Module 7

Application Layer

Contents

- WAP Model
- Mobile Location based services
- WAP Gateway
- WAP protocols
- WAP user agent profile
- Caching model
- Wireless bearers for WAP
- WML
- WMLScripts
- WTA.

Mobile Application Layer

- **Additional Components (as compared to traditional network):**
    - File Systems, databases, security, accounting and billing mechanisms etc.
  - Web browsing for mobile user
  - Problems with HTTP and HTML
  - Wireless Application Protocol 1.x- wireless and mobile web access
    - Commercially unsuccessful- fatal combination of web browsing with a connection oriented transport system
    - Markup and scripting language-WML and WMLScript
  - I-mode- commercially successful protocol for web browsing over cellular network
  - WAP 2.x

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 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 and mobility-Cont..

- **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

- **POSTing (i.e., sending to a server)**
  - can typically not be buffered, very problematic if currently disconnected

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

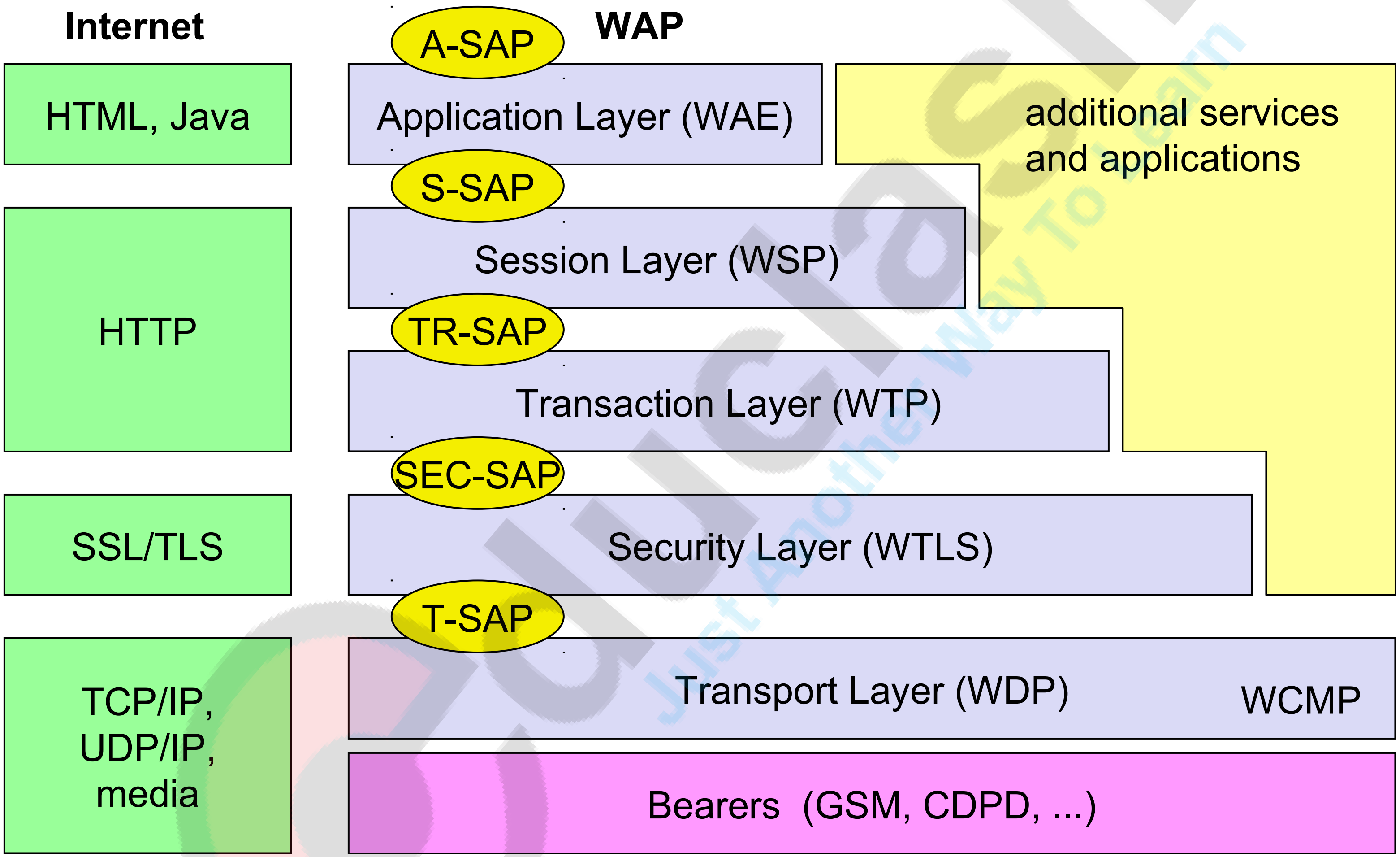
WAP - Wireless Application Protocol

- **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 etc).
- **Forum**
  - was: WAP Forum, co-founded by Ericsson, Motorola, Nokia, Unwired Planet.
  - now: Open Mobile Alliance (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

HTML, Java

HTTP

SSL/TLS

TCP/IP,
UDP/IP,
media

WAP

A-SAP

Application Layer (WAE)

S-SAP

Session Layer (WSP)

TR-SAP

Transaction Layer (WTP)

SEC-SAP

Security Layer (WTLS)

T-SAP

Transport Layer (WDP)

WCMP

Bearers (GSM, CDPD, ...)

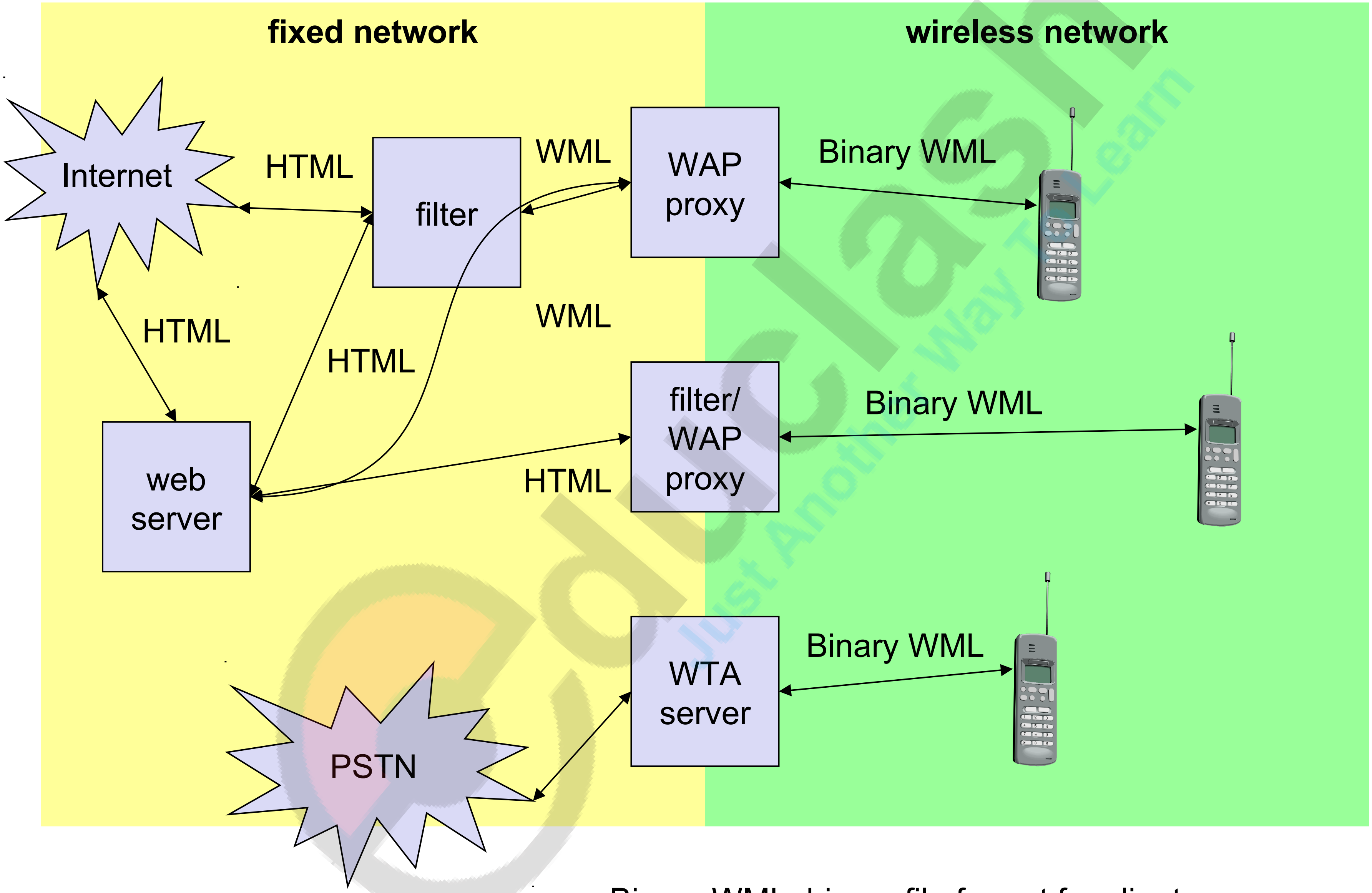
additional services
and applications

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

WAP 1.x - reference model and protocols – Cont..

- Existing bearer services
- Transport Layer- WDP- Wireless Datagram Protocol and WCMP- Wireless Control Message Protocol
- Security Layer- WTLS- Wireless Transport Layer Security- based on Secure Socket layer-SSL
- Transaction layer- WTP- Wireless Transaction Protocol- reliable and unreliable request and asynchronous transactions
- Session Layer- WSP- Wireless Session Protocol
  - 2 services: Connection oriented and connectionless over WDP
- Application Layer- WAE- Wireless Application Environment- integration of different www and mobile telephony applications

WAP - network elements



Binary WML: binary file format for clients

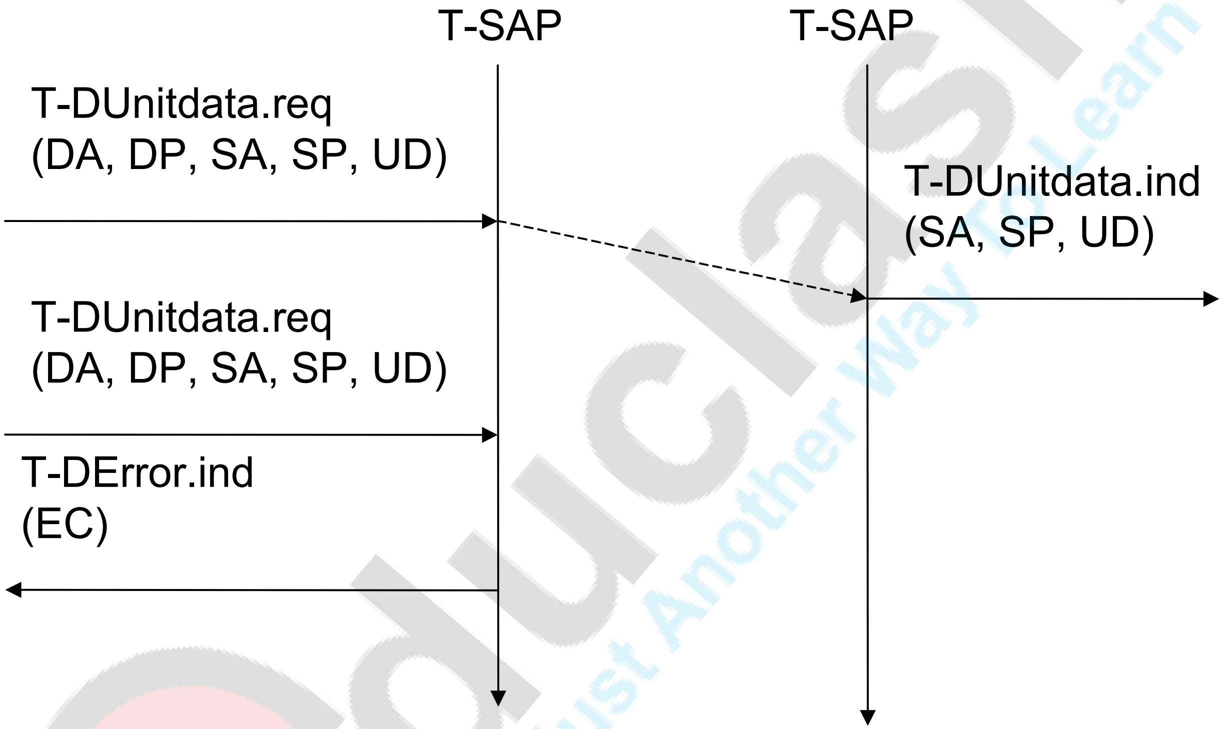
WDP - Wireless Datagram Protocol

- 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

- Additionally, **WCMP (wireless Control Message Protocol) is used for control/error report** (similar to ICMP in the TCP/IP protocol suite)

WDP - Service Primitives



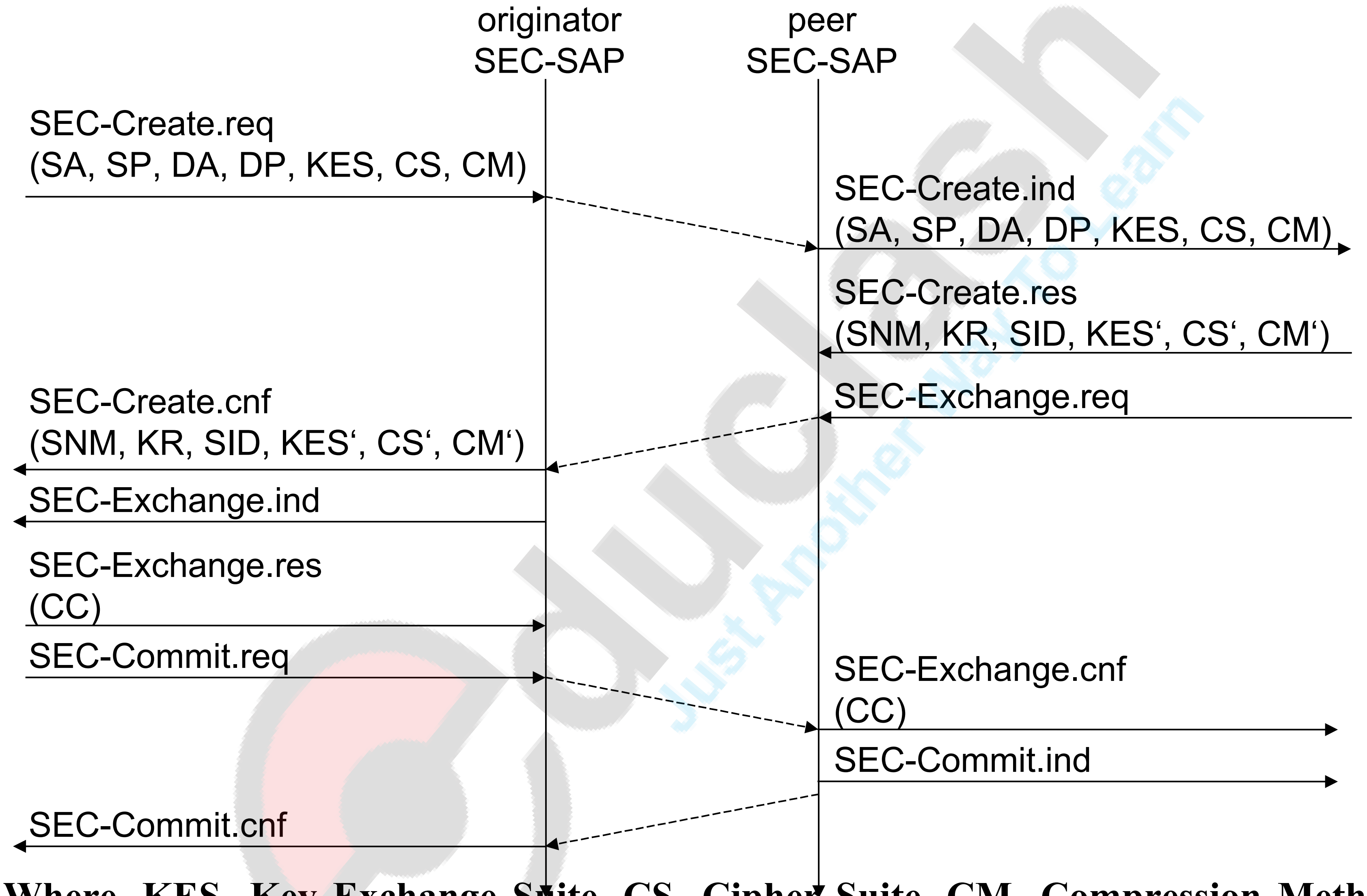
Where, DA- Destination Address, DP- Destination Port, SA- Source Address, SP-Source Port, UD- User Data

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 establishment, 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-Exchange.req

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

SEC-Exchange.ind

SEC-Exchange.res (CC)

SEC-Commit.req

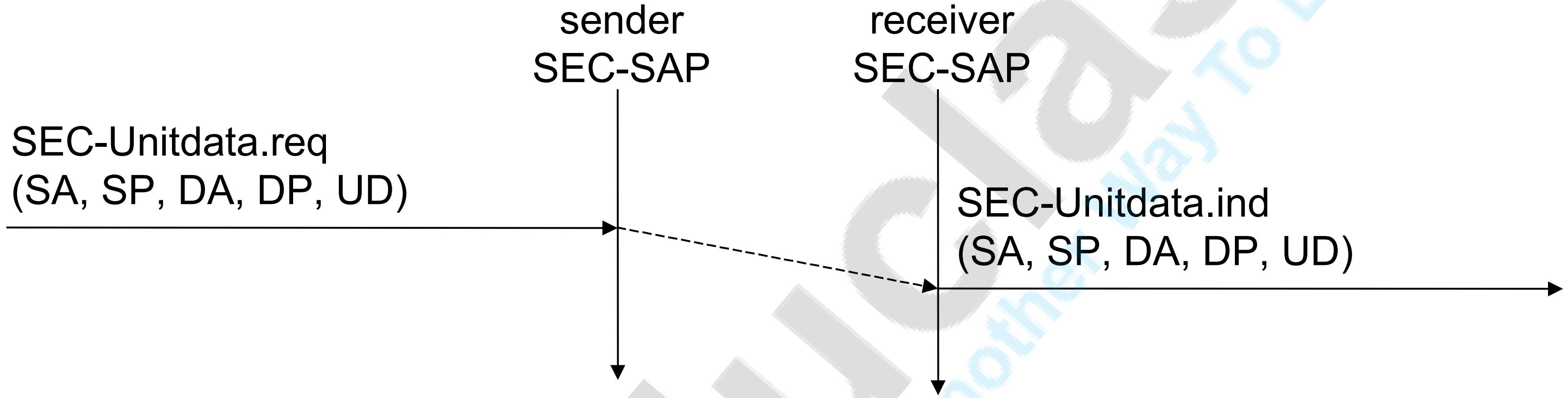
SEC-Exchange.cnf (CC)

SEC-Commit.ind

SEC-Commit.cnf

Where, KES- Key Exchange Suite, CS- Cipher Suite, CM- Compression Method, SNM- Sequence Number Mode, SID- Session Identifier, KR-Key refresh Cycle, CC- Client Certificate

SEC-Unitdata - transferring datagrams



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

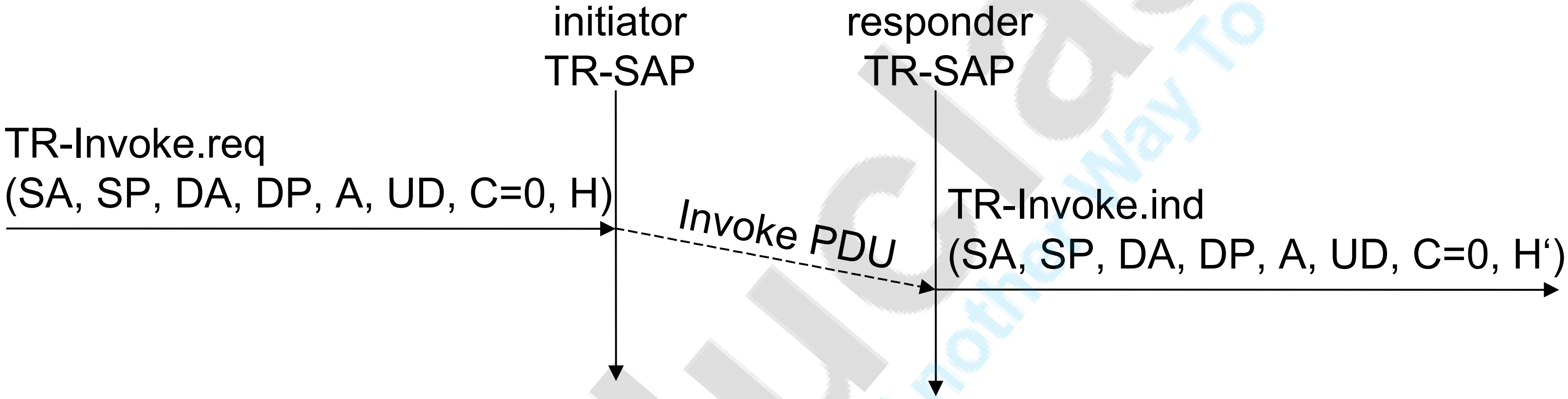
- Support of different communication scenarios
  - Class 0: unreliable message transfer
    - Example: push service
  - Class 1: reliable request
    - An invoke message is not followed by a result message
    - Example: reliable push service
  - Class 2: reliable request/response
    - An invoke message 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-Cont..

- 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)

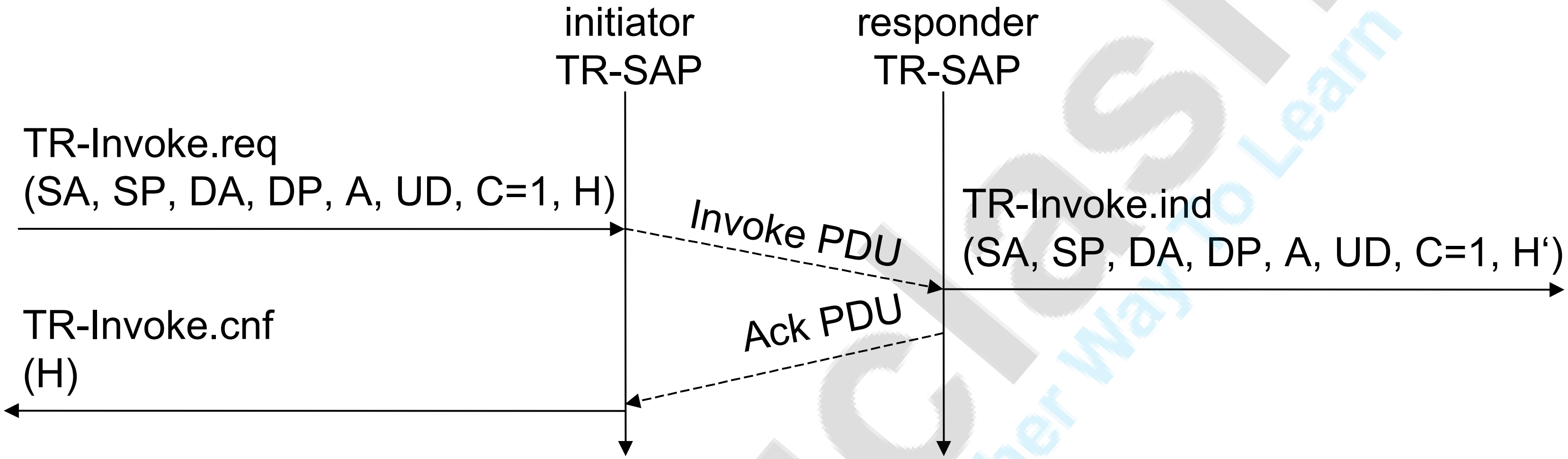
Invoke PDU

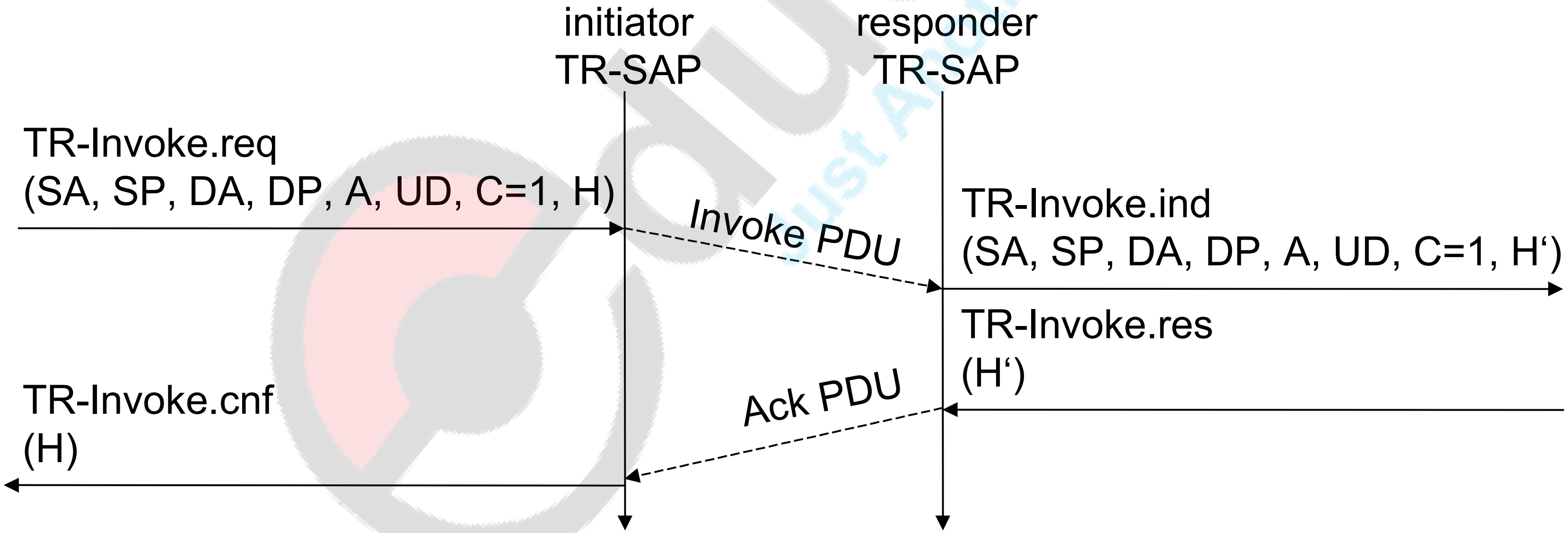
TR-Invoke.ind
(SA, SP, DA, DP, A, UD, C=0, H‘)

Where, C- Class type,

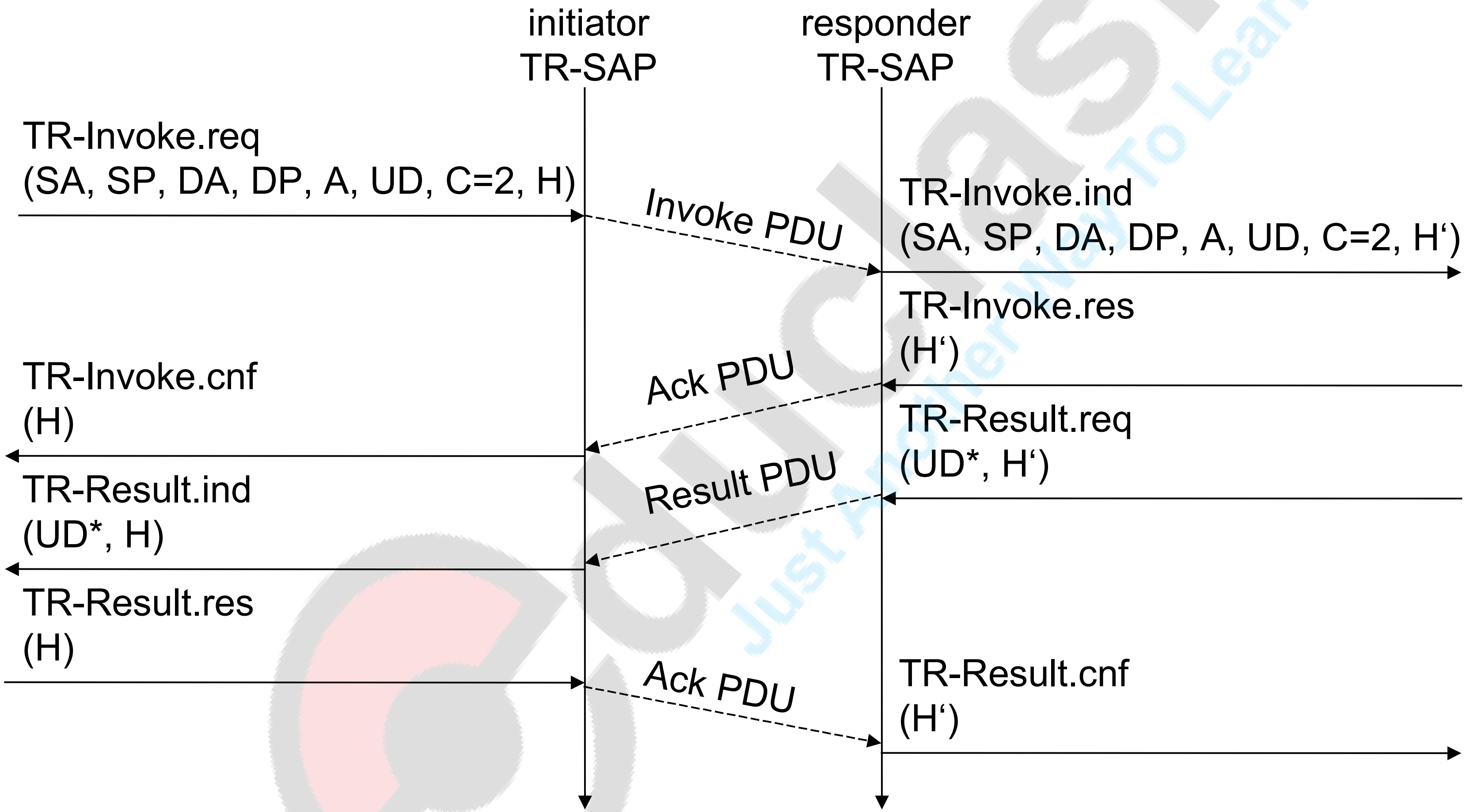
H- handle H to identify transaction uniquely

WTP Class 1 transaction- no user ack and user ack





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‘)

Ack PDU

TR-Invoke.cnf
(H)

TR-Result.req
(UD*, H‘)

Result PDU

TR-Result.ind
(UD*, H)

TR-Result.res
(H)

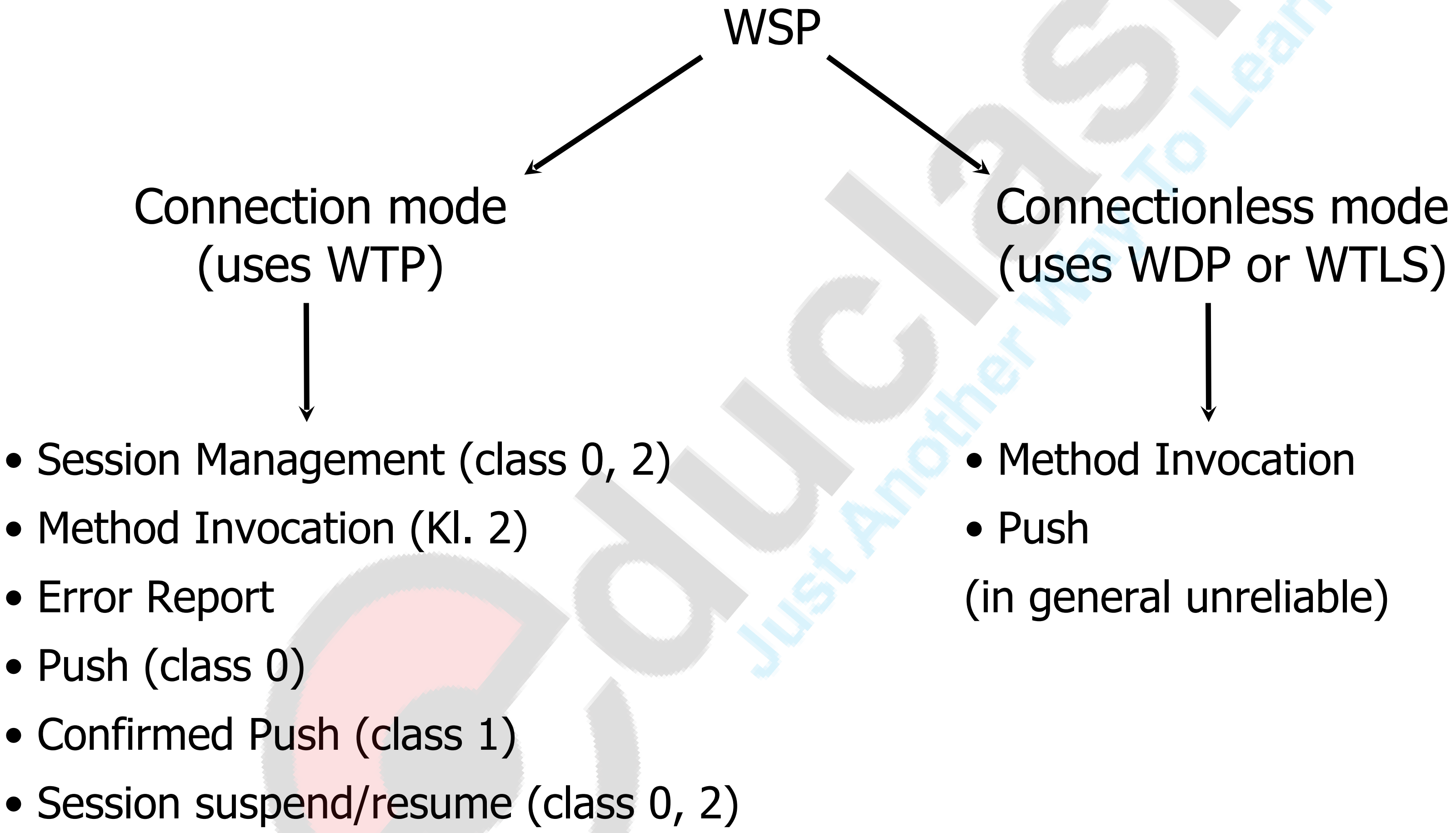
Ack PDU

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,...)
- QoS support
- group communication
- management

WSP protocols



WSP

Connection mode
(uses WTP)

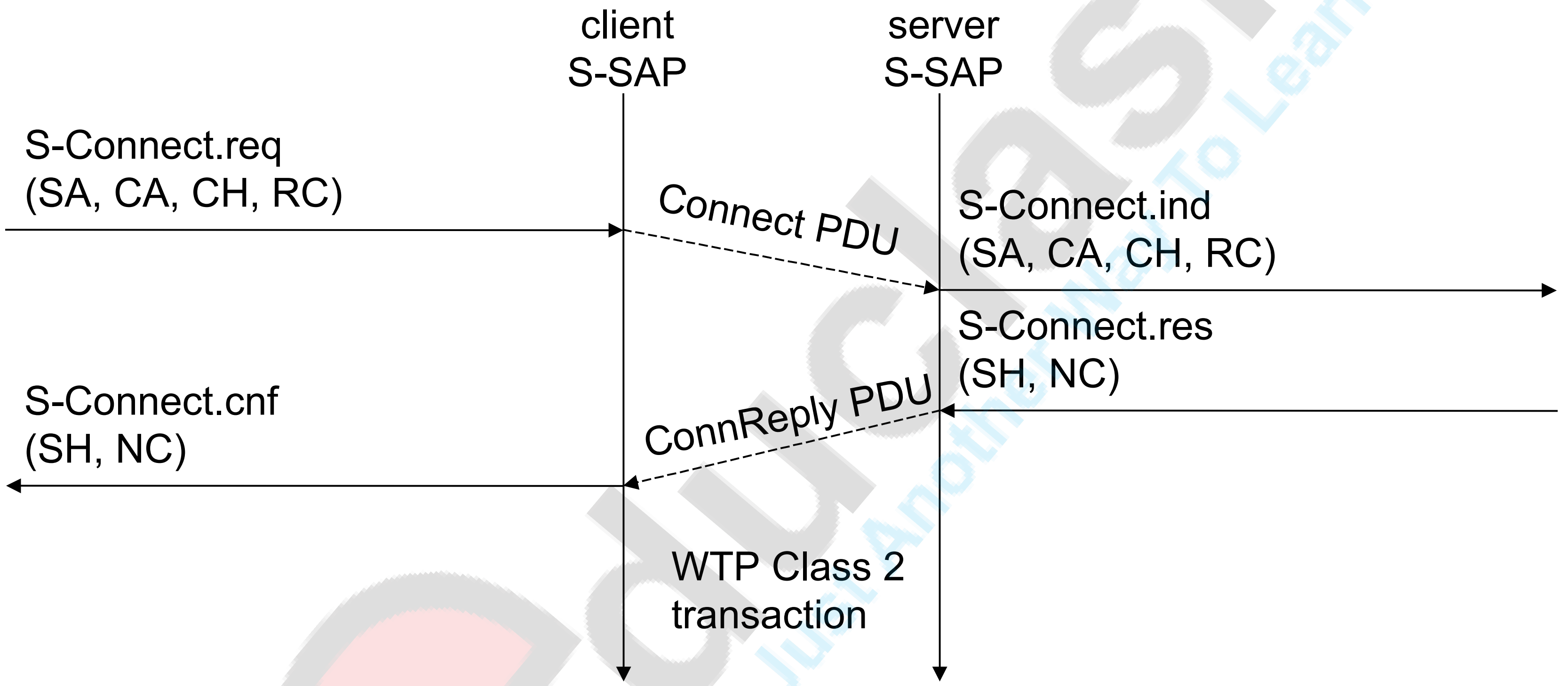
- Session Management (class 0, 2)
- Method Invocation (Kl. 2)
- Error Report
- Push (class 0)
- Confirmed Push (class 1)
- Session suspend/resume (class 0, 2)

Connectionless mode
(uses WDP or WTLS)

- Method Invocation
- Push

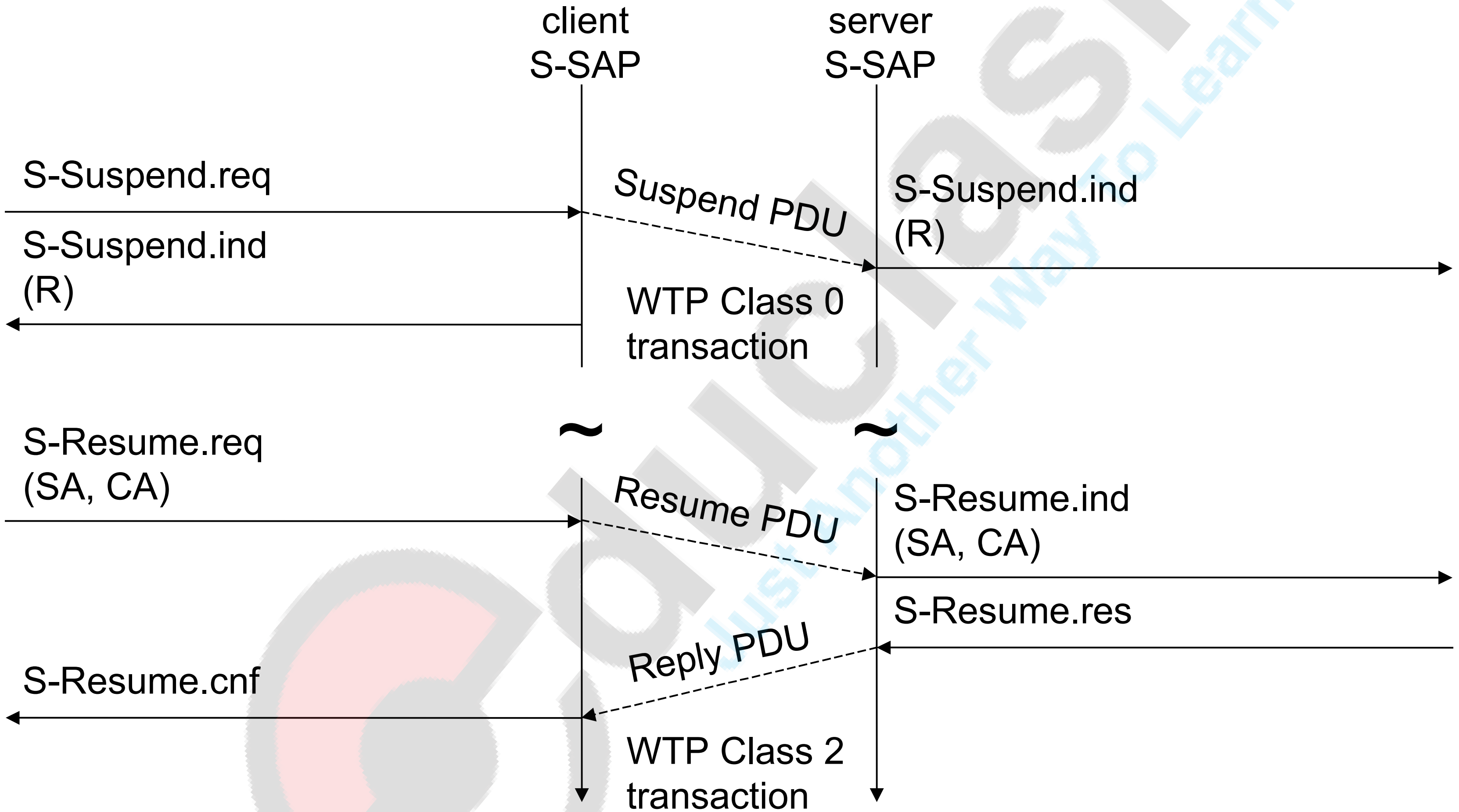
(in general unreliable)

WSP/B session establishment

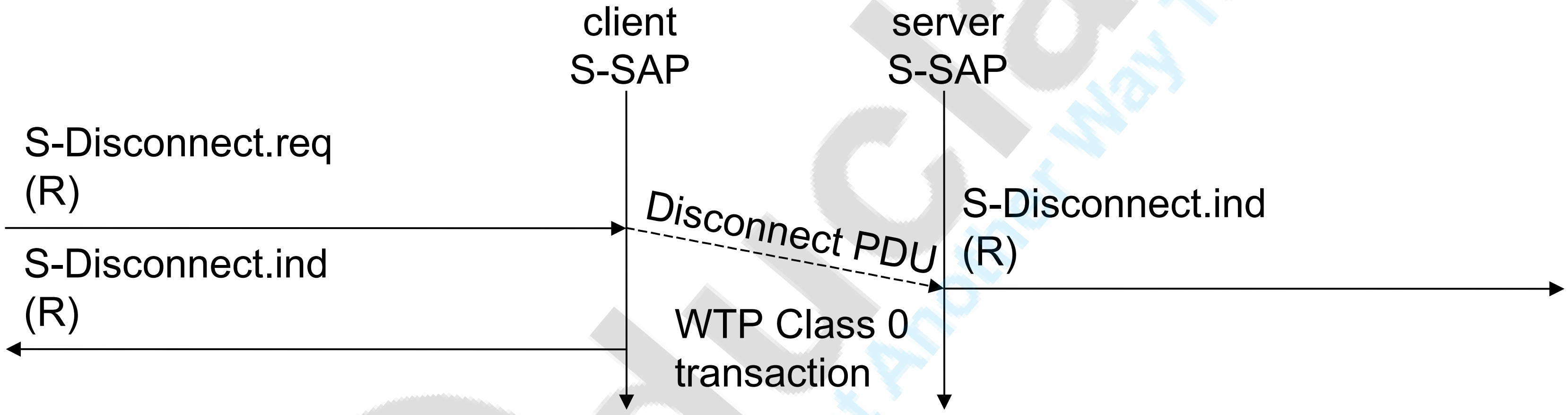


Where, SA-server address, CA- client address, CH- client header-optional, RC- requested capabilities, SH-server header, NC-negotiated capabilities

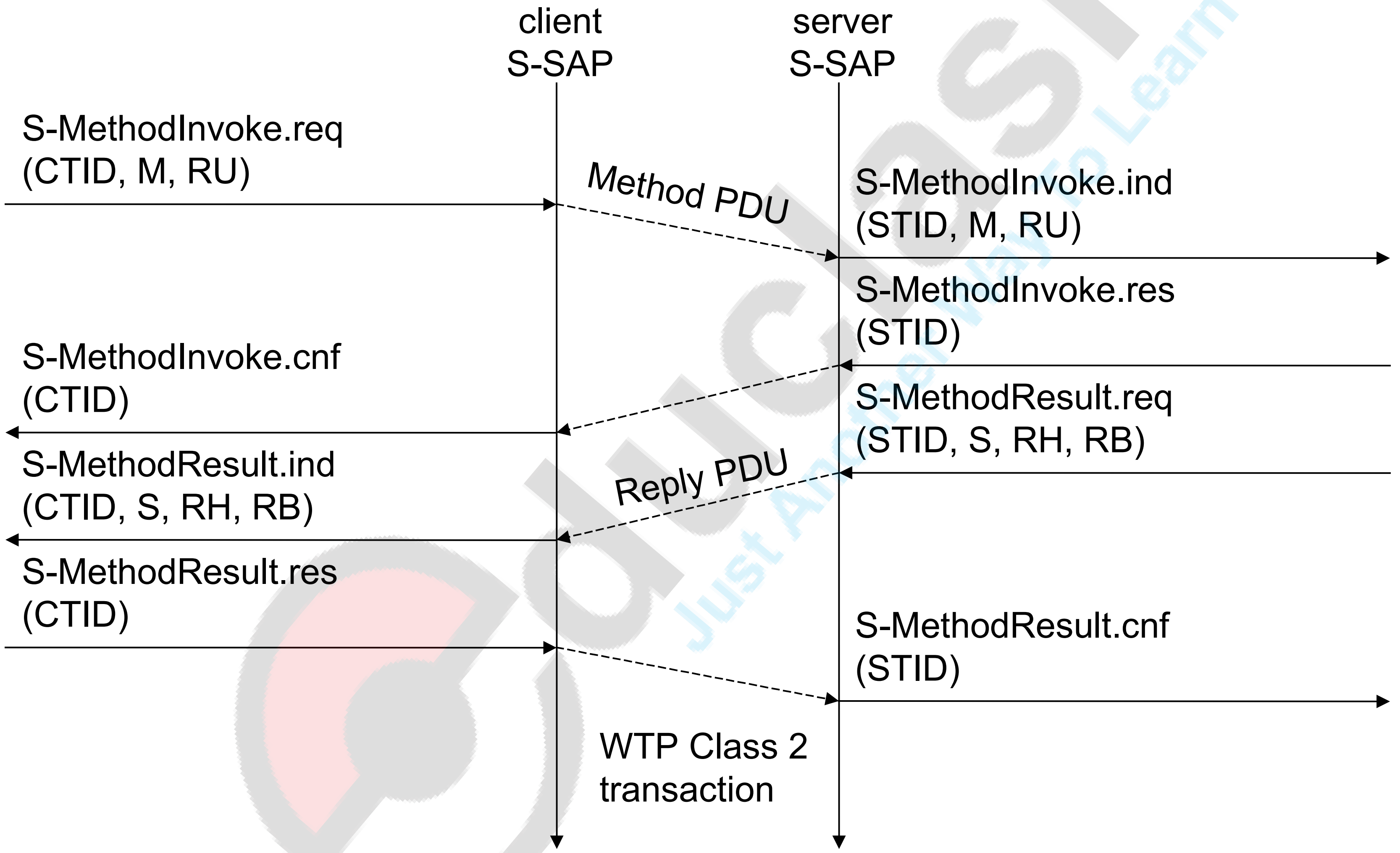
WSP/B session suspend/resume



WSP/B session termination

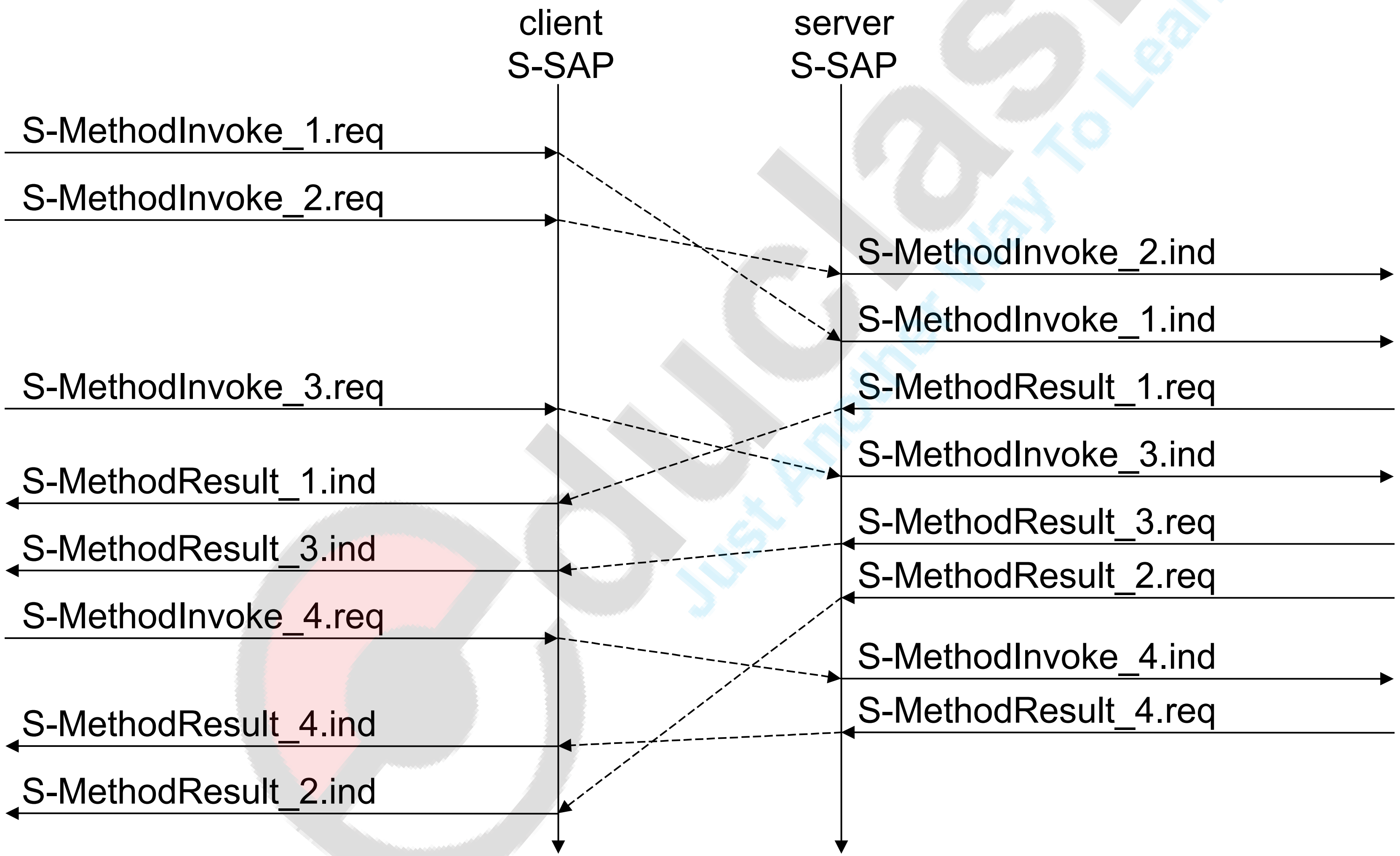


WSP/B method invoke



Where, STID- server transaction identifier, CTID- client TID, M- method, RU- request URI, RH- response header, RB- response body

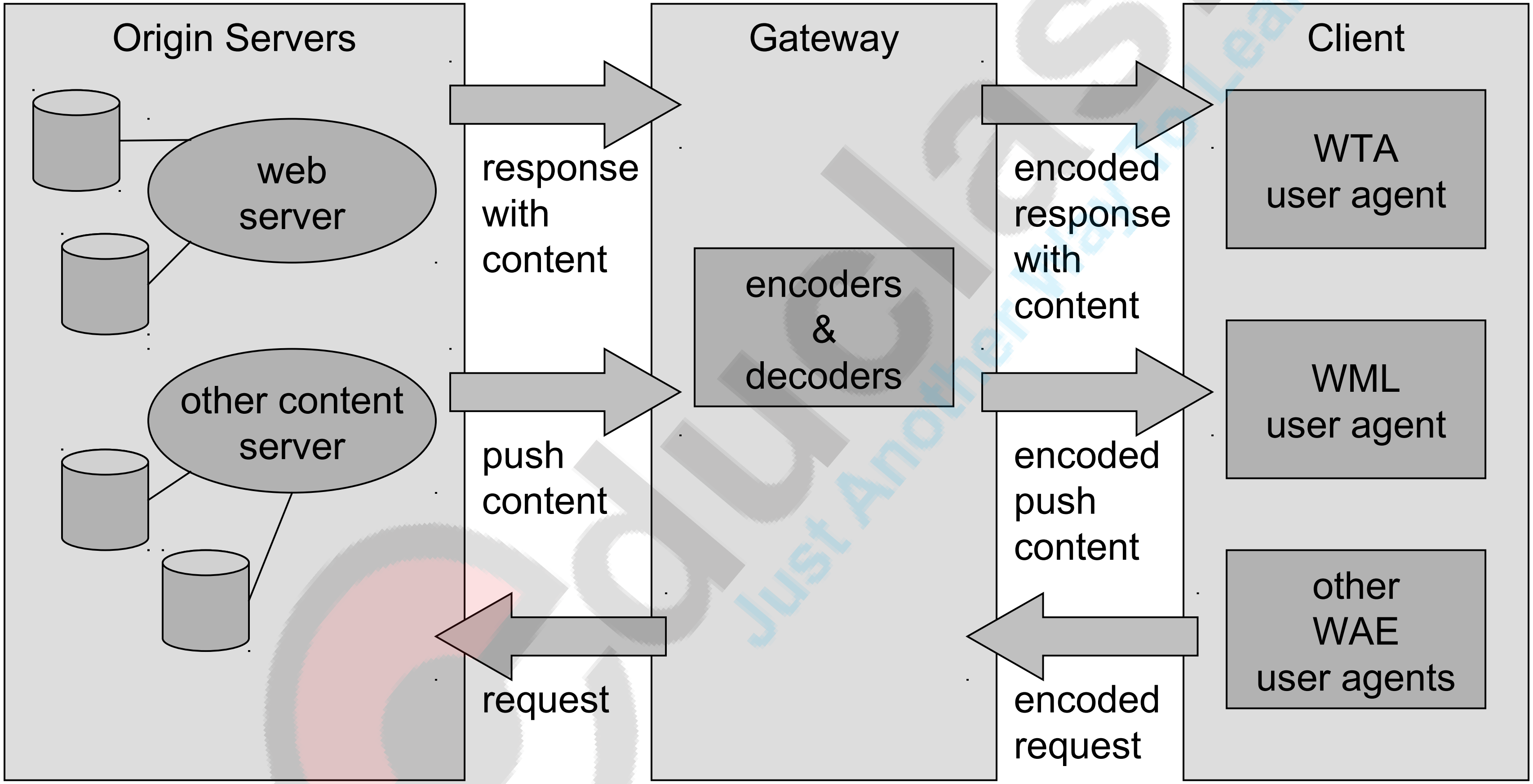
WSP/B over WTP - asynchronous, unordered requests



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;
};
```

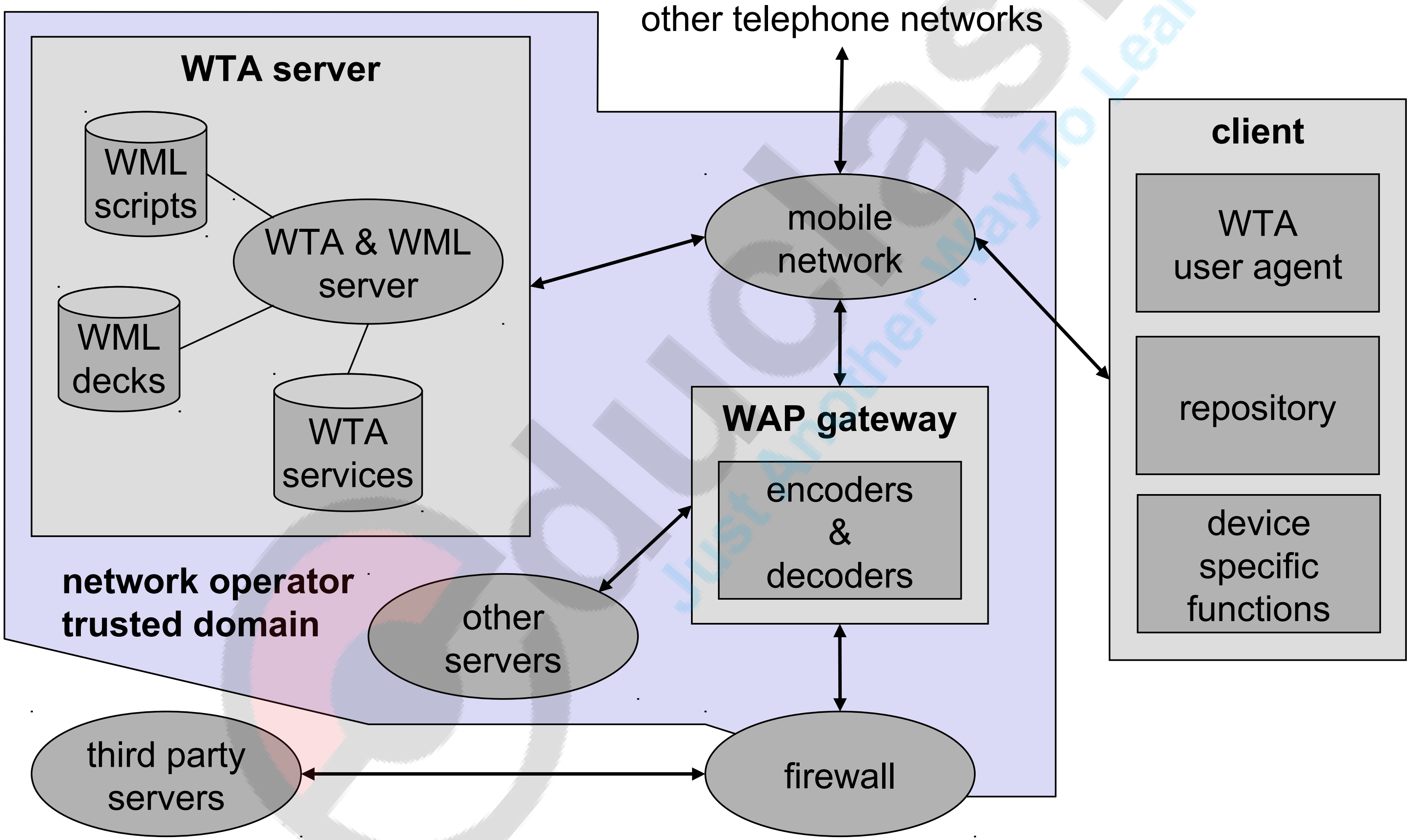
Wireless Telephony Application (WTA)

- 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)

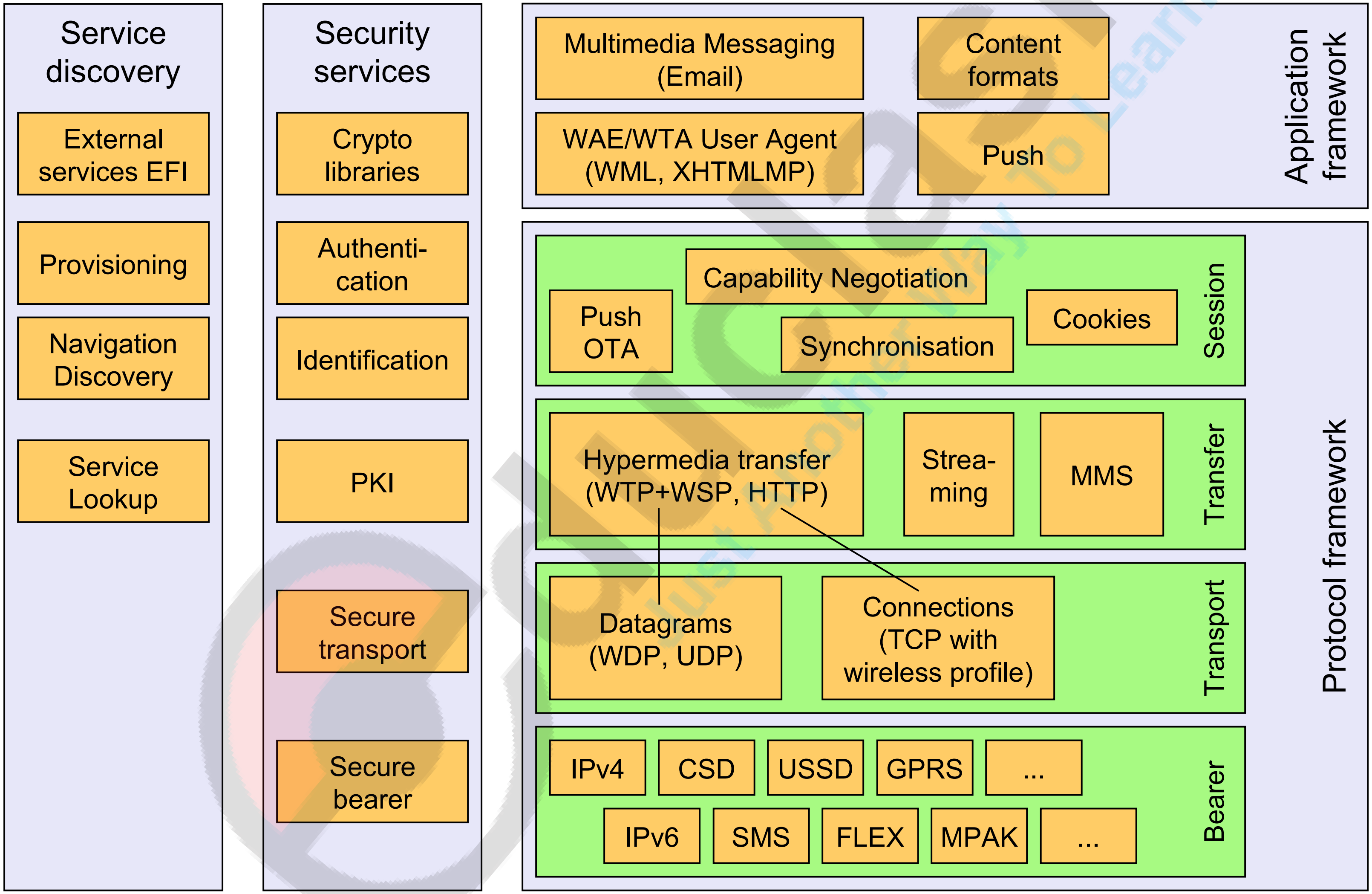
      ```
      wtai://wp/mc;07216086415
      ```
    - calling a number (WMLScript)

      ```
      WTAPublic.makeCall("07216086415");
      ```

WTA logical architecture



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)

Content formats

WAE/WTA User Agent (WML, XHTMLMP)

Push

Protocol framework

Session

Capability Negotiation

Push OTA

Cookies

Synchronisation

Transfer

Hypermedia transfer (WTP+WSP, HTTP)

Streaming

MMS

Transport

Datagrams (WDP, UDP)

Connections (TCP with wireless profile)

Bearer

IPv4

CSD

USSD

GPRS

...

IPv6

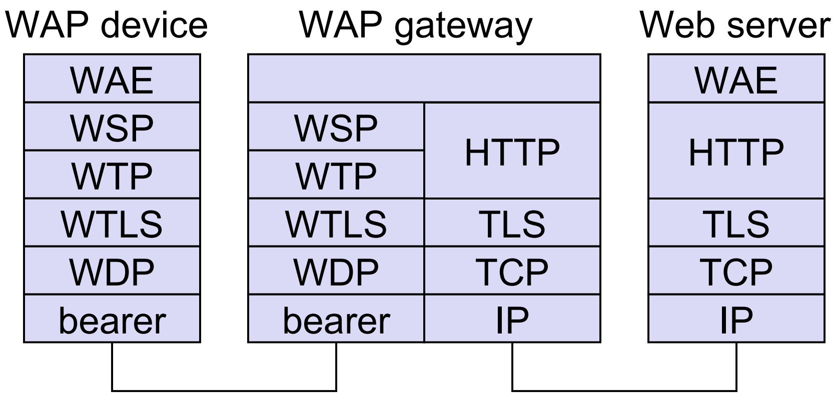
SMS

FLEX

MPAK

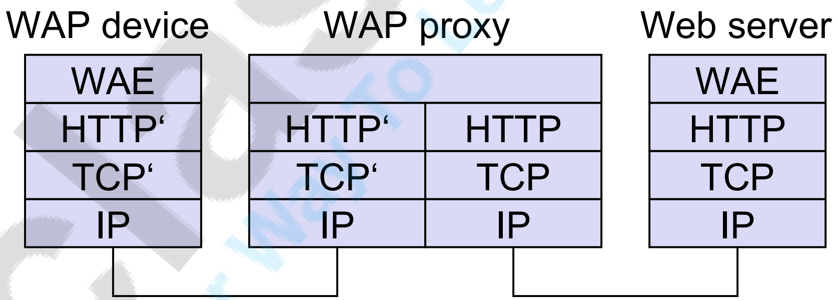
...

WAP 2.0 example protocol stacks



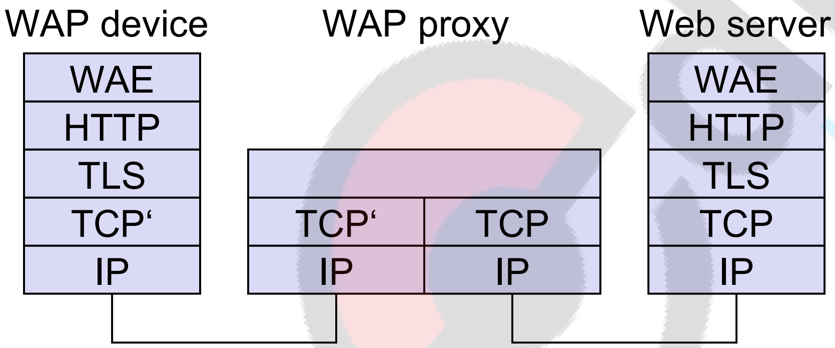
WAP device	WAP gateway		Web server
WAE			WAE
WSP	WSP	HTTP	HTTP
WTP	WTP		
WTLS	WTLS	TLS	TLS
WDP	WDP	TCP	TCP
bearer	bearer	IP	IP

WAP 1.x Server/Gateway/Client



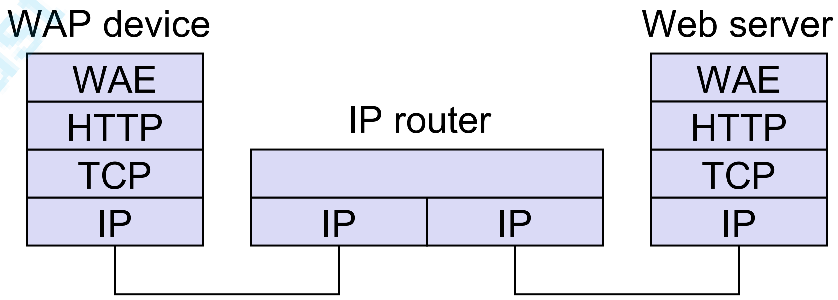
WAP device	WAP proxy		Web server
WAE			WAE
HTTP‘	HTTP‘	HTTP	HTTP
TCP‘	TCP‘	TCP	TCP
IP	IP	IP	IP

WAP HTTP Proxy with profiled TCP and HTTP



WAP device	WAP proxy		Web server
WAE			WAE
HTTP			HTTP
TLS			TLS
TCP‘	TCP‘	TCP	TCP
IP	IP	IP	IP

WAP Proxy with TLS tunneling



WAP device	IP router		Web server
WAE			WAE
HTTP			HTTP
TCP			TCP
IP	IP	IP	IP

WAP direct access

References:

Mobile Communications, Second Edition, Jochen Schiller, Pearson Education- Chapter 10.

University Questions:

- Discuss WAP protocol architecture in detail.-May 16-7M
- Discuss WAP programming model-May 16-7M
- Write a short note on WML and WML Script-May 16-5M
- Write a short note on SyncML-Nov 16-5M