

Table of Contents

- **WAP Process**
- **WAP Architecture**
- **Wireless Transport Protocol Overview**
- **Wireless Session Protocol Overview**
- **Wireless Application environment Overview**

WAP Process

- **Objectives of WAP**
- **Membership benefits**
- **WAP publications**
- **WAP Board of Directors**
- **Participation process up until v1.0 release**
- **Participation process after v1.0 release**

Objectives of WAP Forum

- **Bring Internet content and advanced services to digital cellular phones and other wireless terminals**
- **Create a global wireless protocol specification to work across differing wireless network technologies**
- **Submit specifications for adoption by appropriate industry and standards bodies**
- **Enable applications to scale across a variety of transport options and device types**

Membership benefits

- **Contribute to current specification work**
 - see all drafts before the public version comes out
 - provide comments on drafts to technical chairmen
 - provide input for consideration
- **Participate in driving future evolution of WAP**
 - participate directly in future working groups
- **Nominate and elect new directors to the WAP Forum board**
 - eight positions for new directors will be available

WAP Publications

**Publication on the Web:
<http://www.wapforum.org>**

WAP History & Schedule

26 Jun 1997	WAP effort launched by founders
15 Sep 1997	WAP architecture published
30 Dec 1997	WAP Forum LTD. legally registered
07 Jan 1998	Briefing for early endorsers
30 Jan 1998	Draft spec available for members to review
13 Feb 1998	1st Members review and briefing meeting
Feb 1998	Draft WAP specifications available to public
Mar 1998	2nd Members review and briefing meeting
May 1998	New working groups created for future evolution

WAP Board of Directors

Joakim Nelson

Ericsson

joakim.nelson@ecs.ericsson.se

Jerry Upton

Motorola

jerry_upton@wes.mot.com

Mikko Terho

Nokia

mikko.terho@nmp.nokia.com

Chuck Parrish

Unwired Planet

chuck@uplanet.com

Participation process up until v1.0 release

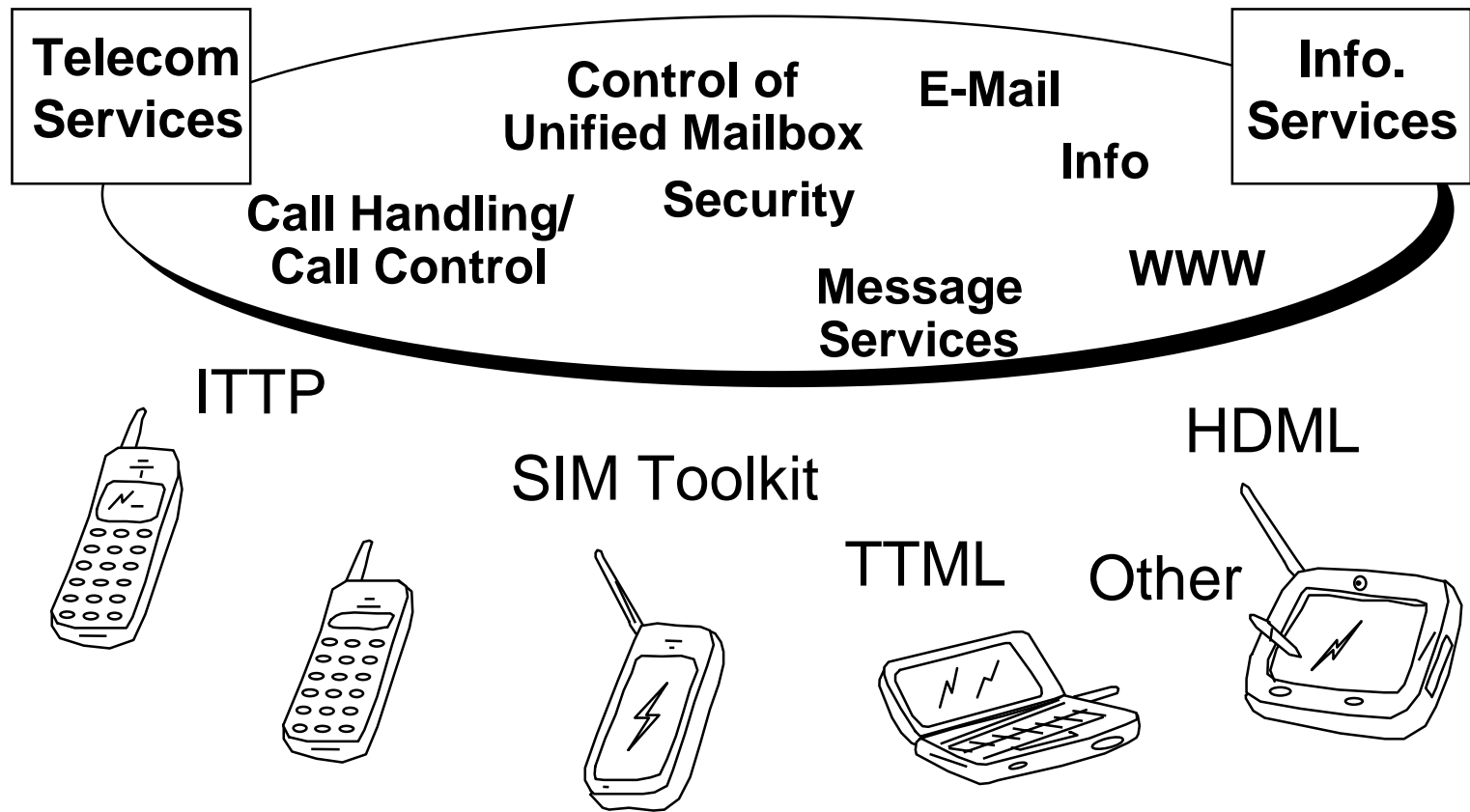
- **New members of WAP Forum will be able to influence the finalization of the WAP specifications**
- **Overview meeting for new and prospective members from 1 to 5 PM on February 12**
- **First detailed technical briefing for members, February 13 from 9 to 5 PM**
- **Second detailed technical briefing for members, week of March 9**

Participation process after v1.0 release

- **Working groups will be reconstituted to include a broader representation**
- **Additional working groups may be created**
- **An expanded input process will be developed to support a larger number of participants with greater interaction**
- **Efforts will be launched to address**
 - **compliance**
 - **interoperability**

WAP Architecture Overview

Service Development - Concept Overview

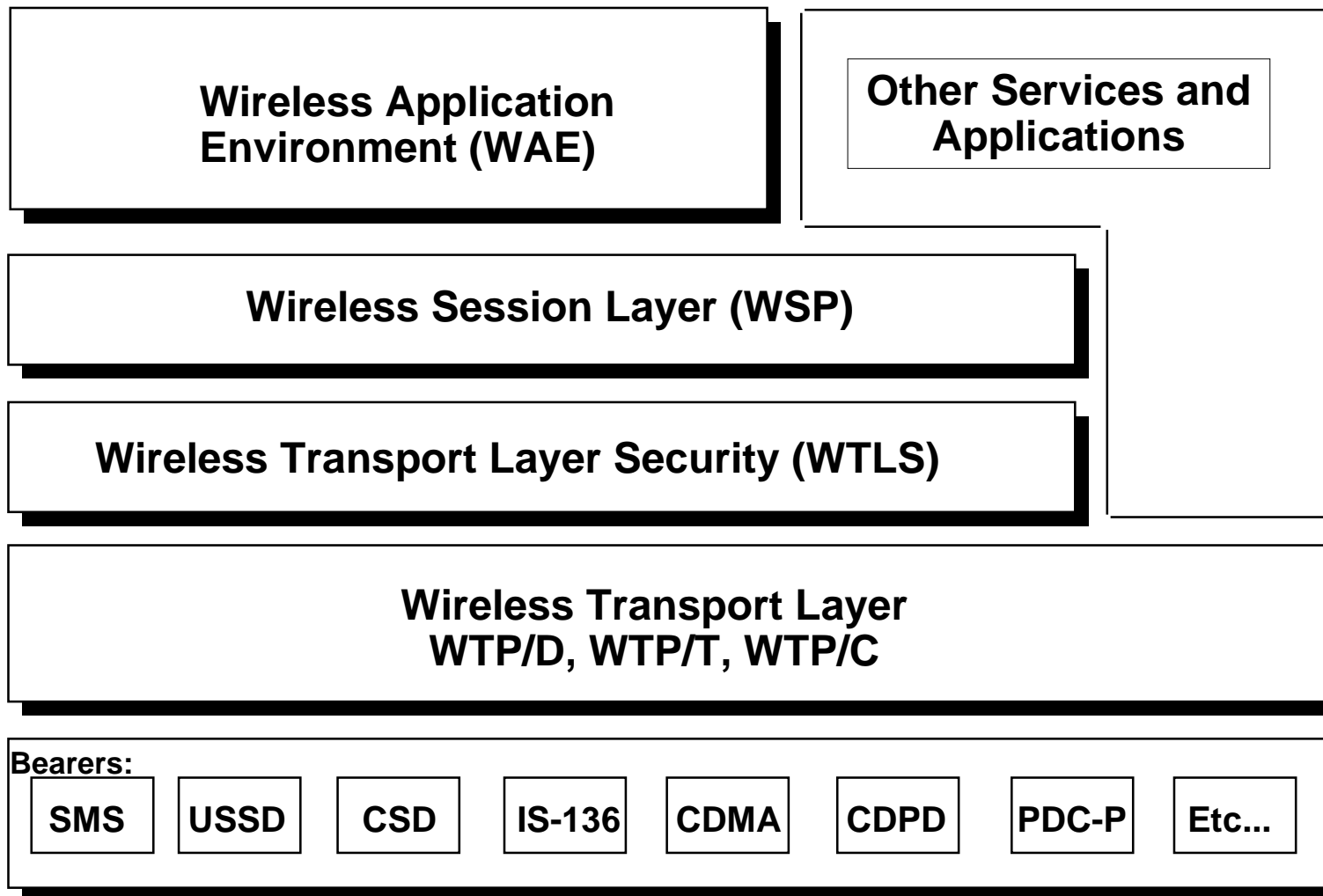


What is defined in WAP ?

- **Micro-Browser**
 - concept that is similar to the Internet browsing
- **Scripting similar to JavaScript**
 - provide means for dynamically enhancing MS capabilities
- **WTA / WTAI**
 - access telephone functionality e.g. call chains
- **Content formats e.g.**
 - business card (vCard)
 - calendar event (vCalendar)
- **Layered telecommunication stack including**
 - transport
 - security
 - session

WAP Architecture

Wireless Application Protocol



The new W-acronyms

WAE = Wireless Application Environment

WML = Wireless Markup Language

WMLScript = Wireless Markup Language Script

WTA = Wireless Telephony Application

WSP = Wireless Session Protocol

WTLS = Wireless Transport Layer Security

WTP = Wireless Transport Protocol

Other Terms

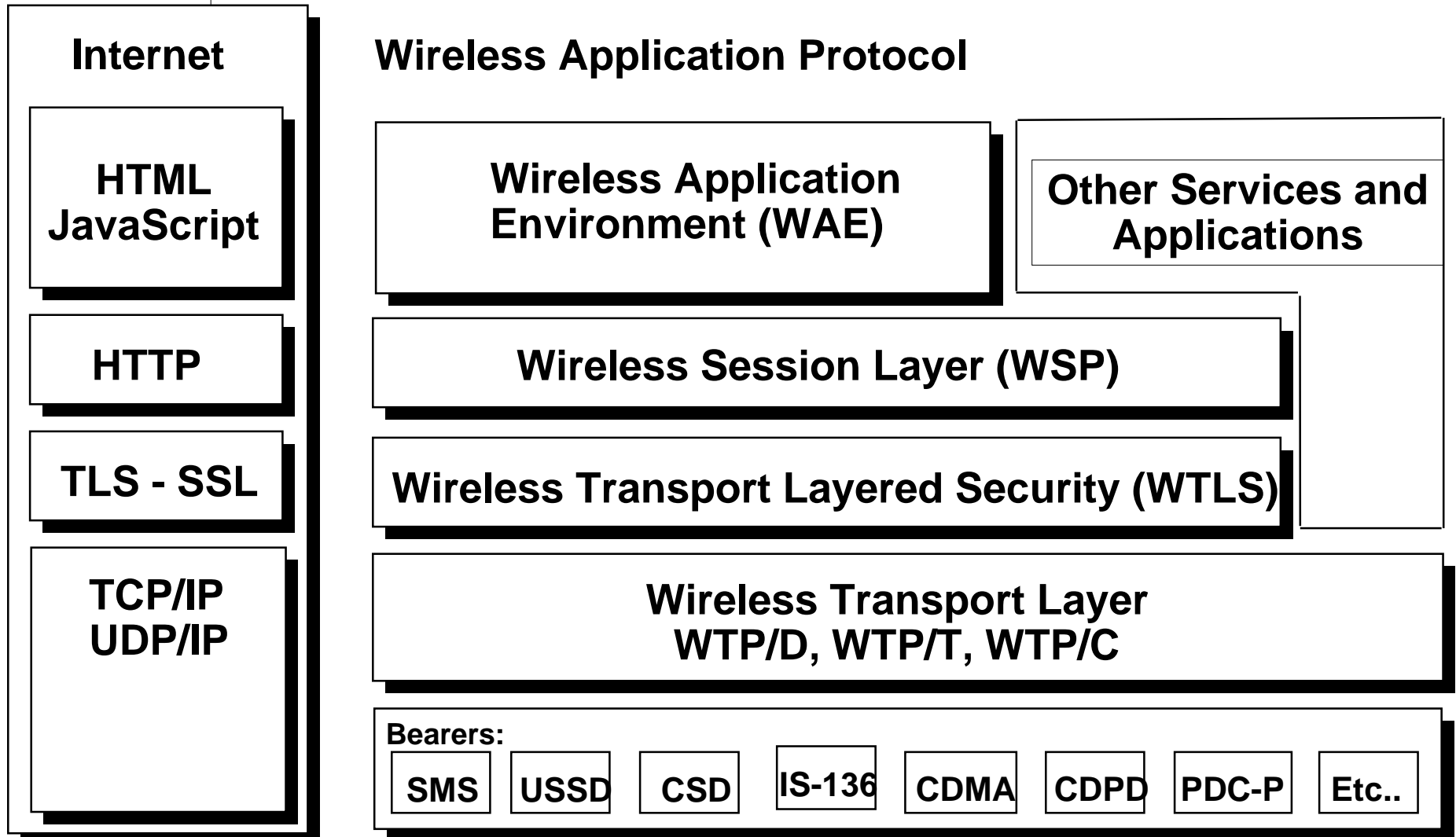
Other applications = applications using WAP services like session or application layer

URL = Universal Resource Locators, which are the key addressing mechanism in internet as well as in WAP

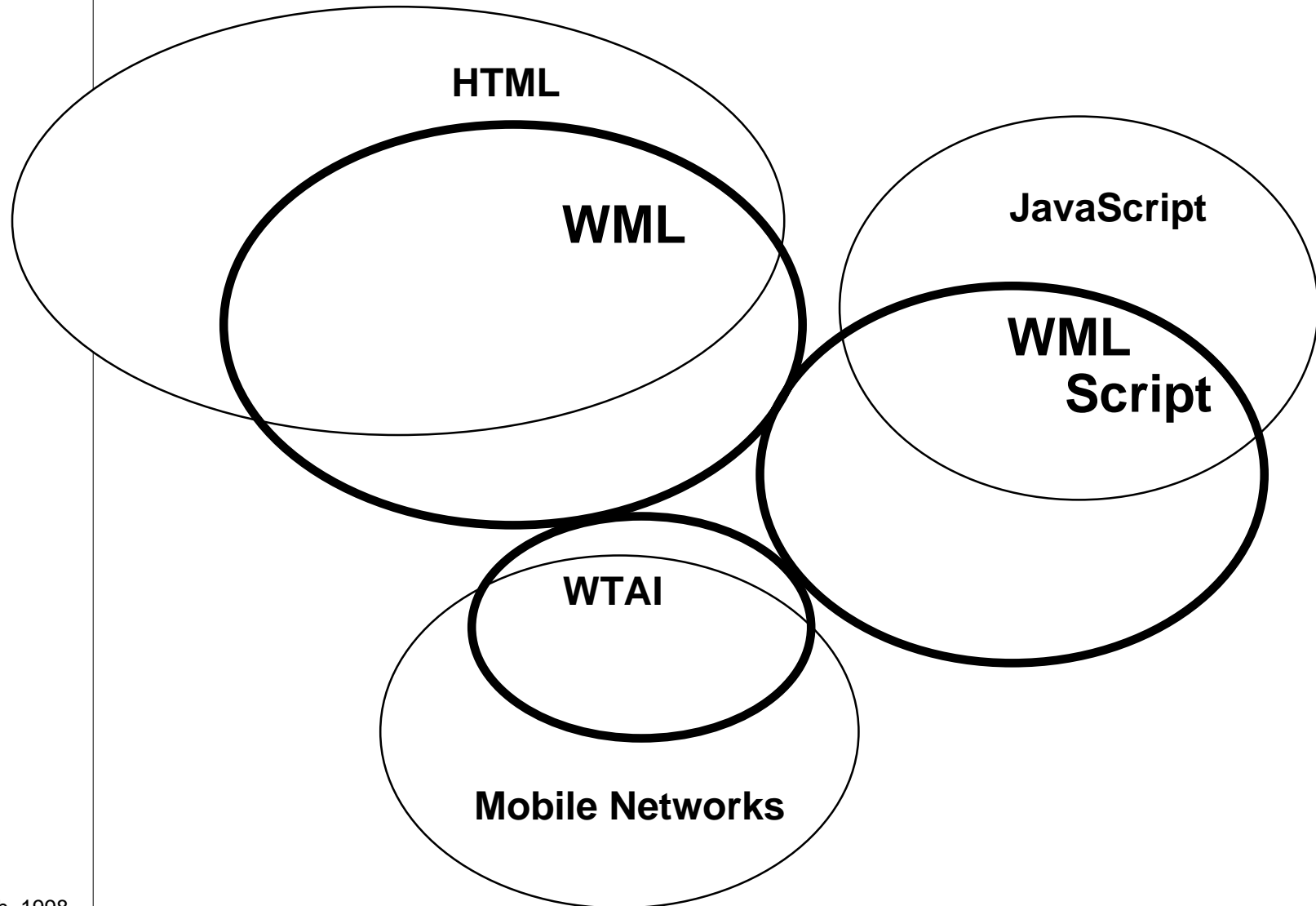
Content formats = formats to exchange data e.g. image, calendar (vCalendar), and phone book (vCard) data

**Events = State Transitions in:
WML Browser (e.g. User Input or Navigation),
WTA Browser (e.g. Network Events)**

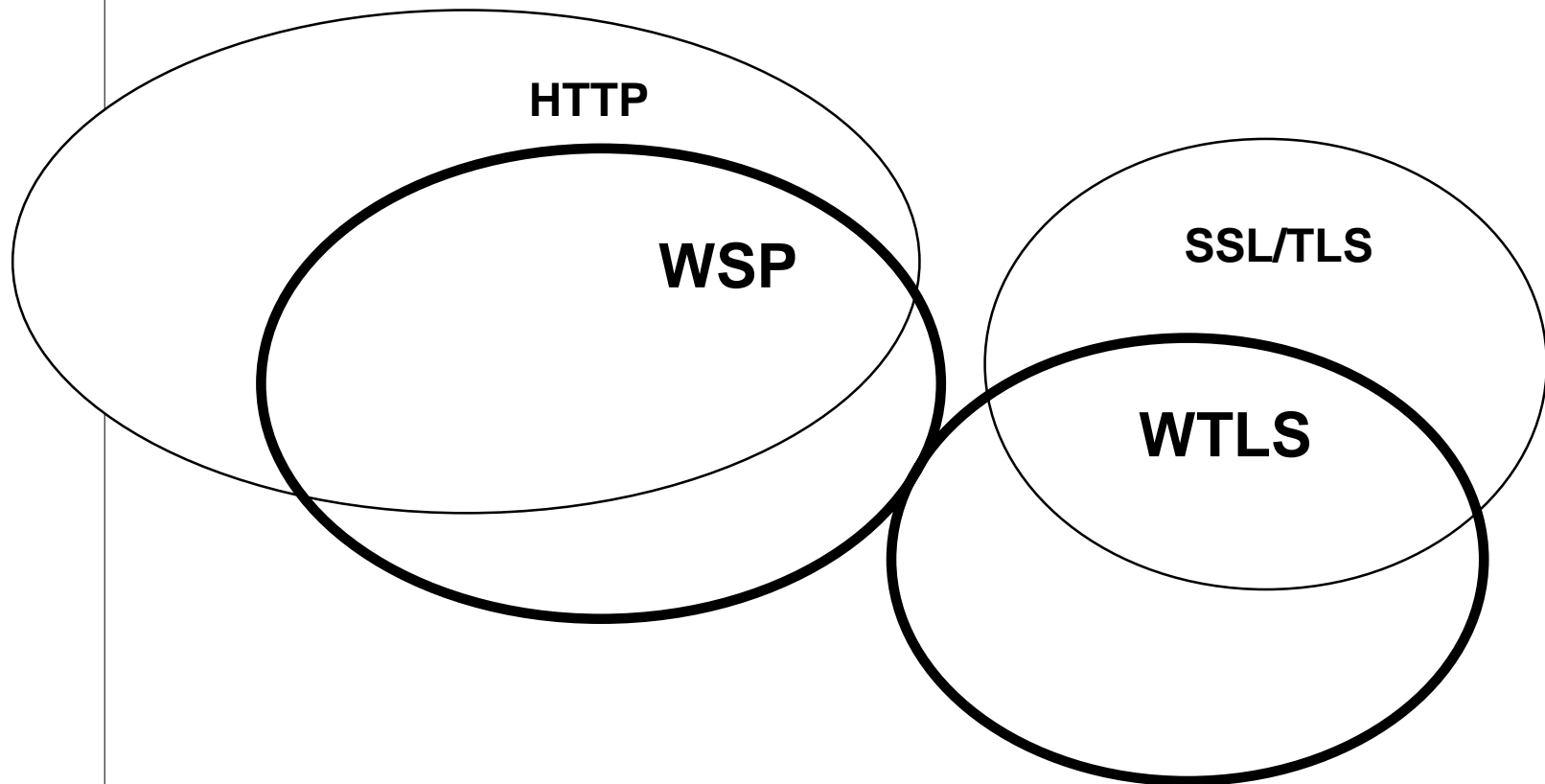
Comparison between Internet and WAP Architecture



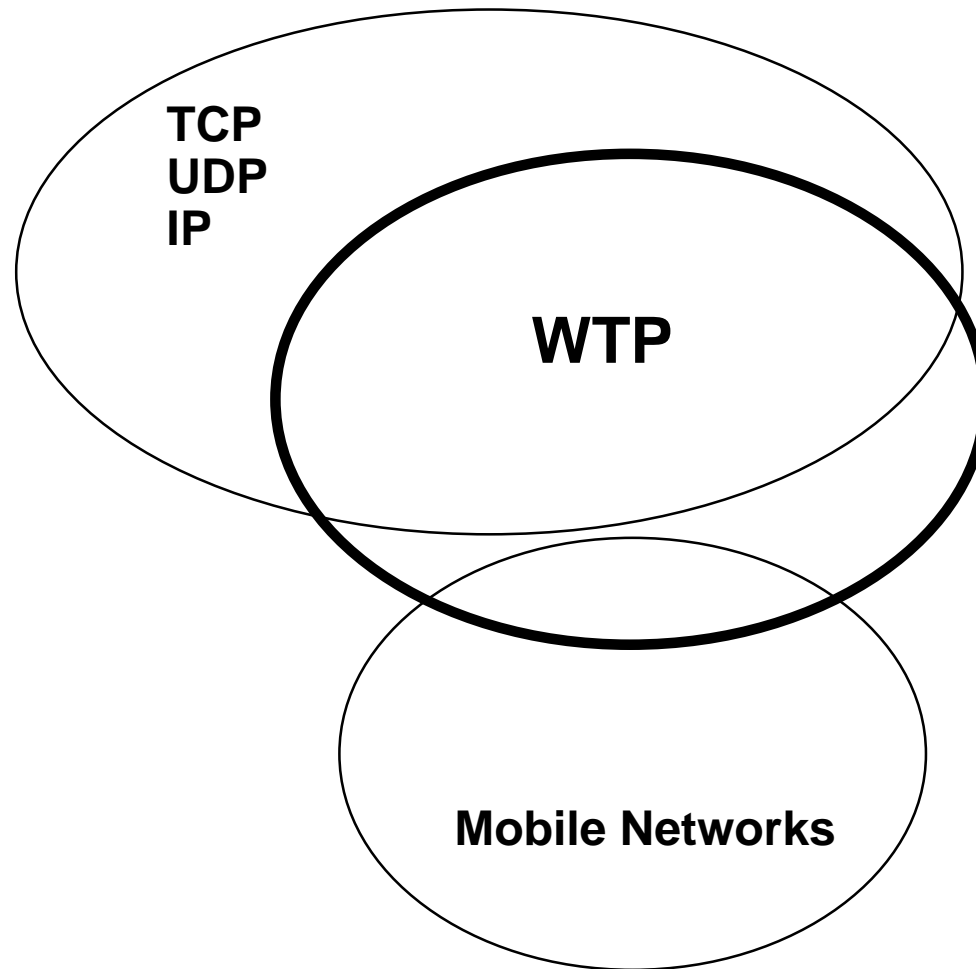
Concept Comparison - WAE



Concept Comparison - WSP and WTLS

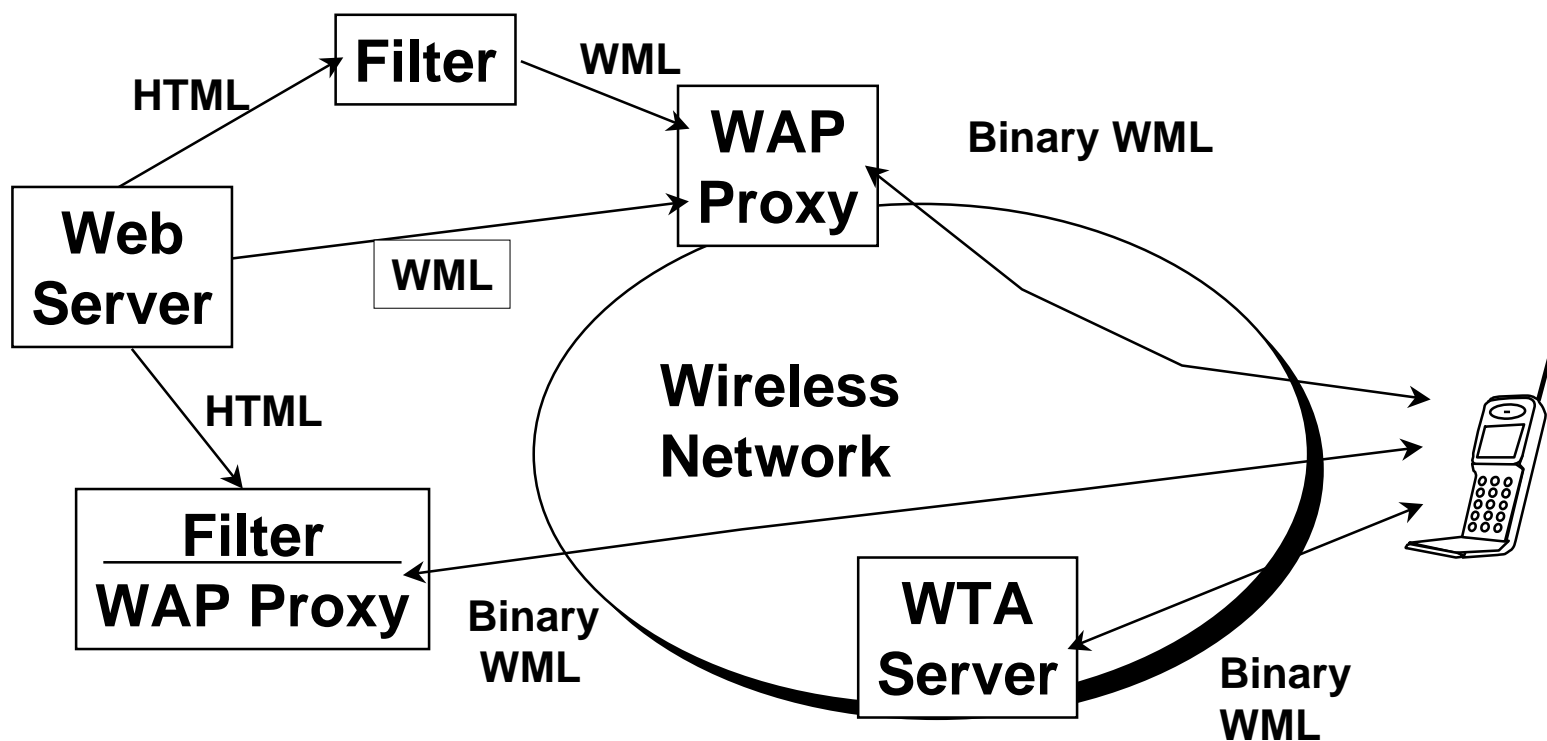


Concept Comparison: WTP

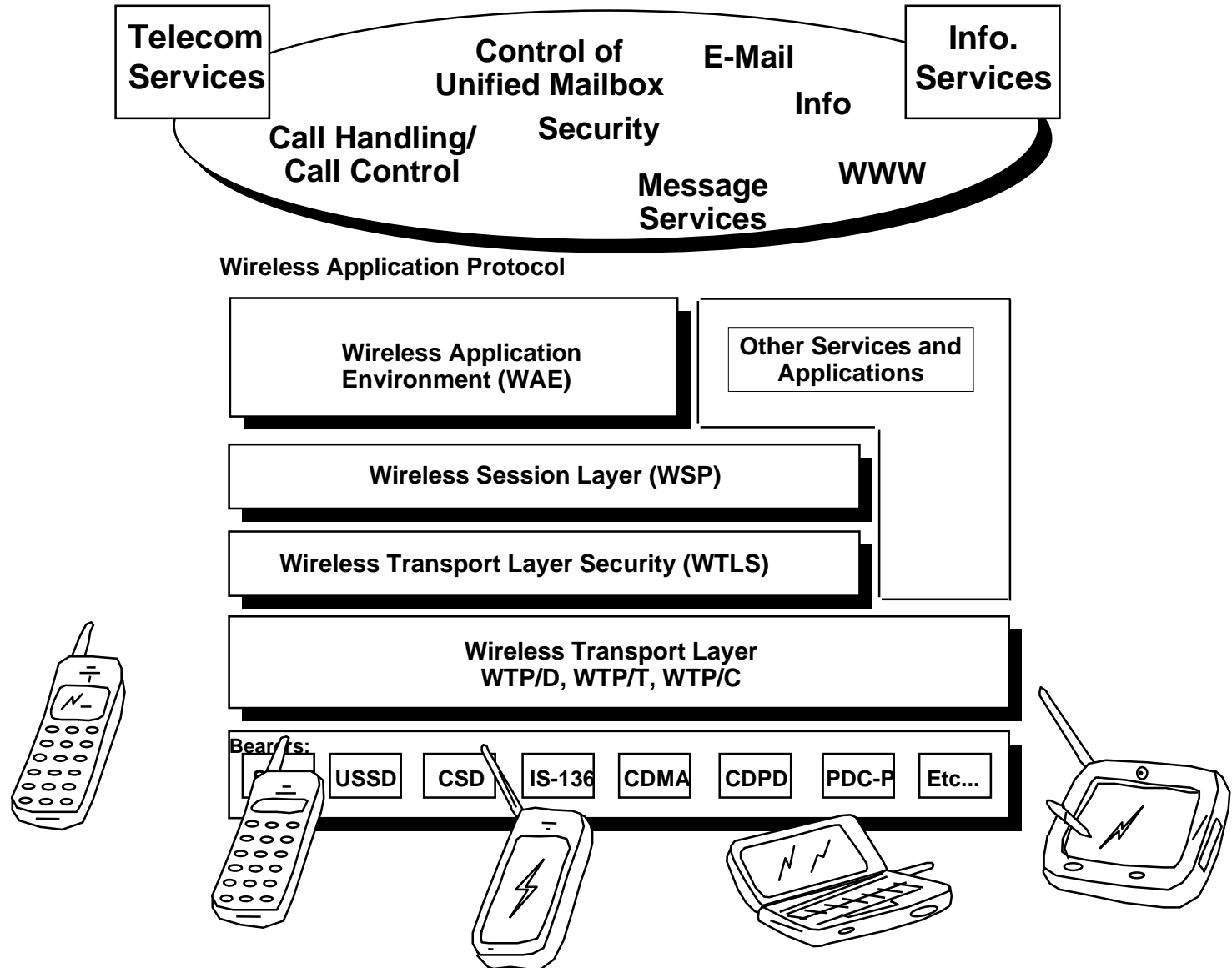


WAP and Related Network Elements

- Binary format to the client
- Filtering and proxy technologies



Concept for Wireless



Wireless Transport Protocol Technical Overview

Overview

- **WTP Spec for January 1998**
 - **Goals and Requirements**
 - **Architecture**
 - **Services**
 - **Protocols**
 - **Beareres**
- **Work Items for 1998**
 - **work items to be completed by 1Q98**
 - **work items to be completed by 4Q98**

Goals for WTP

- a common interface to physical transport mechanisms
- provide a port-based abstract interface to upper layer protocols
- allow applications designed independent from specific transports
- extensible to a variety of digital wireless networks and future transport options
- supports connection-oriented, transaction, and datagram modes
- optimize for narrow to medium bandwidth channels
- ensure multi-vendor interoperability
- allow peer-to-peer, client/server, and one-to-many applications
- to be capable of implementation in a low memory footprint, suitable for “standard” or “low-IQ” handsets

Requirements for WTP

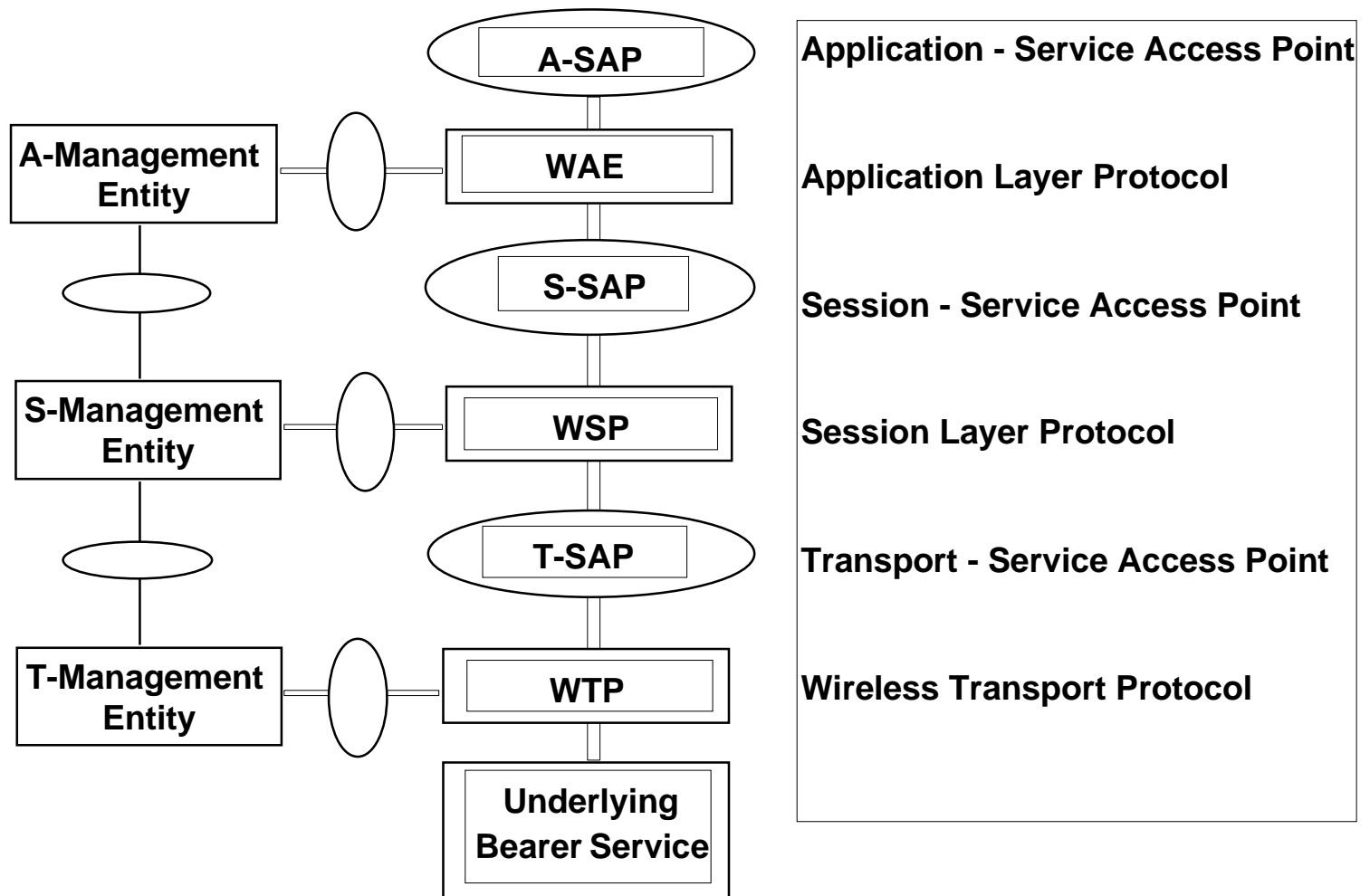
- **WTP/C provides a reliable connection-oriented service to the upper layer**
- **WTP/T provides a reliable transaction-oriented service suitable for web browsing**
- **WTP/D provides a datagram service**
- **lightweight : implementable in $\leq 10\text{Kb}$ (low memory/computational needs)**
- **a minimum implementation of the protocol should be possible**
- **a reference implementation will be used to evaluate the acceptance criteria**
- **support the selection of an underlying bearer by the upper layer**

Requirements (continued)

- **eliminate need for applications to be aware of the specifics of available transports**
- **protocol must be modular to allow various security solutions**
- **port numbers must be supported**
- **efficiency with respect to over-the-air transmission**
 - **segmentation and reassembly**
 - **selective retransmission**
 - **header compression**
 - **low transmission overhead (reduce 3-way handshakes)**
 - **optimistic handshake (call setup with data)**
- **scalability to operate over a range of wireless networks and devices**

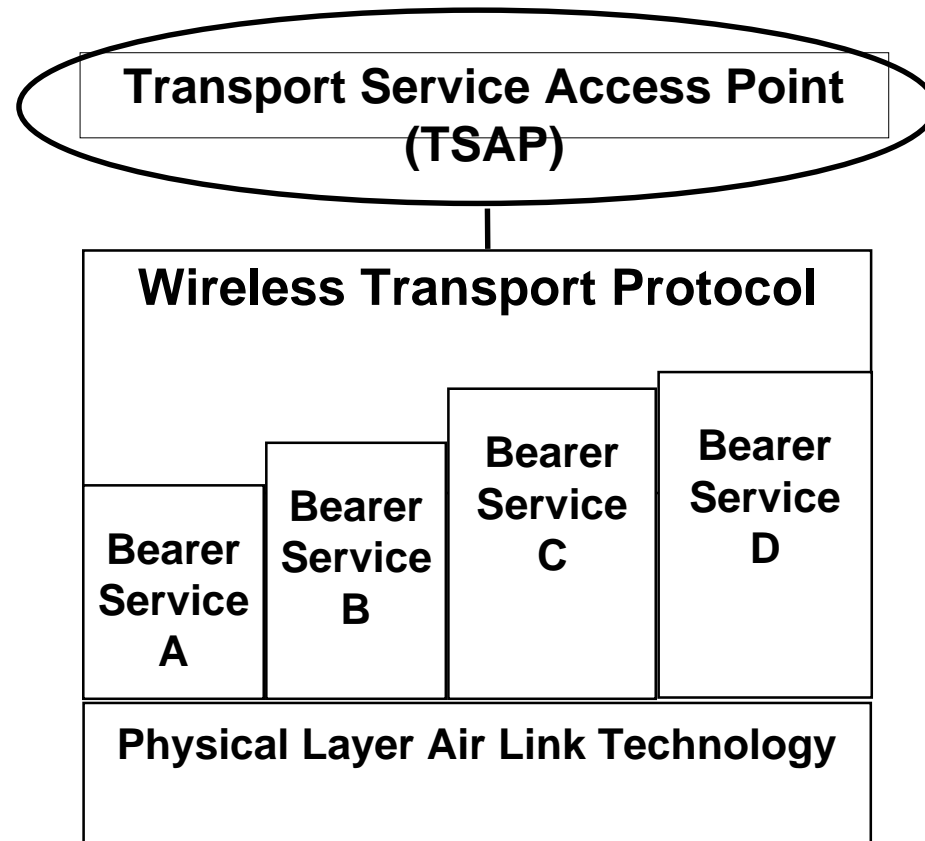
Architecture

- **WAP Architecture uses concepts of Service Interface and Management Entities**



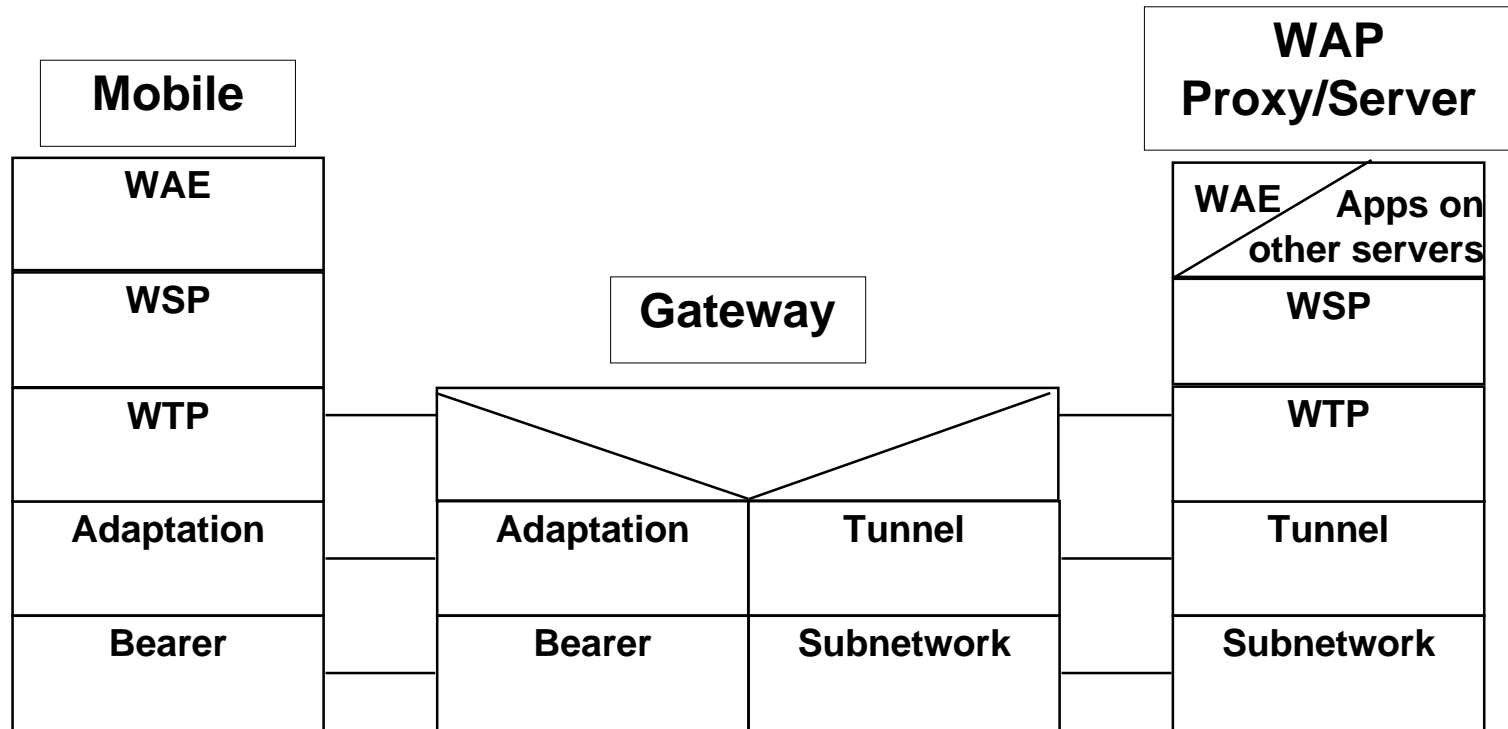
Architecture

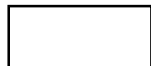
- **WTP Architecture illustrates the integration of underlying bearers to a common service**



Architecture

WTP Detailed Architecture



 defined in the WTP Specification

Services

WTP consists of 3 protocols offering 3 different services to the upper layer

- **WTP/D is the Datagram protocol**
- **WTP/T is the Transaction-Oriented protocol**
- **WTP/C is the Connection-Oriented protocol**
 - **under development, some risk for completion by v1.0**

Services and Protocols

WTP/D (Datagram)

- provides a connection-less, unreliable datagram service
- WTP/D is replaced by UDP when used over an IP network layer.
- uses the Service Primitive
 - T-UnitData.req .ind

Services and Protocols

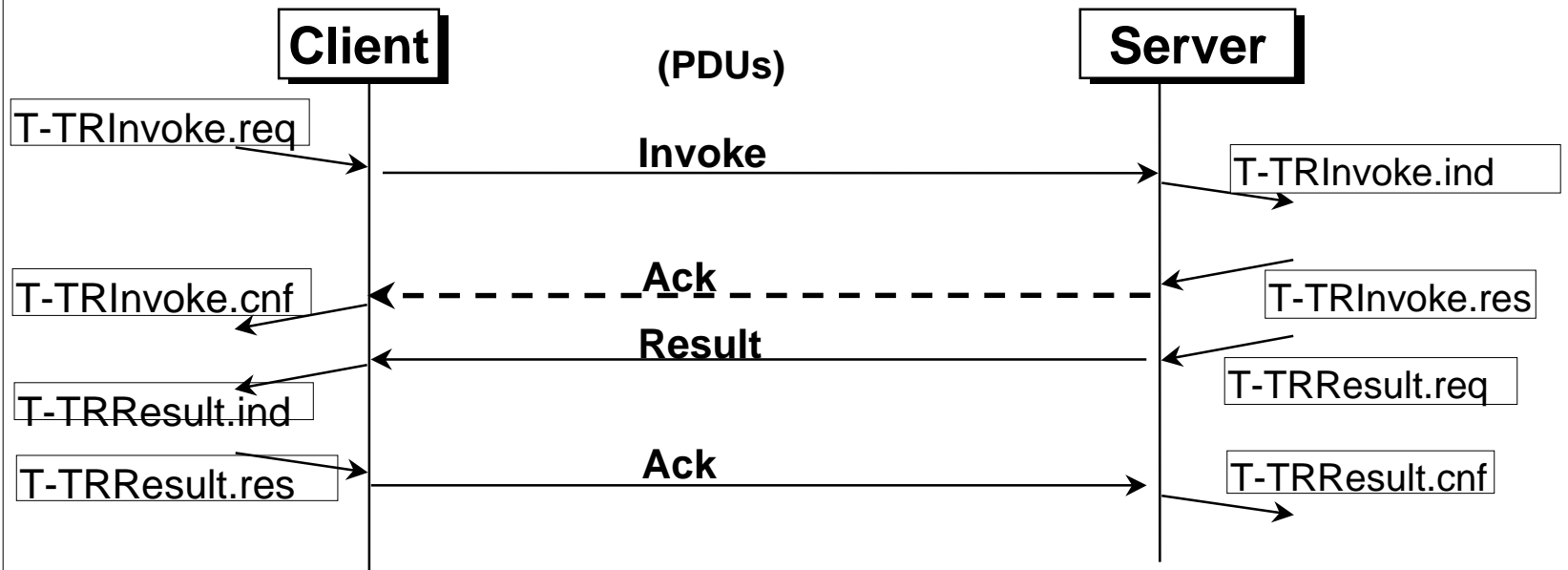
WTP/T (Transaction)

- **provides reliable data transfer based on request/reply paradigm**
 - **no explicit connection setup or tear down**
 - **data carried in first packet of protocol exchange**
 - **seeks to reduce 3-way handshake on initial request**
 - **supports**
 - **retransmission of lost packets**
 - **selective-retransmission**
 - **segmentation / re-assembly**
 - **port number addressing (specific to WTP/T)**
 - **flow control**
 - **message oriented (not stream)**
 - **supports an Abort function for outstanding requests**
 - **supports concatenation of PDUs**

Services and Protocols

WTP/T continued

- uses the service primitives
 - T-TRInvoke.req .cnf .ind .res
 - T-TRResult.req .cnf .ind .res
 - T-Abort.req .ind
- an example of a WTP/T protocol exchange



Services and Protocols

WTP/C (Connection-Oriented)

- **reliable data transfer based on long duration connections requiring full-duplex data exchange**
- **primary characteristics (TBD) of WTP/C**
 - **connection setup with negotiation of parameters**
 - **deterministic flow control appropriate for over-the-air transmissions**
 - **supports**
 - **retransmission of lost packets**
 - **selective-retransmission**
 - **segmentation / re-assembly**
 - **port number addressing**
 - **flow control**
 - **message oriented (not stream)**
 - **supports concatenation of PDUs**
 - **explicit connection disconnect**

Services and Protocols

WTP/C continued

- **uses service primitives**
 - **T-Connect.Req .Cnf .Ind .Res**
 - **T-Data.Req .Cnf .Ind .Res**
 - **T-Disconnect.Req .Cnf .Ind .Res**

Bearers

Bearer definition status for WTP

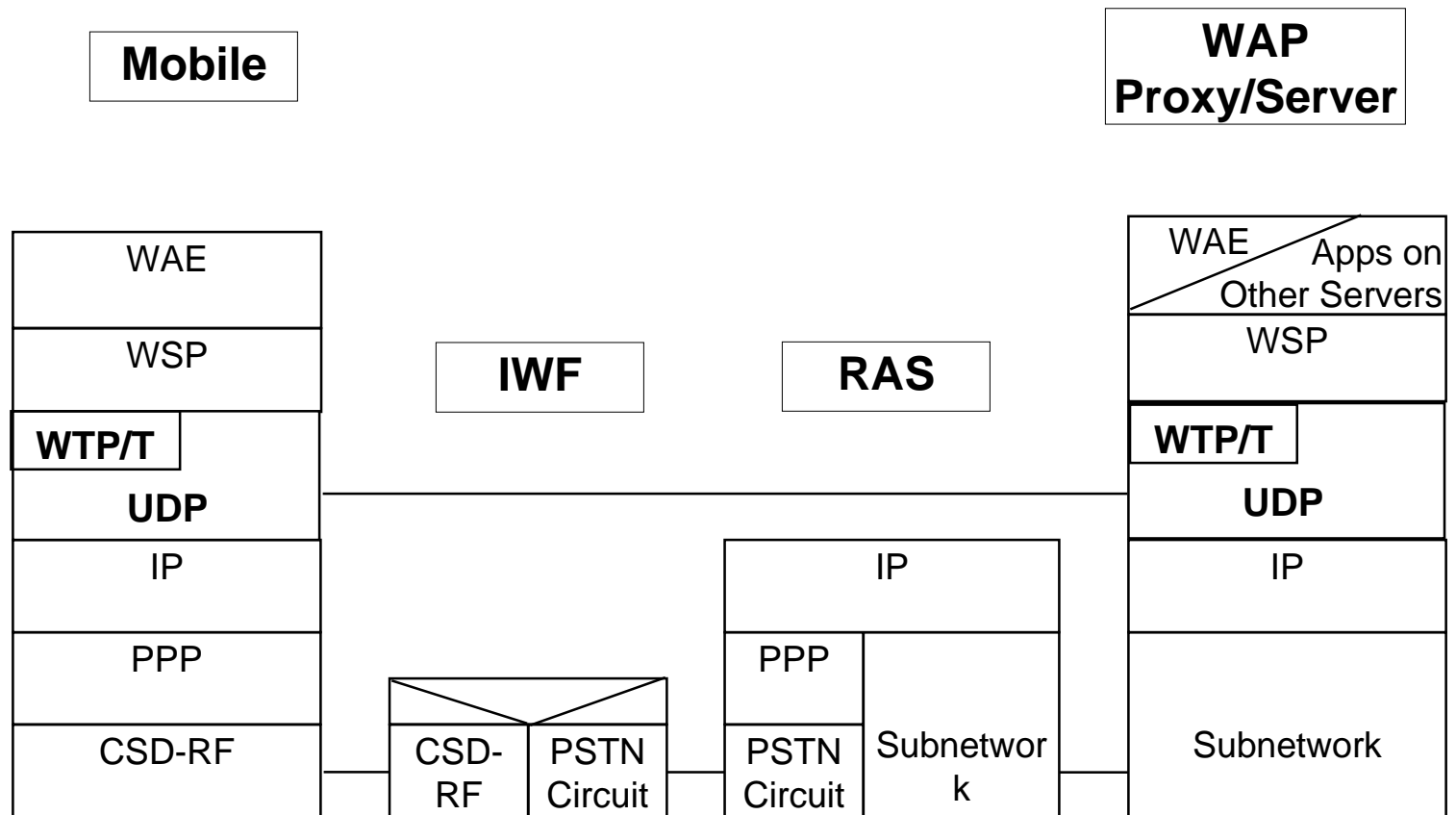
Bearer	Services/Protocols			Comments
	Datagram Service WTP/D	Transaction Service WTP/T	Connection-Oriented Service WTP/C	
GSM SMS	Defined	Defined	Targeted 1998	
GSM USSD	Defined	Defined	Targeted 1998	
GSM C-S Data	Defined	Defined	Targeted 1998	Note 1, 2
GSM GPRS	Defined	Defined	Targeted 1998	Note 1, 2
IS-136 R-Data	Defined	Defined	TBD	
IS-136 C-S Data	Defined	Defined	TBD	
IS-136 Packet	Defined	Partial Def.	TBD	
CDPD	Defined	Defined	TBD	Note 1, 2
CDMA SMS	Partial Def.	Started	TBD	
CDMA C-S Data	Partial Def.	Started	TBD	
PDC	Started	Started	TBD	
iDEN	Defined	Defined	Available 1Q98	Note 1, 2
Mobitex	TBD	TBD	TBD	
DataTAC	TBD	TBD	TBD	

Note 1: any bearer supporting IP will use UDP as the datagram protocol for WTP/D.
i.e. WTP/D over IP = UDP/IP.

Note 2: IP is assumed to be the network layer protocol for this bearer.

Service, Protocol, and Bearer Example

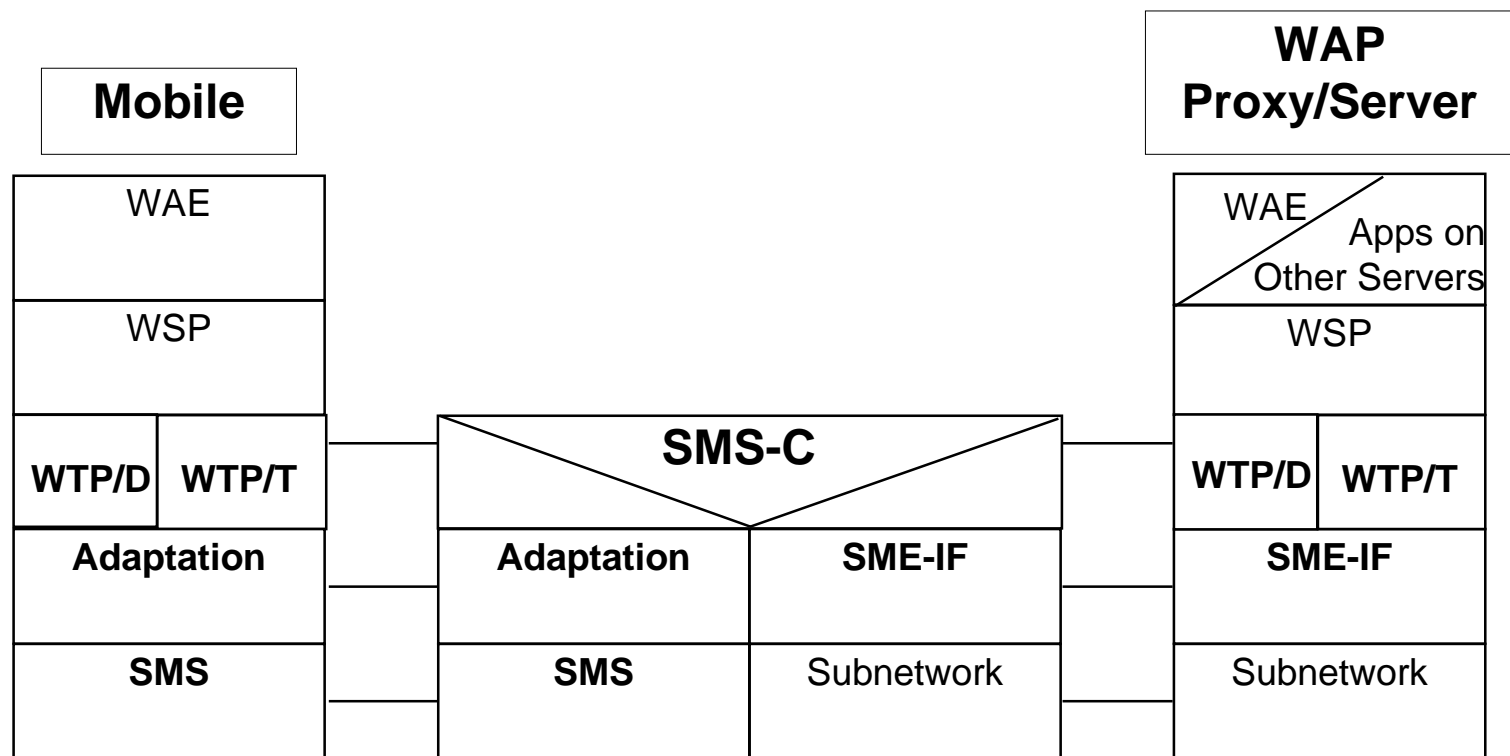
WTP Over GSM Circuit-Switched



RAS - Remote Access Server
IWF - InterWorking Function

Service, Protocol, and Bearer Example

WTP Over GSM Short Message Service



defined in the WTP Specification

Work Items for 1998

Work items for WTP

1Q98 - v1.0

- complete protocol definition for WTP/T
- include architecture and protocol solutions for IS-136, CDMA, PDC
- service primitive definition finalization
- quality of service
- conformance statement

during 1998

- complete protocol definition for WTP/C
- management entity definition (functions and interface)
- tuning/performance data required (timer default settings)
- protocol verification (reference implementation)
- API for WTP

Wireless Session and Security Protocols Overview

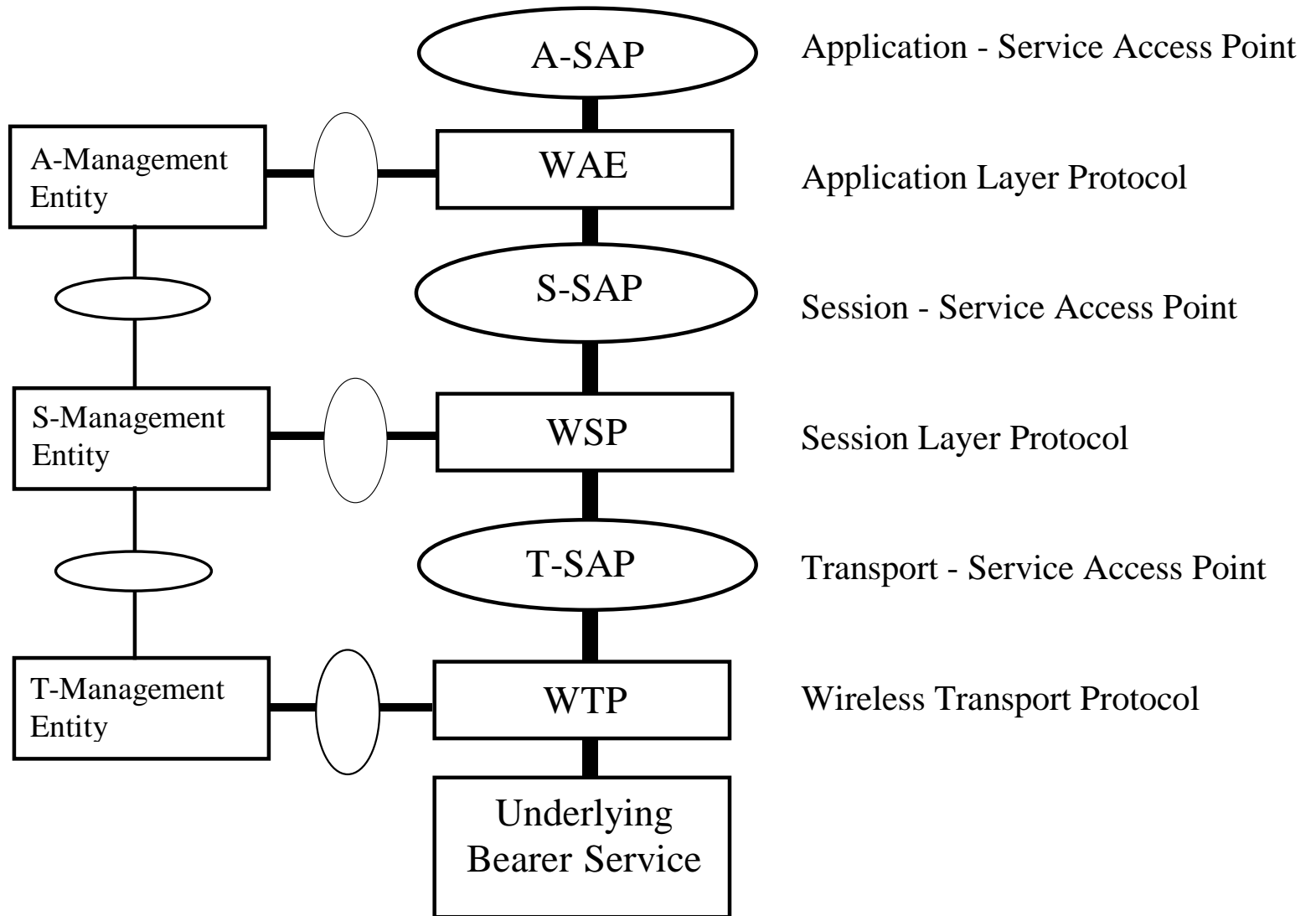
Areas of responsibility

- WSP/B Specification
 - Provides upper layers with session services and management
 - Provides semantics and mechanisms based on HTTP 1.1
- WTLS Specification
 - Provides mechanisms for secure and authenticated communication
 - Based on SSL and TLS, optimized for use over wireless networks

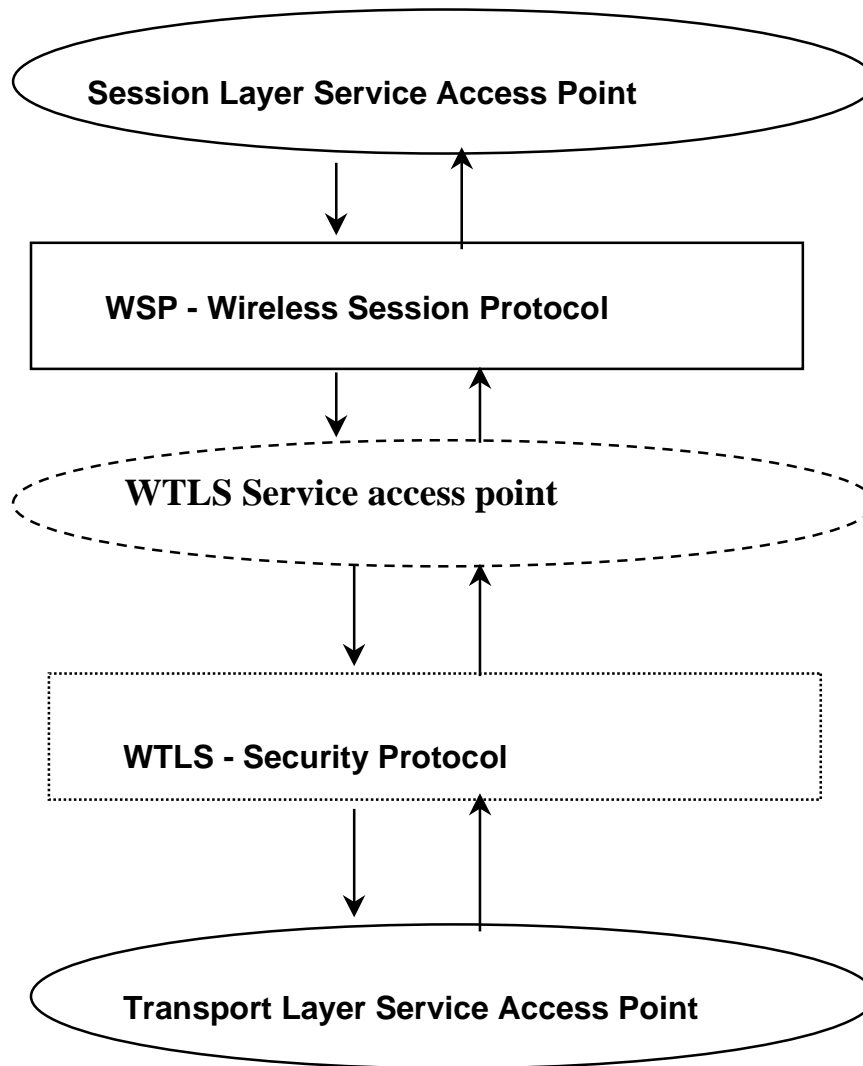
Areas of responsibility (cont.)

- **Possible other session protocols**
 - **None identified yet; WSP/B provides necessary functionality for currently identified application protocols**

WAP Protocol Architecture



Internal Layer Architecture



Goals and Requirements for WSP/B

- **Support current WAP transport protocols**
- **Provide HTTP 1.1 functionality**
 - Extensible Request/reply methods
 - Composite objects
 - Content type negotiation
- **Exchange client and server session headers**
- **Interrupt transactions in process on the server**
- **Push content from server to client asynchronously**
- **Key management**
- **Authentication**
- **Built on existing Internet security specifications**
 - Proven and understood technology
- **Negotiate support for multiple, simultaneous, asynchronous transactions**
- **Application acknowledgement**
- **Session suspend and resume**

Goals and Requirements for WSP/B

- **Support low-capacity bearer networks and devices with limited processor and memory resources**
- **Ensure interoperability between products from different vendors**

WSP/B Specification Status

- **Stable WAP Members' Draft of WSP/B will be available during January '98**
- **Will not be specified**
 - **Management entity**
 - **WSP API**

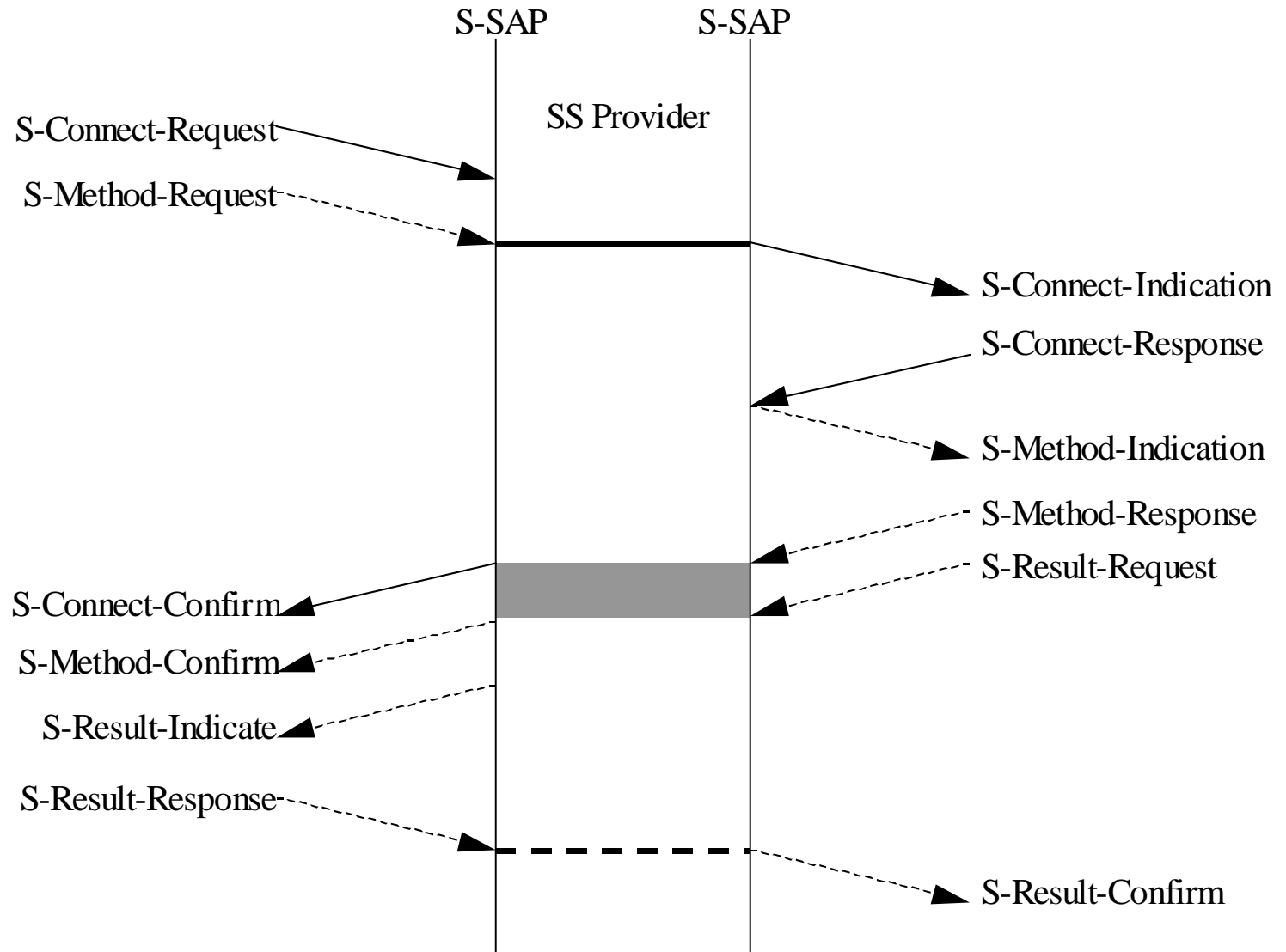
WSP Technical highlights

- **Based on HTTP 1.1**
- **Incorporates features from HDTP 1.1**
- **Builds on a request/reply mechanism**
- **Efficient usage of bandwidth is made possible by introducing compact binary encodings of well-known headers and content types**
- **Session suspend and resume**
- **Provides also a light-weight unreliable session service on top of datagram transports for implementation in low-end devices**
- **Supports performance enhancement features such as asynchronous requests and pipelining**

Session creation and method invocation

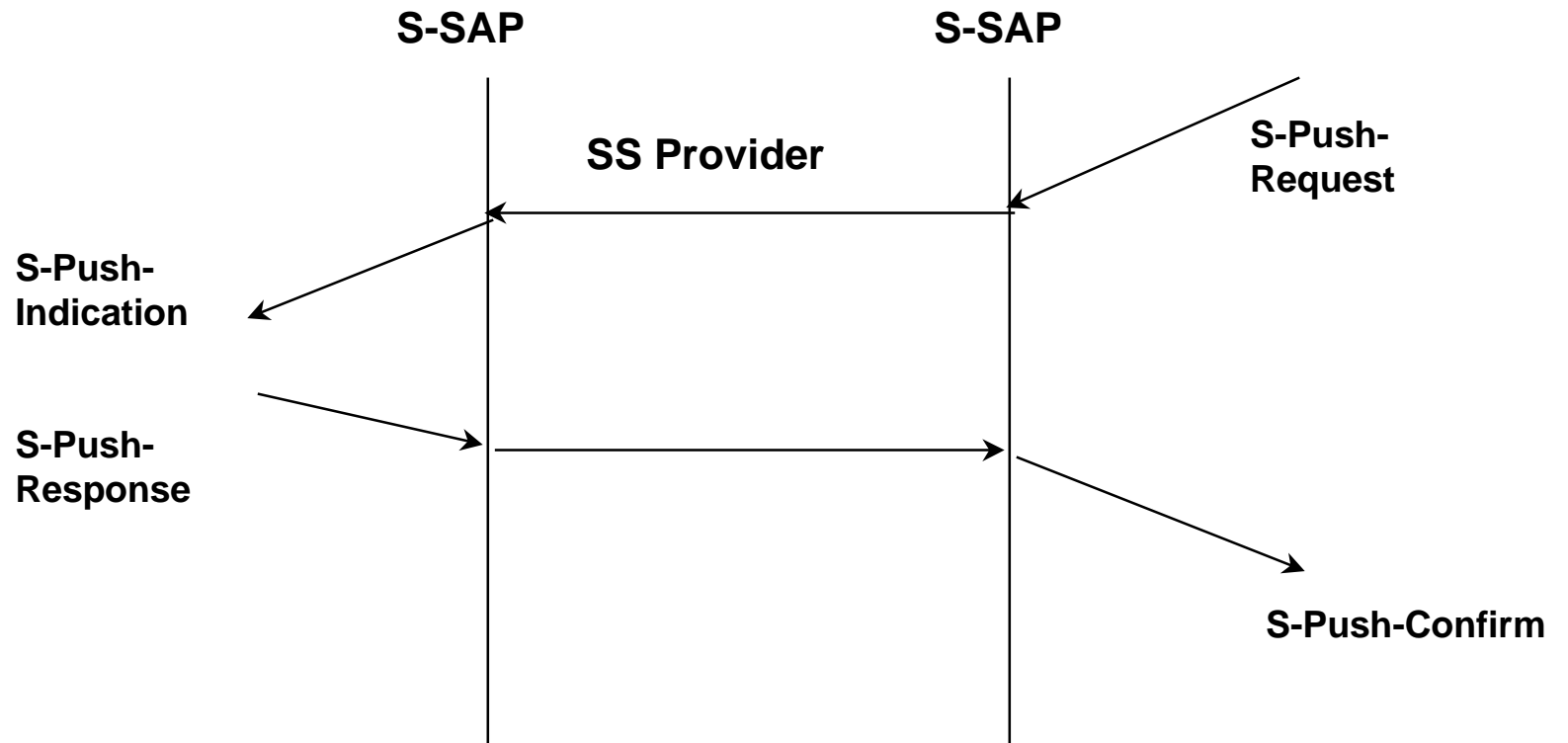
- **The session establishment phase is entered when the session is created. During this phase, the session layer performs the following functions:**
 - Exchange of the client and server session headers
 - Exchange of the session identifier
 - Negotiation of session protocol capabilities
 - Optional data transfer

Session creation and method invocation



Data push mechanisms

The example given below shows the case of unidirectional, reliable data push.



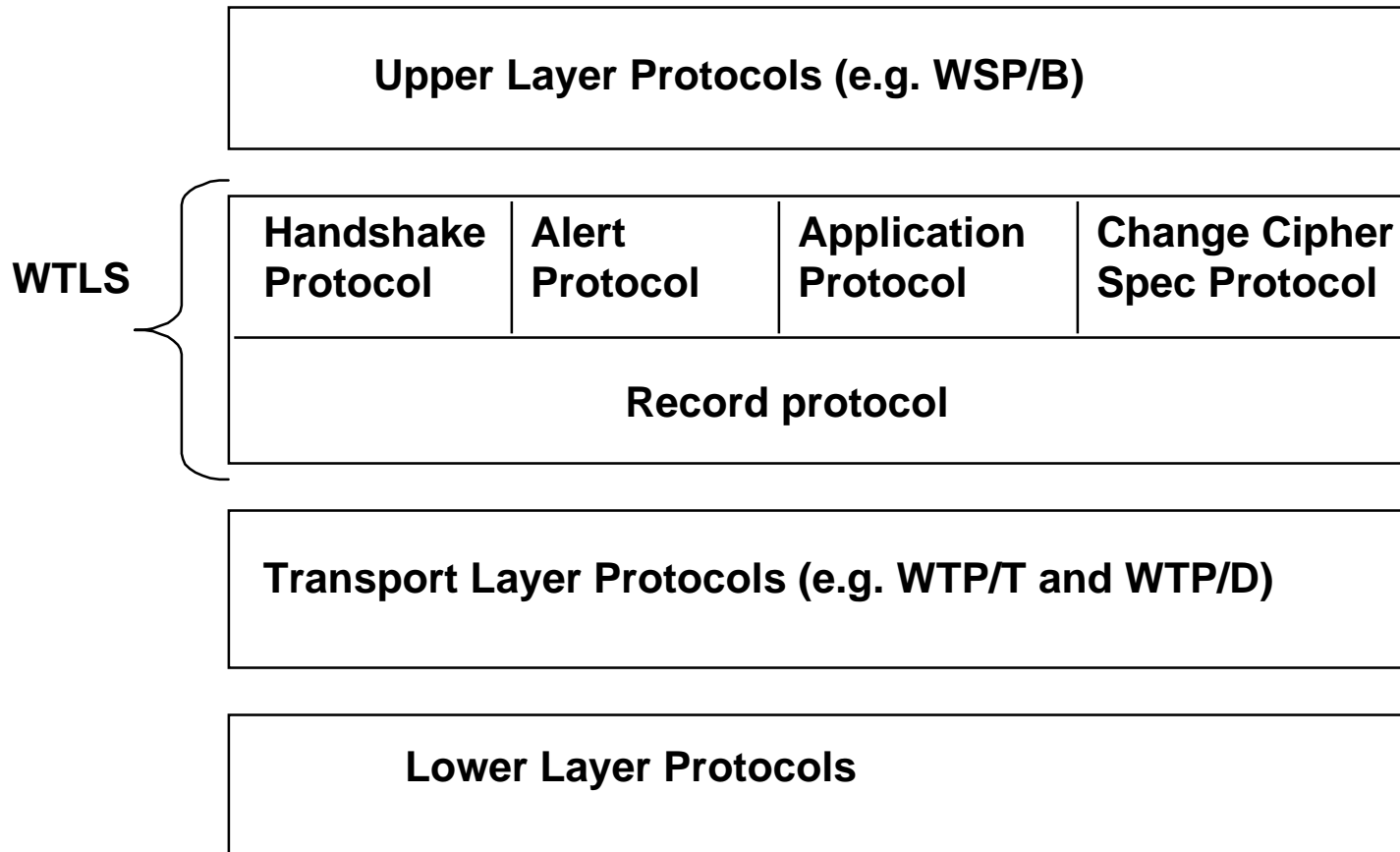
WSP Work items

- **Support for Quality of Service parameters**
- **Multicast data**
- **Ordered pipelining**
- **Chunked data transfer**
- **WSP Management entity**
- **Support for isochronous multimedia objects**
- **Other extended functionality that is not a part of the first public version**

Goals and Requirements for WTLS

- **Provide mechanisms for secure, authenticated communication**
 - Data encryption using available encryption algorithms
- **Provides an extensible architecture for security protocols**
 - New cryptographic algorithms can be added
- **Interoperability**
- **Lightweight and efficient with respect to bandwidth and processing power**
- **Support current WAP transport protocols**

WTLS Internal Architecture



WTLS Specification Status

- **Independent third party security review will be arranged**
- **WAP Members' Draft available during January**

WTLS Future evolution

- **Recommend cryptographic algorithms**
- **Smartcard support**

Wireless Application Environment Overview

WAE Overview

- **Application framework**
 - **For network-oriented applications;**
 - **On small, narrowband devices**
- **Developed by**
 - **Wireless Applications Group (WAG);**
 - **A WAP technical working group.**

WAE Goals

- **Network-neutral application environment;**
- **For narrowband wireless devices;**
- **With an Internet (WWW) programming model;**
- **And a high degree of interoperability.**

WAE Requirements

- **Leverage WSP and WTP**
- **Leverage Internet standard technology**
- **Device Independent**
- **Network Independent**
- **International Support**

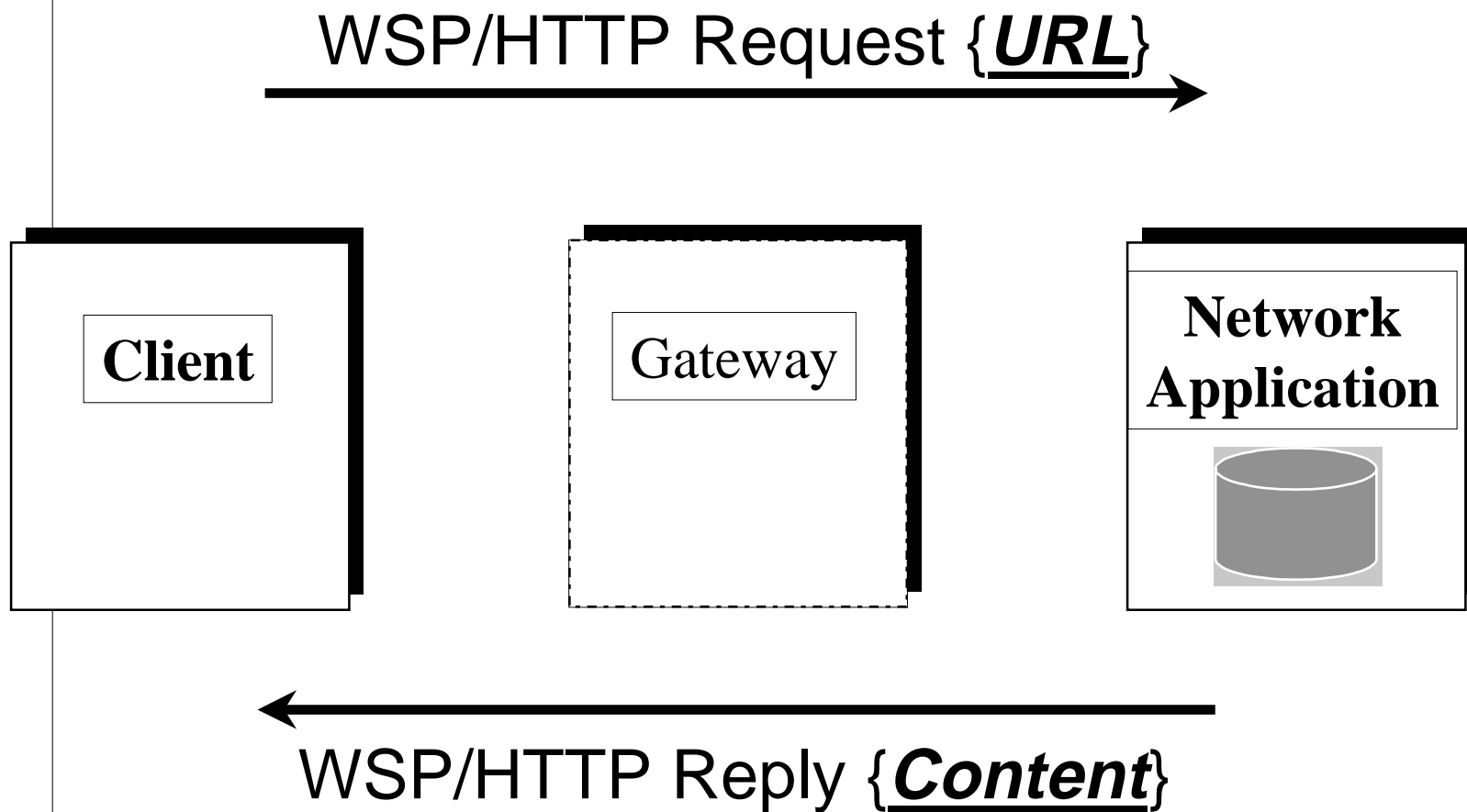
Requirements (cont.)

- **Vendor-controlled MMI**
- **Initial focus on phones**
 - **Slow bearers**
 - **Small memory**
 - **Limited CPU**
 - **Small screen**
 - **Limited input model**

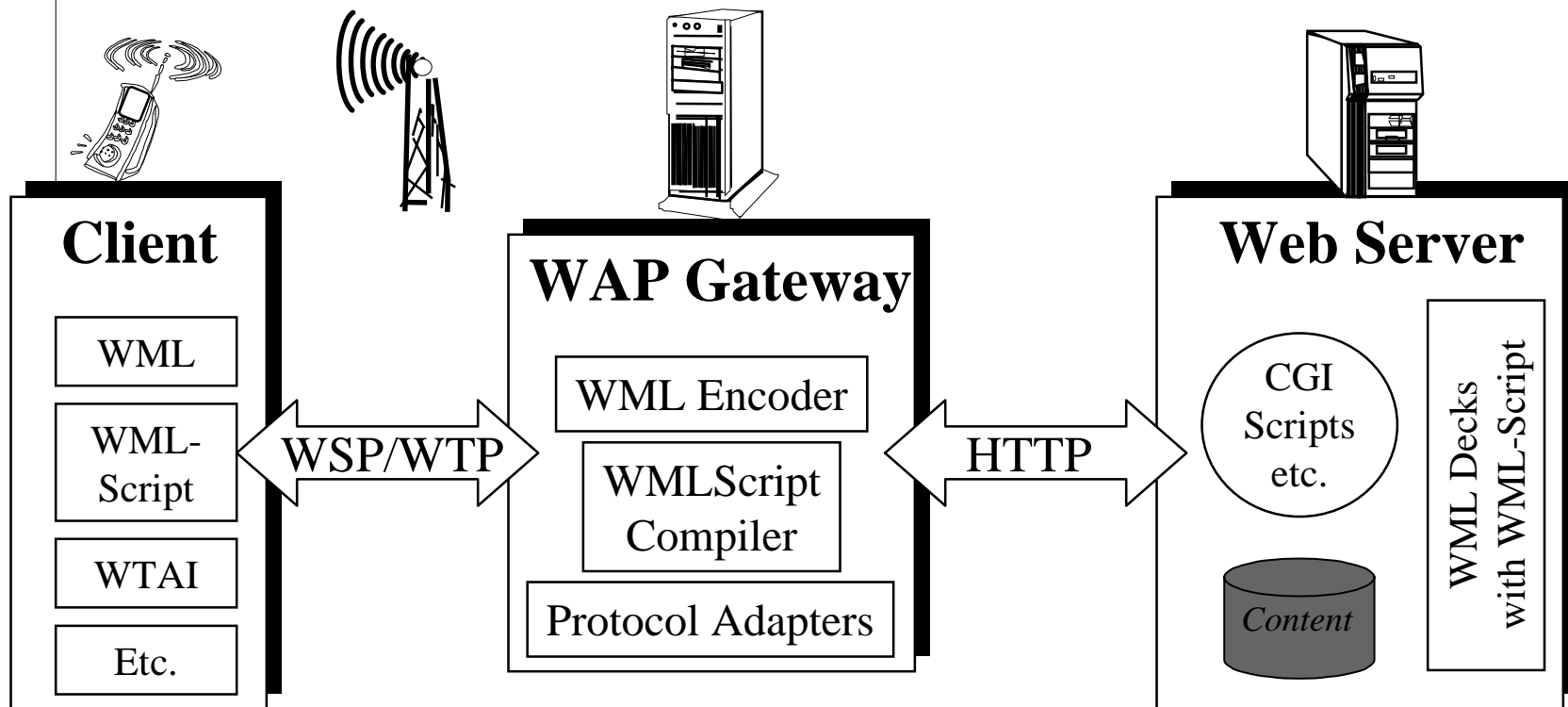
WAE Components

- **Architecture**
 - Application model
 - Browser, Gateway, Content Server
- **WML**
 - Display language
- **WMLScript**
 - Scripting language
- **WTA**
 - Telephony services API and architecture
- **Content Formats**
 - Data exchange

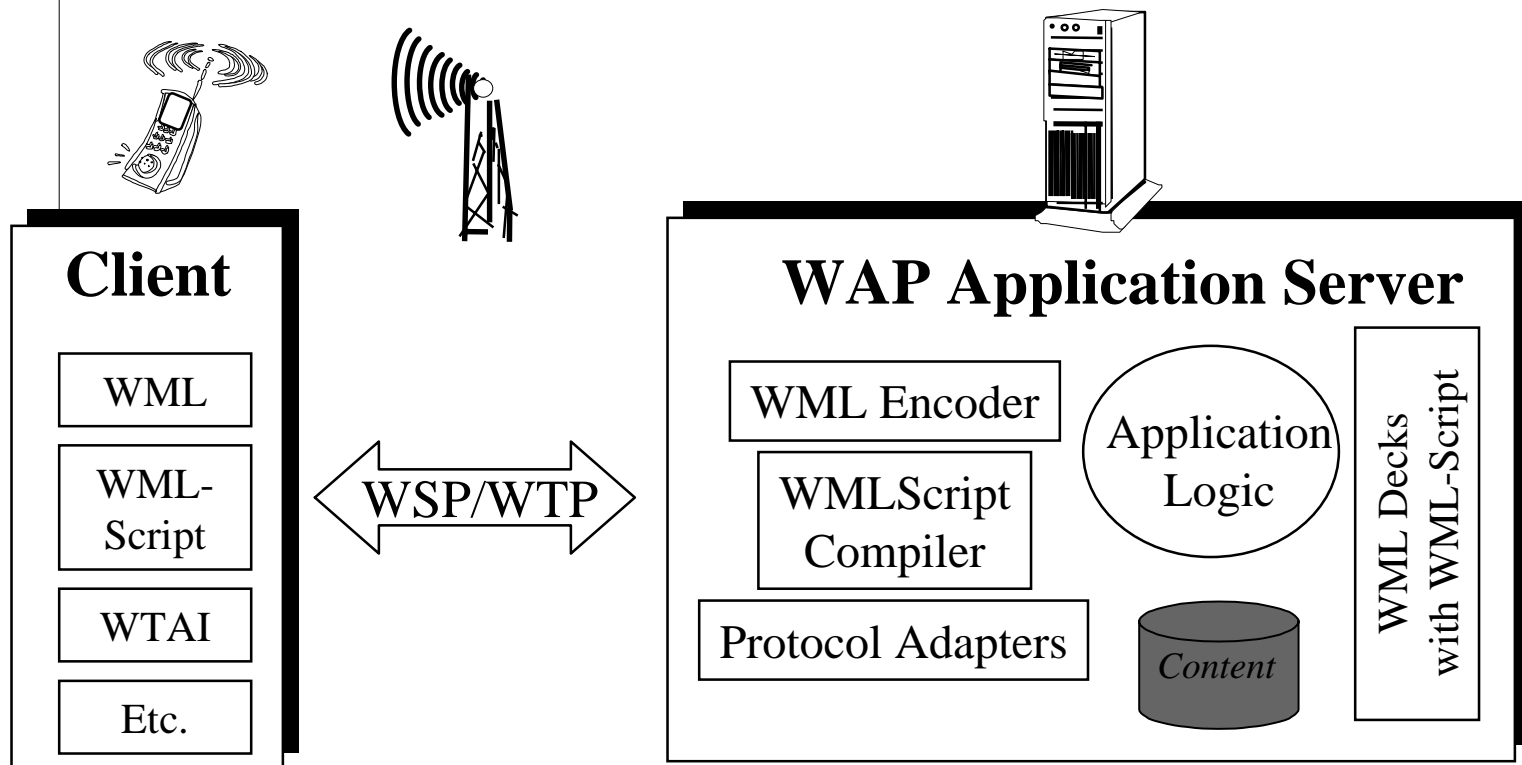
WAE Abstract Network Architecture



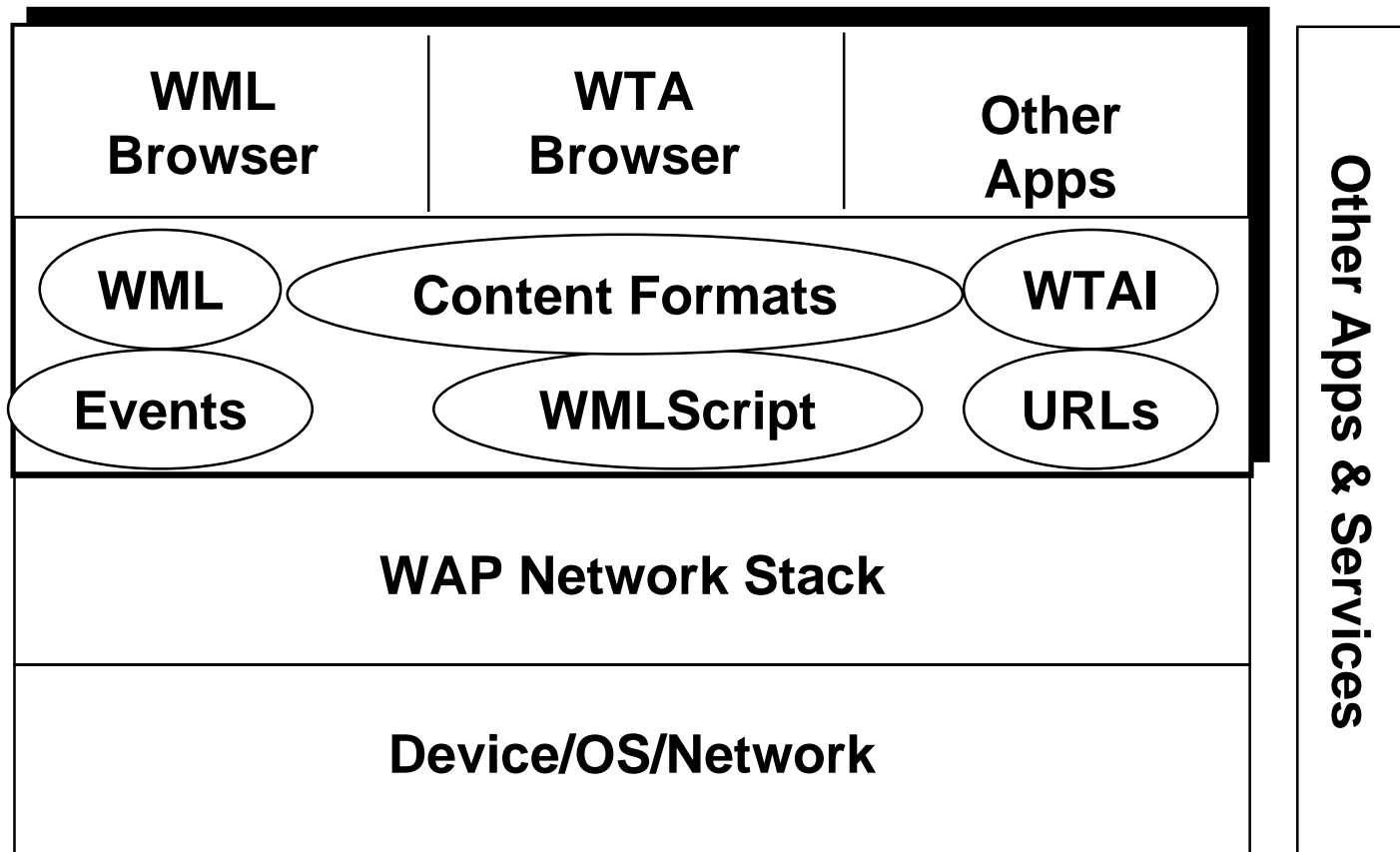
Network Example #1: WAP Gateway



Network Example #2: WAP Application Server



WAE In-Device Architecture



WML

- **Tag-based browsing language:**
 - **Screen management (text, images)**
 - **Data input (text, selection lists, etc.)**
 - **Hyperlinks & navigation support**
- **Syntax: W3C XML (SGML derivative)**
- **Inherits technology from HDML and HTML**

WML (cont.)

- **Card metaphor**
 - User interactions are split into *cards*
 - Navigation occurs between cards
- **Explicit inter-card navigation model**
 - Hyperlinks
 - UI Event handling
 - History
- **State management and variables**
 - Reduce network traffic
 - Results in better caching

WML Example

Navigation

Variables

Input
Elements

```
<WML>
  <FORM>
    <DO TYPE="ACCEPT" TASK="GO"
      URL="#eCard" />
    Welcome!
  </FORM>
  <FORM NAME="eCard">
    <DO TYPE="ACCEPT" TASK="GO"
      URL="/submit?N=$(N)&S=$(S)" />
    Enter name: <INPUT KEY="N" />
    Choose speed:
    <SELECT KEY="S">
      <OPTION VALUE="0">Fast</OPTION>
      <OPTION VALUE="1">Slow</OPTION>
    <SELECT>
  </FORM>
</WML>
```

Card

Deck

WMLScript

- **Scripting language:**
 - Procedural logic, loops, conditionals, etc.
 - Optimized for small-memory, small-cpu devices
- **Derived from JavaScript™**
- **Integrated with WML**
 - Powerful extension mechanism
 - Reduces overall network traffic

WMLScript (cont.)

- **Bytecode-based virtual machine**
 - **Stack-oriented design**
 - **ROM-able**
 - **Designed for simple, low-impact implementation**
- **Compiler in network**
 - **Better network bandwidth use**
 - **Better use of terminal memory/cpu.**

WMLScript Standard Libraries

Lang - VM constants, general-purpose math functionality, etc.

String - string processing functions

URL - URL processing

Browser - WML browser interface

Dialog - simple user interface

Float - floating point functions

WMLScript Example Uses

In general: reduce network round-trips and enhance functionality.

Field validation

- *Check for formatting, input ranges, etc.*

Device extensions

- *Access device or vendor-specific API*

Conditional logic

- *Download intelligence into the device*

WMLScript Example

WMLScript is very similar to JavaScript™

Functions

```
function currencyConvertor(currency, exchRate) {  
    return currency*exchangeRate;  
}
```

Variables

```
function myDay(sunShines) {  
    var myDay;  
    if (sunShines) {  
        myDay = "Good";  
    } else {  
        myDay = "Not so good";  
    };  
    return myDay;  
}
```

**Programming
Constructs**

WML & WMLScript Example

**WMLScript
Function**

```
<WML>  
<SCRIPT TYPE="text/wmlscript">  
  function check(N) {  
    var i = parseInt(N);  
    if (i == "NaN" || i < 0 || i > 100)  
      Browser.setVar("Msg", "Error");  
    else  
      Browser.setVar("Msg", "OK");  
    Browser.go("#displayMsg");  
  }  
</SCRIPT>
```

WML Cards

```
<FORM>  
  <DO TYPE="ACCEPT" TASK="GO" URL="#check($N)"/>  
  Enter a number (0 to 100): <INPUT KEY="N"/>  
</FORM>  
<FORM NAME="displayMsg">  
  Number entry: $(Msg)  
</FORM>  
</WML>
```

WTA

- **Tools for building telephony applications**
- **Designed primarily for:**
 - **Network Operators / Carriers**
 - **Equipment Vendors**
- **Network security and reliability a major consideration**

WTA (cont.)

- **WTA Browser**
 - **Separate WML/WMLScript browser**
 - **Exposes additional API (WTAI)**
- **WTAI includes:**
 - **Call control**
 - **Network text messaging**
 - **Phone book interface**
 - **Indicator control**
 - **Event processing**

WTA (cont.)

- **Network model for client/server interaction**
 - Event signaling
 - Client requests to server
- **Security model: segregation**
 - Separate WTA browser
 - Separate WTA port
- **WTAI available in WML & WMLScript**

WTA Example

Placing an outgoing call with WTAI:

WTAI Call
Input Element

```
<WML>  
<FORM>  
  <DO TYPE="ACCEPT" TASK="GO"  
    URL="wtai:cc/mc;$(N)"/>  
  
  Enter phone number:  
  <INPUT TYPE="TEXT" KEY="N"/>  
</FORM>  
</WML>
```

WTA Example

Placing an outgoing call with WTAI:

WTAI Call {

```
<WML>
<COMMON>
<SCRIPT>
  function checkNumber(N) {
    if (Lang.isInt(N))
      WTAI.makeCall(N);
    else
      Dialog.alert("Bad phone
number");
  }
</SCRIPT>
</COMMON>
<FORM>
```

Script URL {

```
<DO TYPE="ACCEPT" TASK="GO"
  URL="#checkNumber($N)" />
```

Enter phone number:

Input Element {

```
<INPUT TYPE="TEXT" KEY="N" />
```

```
</FORM>
```

```
</WML>
```


Content Formats

Common interchange formats

Promoting interoperability

Formats:

- Business cards: IMC **vCard** standard
- Calendar: IMC **vCalendar** standard
- Images: WBMP (Wireless BitMaP)
- Compiled WML, WMLScript

New WAP Content Formats

Newly defined formats:

- WML text and tokenized format
- WMLScript text and bytecode format
- WBMP image format

Binary format for size reduction

- Bytecodes/tokens for common values and operators
- Compressed headers
- Data compression (e.g. images)

**General-purpose transport compression
can still be applied**

Content Format Example

Example Use of an Image:

Image Element {

```
<WML>
<FORM>
  Hello World!<BR/>
  <IMG SRC="/world.wbmp"
      ALT="[Globe]" />
</FORM>
</WML>
```

Summary: WAE Status

- **Documents approaching WAP Members' Draft status**
- **Primary documents:**
 - **WAE specification**
 - **WAE architecture**
 - **WML specification**
 - **WMLScript specification**
 - **WTAI specification**