WIRELESS APPLICATION PROTOCOL (WAP)

Dr. A. Beulah

AP/CSE

Introduction

- 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 + …)

Introduction

- A protocol suite should enable global wireless communication across different wireless network technologies, e.g., GSM, CDPD, UMTS etc.
- **Interoperable**
  - allowing terminals and software from different vendors to communicate with networks from different providers;
- **Scalable**
  - protocols and services should scale with customer needs and number of customers;
- **Efficient**
  - provision of QoS suited to the characteristics of the wireless and mobile networks;
- **Reliable**
  - provision of a consistent and predictable platform for deploying services;
- **Secure**
  - preservation of the integrity of user data, protection of devices and services from security problems.

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 Device Characteristics

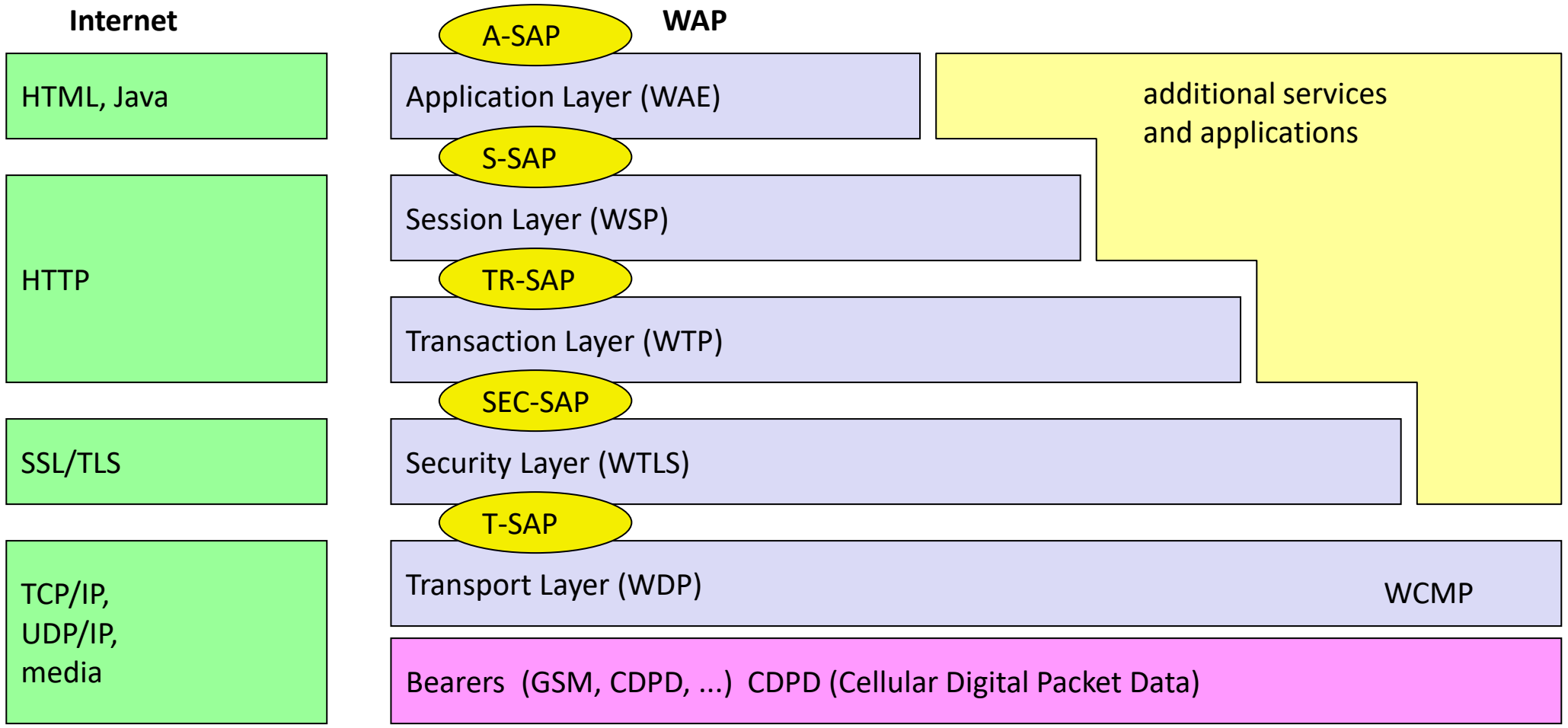
- WAP device has limited CPU power, RAM, and ROM.
- Exact numbers are not given in the specification

- The telephone has “Yes” and “No” buttons and arrow keys for navigating from one screen to another.

Accessing a Web Site

- Here's what happens when accessing a Web site using a WAP-enabled device:
  - Turn on the device and open the minibrowser.
  - The device sends out a radio signal, searching for service.
  - A connection is made with the service provider.
  - Select a Web site to view.
  - A request is sent to a gateway server using WAP.
  - The gateway server retrieves the information via HTTP from the Web site.
  - The gateway server encodes the HTTP data as WML.
  - The WML-encoded data is sent to the device.
  - The wireless Internet version of the Web page selected is visible now.

WAP Architecture



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

WAP Architecture

- **WAE** (Wireless Application Environment)
  - The Wireless Application Environment holds the tools that wireless Internet content developers use.
  - These include WML and WMLScript, which is a scripting language used in conjunction with WML. It functions much like JavaScript.
- **WSP** (Wireless Session Protocol)
  - The Wireless Session Protocol determines whether a session between the device and the network will be **connection-oriented** or **connectionless**.
  - In a connection-oriented session, data is passed both ways between the device and the network; WSP then sends the packet to the Wireless Transaction Protocol layer
  - If the session is connectionless, commonly used when information is being broadcast or **streamed** from the network to the device, then WSP redirects the packet to the Wireless Datagram Protocol layer

WAP Architecture

- **WTP** (Wireless Transaction Protocol)
  - The **WTP** offers a lightweight transaction service at the **transaction SAP (TR-SAP)** **(**transaction request- Service Access Point**)**
  - It also determines how to classify each transaction request: Reliable two-way, Reliable one-way, Unreliable one-way
- **WTLS** (Wireless Transport Layer Security)
  - The **WTLS offers its service at the security SAP (SEC-SAP).**
  - **WTLS** is based on the transport layer security (TLS, formerly SSL, secure sockets layer)
  - WTLS has been optimized for use in wireless networks
  - It can offer data integrity, privacy, authentication, and (some) denial-of-service protection.

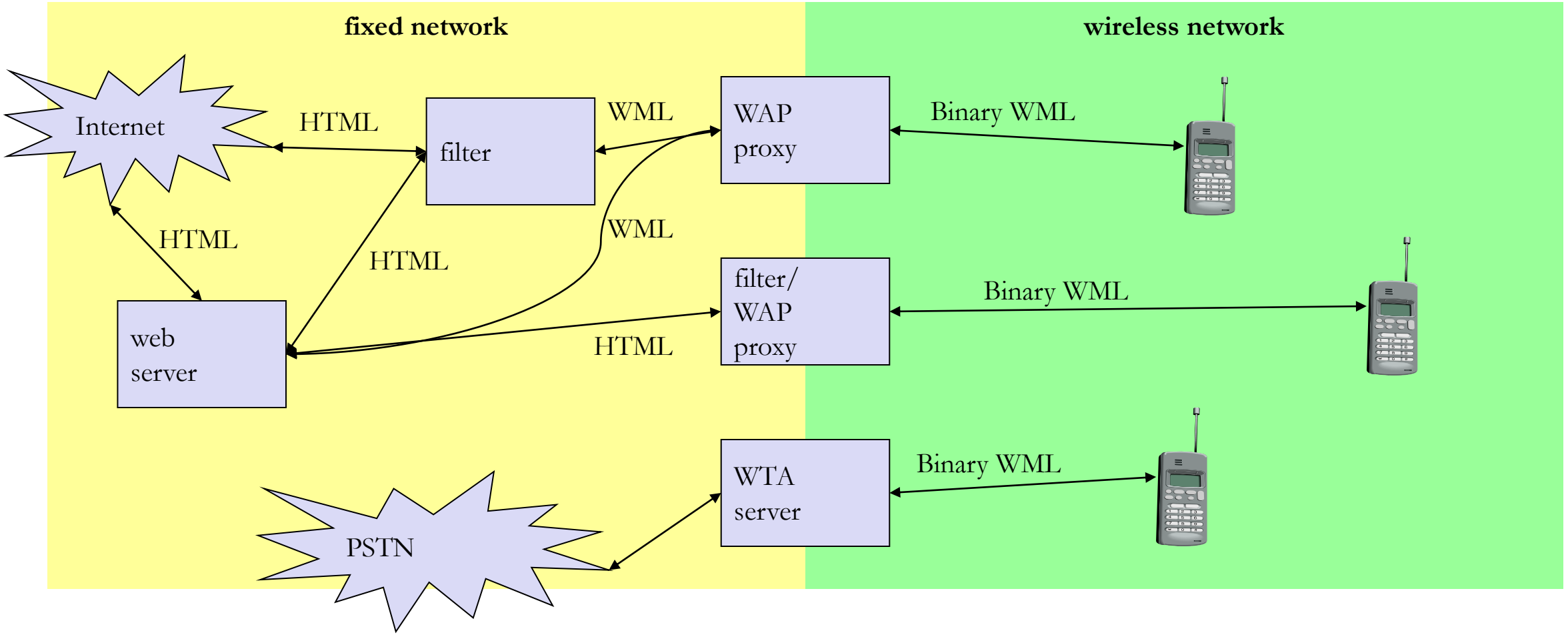
WAP Architecture

- **WDP** (Wireless Datagram Protocol)
  - works in conjunction with the network **bearer** layer
  - WDP makes it easy to adapt WAP to a variety of bearers because all that needs to change is the information maintained at this level.
  - Communication is done transparently over one of the available bearer services.
  - The **transport layer service access point (T-SAP) is the** common interface to be used by higher layers independent of the underlying network

WAP Architecture

- **Network bearers**
  - Also called **bearers**, these can be any of the existing technologies that wireless providers use.
  - The basis for transmission of data is formed by different **bearer services.**
  - **WAP** does not specify bearer services, but uses existing data services and will integrate further services.
  - Examples are message services, such as:
    - short message service (SMS) of GSM,
    - circuit-switched data, such as high-speed circuit switched data (HSCSD) in GSM
    - packet switched data, such as general packet radio service (GPRS) in GSM

WAP - network elements



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, ...)

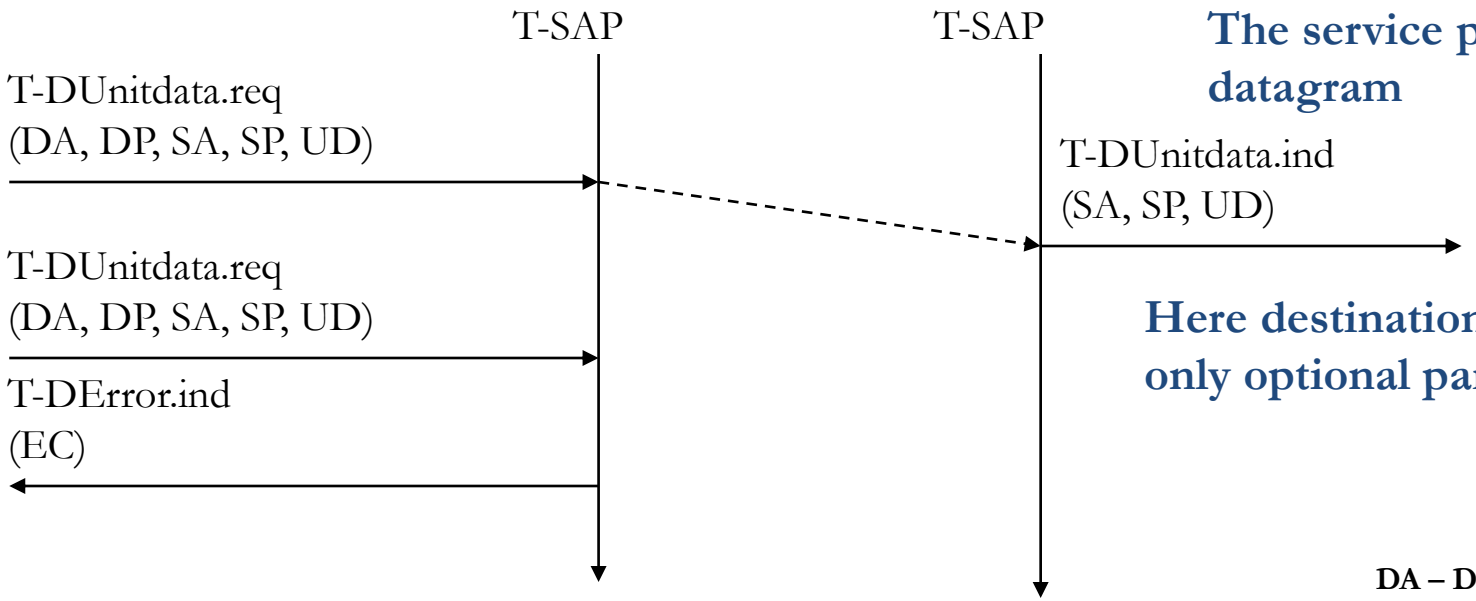
- Goals of WDP
  - create a worldwide interoperable transport system with the help of WDP adapted to the different underlying technologies
  - transmission services such as SMS, GPRS in GSM might change, new services can replace the old ones

WDP - Wireless Datagram Protocol

- WDP offers source and destination port numbers used for multiplexing and demultiplexing of data respectively.
- The service primitive to send a datagram **is TDUnitdata.req** with the destination address (DA), destination port (DP), Source address (SA), source port (SP), and user data (UD) as mandatory parameters
- Destination and source address are unique addresses for the receiver and sender of the user data.
- These could be MSISDNs (i.e., a telephone number), IP addresses, or any other unique identifiers.
- The **T-DUnitdata.ind** service primitive indicates the reception of data. Here destination address and port are only optional parameters.
- If a higher layer requests a service the WDP cannot fulfill, this error is indicated with the **T-DError.ind** service primitive.
- An error code (EC) is returned indicating the reason for the error to the higher layer.

WDP - Service Primitives

The service primitive to send a datagram



The service primitive to receive a datagram

Here destination address and port are only optional parameters

DA – Destination Address
DP – Destination Port
SA – Source Address
SP – Source Port
UD – User Data
EC – Error Code

WDP - Wireless Datagram Protocol

- Additionally, WCMP (Wireless Control Message Protocol) is used for control/error report (similar to ICMP in the TCP/IP protocol suite)
- Typical WCMP messages are
  - destination unreachable (route, port, address unreachable),
  - parameter problem (errors in the packet header),
  - message too big,
  - reassembly failure,
  - echo request/reply.

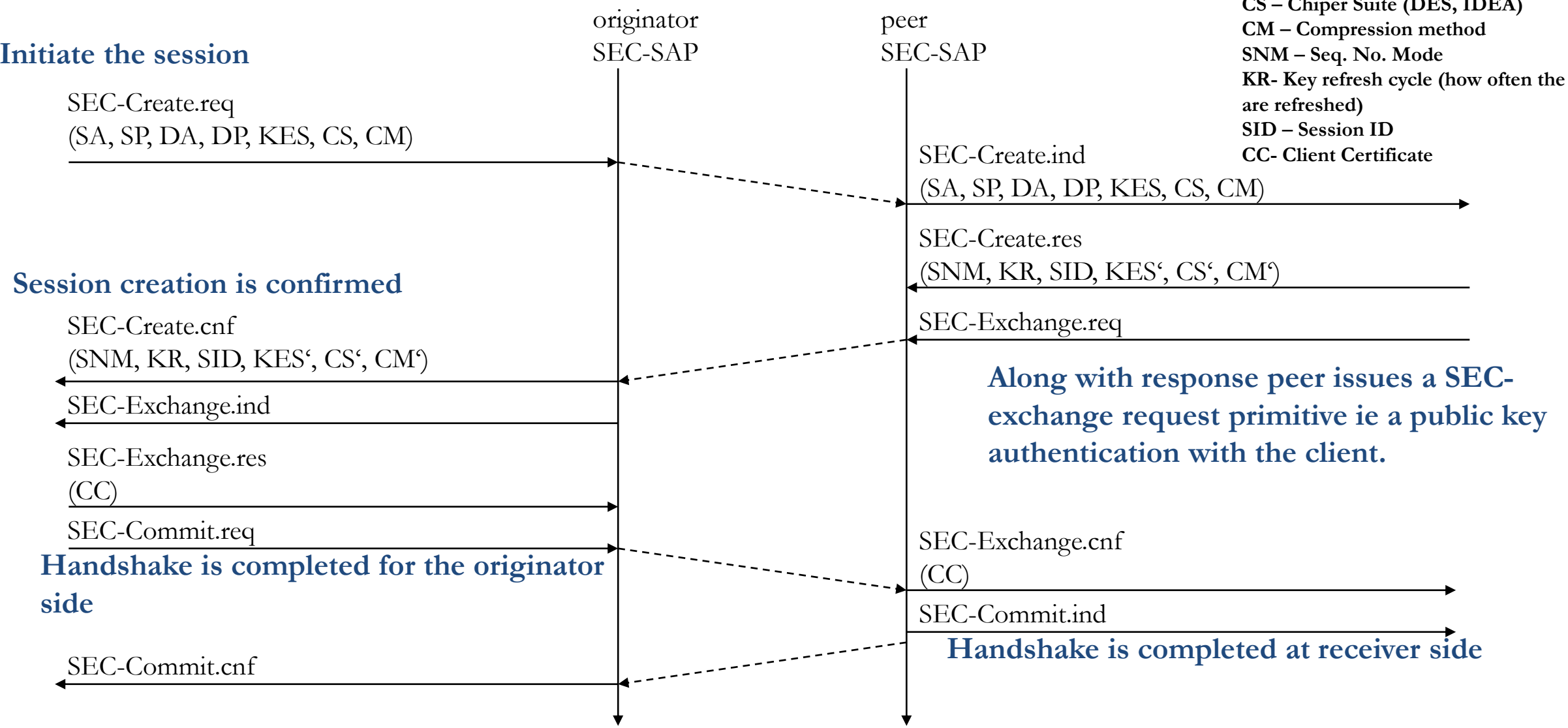
WTLS (Wireless Transport Layer Security)

- 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
  - Before data can be exchanged via WTLS, a secure session has to be established.

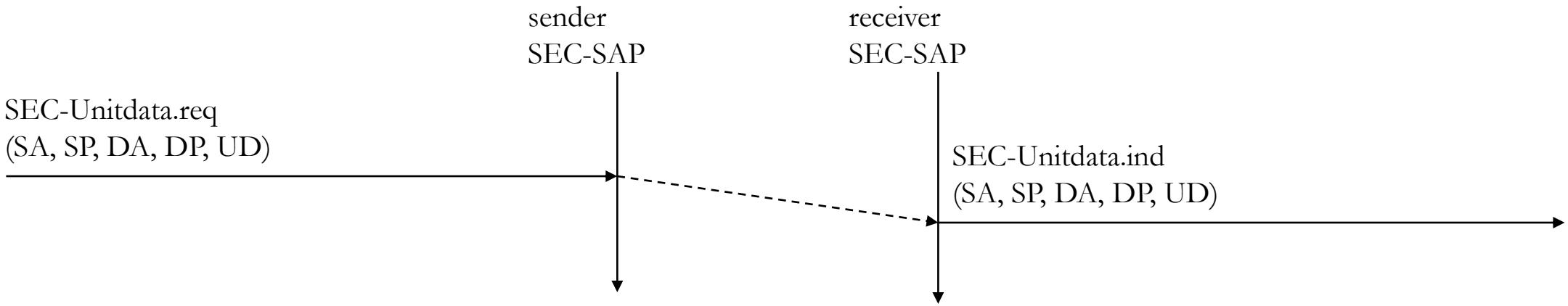
WTLS (Wireless Transport Layer Security)

KES- Key exchange suite (RSA, DH, ECC)
CS – Chiper Suite (DES, IDEA)
CM – Compression method
SNM – Seq. No. Mode
KR- Key refresh cycle (how often the keys are refreshed)
SID – Session ID
CC- Client Certificate



WTLS establishing a secure session

WTLS datagram transfer



Wireless Transaction Protocol (WTP)

- Goals
  - Different transaction services, offloads applications
    - Advantages to higher layers: reliability over datagram services, improved efficiency over connection oriented services.
  - Support of different communication scenarios
    - class 0: unreliable message transfer (unreliable one way)
    - class 1: reliable message transfer without result message (reliable one way)
    - class 2: reliable message transfer with exactly one reliable result message (reliable two way)
  - 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)

Wireless Transaction Protocol (WTP)

- 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

Wireless Transaction Protocol (WTP)

- 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)
- PDUs exchanged between two WTP entities
  - invoke PDU, ack PDU, and result PDU

Wireless Transaction Protocol (WTP)

WTP Class 0 transaction

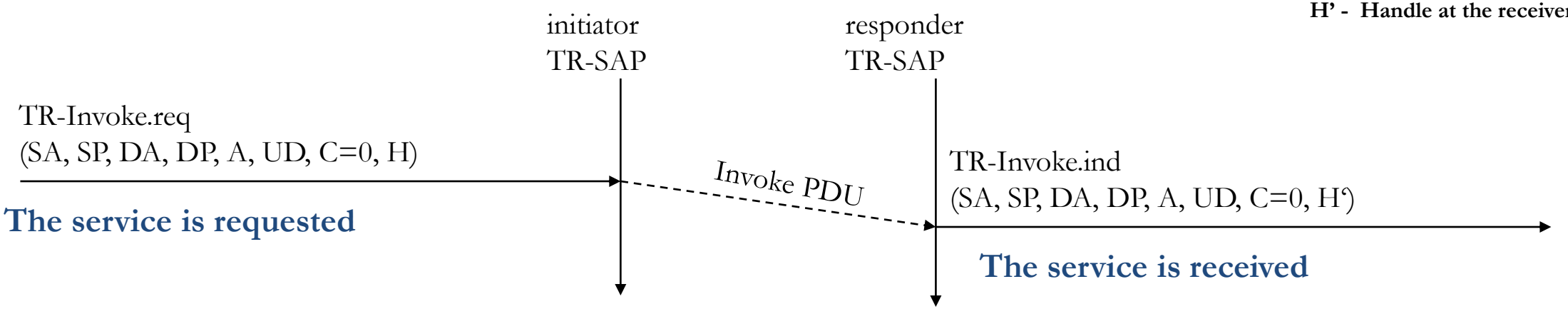
A – flag – If the responder should generate an ACK or not

C – Class

H – Handle – Unique identifier for the transaction

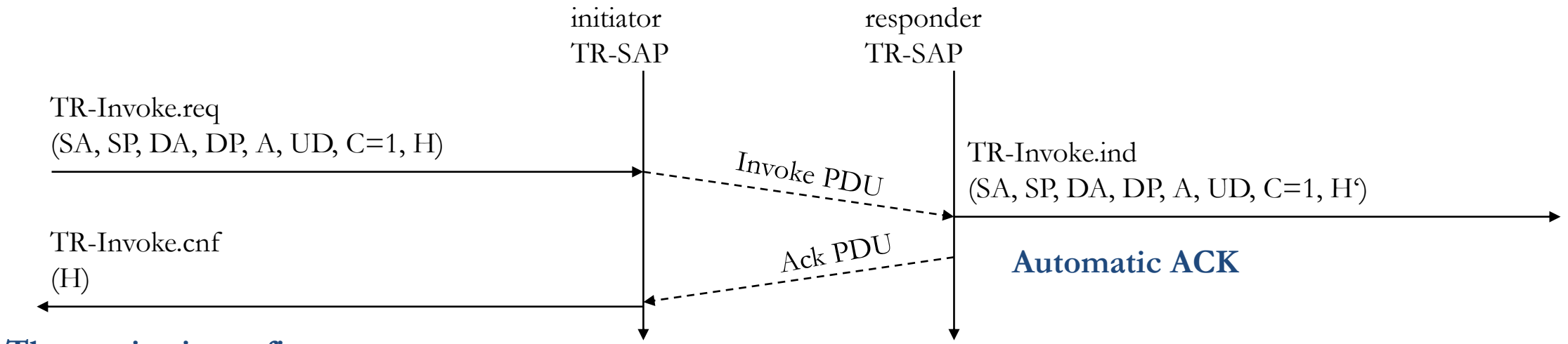
H' - Handle at the receiver side

Unreliable service without result message

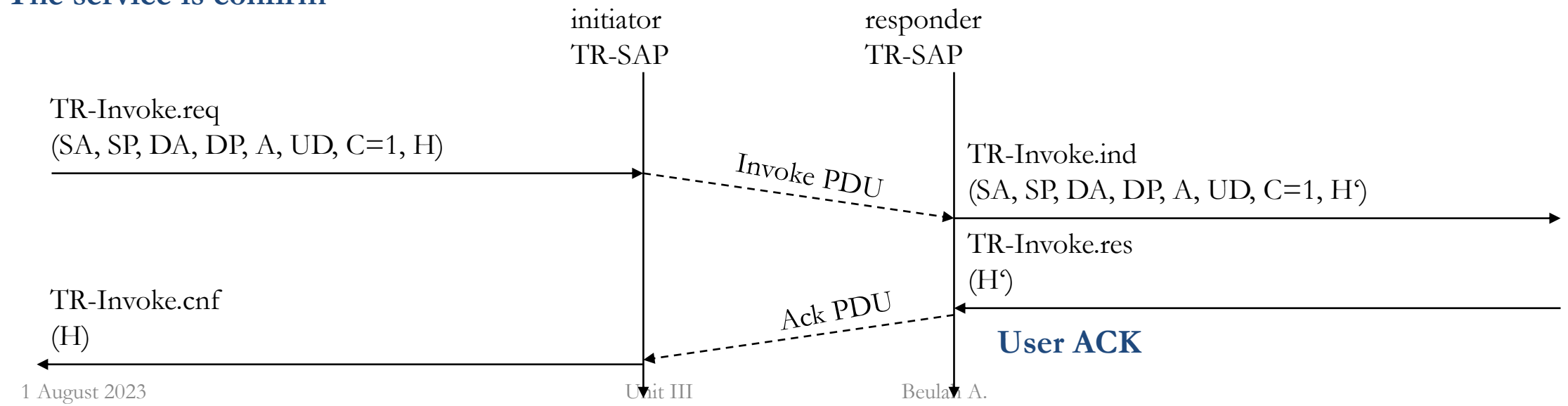


Wireless Transaction Protocol (WTP)

WTP Class 1 transaction, Automatic ack & user ack

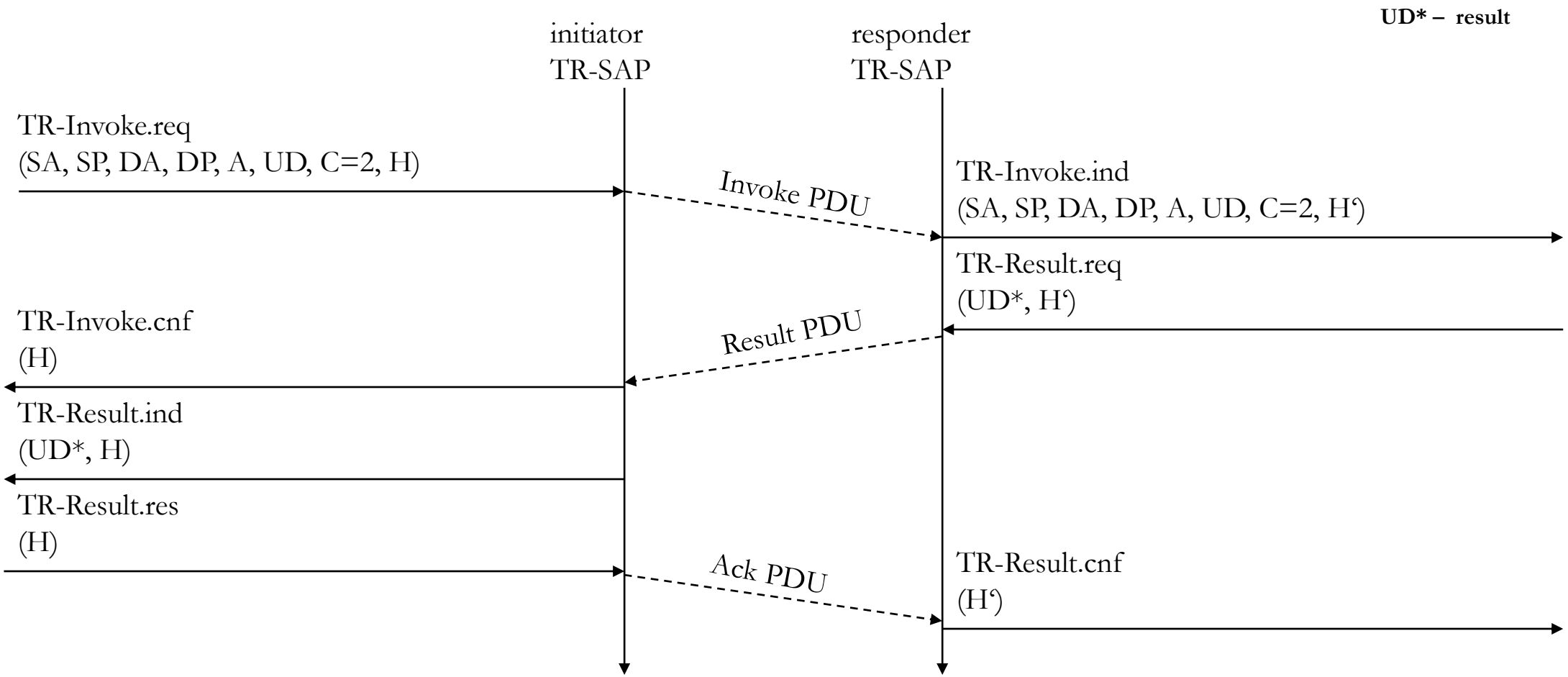


The service is confirm



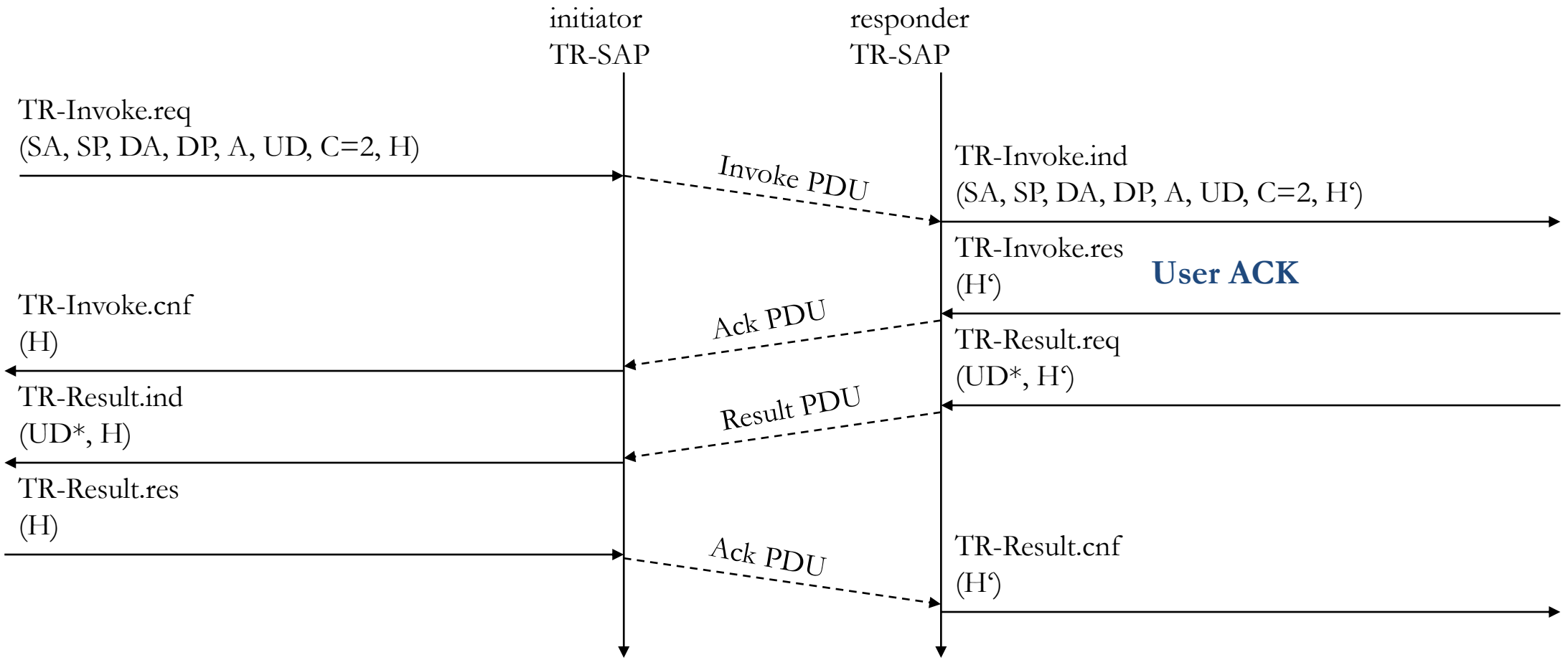
Wireless Transaction Protocol (WTP)

WTP Class 2 transaction, no user ack



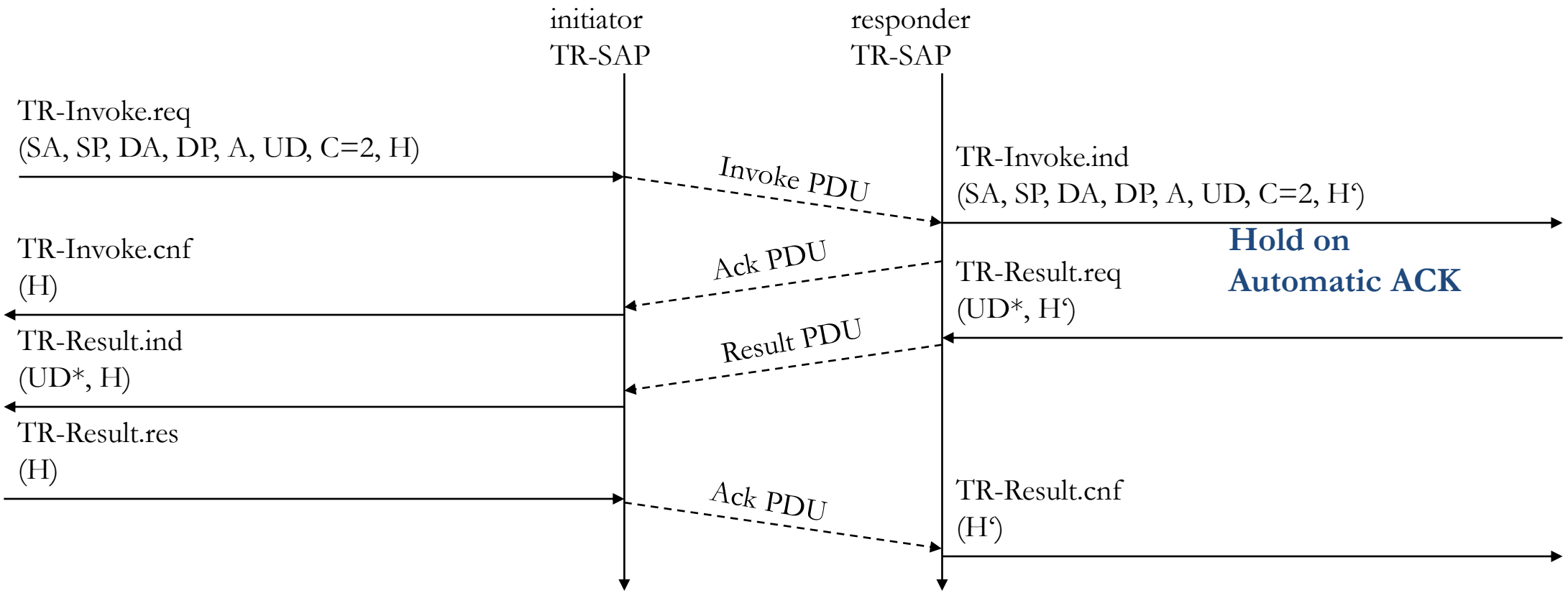
Wireless Transaction Protocol (WTP)

WTP Class 2 transaction, user ack



Wireless Transaction Protocol (WTP)

WTP Class 2 transaction, hold on, no user ack



Hold on → to prevent a retransmission of the invoke PDU (If no result is sent before the timer expire)

Wireless Session Protocol (WSP)

- WSP → general purpose session protocol
- WSP/B → protocols and services suitable for browsing
- 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

Wireless Session Protocol (WSP)

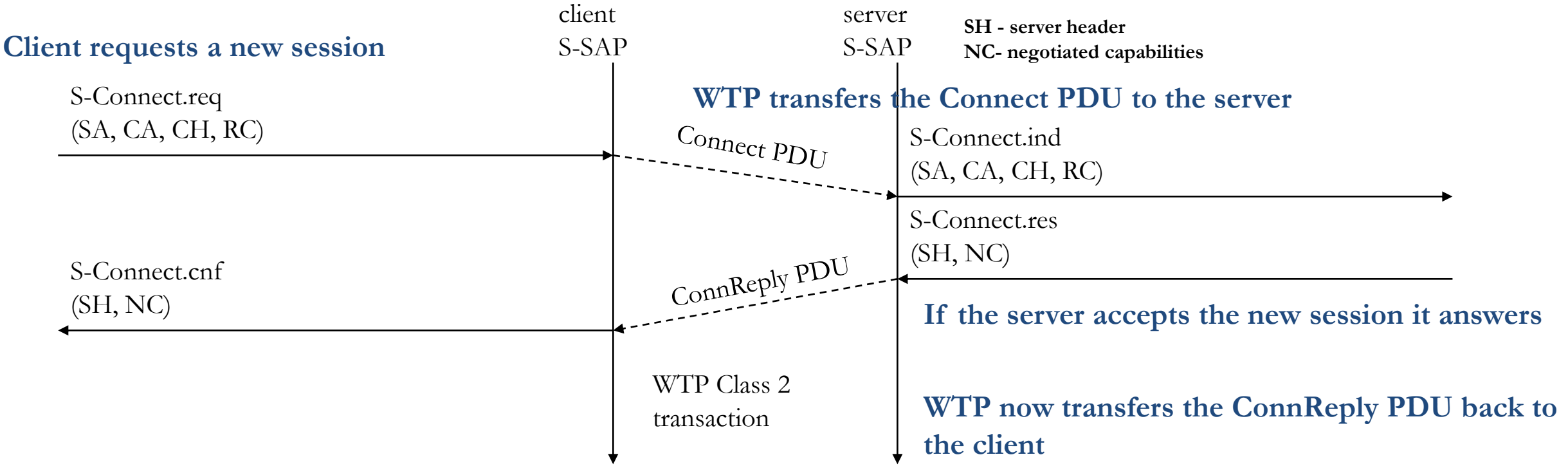
- WSP/B (browsing) over WTP
  - Uses 3 services classes of WTP
  - Class 0 is used for:
    - Unconfirmed push
    - Session resume, and
    - Session management.
  - Class 1 is used for:
    - Confirmed push
  - Class 2 is used for:
    - Confirmed push's
      - method invocation,
      - Session resume
      - Session management

WSP/B session establishment using WTP class 2

SA server address
CA - client address
CH - optional client header
User to user information compatible with HTTP header

RC- requested capabilities
S&C SDU size, outstanding request, protocol options

SH - server header
NC- negotiated capabilities



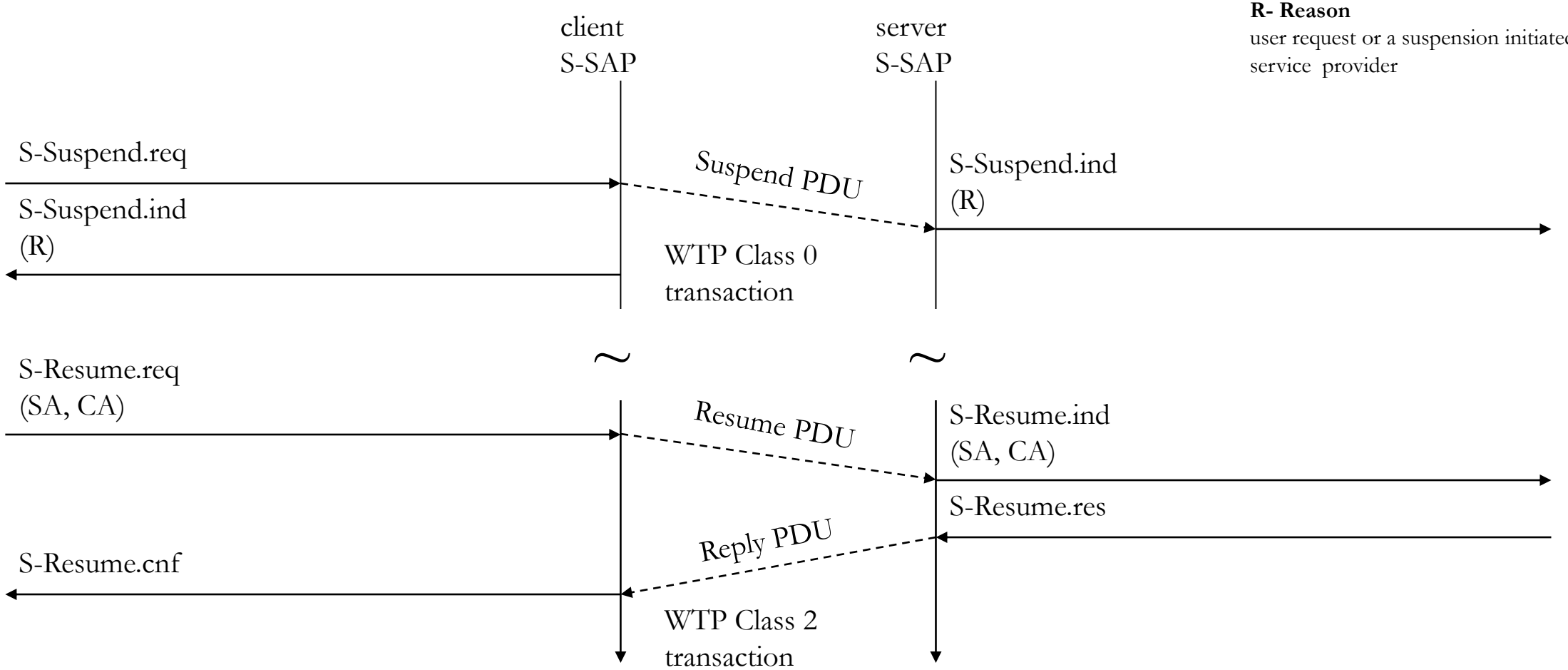
Wireless Session Protocol (WSP)

WSP/B session suspend/resume

- A client notices that it will soon be unavailable,
  - Ex: The bearer network will be unavailable due to roaming to another network
  - or the user switches off the device,
- So, the client can suspend the session.
- Session suspension will automatically abort all data transmission and freeze the current state of the session on the client and server side.

Wireless Session Protocol (WSP)

WSP/B session suspend/resume

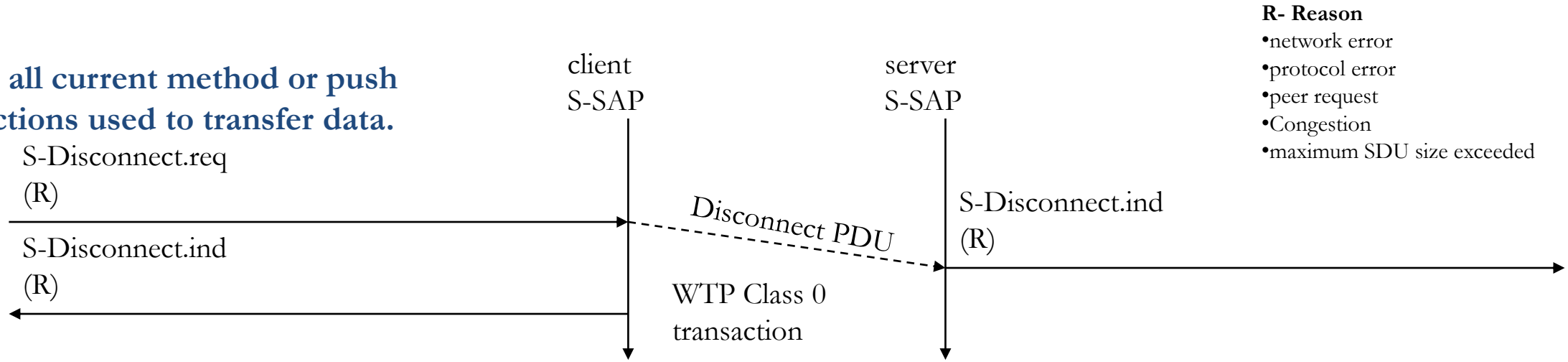


R- Reason
user request or a suspension initiated by the service provider

Wireless Session Protocol (WSP)

WSP/B session termination

Aborts all current method or push transactions used to transfer data.



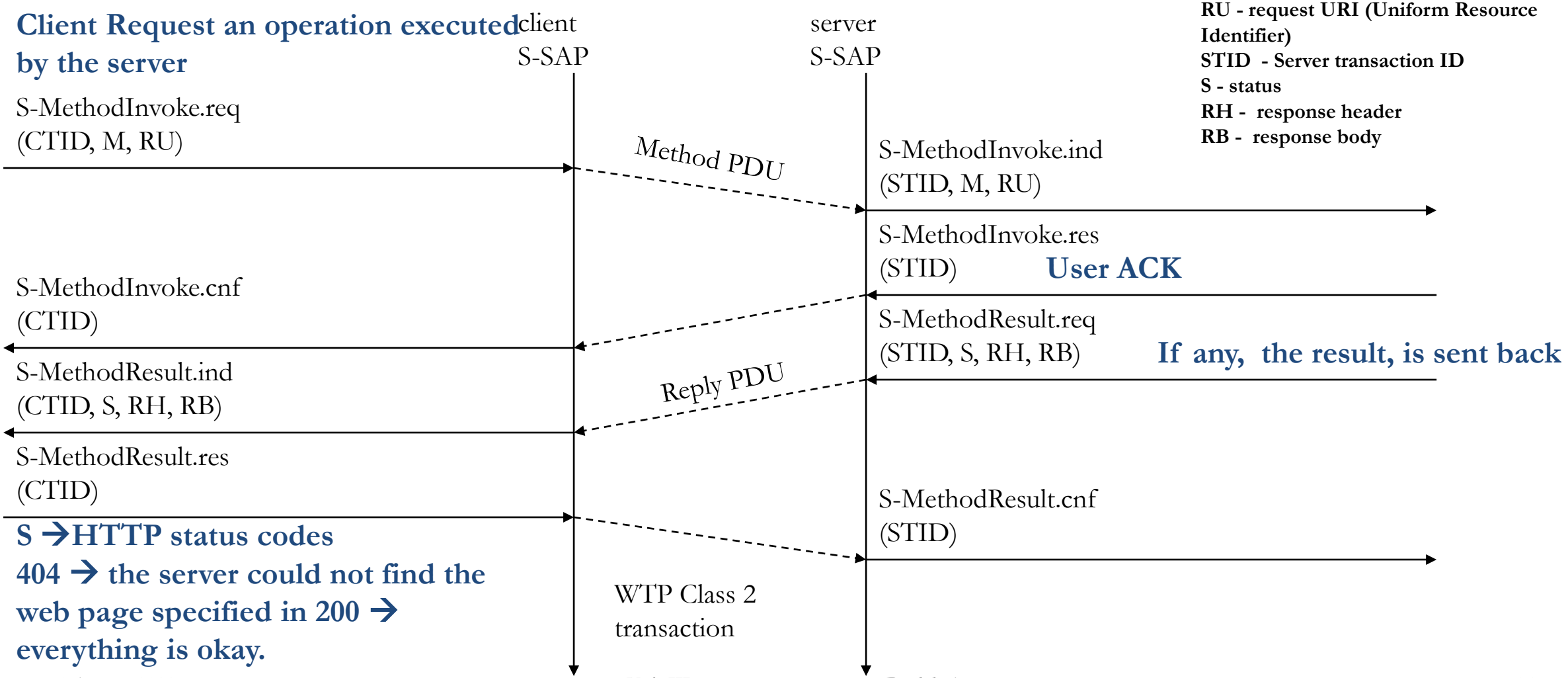
R- Reason
- network error
- protocol error
- peer request
- Congestion
- maximum SDU size exceeded

Wireless Session Protocol (WSP)

WSP/B method invoke

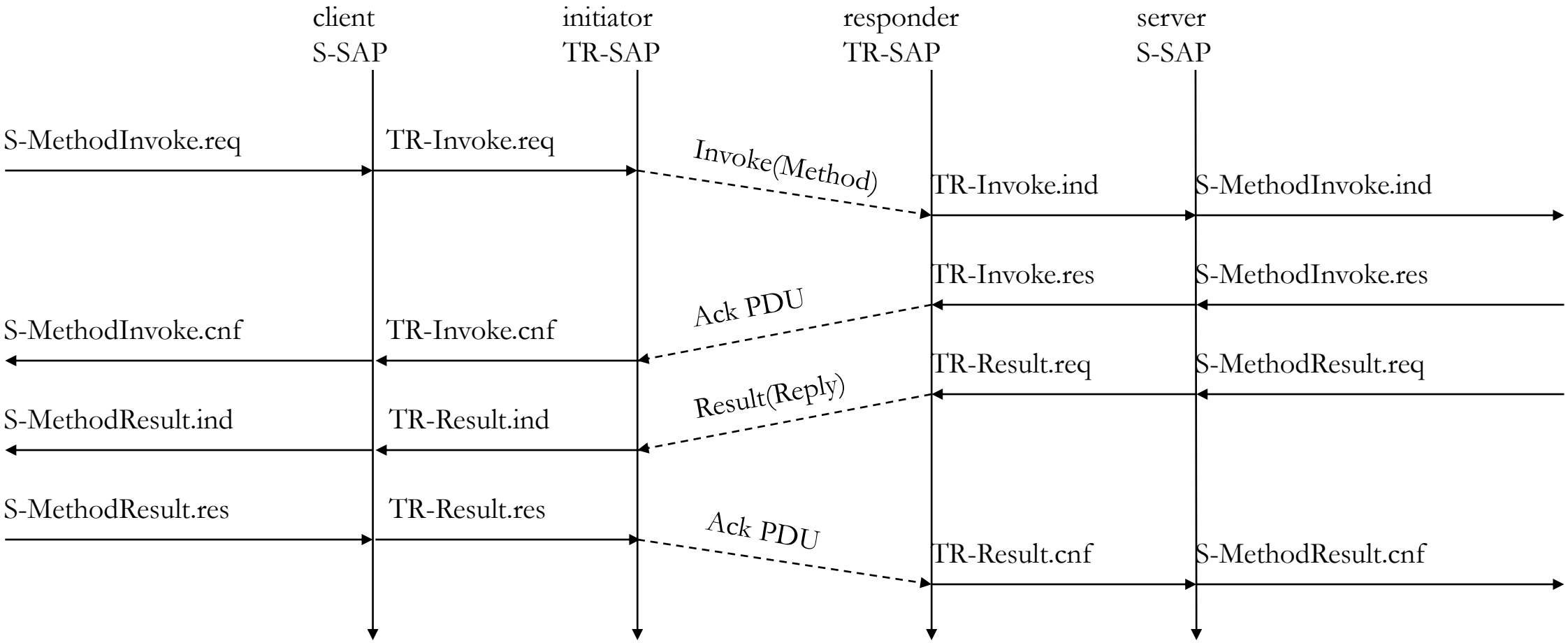
CTID - client transaction identifier (to distinguish between pending transactions)
M - method
RU - request URI (Uniform Resource Identifier)
STID - Server transaction ID
S - status
RH - response header
RB - response body

Client Request an operation executed by the server

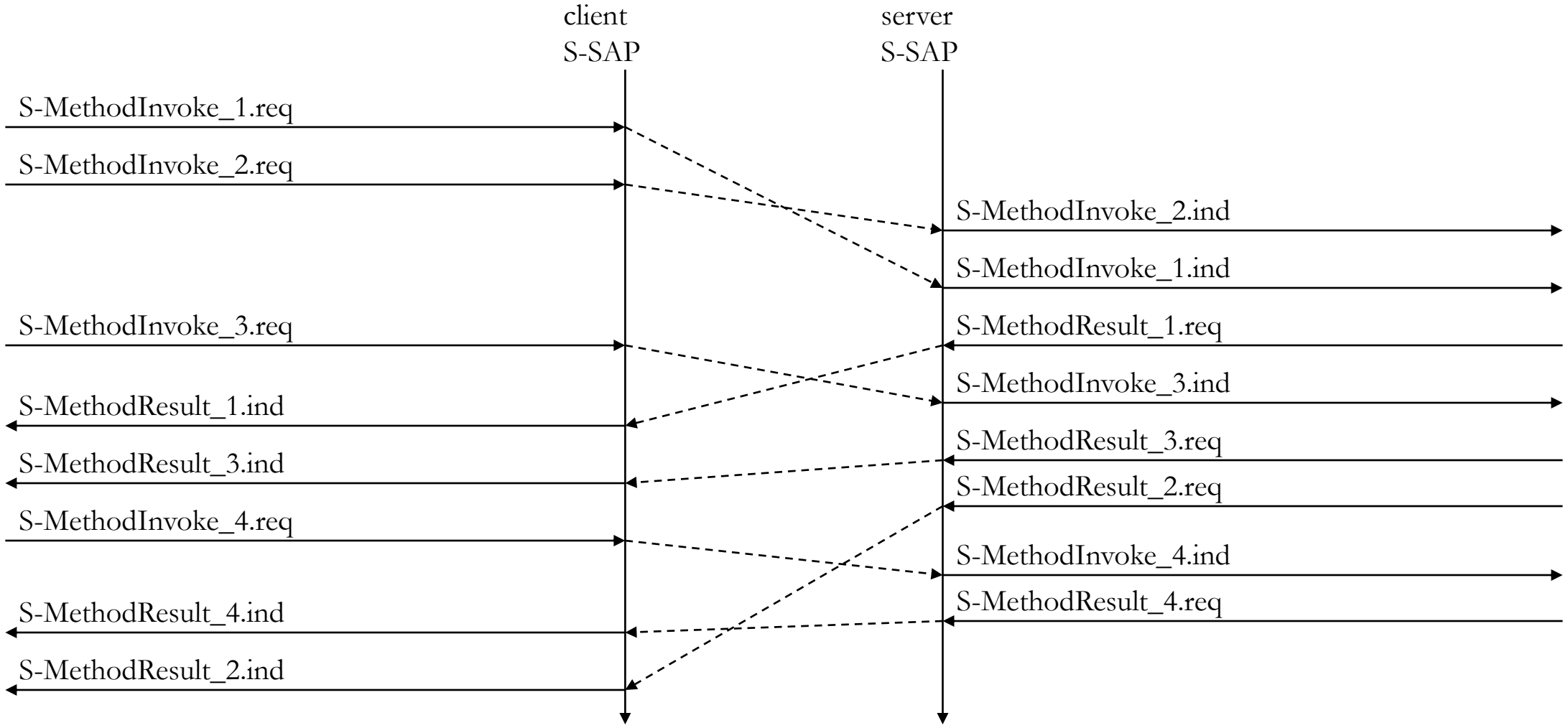


S → HTTP status codes
404 → the server could not find the web page specified in 200 → everything is okay.

WSP/B over WTP - method invocation

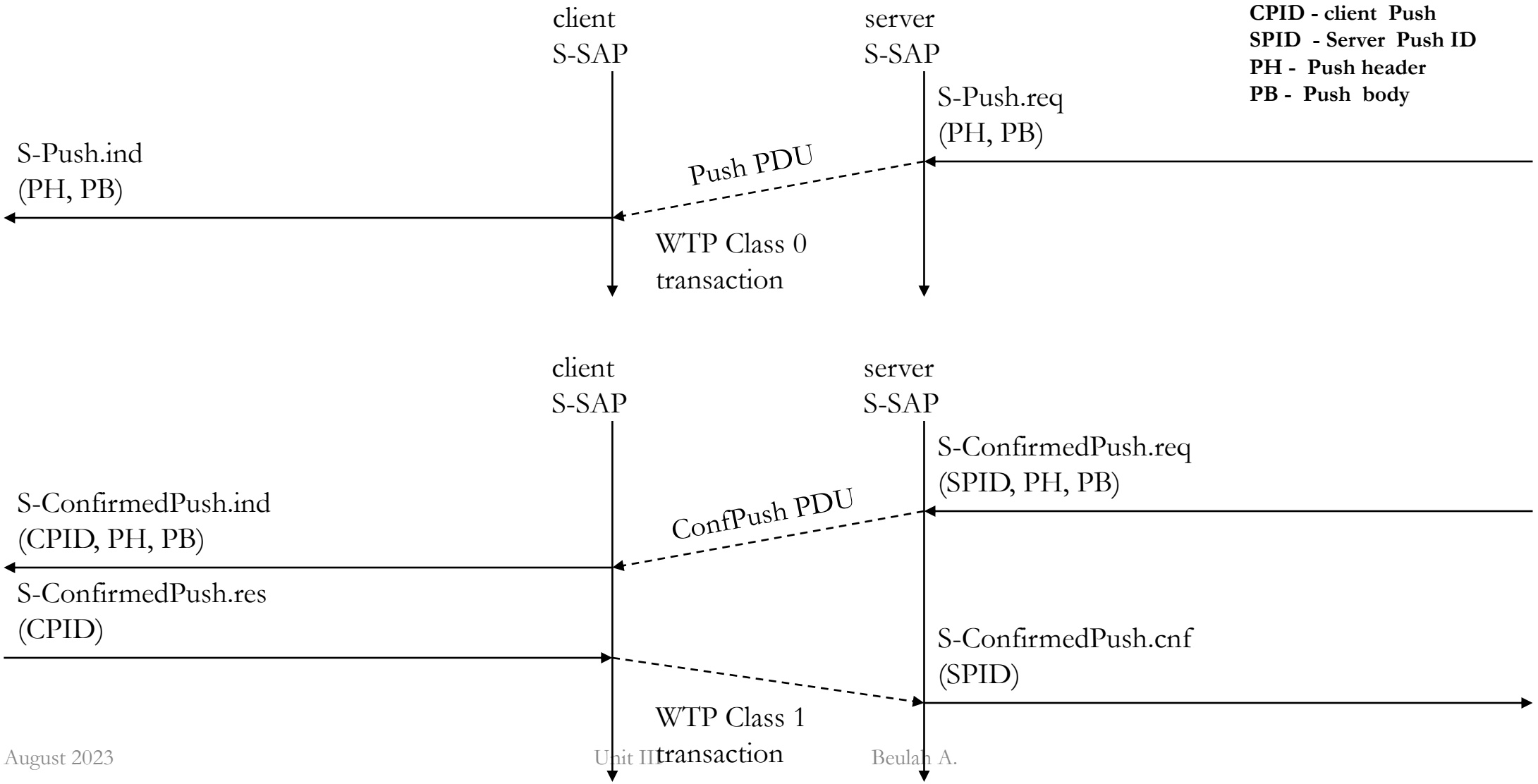


WSP/B over WTP - asynchronous, unordered requests



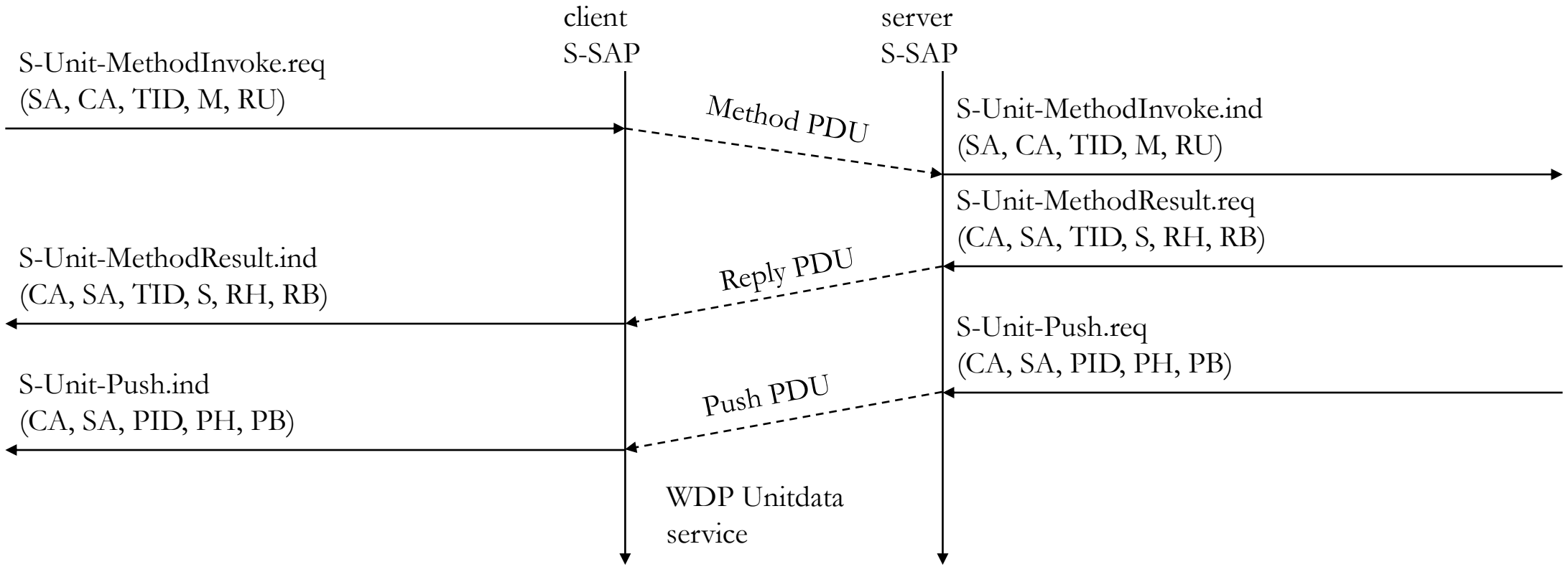
Wireless Session Protocol (WSP)

WSP/B - confirmed/non-confirmed push



Wireless Session Protocol (WSP)

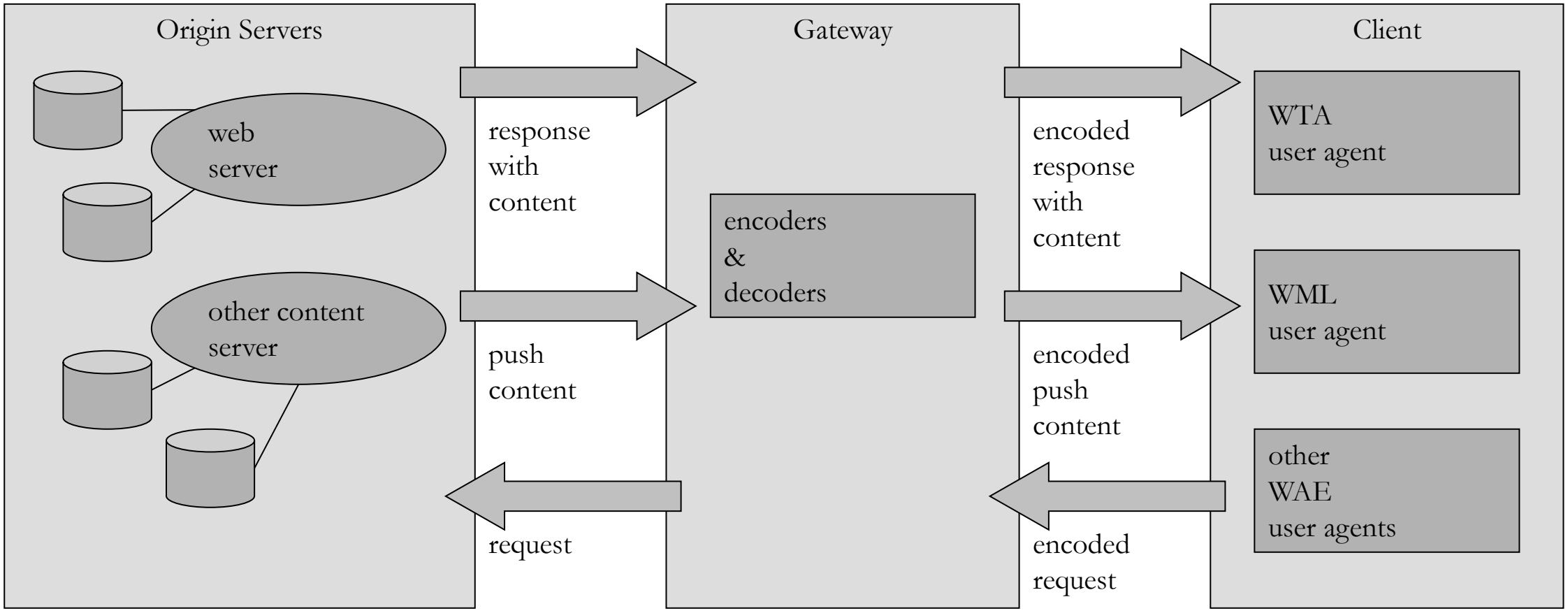
WSP/B over WDP



Wireless Application Environment (WAE)

- 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, ...

Wireless Application Environment (WAE)



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

Wireless Markup Language (WML)

```
<?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>
```

Wireless Markup Language (WML)

```
<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

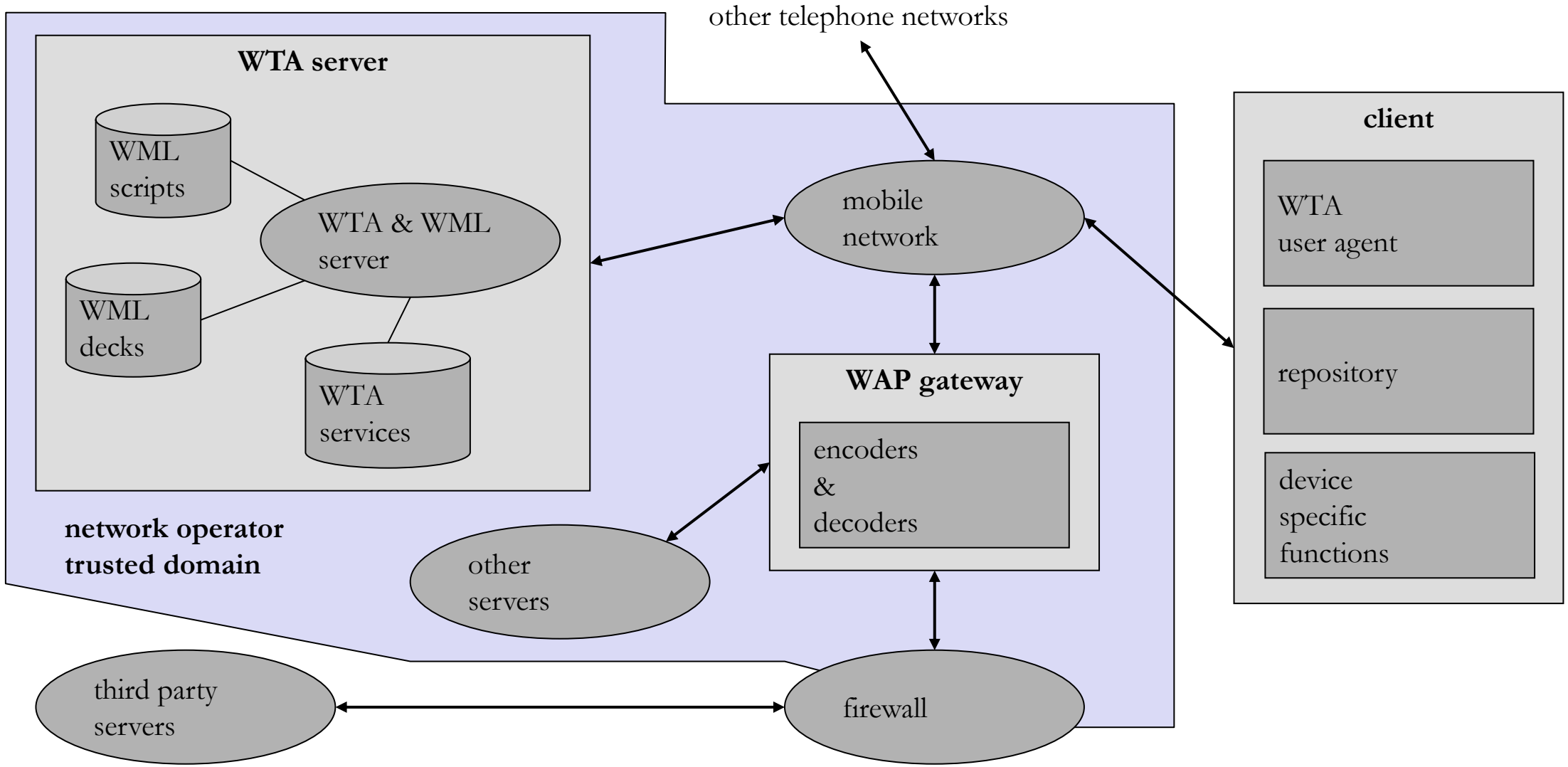
WMLScript

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

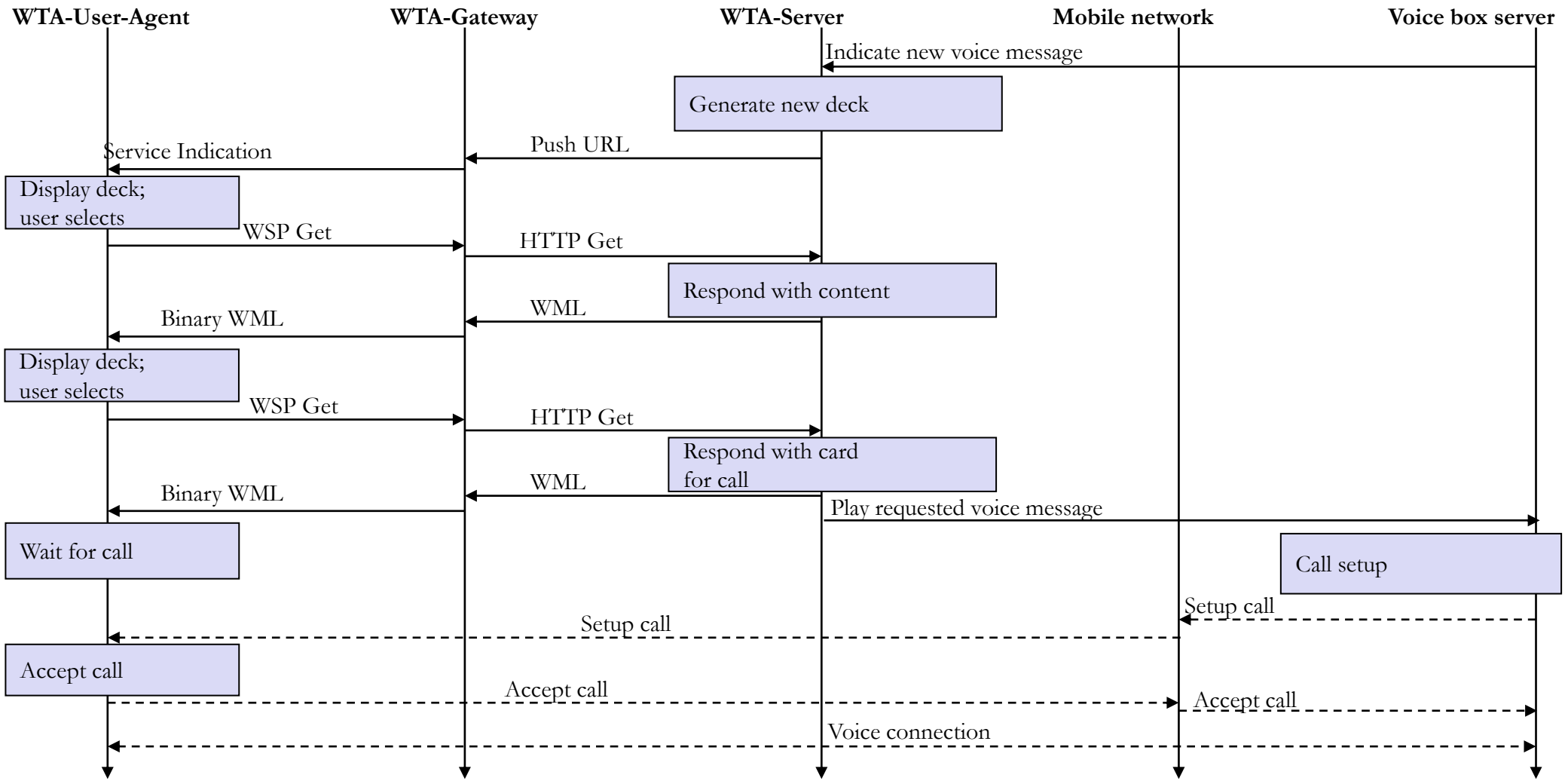
Wireless Telephony Application (WTA)

- 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



Voice box example



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

```
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

```
<?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>
```

Test your Knowledge

- What are the primary goals of the WAP forum efforts and how they reflected in the initial WAP protocol architecture?

Summary

- Wireless application protocol (version 1.x)
  - 10.3.1 Architecture
  - Wireless datagram protocol
  - Wireless transport layer security
  - Wireless transaction protocol
  - Wireless session protocol
  - Wireless application environment
  - Wireless markup language
  - WMLScript
  - Wireless telephony application

References

Jochen H. Schller, “Mobile Communications”, Second Edition, Pearson Education, New Delhi, 2007.