# WAP Overview

An overview of the *'Wireless Application Protocol' to the IAB*.

# WAP Overview

## *Introduction*

WAP enabled
GSM Phone

Request for Information

Response

Application
Server

WAP enabled
PDAs

Mobile device with wireless communications, WAP protocol stack and Microbrowser environment to render information

Application server delivering appropriate content meeting the clients needs

# WAP Overview

## *WAP Forum ™*

- **What is the WAP Forum™?**
  - An Industry Forum established in 1997 whose stated aim is *'..to develop the de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals.'*.
  - Comprised over 200 members
    - Network Operators, e.g. DeTeMobile, Cellnet, Sonera..
    - Network Infrastructure suppliers, e.g. Ericsson, Nokia...
    - Mobile Device suppliers, e.g. Ericsson, Nokia, Panasonic....
    - IT companies, e.g. IBM, HP, Microsoft, Sun...
    - Content developers

# WAP Overview

## *WAP Technology*

- **What is WAP Technology ?**
  - WAP Technology is an architecture and set of specifications that provide :
    - Access to information/services *efficiently*
    - Wireless Application Environment (WAE) targeted at devices with constrained size, power, MMI etc..
    - Protocol stack permitting network and bearer agnostic use by the Wireless Application Environment.
    - Transport layer security with authentication and encryption suitable for wireless devices.
    - Additional specifications for Wireless Telephony Applications (WTA), PUSH etc..
    - Workplan addressing many emerging network, bearer, application and security improvements

# WAP Overview

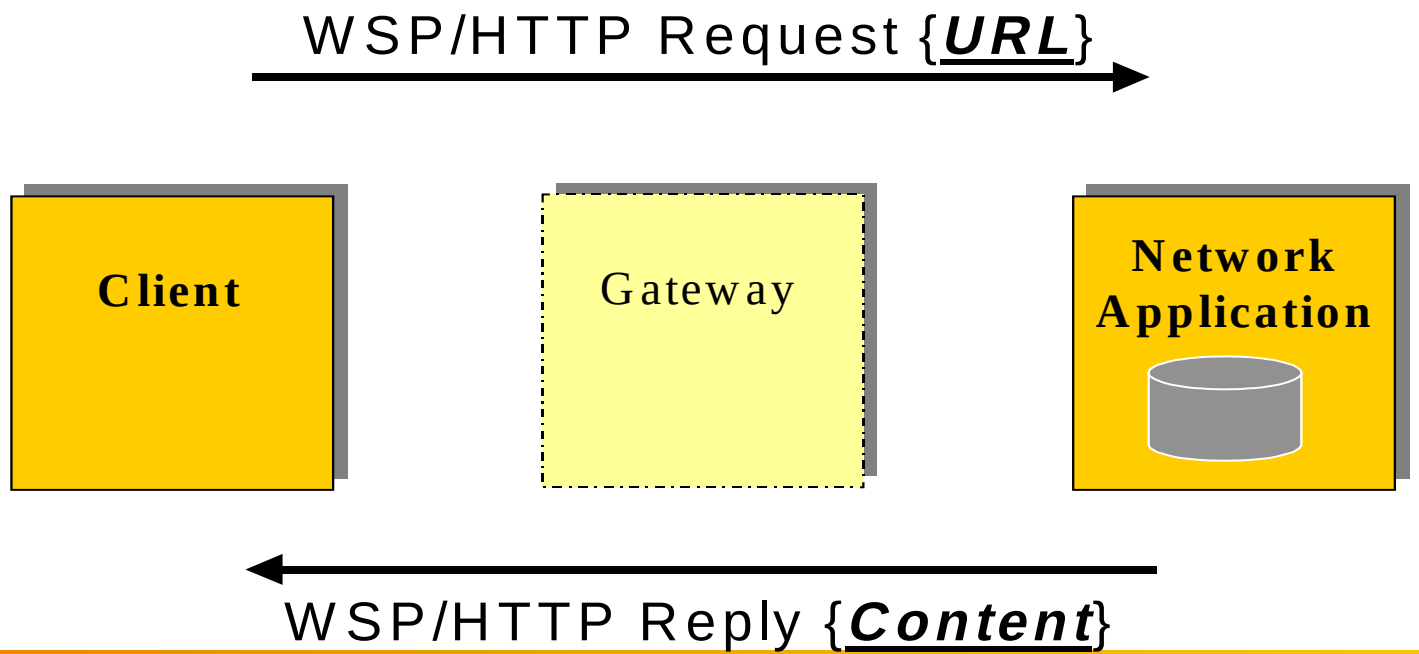
## *The Inspirations*

- **What are/were the inspirations for WAP Technology ?**
  - The Internet
    - Browser as thin client,
    - IP as transport,
    - Transport security.
  - Nokia Smart Messaging
    - Efficient use of SMS for delivery of information,
    - Deployed in many networks.
  - Narrow Band Sockets & TTML
  - Phone.com's HDML browser and UP.Link server
    - Lightweight browser,
    - Services centric.

# WAP Overview

## *Abstract Network Architecture*

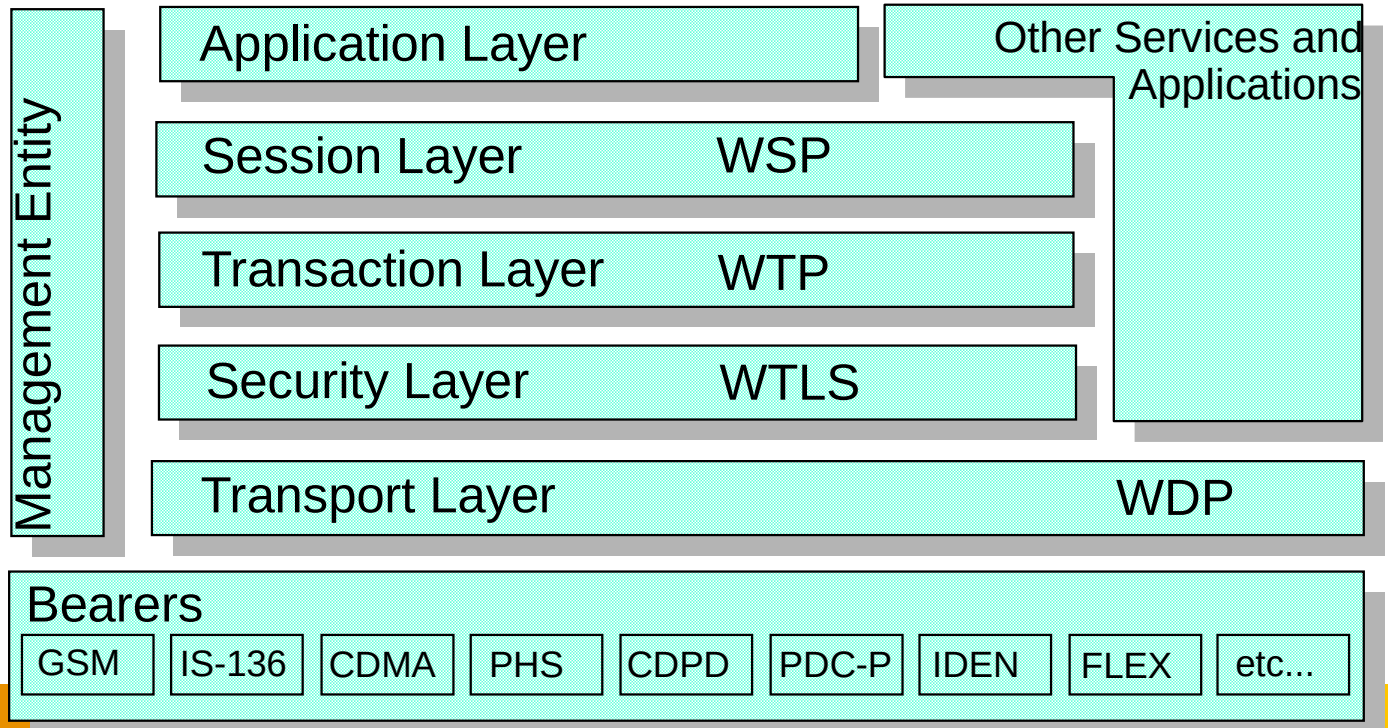
- **What is the WAP Abstract Network Architecture ?**
  - The WAP Abstract Network Architecture resembles the Internet Browser Request-Response model.



# WAP Overview

## *WAP Stack*

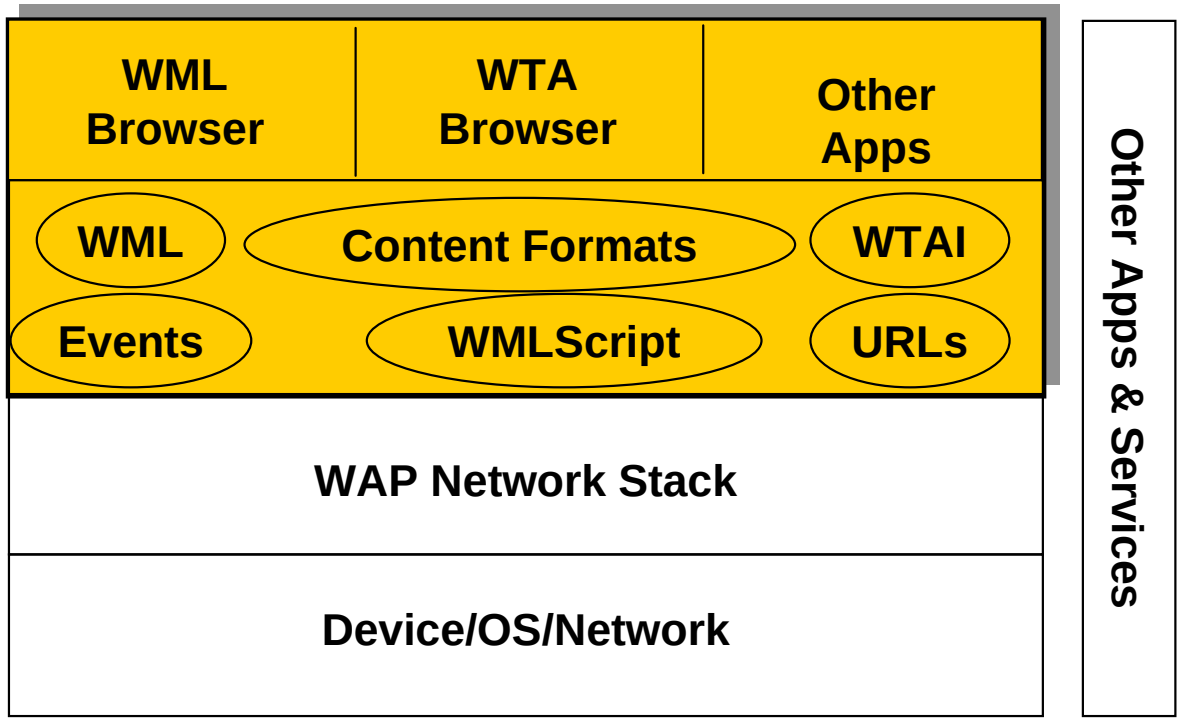
- **What is the WAP Protocol Stack ?**
  - The WAP Protocol Stack is based on the Internet protocol stack as used by a Browser application.



# WAP Overview

## *Application Environment*

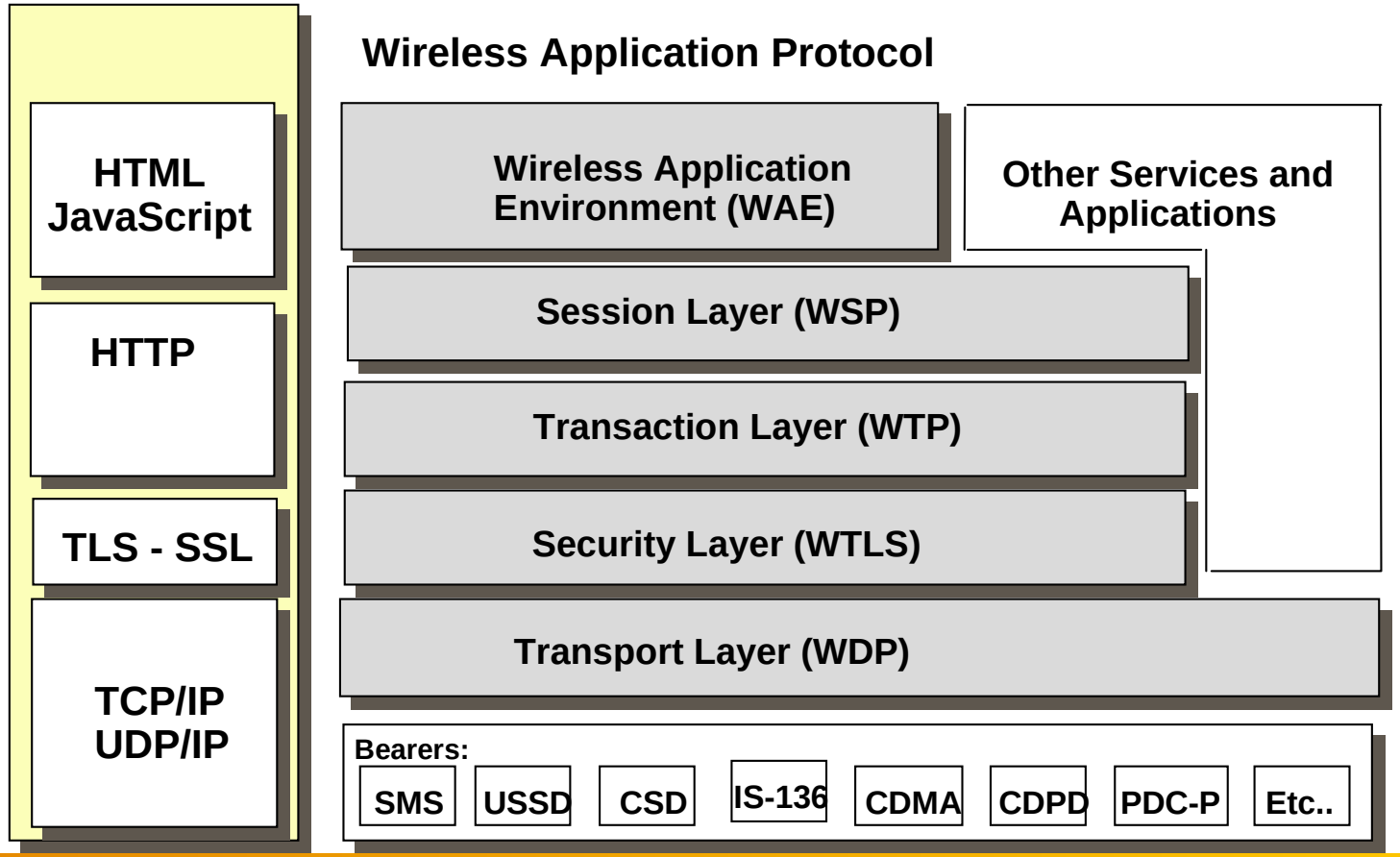
- **What is the WAP Application Environment ?**
  - WAE consists of the WML Browser, optional WTA Browser, Script, telephony support, events etc..



# WAP Overview

## *Comparison with the Internet*

- **This all looks similar to the Internet - what are the differences ?**



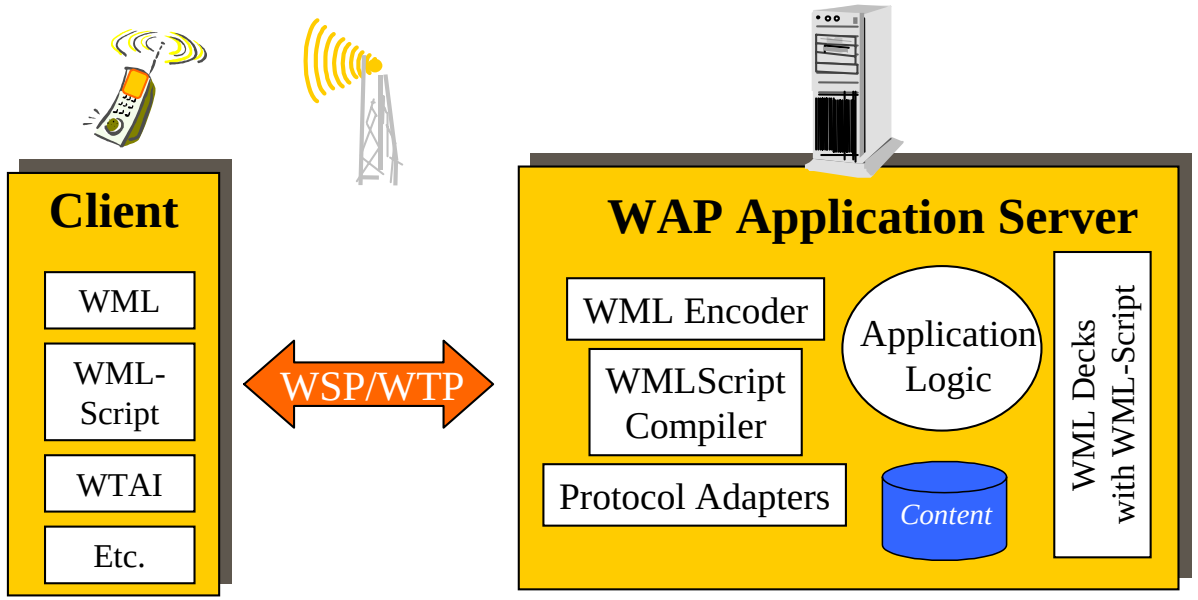
# WAP Overview

## *Deployment Models*

- **WAP deployment models.**
  - Two basic approaches
    - WAP WSP Server :
      - WAP Protocol Stack, User Admin, Content management
      - Web Server
    - WAP Proxy :
      - WAP Procotol Stack, User Admin, Content Management
      - HTTP Client to request content from other servers

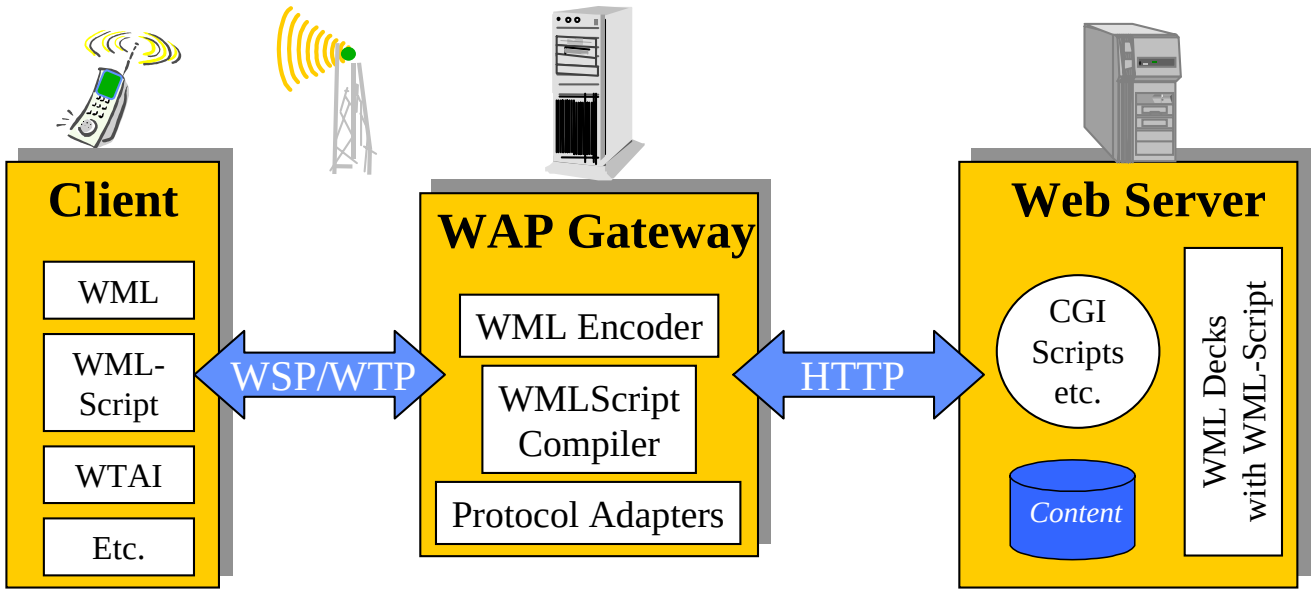
# WAP Overview

## *Deployment Models - WSP Server*



# WAP Overview

## *Deployment Models - WSP Proxy*



# WAP Overview

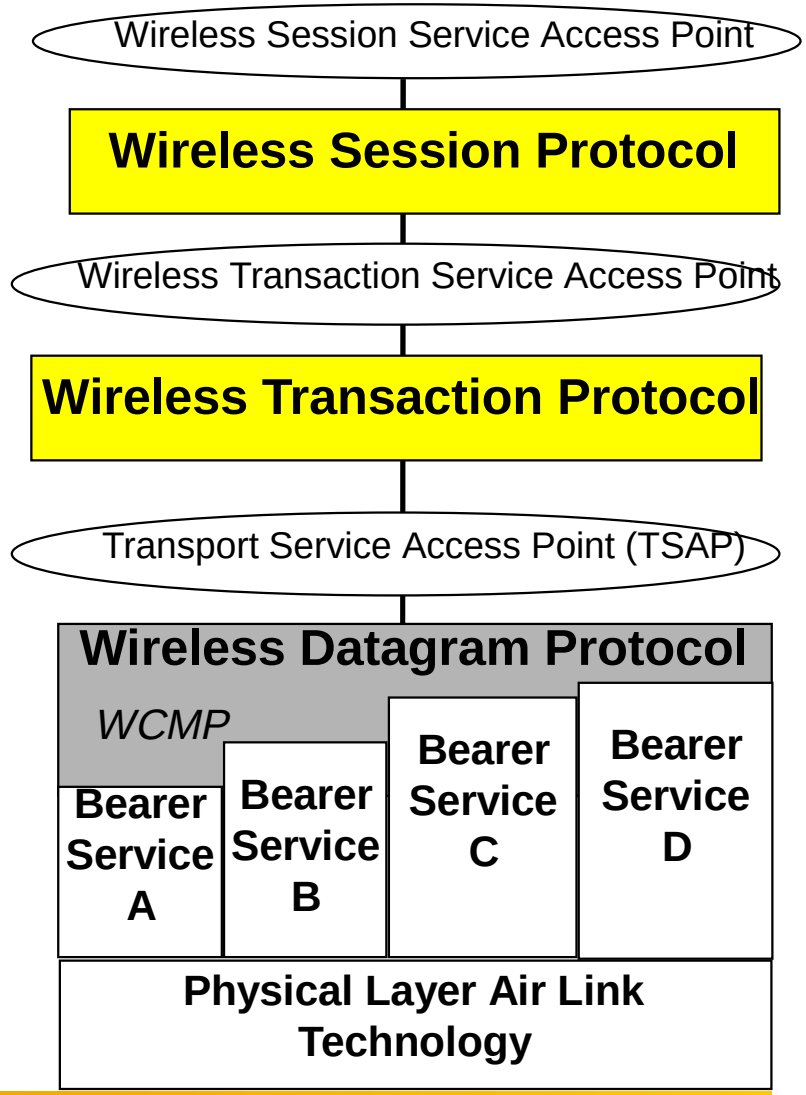
## *Protocol Stack*

# Protocol Stack

# WAP Overview

## *Protocol Stack*

- **Protocol Stack :**
  - WSP - Session Protocol
  - WTP - Transaction Protocol
  - WDP - Datagram Protocol
    - WCMP - Control Message Protocol



# WAP Overview

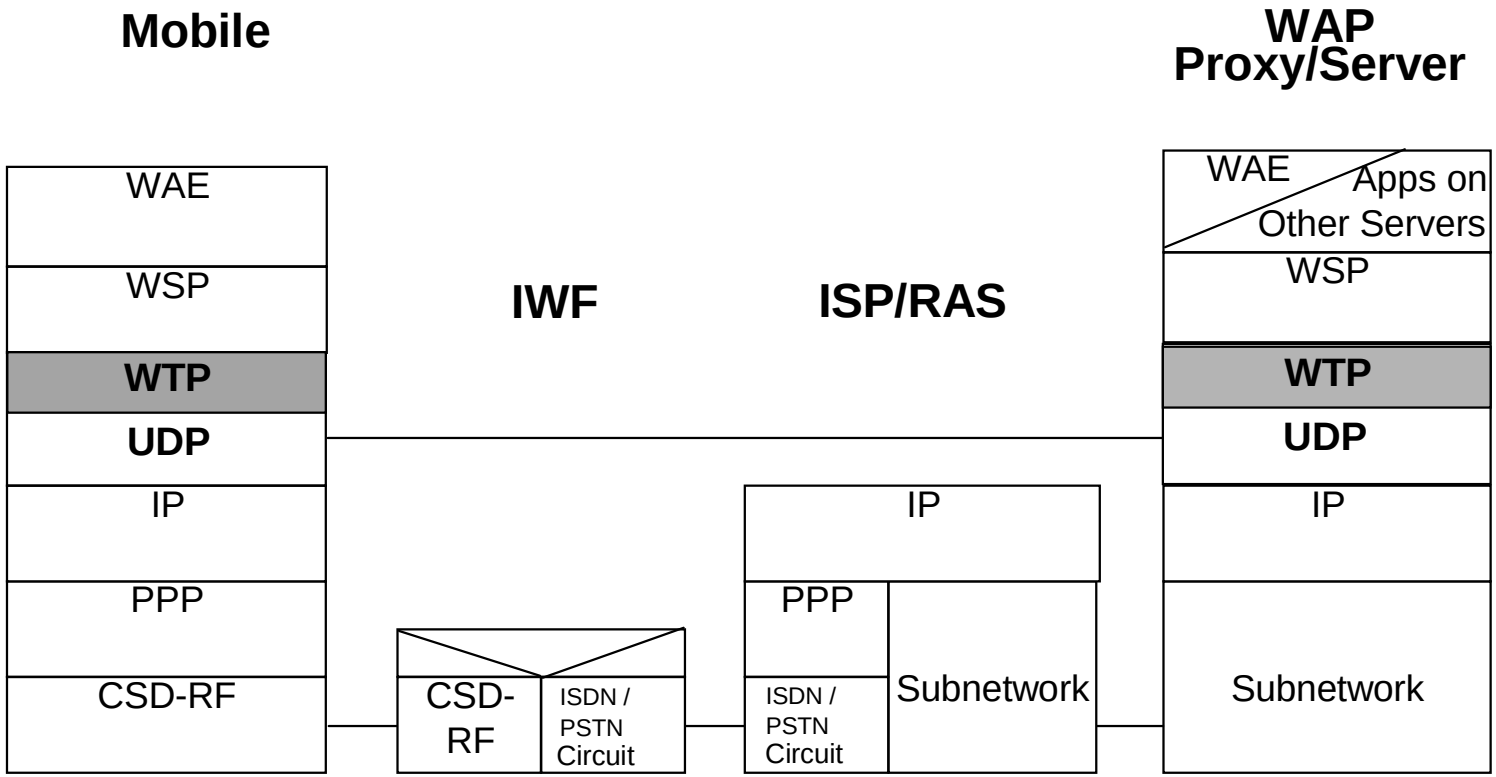
## *Protocol Stack - Datagram (WDP)*

- **WDP - Wireless Datagram Protocol :**
  - Provides a connection-less, unreliable datagram service,
  - WDP is replaced by UDP when used over an IP network layer (UDP/IP)
  - WDP uses the Service Primitive **T-UnitData.req.ind**
  - Bearers currently supported
    - GSM SMS, USSD, C-S Data, GPRS
    - ANSI-136 R-Data, C-S Data, Packet
    - CDMA SMS, C-S Data
    - PDC C-S Data, Packet
    - PHS C-S Data
    - CDPD
    - iDEN SMS, C-S Data, Packet
    - FLEX and ReFLEX
    - DataTAC, Mobitex, etc.

# WAP Overview

## *Protocol Stack - Datagram (WDP)*

- **WDP - Wireless Datagram Protocol :**
  - Example of connectivity : GSM Circuit-Switched.

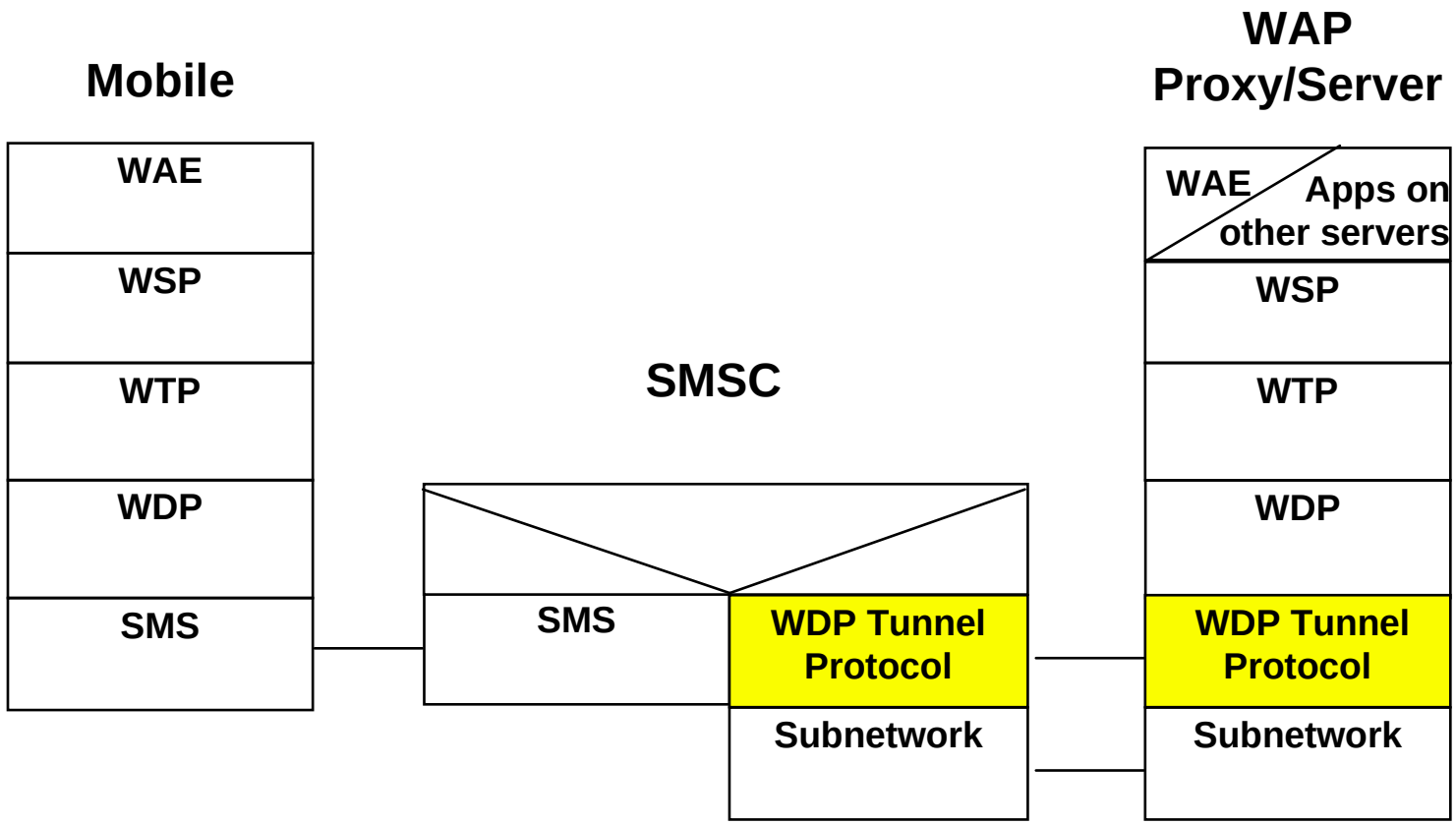



IWF - InterWorking Function  RAS - Remote Access Server

# WAP Overview

## *Protocol Stack - Datagram (WDP)*

- **WDP - Wireless Datagram Protocol :**
  - Example of connectivity : GSM SMS.




under development

# WAP Overview

## *Protocol Stack - Transaction (WTP)*

- **WTP - Wireless Transaction Protocol :**
  - What is its purpose ?
    - Provides efficient request/reply based transport mechanism suitable for devices with limited resources over networks with low to medium bandwidth.
  - What are the advantages ?
    - Operator Perspective - Load more subscribers on the same network due to reduced bandwidth utilization.
    - Individual User - Performance is improved and cost is reduced.
  - Why not TCP/IP ?
    - Less efficient (connection oriented with opens/closes, problems with long latency networks etc..
    - TTCP is an inspiration for WTP.

# WAP Overview

## *Protocol Stack - Transaction (WTP)*

- **WTP - Wireless Transaction Protocol :**
  - Classes of Operation
    - WTP Classes of Service
    - Class 0 Unconfirmed Invoke message with no Result message
      - a datagram that can be sent within the context of an existing WSP (Session) connection
    - Class 1 Confirmed Invoke message with no Result message
      - used for data push, where no response from the destination is expected
    - Class 2 Confirmed Invoke message with one confirmed Result message
      - a single request produces a single reply

# WAP Overview

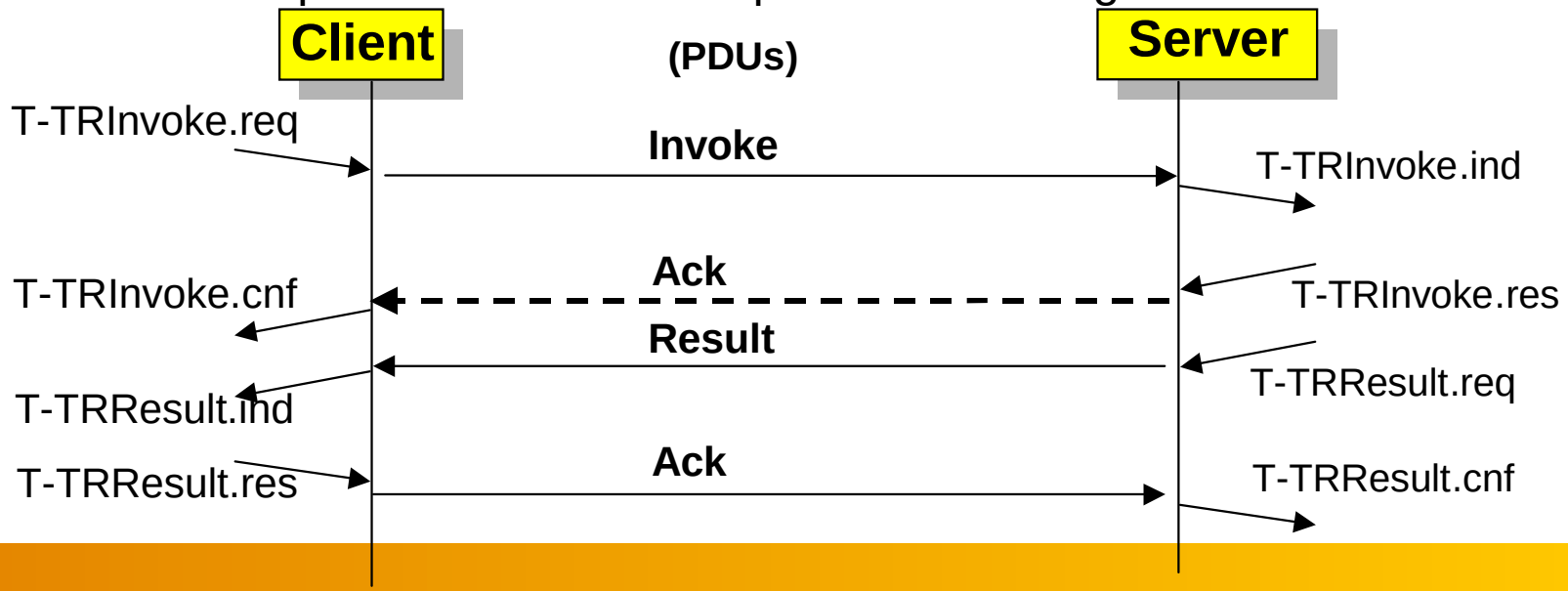
## *Protocol Stack - Transaction (WTP)*

- **WTP - Wireless Transaction Protocol :**
  - 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 (UDP ports numbers)
    - flow control
  - message oriented (not stream)
  - supports an Abort function for outstanding requests
  - supports concatenation of PDUs

# WAP Overview

## *Protocol Stack - Transaction (WTP)*

- **WTP - Wireless Transaction Protocol :**
  - 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 class 2 protocol exchange



# WAP Overview

## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Overview
    - Provides shared state between client and server used to optimize content transfer
    - Based on HTTP V1.1
    - Enhancements for WAE, wireless networks and “low-end” devices
      - Compact encoding
      - Efficient negotiation
      - Push
      - Capability negotiation
      - Suspend and resume
      - Fully asynchronous requests
      - Connectionless service

# WAP Overview

## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Enhancements beyond HTTP V1.1
    - Binary header encoding
    - Session headers
    - Confirmed and non-confirmed data push
    - Capability negotiation
    - Suspend and resume
    - Fully asynchronous requests
    - Connectionless service

# WAP Overview

## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Why Not HTTP V1.1 ?
    - Encoding not compact enough
    - No push facility
    - Inefficient capability negotiation
  - Header Encoding
    - Defined compact binary encoding of headers, content type identifiers and other well-known textual or structured values
    - Reduces the data actually sent over the network

# WAP Overview

## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Capabilities are defined for:
    - Message Size, client and server
    - Protocol Options: Confirmed Push Facility, Push Facility, Session Suspend Facility, Acknowledgement headers
    - Maximum Outstanding Requests
    - Extended Methods
    - Header Code Pages

# WAP Overview

## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Suspend and Resume
    - Server knows when client can accept a push
    - Multi-bearer devices
    - Dynamic addressing
    - Allows the release of underlying bearer resources
  - Session Context and Push
    - Push can take advantage of session headers
    - Server knows when client can accept a push

# WAP Overview

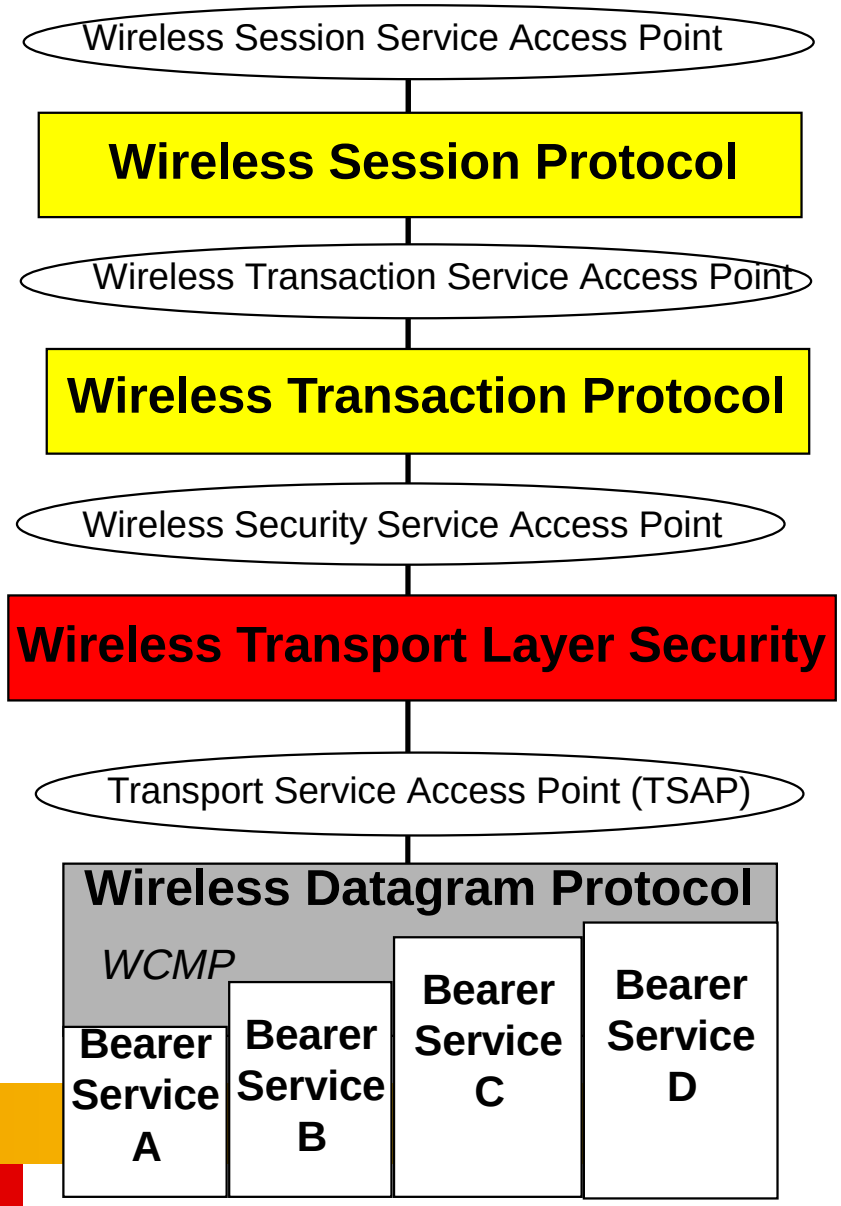
## *Protocol Stack - Session (WSP)*

- **WSP - Wireless Session Protocol :**
  - Connection and Connectionless Modes
    - Connection-mode
      - Long-lived communication
      - Benefits of the session state
      - Reliability
    - Connectionless
      - Stateless applications
      - No session creation overhead
      - No reliability overhead

# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - WTLS Provides Mechanisms for :
    - secure transfer of content for allow for applications needing privacy,
    - identification,
    - verified message integrity
    - and non-repudiation
  - WTLS is Transport level security,
    - based on SSL and TLS from the Internet community



# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - WTLS Services and Characteristics:
    - Specifies a framework for secure connections, using protocol elements from common Internet security protocols like SSL and TLS.
    - Provides security facilities for encryption, strong authentication, integrity, and key management
    - Compliance with regulations on the use of cryptographic algorithms and key lengths in different countries
    - Provides end-to-end security between protocol end points

# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - WTLS Services and Characteristics:
    - Provides connection security for two communicating applications
      - privacy (encryption)
      - data integrity (MACs)
      - authentication (public-key and symmetric)
    - Lightweight and efficient protocol with respect to bandwidth, memory and processing power
    - Employs special adapted mechanisms for wireless usage
      - Long lived secure sessions
      - Optimised handshake procedures
      - Provides simple data reliability for operation over datagram bearers

# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - Goals and Requirements:
    - Interoperable protocols
    - Scalability to allow large scale application deployment
    - First class security level
    - Support for public-key certificates
    - Support for WAP transport protocols

# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - Services & Protocols
    - Provide 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 port number addressing
    - Message oriented (not stream)
    - Supports an Abort function for outstanding requests
    - Supports concatenation of PDUs
    - User acknowledgement or Stack acknowledgement option
      - acks may be forced from the WTP user (upper layer)
      - default is stack ackh

# WAP Overview

## *Protocol Stack - Security (WTLS)*

- **WTLS - Wireless Transport Layer Security :**
  - Uses WDP (UDP/IP) rather than TCP/IP;
  - Supports quick negotiation;
  - Supports DES and ECC;
    - ECC provides stronger encryption strength per key bit;
    - Limits the processing and memory requirements;
  - WTLS security will be between the client and Proxy or WSP Server;
  - WTLS provide end-to-end security only between WTLS endpoints;
  - Does not interoperate with TLS/SSL;
  - Certificate management is an issue at present;
  - Good system design required to achieve overall security of which WTLS is a major part.

# WAP Overview

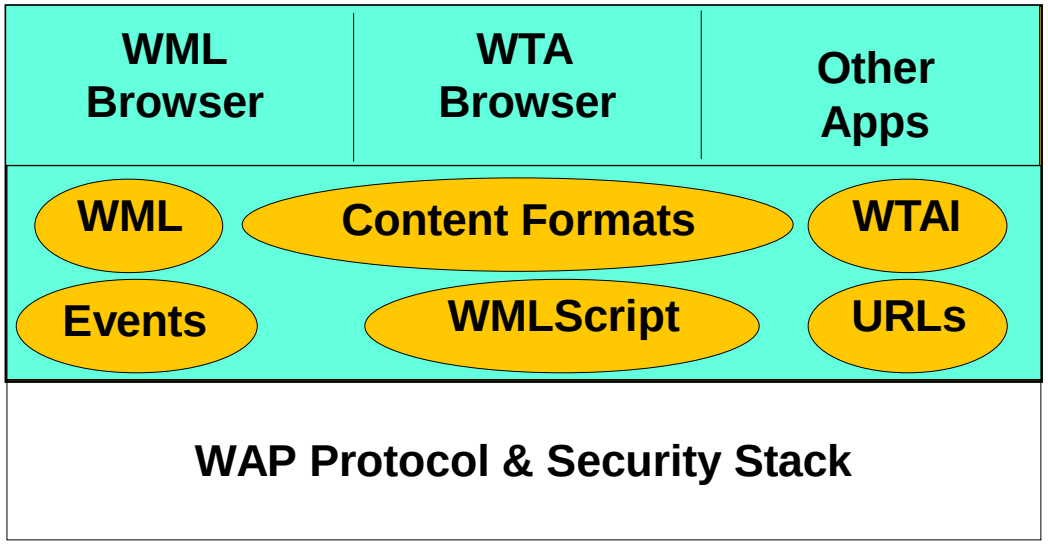
## *Application Environment (WAE)*

- **WAE - Wireless Application Environment :**
  - Objective :
    - Network-neutral application environment,
    - Suitable for narrow-band wireless devices,
    - Permit a high degree of device independence,
    - Use an Internet/WWW programming model,
      - leverage Internet Standard technology,
      - WAP Protocol stack for efficiency,
    - Enable telephony aware applications
    - And a high degree of interoperability and internationalisation.

# WAP Overview

## *Application Environment (WAE)*

- **WAE - Wireless Application Environment :,**



# WAP Overview

## *Application Environment (WAE)*

- **WAE - Content Formats :**
  - Tokenised WML & Compiled WMLScript
  - WBXML
  - Images: WBMP (Wireless BitMaP)
  - Business cards: IMC vCard standard
  - Calendar: IMC vCalendar standard

# WAP Overview

## *Application Environment (WAE)*

- **WAE - WML Explained :**
  - Tag-based browsing language:
  - Screen management (text, images)
  - Data input (text, selection lists, etc.)
  - Hyperlinks & navigation support
  - W3C XML-based language

# WAP Overview

## *Application Environment (WAE)*

- **WAE - WML Card Metaphor Explained :**
  - 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

# WAP Overview

## *Application Environment (WAE)*

- **WAE - WMLScript design point :**
  - Scripting language:
    - Procedural logic, loops, conditionals, etc.
    - Optimized for small-memory, small-cpu devices
  - Based on ECMAScript™
  - Integrated with WML
  - Powerful extension mechanism
  - Reduces overall network traffic

# WAP Overview

## *Application Environment (WAE)*

- **WAE - WMLScript in practice :**
  - Bytecode-based virtual machine
  - ROM-able
  - Designed for simple, low-impact implementation
  - Compiler in network
  - Better network bandwidth use
  - Better use of terminal memory/cpu.

# WAP Overview

## *Application Environment (PUSH)*

- **PUSH framework:**
  - permits content to be pushed in order to avoid cost and bandwidth penalty of polling,
  - Special security considerations provided,
  - Several control mechanisms provided for:
    - Lifetime of PUSHed content
    - Removal of PUSHed content
    - etc.
  - Can be applied to unprovisioned device.

# WAP Overview

## *Application Environment (UAPROF)*

- **User Agent PROFile:**
  - Permits devices to advertise their capabilities to WAP Proxies or WSP Servers and Application Servers,
  - Powerful extensible mechanism for relating many aspects of applications, preferences etc.,
  - Developed jointly with W3C (W3C calls it CC/PP)

# WAP Overview

## *Application Environment (WTA)*

- **WTA - Wireless Telephony Environment :**
  - Tools for building telephony applications
  - Initialy designed primarily for:
    - Network Operators / Carriers
    - Equipment Vendors
  - Network security and reliability a major consideration
  - WTA Browser function:
    - Extensions added to standard WML/WMLScript browser
    - Exposes additional API (WTAI) which includes call control, network text messaging, phone book interface
    - WTAI available in WML & WMLScript

# WAP Overview

## *Futures*

- **WAP Futures (Short Term):**
  - Protocols
    - More networks and bearers
  - Security
    - Enhance current security offering etc.
  - Application Environment
    - Extensions for new types of services
    - Extend to support off-line operation
    - Enrichen the experience

# WAP Overview

## *Futures*

- **WAP Futures (Longer Term):**
  - Convergence !

# WAP

## *Why was WAP needed? Why were the IP insufficent?*

- **Why was WAP needed ?**
  - **To meet the needs of wireless devices, e.g. constrained bandwidth and device resources, but with scaleability.**
    - **Scales from 'Dilbert Ring' through PDA and notebook.**
  - **Concept of Interoperability Certification useful.**
- **Why were the IP insufficient ?**
  - **IP seen as an overhead in SMS/USSD**
  - **TCP seen as too verbose when used for browser, Datagram concept well liked. WTP perceived as offering sufficient reliability.**
  - **HTTP V1.1 well liked but long life sessions and efficiency investigated**
  - **TLS pre-req of reliable transport inhibited acceptibility.**
  - **HTML V3.2 & 4.0 too rich and lack telephony awareness**

# WAP

## *A Personal Critique*

- **Plus's**
  - **Addresses the delivery of information to constrained devices, e.g. bandwidth and resources**
  - **Adopts Internet technology asis where appropriate and optimises otherwise**
  - **Scales from 'Dilbert Ring' through PDA**
  - **Has developed the need for capability negotiation allowing targetted content**
  - **Concept of Interoperabiltiy Certification useful.**
- **Minus's**
  - **Interoperability with Internet services.**
    - **protocols, security, etc..**
  - **Issues of using Internet to get to WAP services based on WSP Server or non-operator Proxy**