTN Circuit	
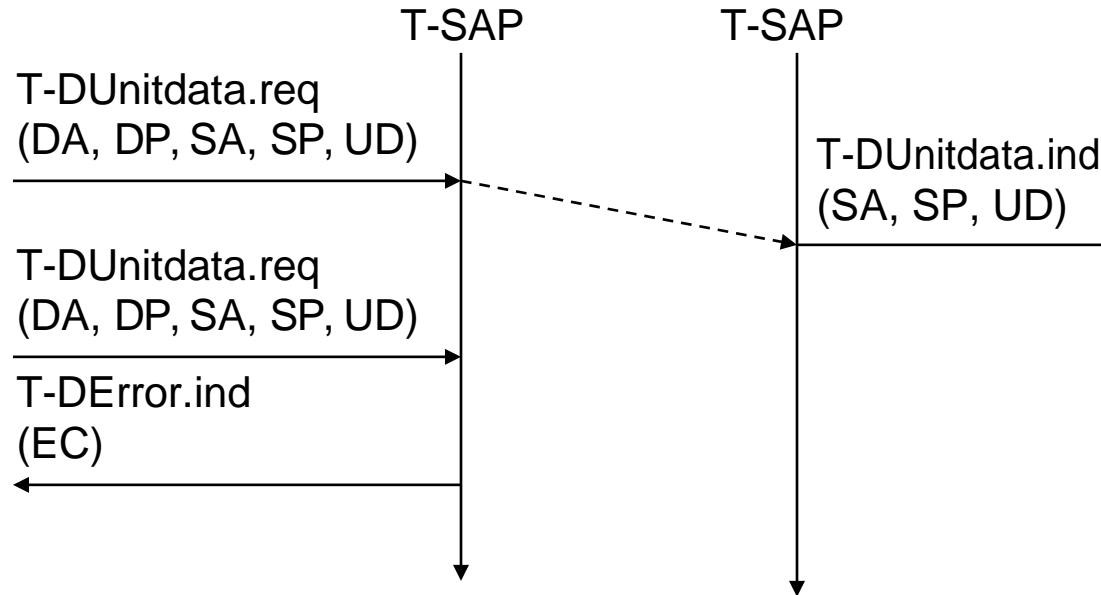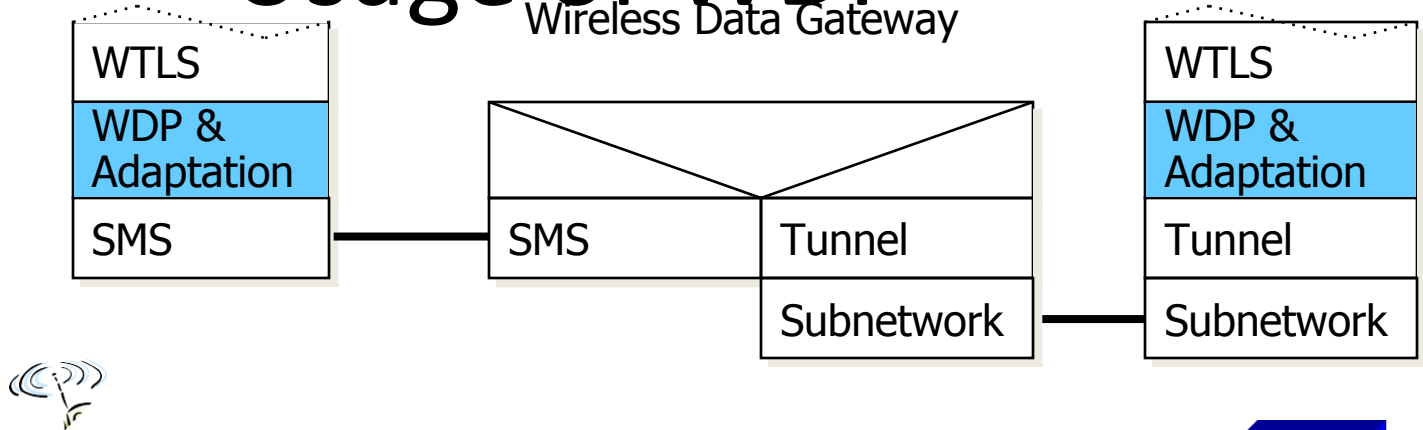
| WTLS |
| UDP |
| IP |
| Subnetwork |
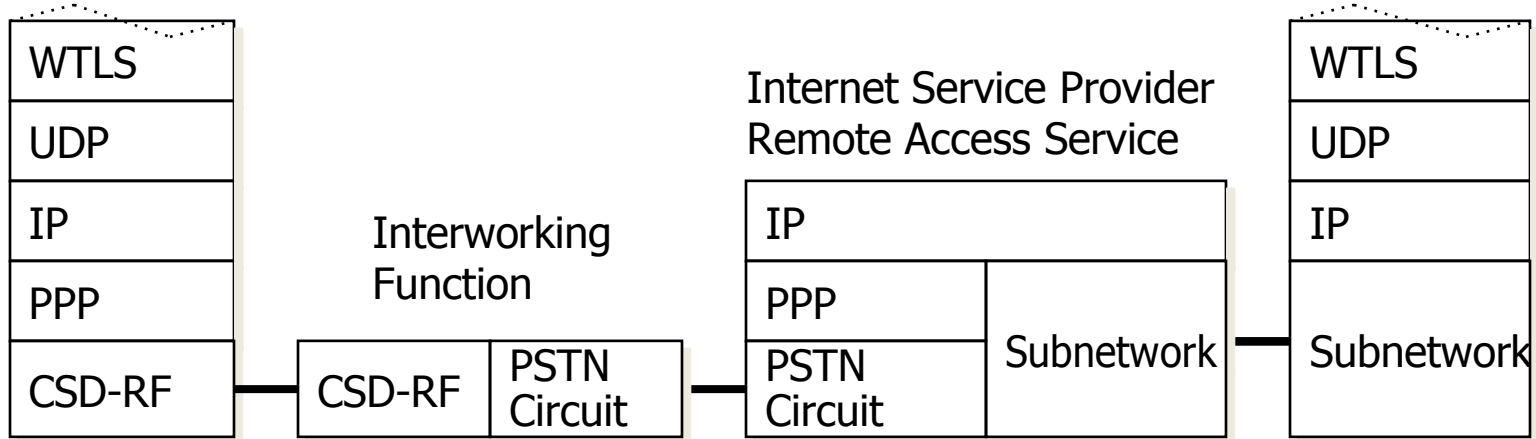
WTLS - Wireless Transport Layer Security

- 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

Secure session, full handshake

originator
SEC-SAP

peer
SEC-SAP

SEC-Create.req
(SA, SP, DA, DP, KES, CS, CM)

SEC-Create.ind
(SA, SP, DA, DP, KES, CS, CM)

SEC-Create.res
(SNM, KR, SID, KES', CS', CM')

SEC-Create.cnf
(SNM, KR, SID, KES', CS', CM')

SEC-Exchange.req

SEC-Exchange.ind

SEC-Exchange.res
(CC)

SEC-Commit.req

SEC-Exchange.cnf
(CC)

SEC-Commit.ind

SEC-Commit.cnf
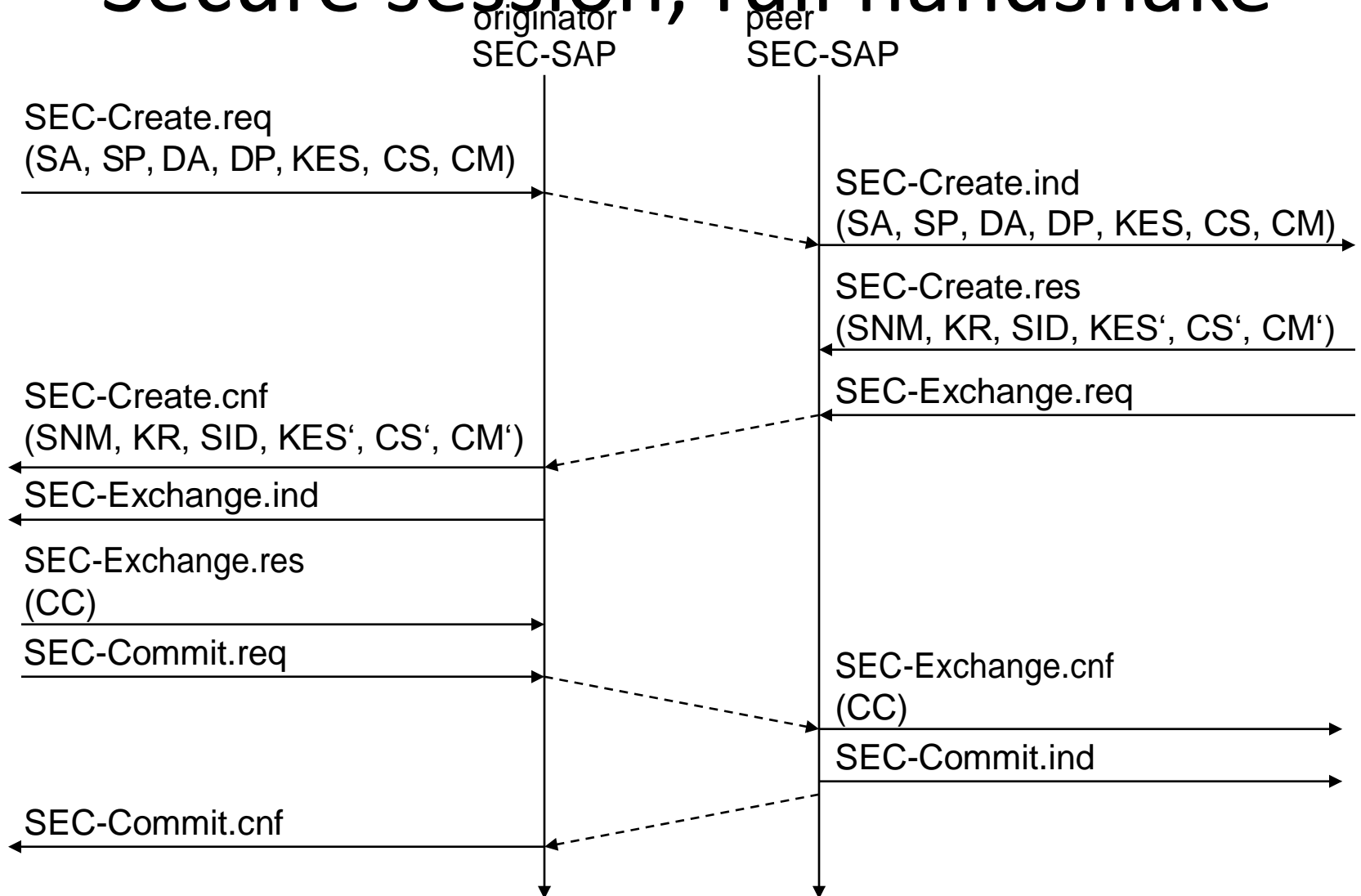
SEC-Unitdata - transferring datagrams

sender
SEC-SAP

receiver
SEC-SAP

SEC-Unitdata.req
(SA, SP, DA, DP, UD)

SEC-Unitdata.ind
(SA, SP, DA, DP, UD)
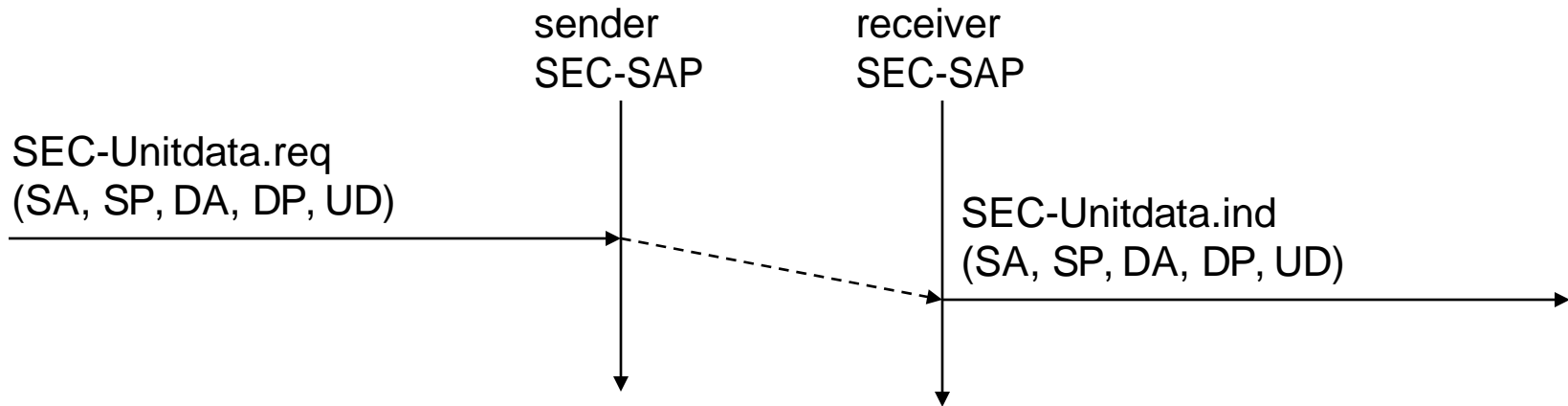
WTP - Wireless Transaction Protocol

- 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)

Details of WTP I

- Support of different communication scenarios

  - Class 0: unreliable message transfer
    - Example: push service

  - Class 1: reliable request
    - An invoke message is not followed by a result message
    - Example: reliable push service

  - Class 2: reliable request/response
    - An invoke message is followed by exactly one result message
    - With and without ACK
    - Example: typical web browsing

- No explicit connection setup or release is available

- Services for higher layers are called events

Details of WTP II

- Used Mechanisms
  - Reliability
    - Unique transaction identifiers (TID)
    - Acknowledgements
    - Selective retransmission
    - Duplicate removal
  - Optional: concatenation & separation of messages
  - Optional: segmentation & reassembly of messages
  - Asynchronous transactions
  - Transaction abort, error handling
  - Optimized connection setup (includes data transmission)

WTP Class 0 transaction

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=0, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=0, H')
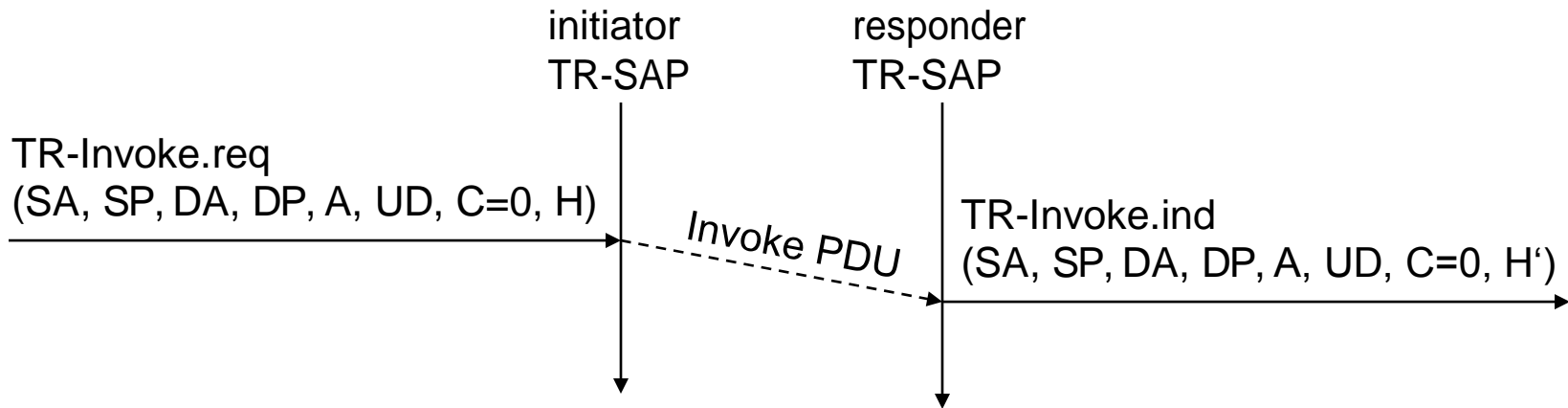
WTP Class 1 transaction, no user ack & user ack

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=1, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=1, H')

TR-Invoke.cnf
(H)

Ack PDU

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=1, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=1, H')

TR-Invoke.res
(H')

TR-Invoke.cnf
(H)

Ack PDU
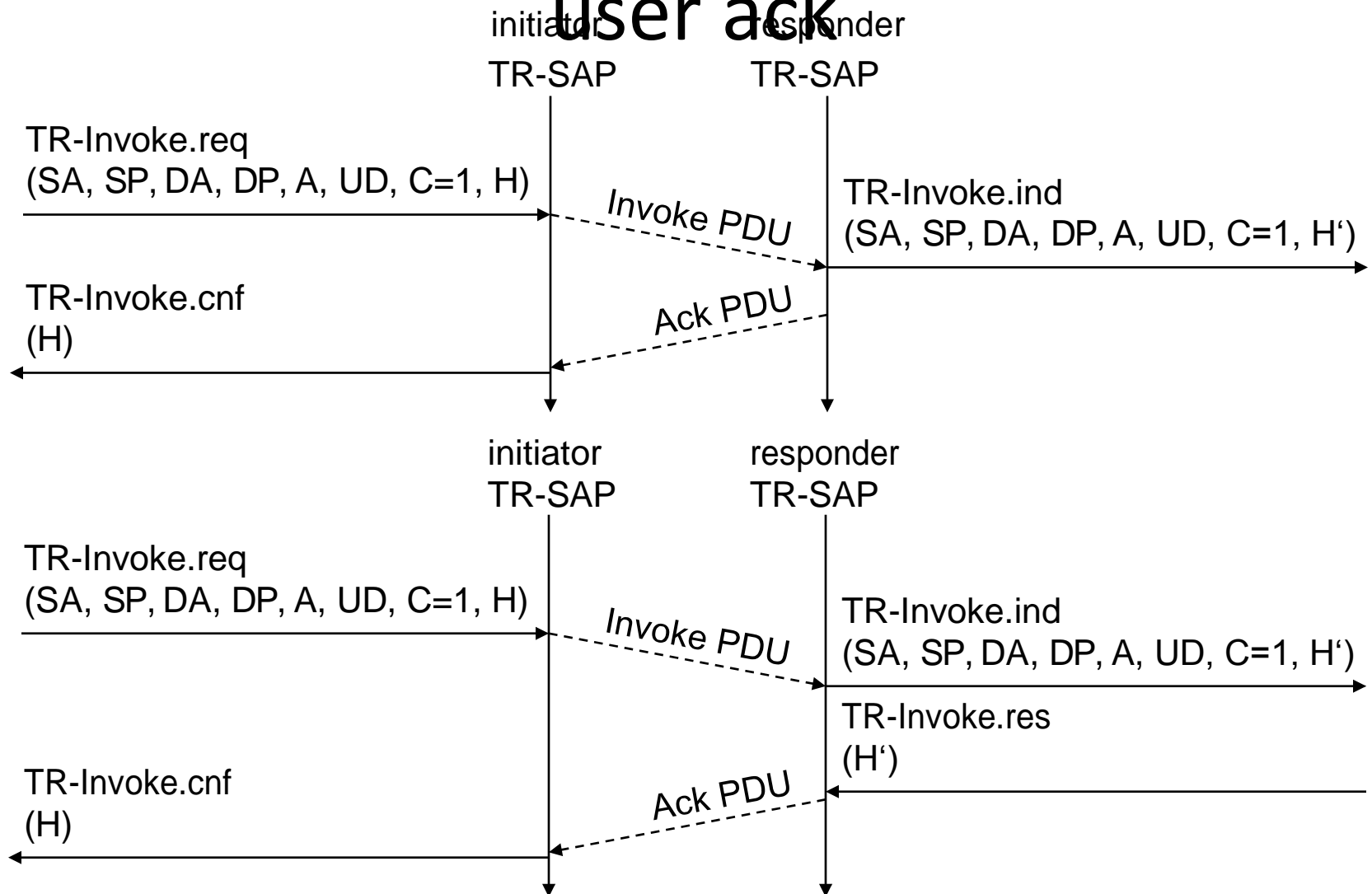
WTP Class 2 transaction, no user ack, no hold on

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=2, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=2, H')

TR-Result.req
(UD*, H')

TR-Invoke.cnf
(H)

Result PDU

TR-Result.ind
(UD*, H)

TR-Result.res
(H)

Ack PDU

TR-Result.cnf
(H')
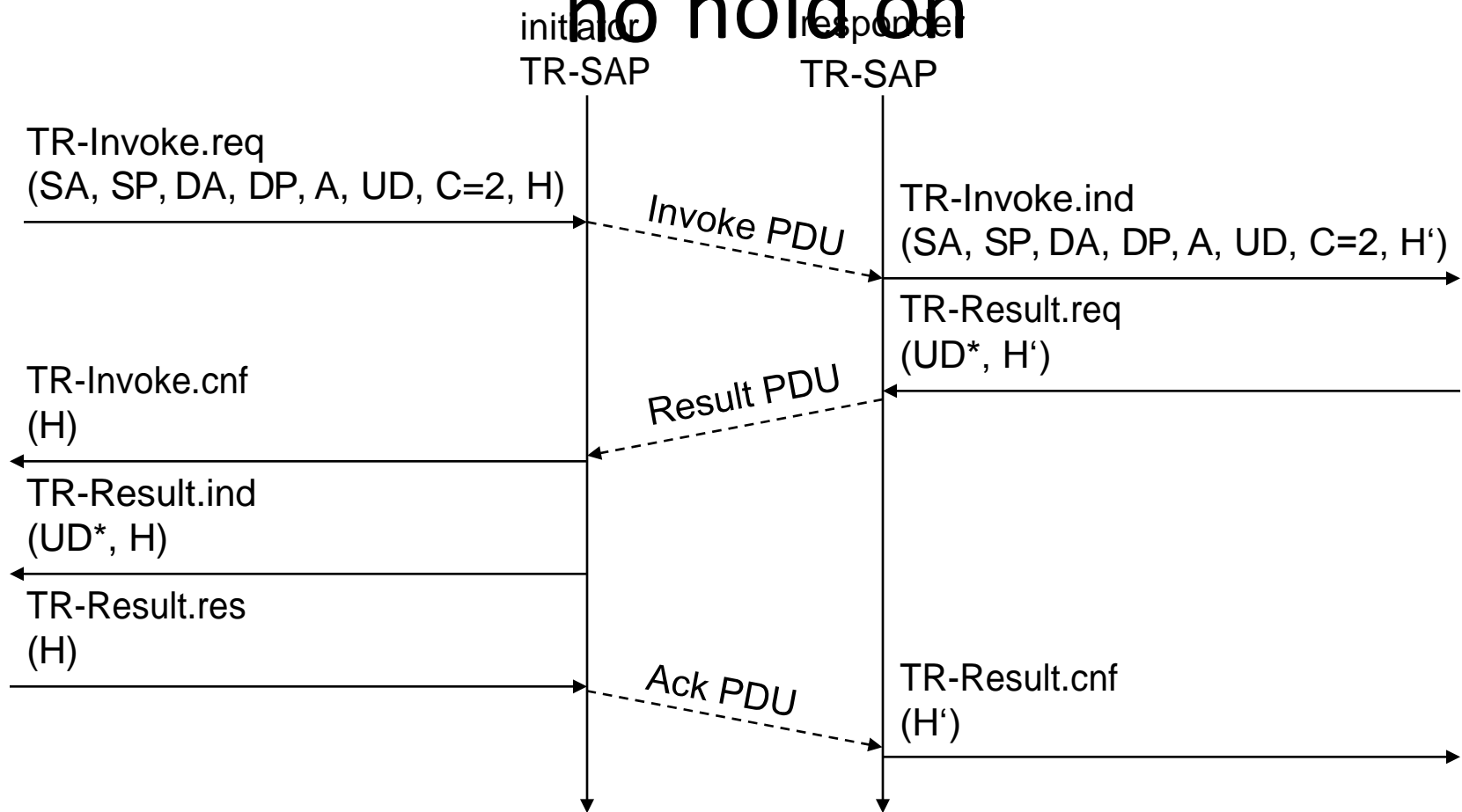
WTP Class 2 transaction, user ack

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=2, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=2, H')

TR-Invoke.res
(H')

TR-Invoke.cnf
(H)

Ack PDU

TR-Result.req
(UD*, H')

TR-Result.ind
(UD*, H)

Result PDU

TR-Result.res
(H)

Ack PDU

TR-Result.cnf
(H')
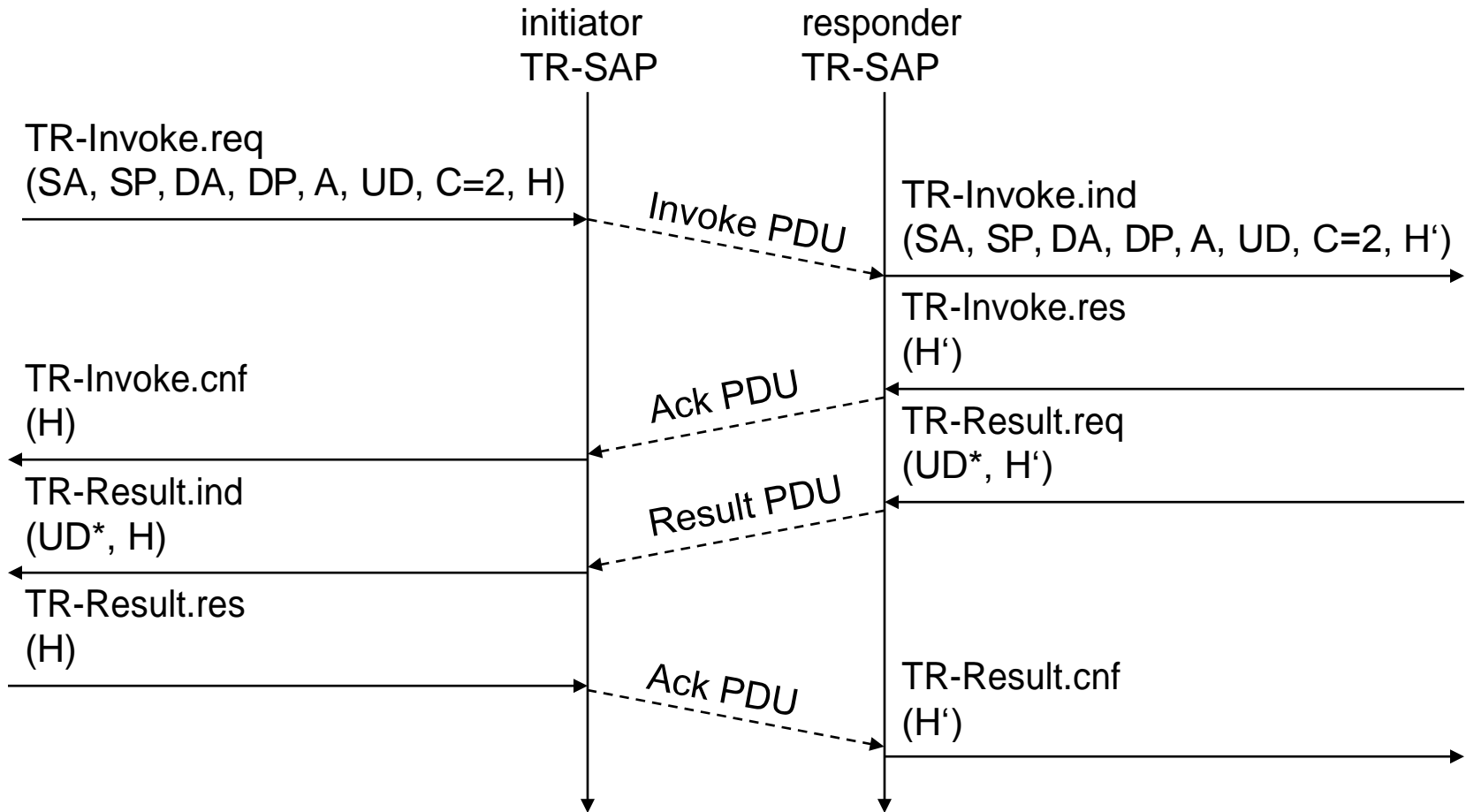
WTP Class 2 transaction, hold on, no user ack

initiator
TR-SAP

responder
TR-SAP

TR-Invoke.req
(SA, SP, DA, DP, A, UD, C=2, H)

Invoke PDU

TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=2, H')

TR-Invoke.cnf
(H)

Ack PDU

TR-Result.req
(UD*, H')

TR-Result.ind
(UD*, H)

Result PDU

TR-Result.res
(H)

Ack PDU

TR-Result.cnf
(H')
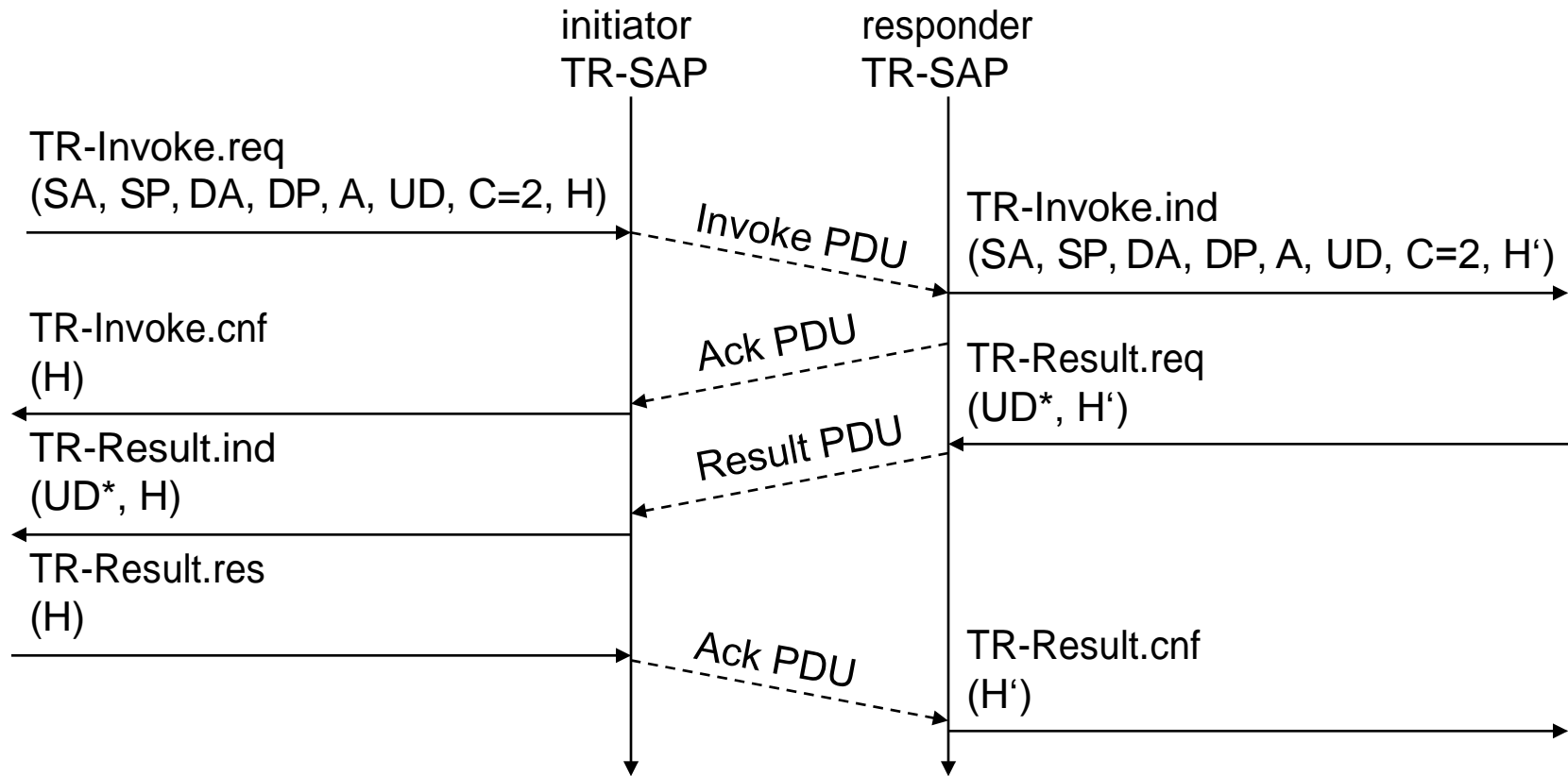
WSP - Wireless Session Protocol

- 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

WSP protocols

WSP

Connection mode
(uses WTP)

Connectionless mode
(uses WDP or WTLS)

- Session Management (class 0, 2)
- Method Invocation (Kl. 2)
- Error Report
- Push (class 0)
- Confirmed Push (class 1)
- Session suspend/resume (class 0, 2)

- Method Invocation
- Push

(in general unreliable)

WSP/B session establishment

WSP/B session suspend/resume

client
S-SAP

server
S-SAP

S-Suspend.req

Suspend PDU

S-Suspend.ind
(R)

S-Suspend.ind
(R)

WTP Class 0
transaction

~ ~

S-Resume.req
(SA, CA)

Resume PDU

S-Resume.ind
(SA, CA)

S-Resume.res

Reply PDU

S-Resume.cnf

WTP Class 2
transaction
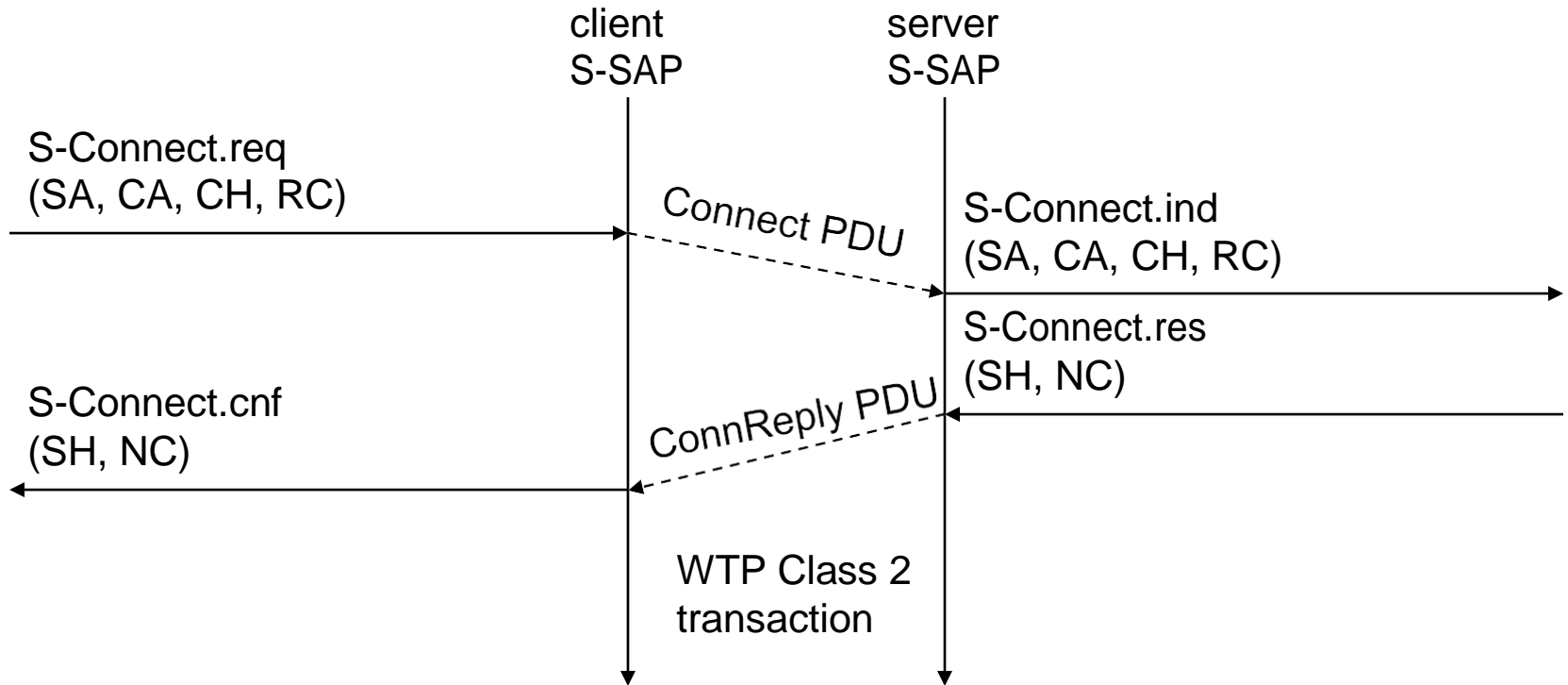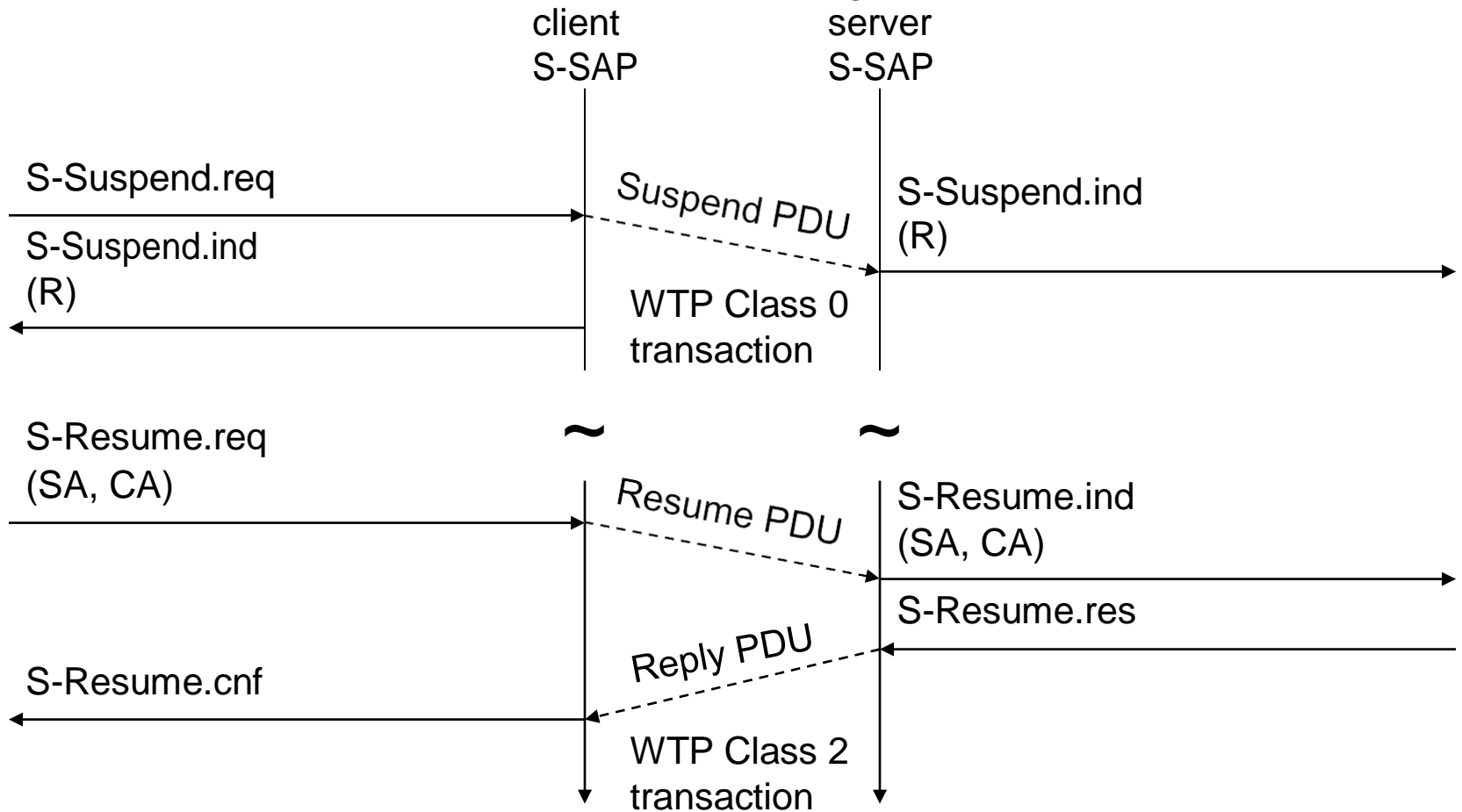
WSP/B session termination

client
S-SAP

server
S-SAP

S-Disconnect.req
(R)

Disconnect PDU

S-Disconnect.ind
(R)

S-Disconnect.ind
(R)

WTP Class 0
transaction

WSP/B method invoke

client
S-SAP

server
S-SAP

S-MethodInvoke.req
(CTID, M, RU)

Method PDU

S-MethodInvoke.ind
(STID, M, RU)

S-MethodInvoke.res
(STID)

S-MethodInvoke.cnf
(CTID)

S-MethodResult.req
(STID, S, RH, RB)

S-MethodResult.ind
(CTID, S, RH, RB)

Reply PDU

S-MethodResult.res
(CTID)

S-MethodResult.cnf
(STID)

WTP Class 2
transaction

WSP/B over WTP - method invocation

client
S-SAP

initiator
TR-SAP

responder
TR-SAP

server
S-SAP

S-MethodInvoke.req → TR-Invoke.req → *Invoke(Method)* → TR-Invoke.ind → S-MethodInvoke.ind →

TR-Invoke.res ← S-MethodInvoke.res ←

S-MethodInvoke.cnf ← TR-Invoke.cnf ← *Ack PDU* ← TR-Result.req ← S-MethodResult.req ←

S-MethodResult.ind ← TR-Result.ind ← *Result(Reply)* ←

S-MethodResult.res → TR-Result.res → *Ack PDU* → TR-Result.cnf → S-MethodResult.cnf →
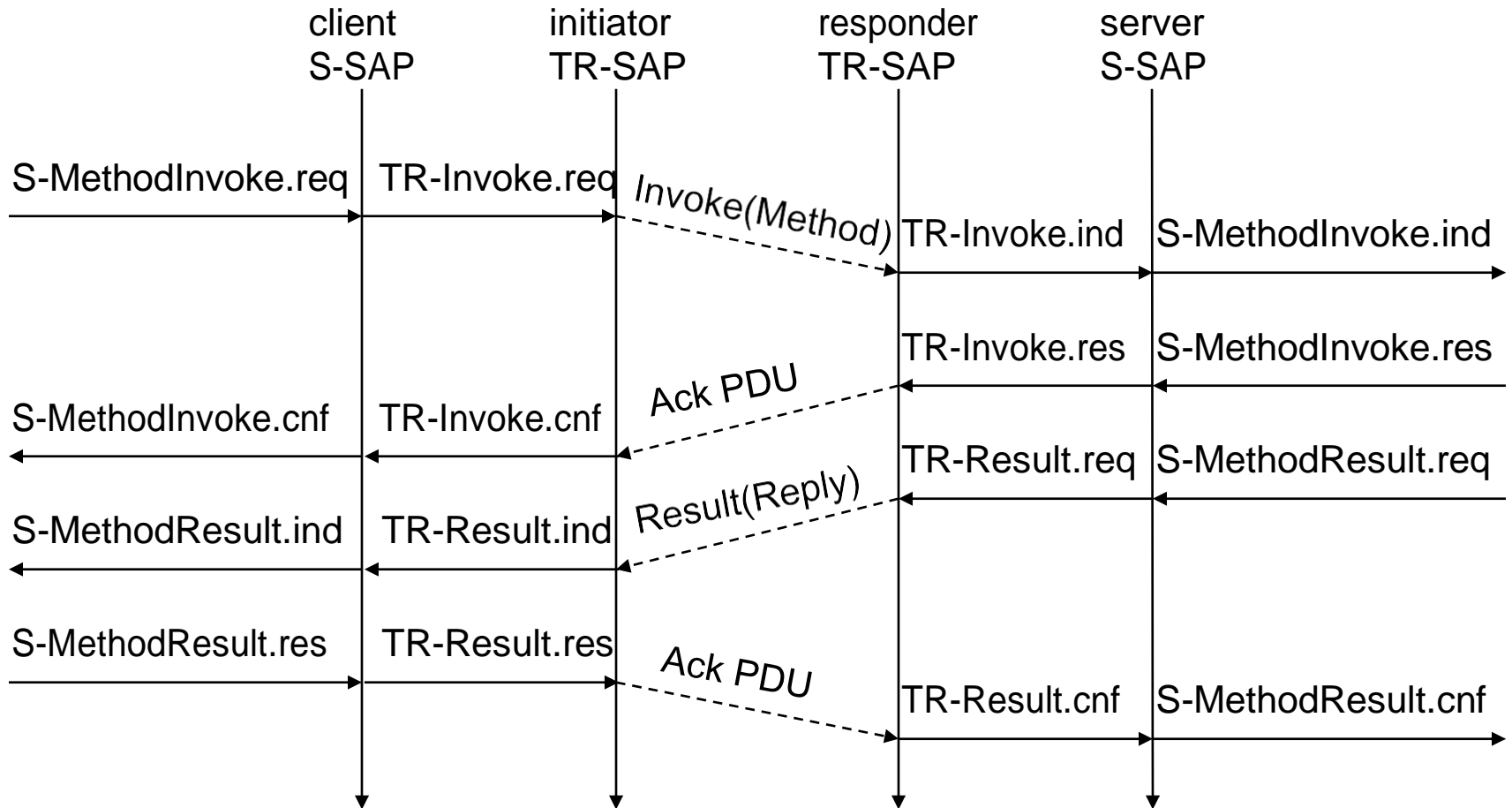
WSP/B over WTP - asynchronous, unordered requests

client
S-SAP

server
S-SAP

S-MethodInvoke_1.req

S-MethodInvoke_2.req

S-MethodInvoke_2.ind

S-MethodInvoke_1.ind

S-MethodInvoke_3.req

S-MethodResult_1.req

S-MethodInvoke_3.ind

S-MethodResult_1.ind

S-MethodResult_3.req

S-MethodResult_3.ind

S-MethodResult_2.req

S-MethodInvoke_4.req

S-MethodInvoke_4.ind

S-MethodResult_4.ind

S-MethodResult_4.req

S-MethodResult_2.ind
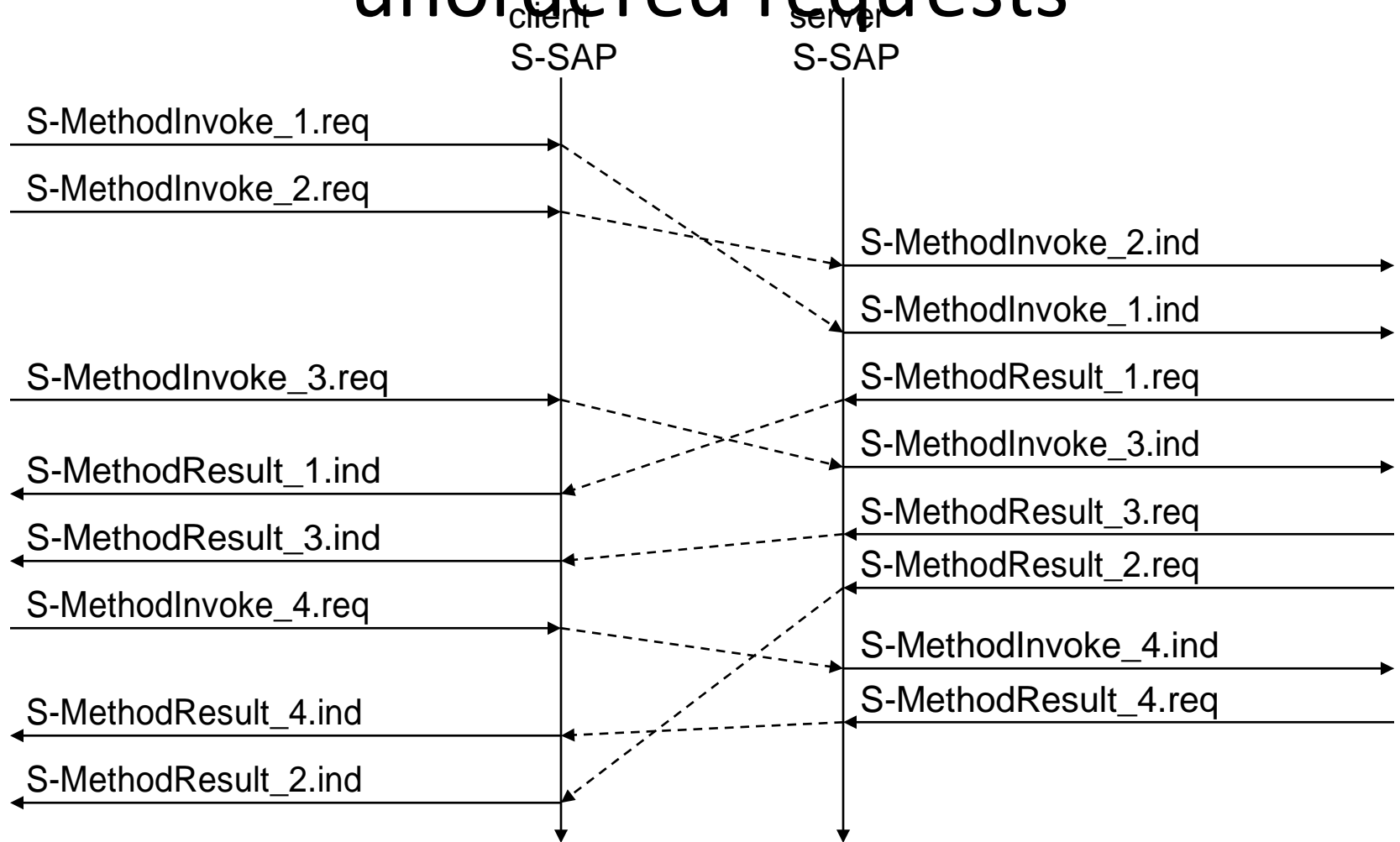
WSP/B - confirmend/non-confirmed push

client
S-SAP

server
S-SAP

S-Push.req
(PH, PB)

S-Push.ind
(PH, PB)

Push PDU

WTP Class 0
transaction

client
S-SAP

server
S-SAP

S-ConfirmedPush.req
(SPID, PH, PB)

S-ConfirmedPush.ind
(CPID, PH, PB)

ConfPush PDU

S-ConfirmedPush.res
(CPID)

S-ConfirmedPush.cnf
(SPID)

WTP Class 1
transaction

WSP/B over WDP

client
S-SAP

server
S-SAP

S-Unit-MethodInvoke.req
(SA, CA, TID, M, RU)

Method PDU

S-Unit-MethodInvoke.ind
(SA, CA, TID, M, RU)

S-Unit-MethodResult.req
(CA, SA, TID, S, RH, RB)

S-Unit-MethodResult.ind
(CA, SA, TID, S, RH, RB)

Reply PDU

S-Unit-Push.req
(CA, SA, PID, PH, PB)

S-Unit-Push.ind
(CA, SA, PID, PH, PB)

Push PDU

WDP Unitdata
service

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
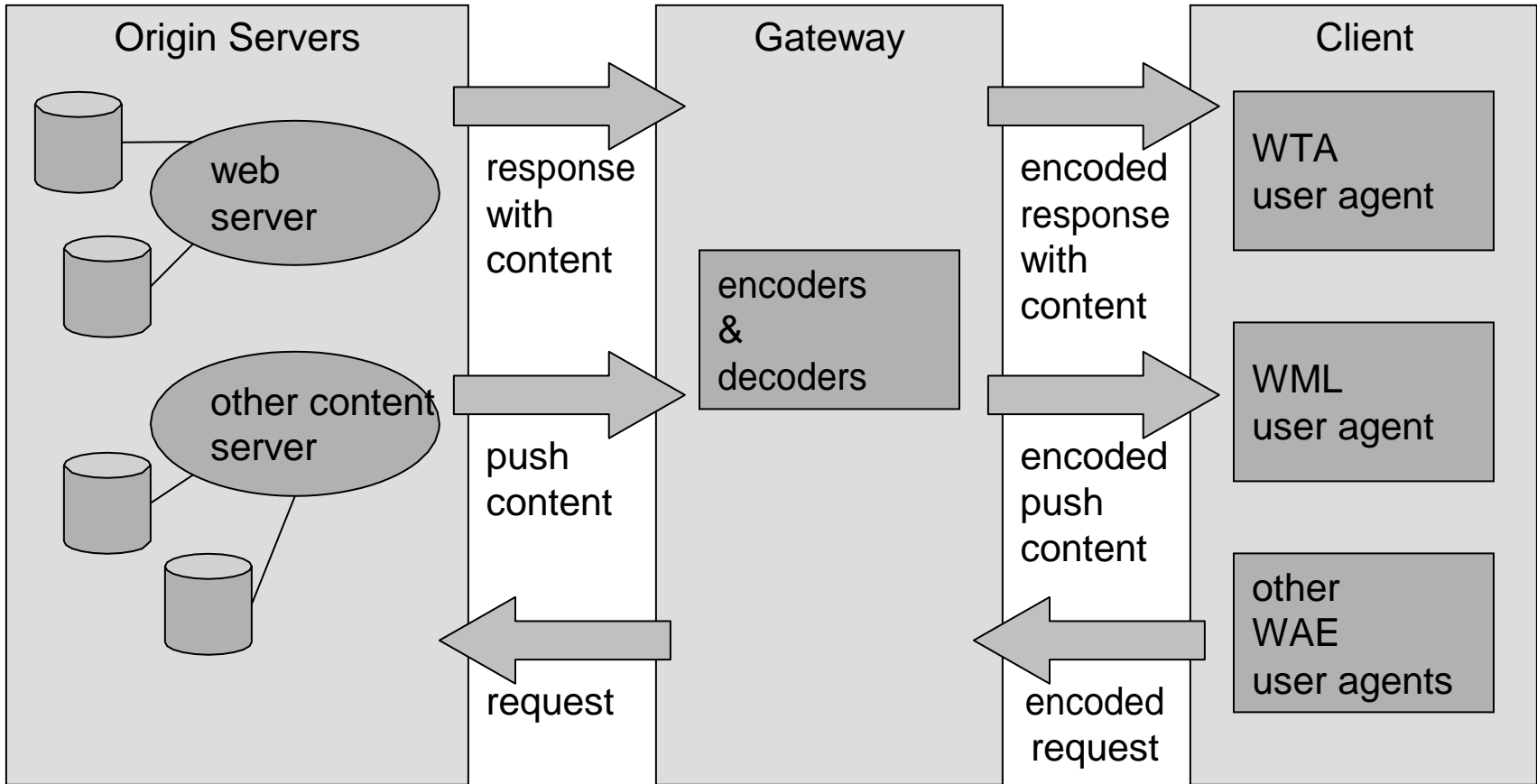
Wireless Markup Language (WML)

- 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

WML – example I

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
            "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="card_one" title="simple example">
        <do type="accept">
            <go href="#card_two"/>
        </do>
        <p>
        This is a simple first card!
        <br/>
        On the next one you can choose ...
        </p>
    </card>
```

WML – example II

```
<card id="card_two" title="Pizza selection">
        <do type="accept" label="cont">
            <go href="#card_three"/>
        </do>
        <p>
        ... your favorite pizza!
        <select value="Mar" name="PIZZA">
            <option value="Mar">Margherita</option>
            <option value="Fun">Funghi</option>
            <option value="Vul">Vulcano</option>
        </select>
        </p>
    </card>
    <card id="card_three" title="Your Pizza!">
        <p>
        Your personal pizza parameter is <b>$(PIZZA)</b>!
        </p>
    </card>
</wml>
```

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
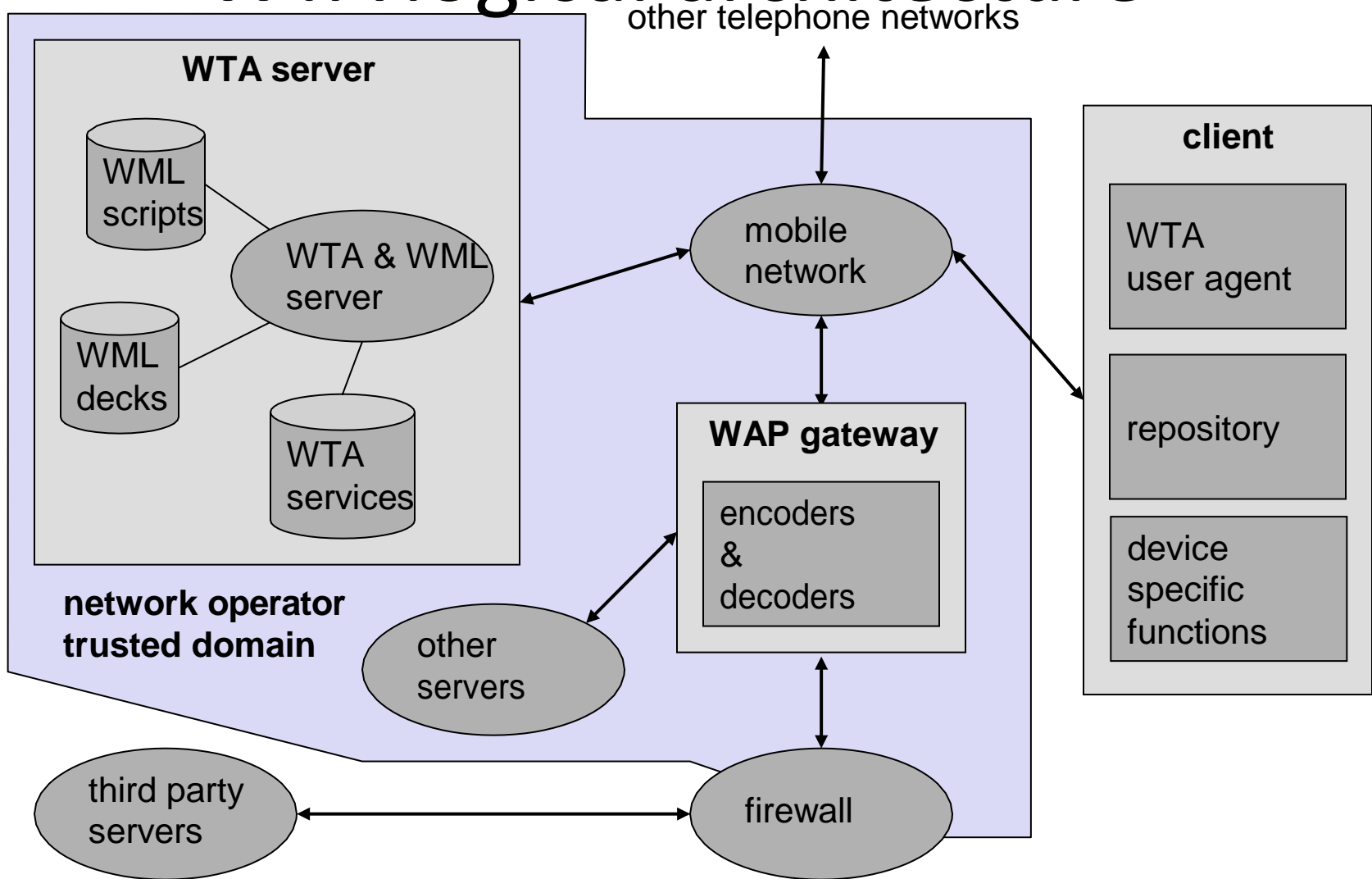
WMLScript - example

```
function pizza_test(pizza_type) {
    var taste = "unknown";
    if (pizza_type = "Margherita") {
        taste = "well... ";
    }
    else {
        if (pizza_type = "Vulcano") {
            taste = "quite hot";
        };
    };
    return taste;
};
```
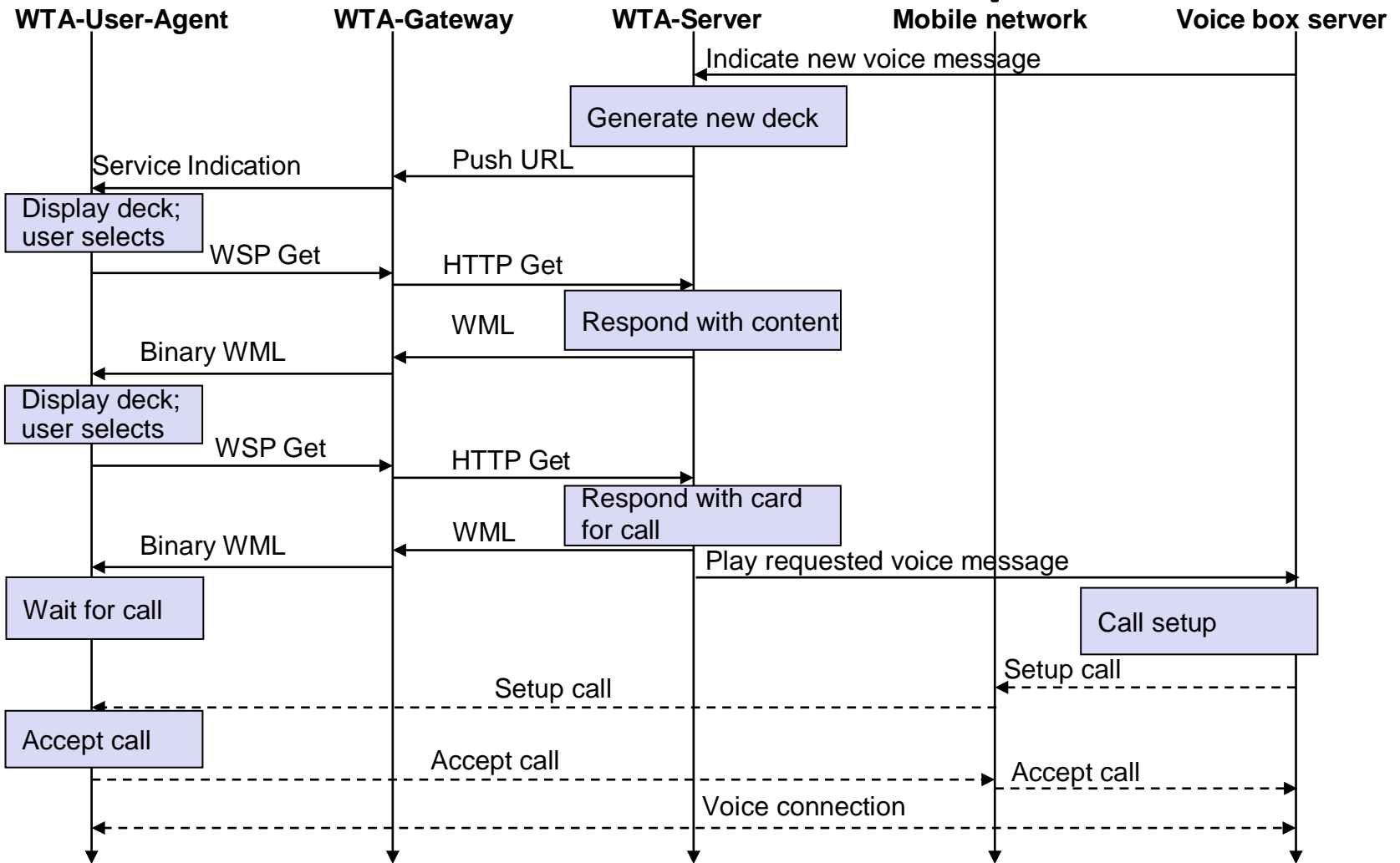
Wireless Telephony Application (WTA)

- Collection of telephony specific extensions
- Extension of basic WAE application model
  - content push
    - server can push content to the client
    - client may now be able to handle unknown events
  - handling of network events
    - table indicating how to react on certain events from the network
  - access to telephony functions
    - any application on the client may access telephony functions
- Example
  - calling a number (WML)
    `wtai://wp/mc;07216086415`
  - calling a number (WMLScript)
    `WTAPublic.makeCall("07216086415");`

WTA logical architecture

other telephone networks

WTA server

WML scripts

WTA & WML server

WML decks

WTA services

network operator trusted domain

mobile network

other servers

WAP gateway

encoders & decoders

third party servers

firewall

client

WTA user agent

repository

device specific functions

Voice box example

WTA-User-Agent	WTA-Gateway	WTA-Server	Mobile network	Voice box server

Indicate new voice message

Generate new deck

Service Indication ← Push URL

Display deck; user selects

WSP Get → HTTP Get →

WML ← Respond with content

Binary WML ←

Display deck; user selects

WSP Get → HTTP Get →

Respond with card for call

WML ← Binary WML ←

Play requested voice message

Wait for call

Call setup

Setup call

Setup call

Accept call

Accept call

Accept call

Voice connection

WTAI - example with WML only

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
            "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="card_one" title="Tele voting">
        <do type="accept">
            <go href="#card_two"/>
        </do>
        <p> Please choose your candidate! </p>
    </card>
    <card id="card_two" title="Your selection">
        <do type="accept">
            <go href="wtai://wp/mc;$dialno"/>
        </do>
        <p> Your selection:
        <select name="dialno">
            <option value="01376685">Mickey</option>
            <option value="01376686">Donald</option>
            <option value="01376687">Pluto</option>
        </select>
        </p>
    </card>
</wml>
```

WTAI - example with WML and WMLScript I

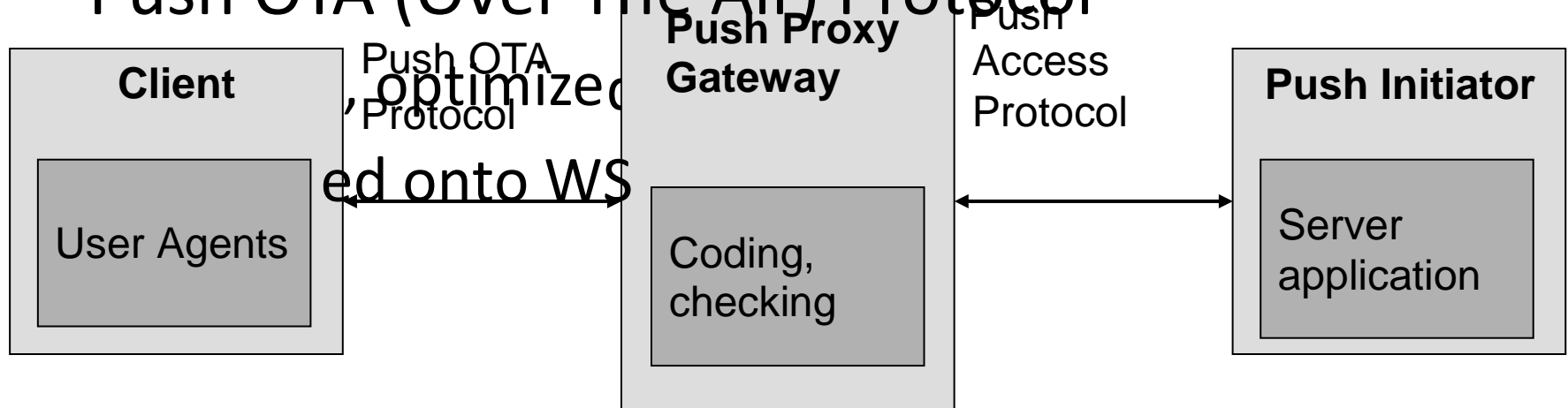```
function voteCall(Nr) {
    var j = WTACallControl.setup(Nr,1);
    if (j>=0) {
        WMLBrowser.setVar("Message", "Called");
        WMLBrowser.setVar("No", Nr);
    }
    else {
        WMLBrowser.setVar("Message", "Error!");
        WMLBrowser.setVar("No", j);
    }
    WMLBrowser.go("showResult");
}
```

# WTAI - example with WML and WMLScript II

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
            "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="card_one" title="Tele voting">
        <do type="accept"> <go href="#card_two"/> </do>
        <p> Please choose your candidate! </p>
    </card>
    <card id="card_two" title="Your selection">
        <do type="accept">
            <go href="/myscripts#voteCall($dialno)"/> </do>
        <p> Your selection:
        <select name="dialno">
            <option value="01376685">Mickey</option>
            <option value="01376686">Donald</option>
            <option value="01376687">Pluto</option>
        </select> </p>
    </card>
    <card id="showResult" title="Result">
        <p> Status: $Message $No </p>
    </card>
</wml>
```

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
  - , optimized
  - ed onto WS

Client	Push OTA Protocol	**Push Proxy Gateway**	Push Access Protocol	**Push Initiator**
User Agents		Coding, checking		Server application

Push/Pull services in WAP I

- Service Indication
  - Service announcement using a pushed short message
  - Service usage via a pull
  - Service identification via a URI

```
<?xml version="1.0"?>
  <!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
    "http://www.wapforum.org/DTD/si.dtd">
<si>
  <indication
  href="http://www.piiiizza4u.de/offer/salad.wml"
    created="2007-10-30T17:45:32Z"
    si-expires="2007-10-30T17:50:31Z">
    Salad special: The 5 minute offer
  </indication>
</si>
```

Push/Pull services in WAP II

- Service Loading
  - short message pushed to a client containing a URI
  - User agent decides whether to use the URI via a pull
  - Transparent for users, always looks like a push

```
<?xml version="1.0"?>
  <!DOCTYPE  sl  PUBLIC  "-//WAPFORUM//DTD  SL
  1.0//EN"
    "http://www.wapforum.org/DTD/sl.dtd">
<sl

  href="http://www.piiiizza4u.de/offer/salad.wm
  l">
</sl>
```
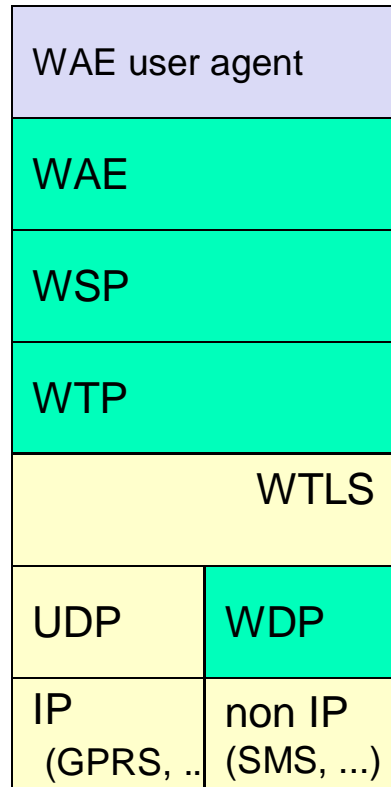
Examples for WAP protocol stacks (WAP 1.x)

WAP standardization

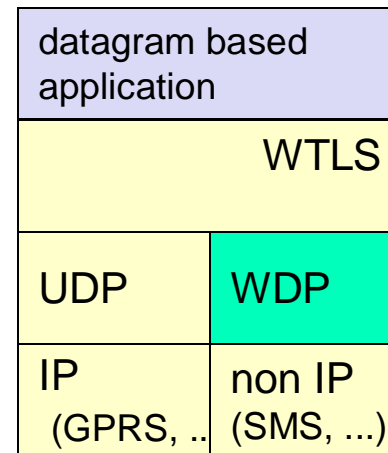| outside WAP |
|---|

1.

WAE user agent
WAE
WSP
WTP

	WTLS
UDP	WDP
IP (GPRS, ..	non IP (SMS, ...)

typical WAP
application with
complete protocol
stack

2.

transaction based application
WTP

	WTLS
UDP	WDP
IP (GPRS, ..	non IP (SMS, ...)

3.

datagram based application

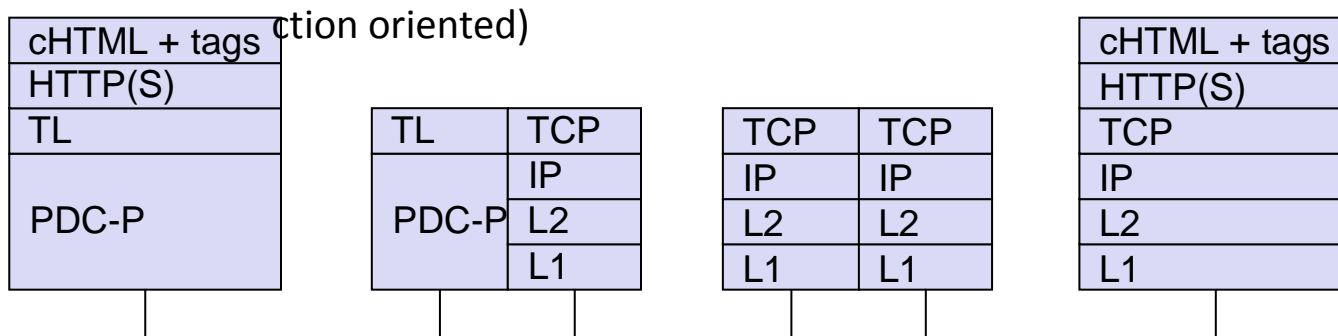|	WTLS
UDP	WDP
IP (GPRS, ..	non IP (SMS, ...)
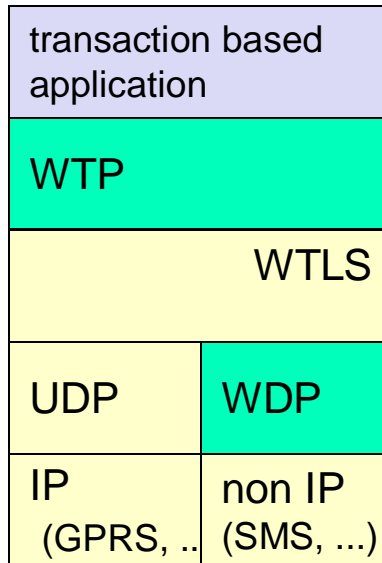
pure data application
with/without
additional security

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 (in some countries) – millions of users
    - Many use i-mode as PC replacement
    - For many this was the first Internet contact
    - Very simple to use, convenient
    - Technology9.6 kbit/s (enhancements with 28.8 kbit/s), packet oriented (PDC-P) Mobile
      ction oriented)

cHTML + tags
HTTP(S)
TL
PDC-P

TL	TCP
	IP
PDC-P	L2
	L1

TCP	TCP
IP	IP
L2	L2
L1	L1

cHTML + tags
HTTP(S)
TCP
IP
L2
L1

-

Email example: i-mode push with SMS

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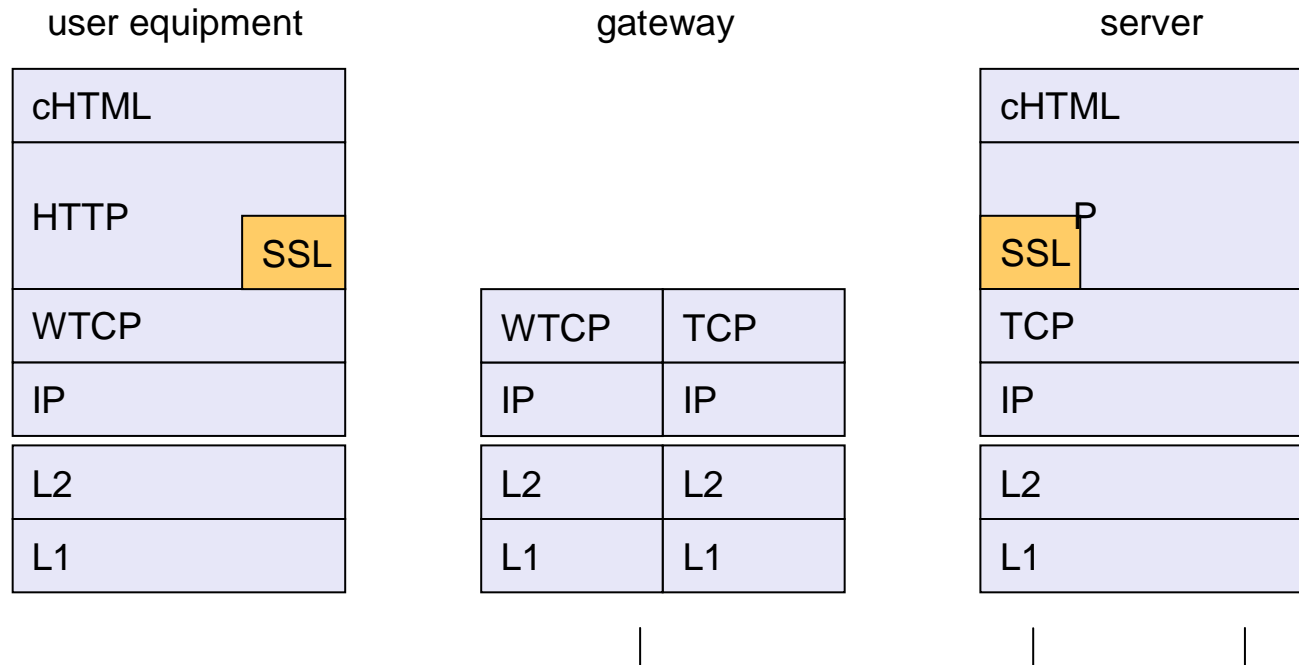
Popular misconception:
WAP was a failure, i-mode is different and a success – wrong from a technology point of view, right from a business point of view…

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!)
- not successful in e.g. Germany (stopped in 2008)

i-mode protocol stack based on WAP 2.0

user equipment

cHTML
HTTP
WTCP
IP
L2
L1

SSL

gateway

WTCP	TCP
IP	IP
L2	L2
L1	L1

server

cHTML
P
TCP
IP
L2
L1

SSL

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

i-mode – technical requirements

| Functions	Descriptions	Status	Requirement
WEB Access	Portal Site / Internet Access	M	i-mode HTML (cHTML+tags)
E-mail	Internet e-mail and inter-terminal email	M	HTTP 1.1
Security	End-End security	O	SSL (Version 2, 3), TLS 1
Java	Java application made available	O	Compatible i-mode JAVA
Ringing tone download	Ringing melody download	M	SMF based
Image download	Stand-by screen download	M	GIF (O: JPEG)
Voice call notification during i-mode session	Voice termination notified and responded during i-mode communications	M	3GPP standard system
Content charge billing	Per content charge billed to user	M	Specifications depend on each operator's billing system
Third party payment collection	Content charge collection on behalf of Content Provider	M	Specifications depend on each operator's billing system
Reverse billing	Packet usage charges can be billed to third party	O	Specifications depend on each operator's billing system
Subscriber ID transmission	Hashed subscriber ID from the operator's portal to the CP transmission on each content access	M	The ID generation algorithm should be determined by each operator and has to be secret
Number of characters per e-mail	Number of characters (byte) per e-mail	M	To be defined by operators (e.g. 500 byte, 1K byte, 10K byte)
Character code set supported	Character code set supported by browser and used to develop content	M	To be defined by operators
User Agent	Browser specifications to be notified	M	HTTP 1.1
i-mode button	Dedicated button	O	Hard or soft key

i- mode – very first examples I

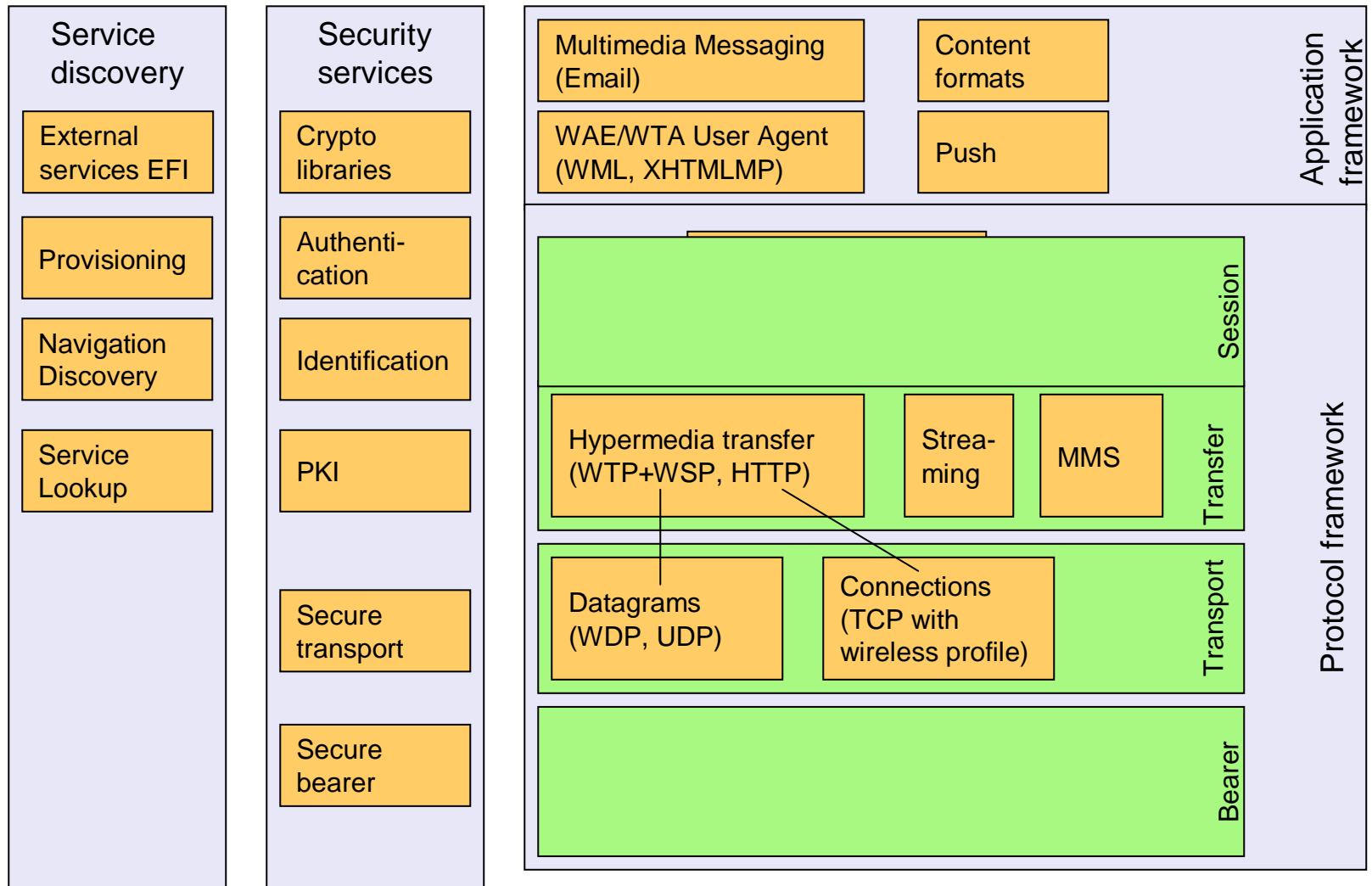# i-mode – very first examples II
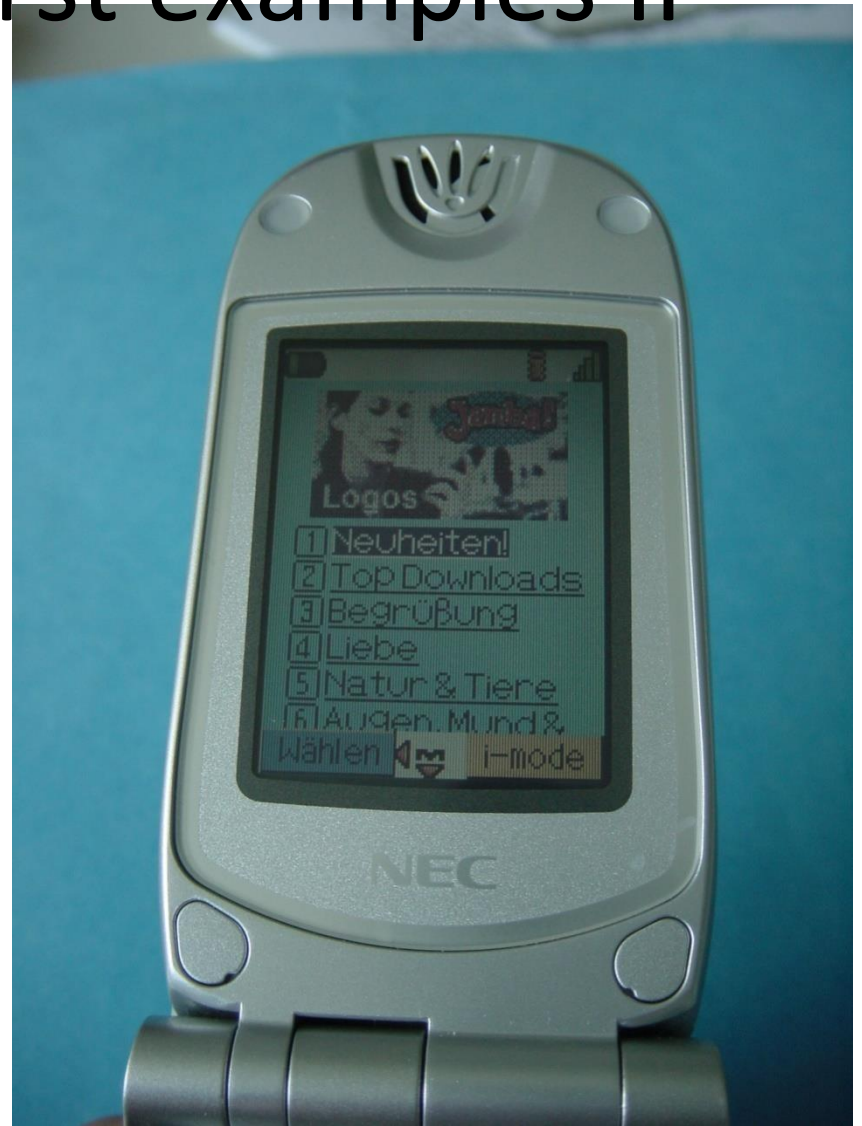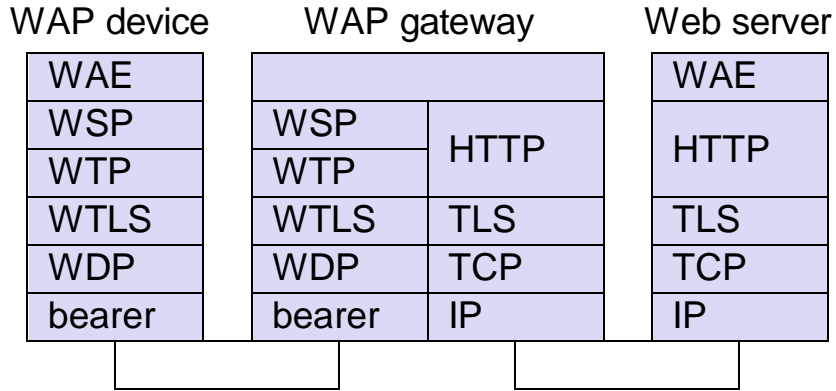
i-mode – very first examples III
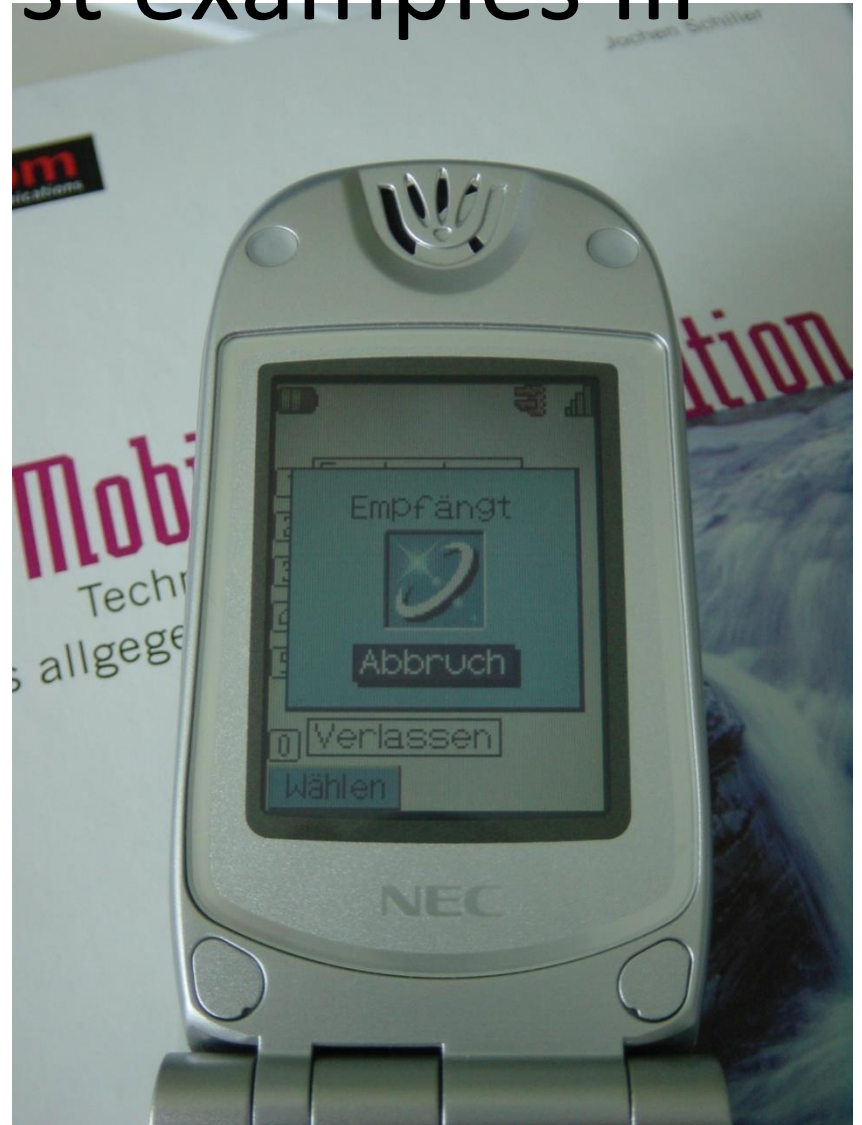
WAP 2.0 (July 2001)

- 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

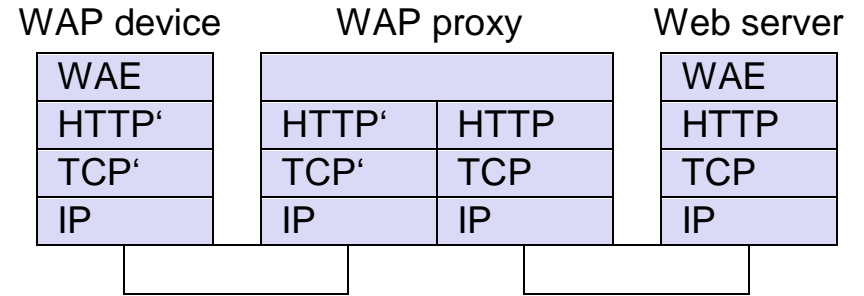- Goal: integration of WWW, Internet, WAP, i-mode

WAP 2.0 architecture

Service discovery

- External services EFI
- Provisioning
- Navigation Discovery
- Service Lookup

Security services

- Crypto libraries
- Authentication
- Identification
- PKI
- Secure transport
- Secure bearer

Application framework

- Multimedia Messaging (Email)
- WAE/WTA User Agent (WML, XHTMLMP)
- Content formats
- Push

Protocol framework

- Session
- Transfer
  - Hypermedia transfer (WTP+WSP, HTTP)
  - Streaming
  - MMS
- Transport
  - Datagrams (WDP, UDP)
  - Connections (TCP with wireless profile)
- Bearer

WAP 2.0 example protocol stacks

WAP device

| WAE |
| WSP |
| WTP |
| WTLS |
| WDP |
| bearer |

WAP gateway

WSP	HTTP
WTP	
WTLS	TLS
WDP	TCP
bearer	IP

Web server

| WAE |
| HTTP |
| TLS |
| TCP |
| IP |

WAP 1.x Server/Gateway/Client

WAP device

| WAE |
| HTTP' |
| TCP' |
| IP |

WAP proxy

HTTP'	HTTP
TCP'	TCP
IP	IP

Web server

| WAE |
| HTTP |
| TCP |
| IP |

WAP HTTP Proxy with profiled TCP and HTTP

WAP device

| WAE |
| HTTP |
| TLS |
| TCP' |
| IP |

WAP proxy

TCP'	TCP
IP	IP

Web server

| WAE |
| HTTP |
| TLS |
| TCP |
| IP |
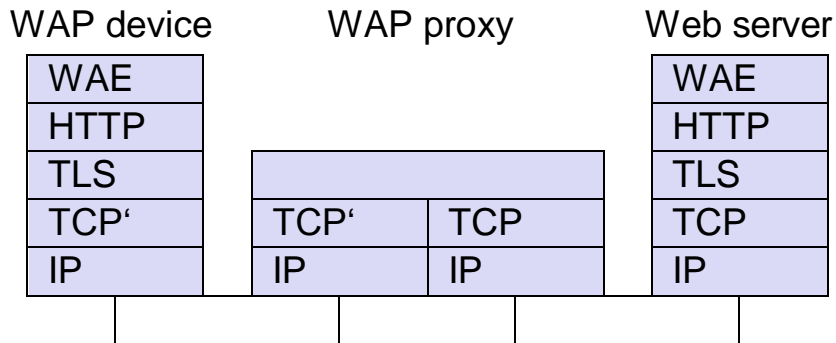
WAP Proxy with TLS tunneling

WAP device

| WAE |
| HTTP |
| TCP |
| IP |

IP router

| |
| IP | IP |

Web server

| WAE |
| HTTP |
| TCP |
| IP |
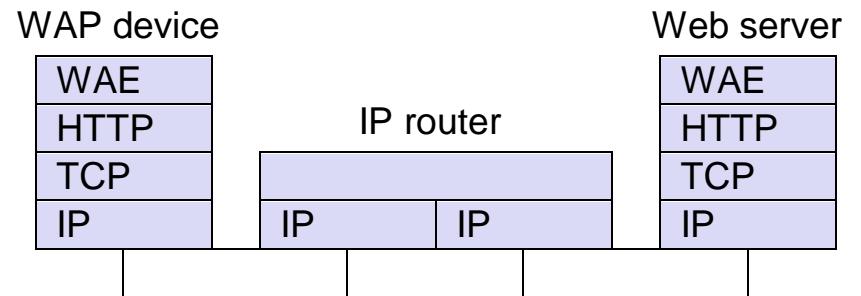
WAP direct access

Java 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

- Java ME (source released as: phone ME / was: J2ME)
  - Provides a uniform platform
  - Restricted functionality compared to standard java platform (JVM)

Applications of Java ME

- Example first 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, … -> MIDlets

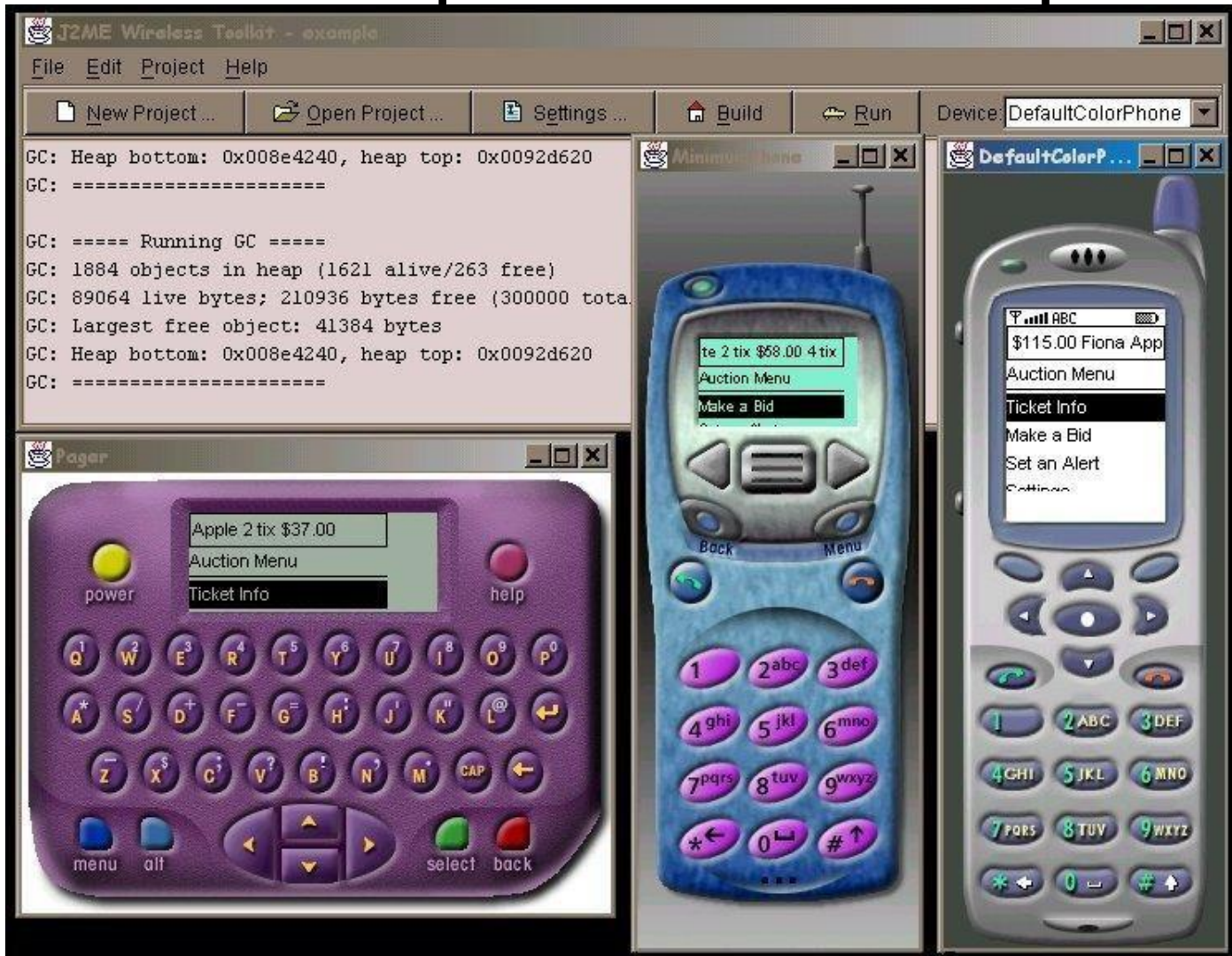| Applications |
| --- |

Profile (**MIDP**)

Configurations (CDC, **CLDC**)

Java Virtual Machine (JVM, **KVM**)

Operating system (EPOC, Palm, WinCE)

Hardware (SH4, ARM, 68k, …)

Hardware independent development

Summary Java ME

- 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

Other mobile application platforms

- Microsoft .NET Compact Framework
  - run-time environment plus class library with focus on mobile devices –
  light-weight version of .NET
  - support of many programming languages (C#, Python, Ruby, C++,
  Haskell, …)
  - typically in connection with Windows CE

- Qualcomm BREW (Binary Runtime Environment for Wireless)
  - run-time environment with main focus on games
  - certified applications only

- Google Android
  - OS + middleware + applications for mobile devices
  - Java/Linux based, open source

THANK YOU

This content is taken from the text books and reference books prescribed in the syllabus.