WAP: Wireless Application Protocol

Bheemarjuna Reddy Tamma
IIT Hyderabad

Adapted from Slides of Sridhar Iyer, IITB

Outline

- Mobile applications
- How are mobile/wireless environments different?
- What is WAP?
- WAP Architecture
- WAE (WML/WMLScript)
- WTA Framework
- WAP Push Services
- WAP Protocol Stack
- Hype v/s Reality
- References and Resources

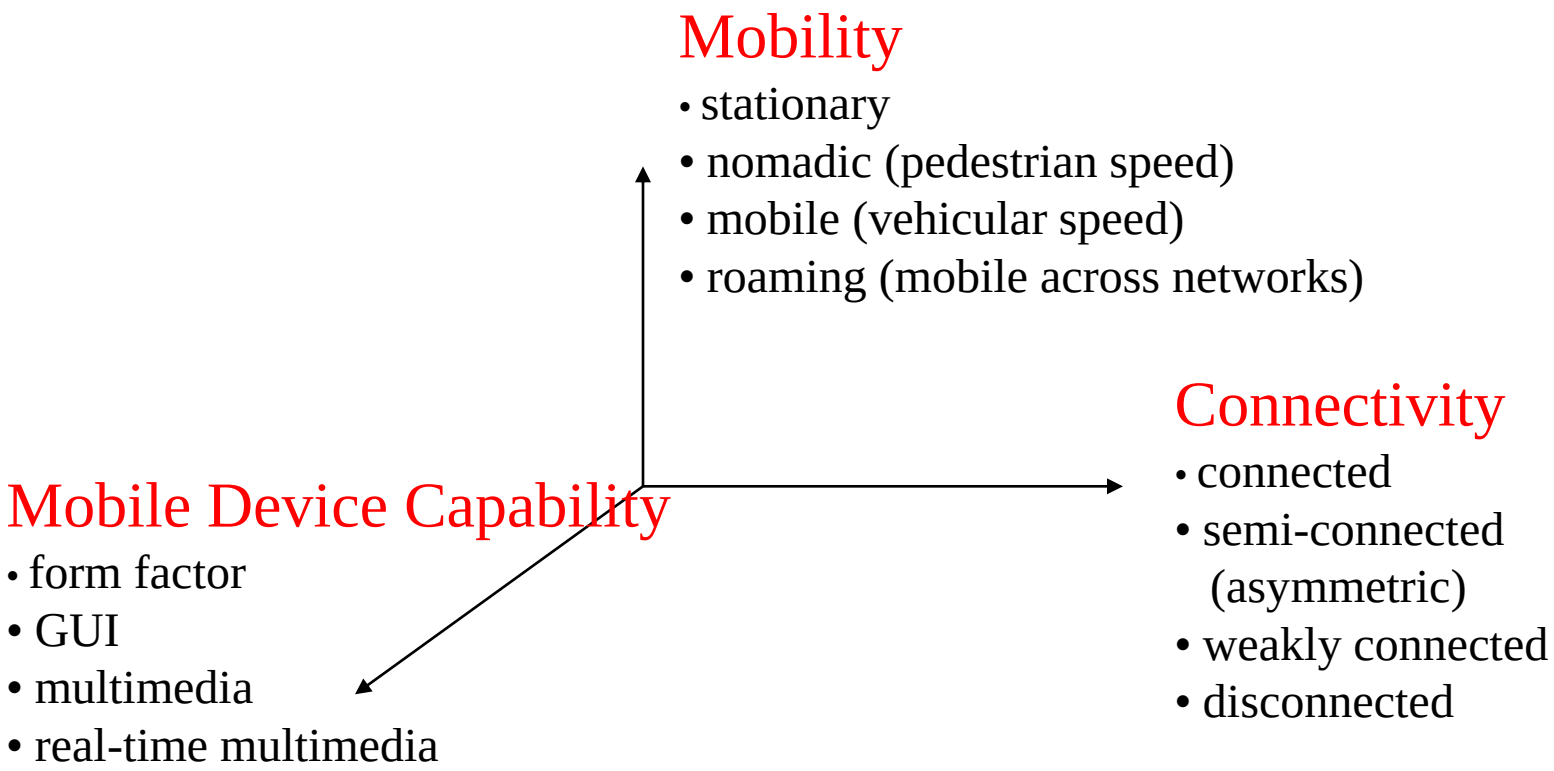
Mobile Applications - 1

- Vehicles
  - transmission of news, road condition etc
  - ad-hoc network with near vehicles to prevent accidents
- Emergencies
  - early transmission of patient data to the hospital
  - ad-hoc network in case of earthquakes, cyclones
  - military ...
- Traveling salesmen
  - direct access to central customer files
  - consistent databases for all agents
  - mobile office

Mobile Applications - 2

- Web access
  - outdoor Internet access
  - intelligent travel guide with up-to-date location dependent information
- Information services
  - push: stock quotes; pull: nearest cash ATM
- Disconnected operations
  - file-system caching for off-line work
  - mobile agents, e.g., shopping
- Entertainment
  - games, etc

Variability of the Mobile Environment



World Wide Web and Mobility

HTTP/HTML have not been designed for mobile applications/devices

- HTTP 1.0 characteristics
  - designed for large bandwidth, low delay
  - stateless, client/server, request/response communication
  - connection oriented, one connection per request
  - TCP 3-way handshake, DNS lookup overheads
  - big protocol headers, uncompressed content transfer
  - primitive caching (often disabled, dynamic objects)
  - security problems (using SSL/TLS with proxies)
- HTML characteristics
  - designed for computers with “high” performance, color high-resolution display, mouse, hard disk
  - typically, web pages optimized for design, not for communication; ignore end-system characteristics

System Support for Mobile Internet

- Enhanced browsers
  - client-aware support for mobility
- Proxies
  - Client proxy: pre-fetching, caching, off-line use
  - Network proxy: adaptive content transformation for connections
  - Client and network proxy
- Enhanced servers
  - server-aware support for mobility
  - serve the content in multiple ways, depending on client capabilities
- New protocols/languages
  - WAP/WML

Wireless Application Protocol (WAP)

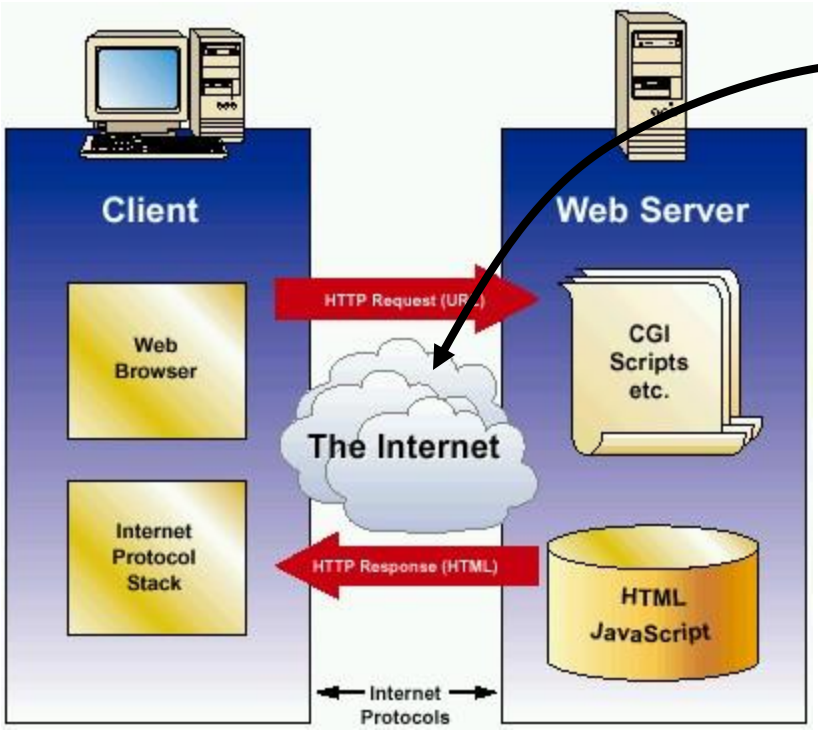
- Empowers mobile users with wireless devices to easily access and interact with information and services.
- A “standard” created by wireless and Internet companies to enable Internet access from a cellular phone

- wapforum.org:
  - co-founded by Ericsson, Motorola, Nokia, Phone.com
  - 450 members in 2000, comprise of Handset manufacturers, Wireless service providers, ISPs, Software companies in the wireless industry
  - Goals
    - deliver Internet services to mobile devices
    - enable applications to scale across a variety of transport options and device types
    - independence from wireless network standards
    - GSM, CDMA IS-95, TDMA IS-136, 3G systems (UMTS, W-CDMA)

WAP: Main Features

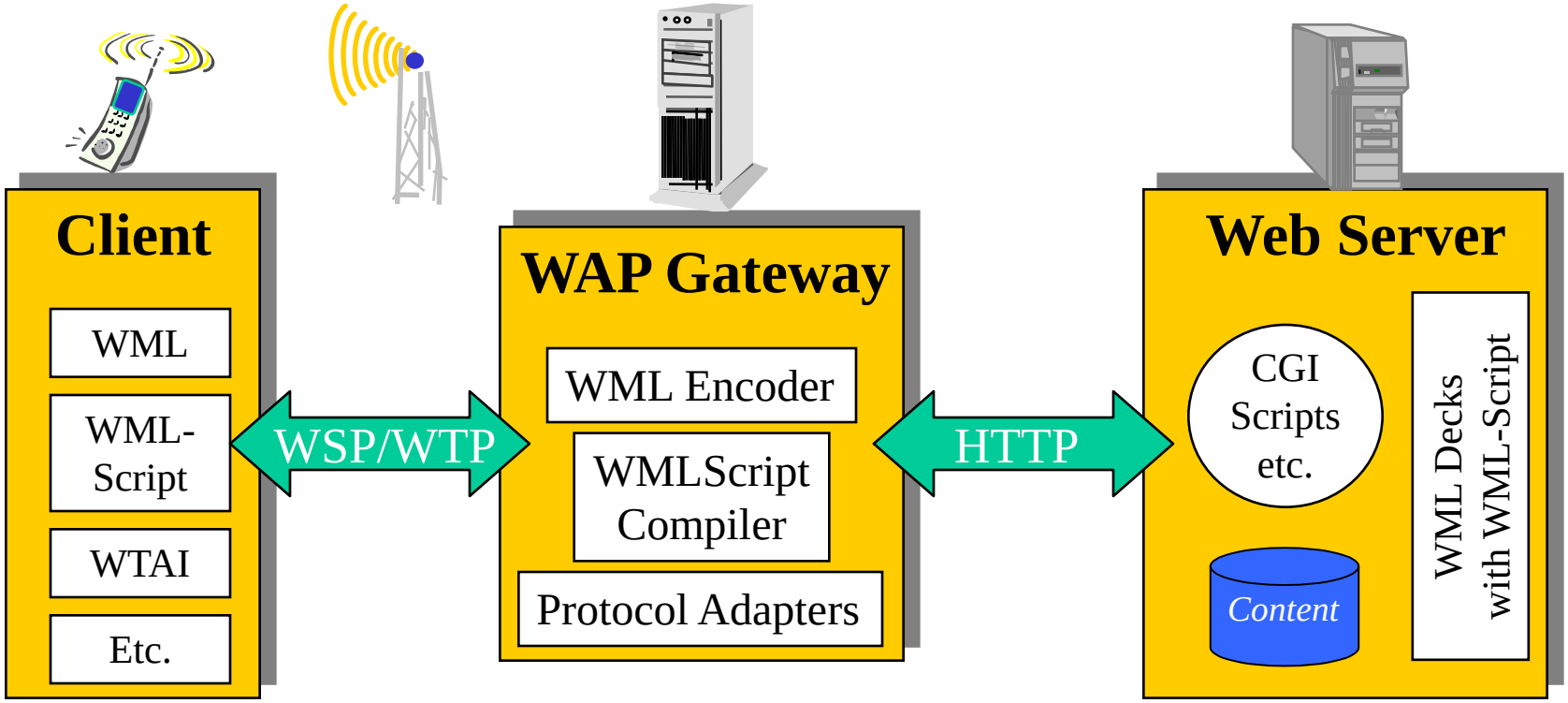
- Browser
  - "Micro browser", similar to existing web browsers
- Markup language
  - Similar to HTML, adapted to mobile devices
- Script language
  - Similar to Javascript, adapted to mobile devices
- Gateway
  - Transition from wireless to wired world
- Server
  - "Wap/Origin server", similar to existing web servers
- Protocol layers
  - Transport layer, security layer, session layer etc.
- Telephony application interface
  - Access to telephony functions

Internet Model



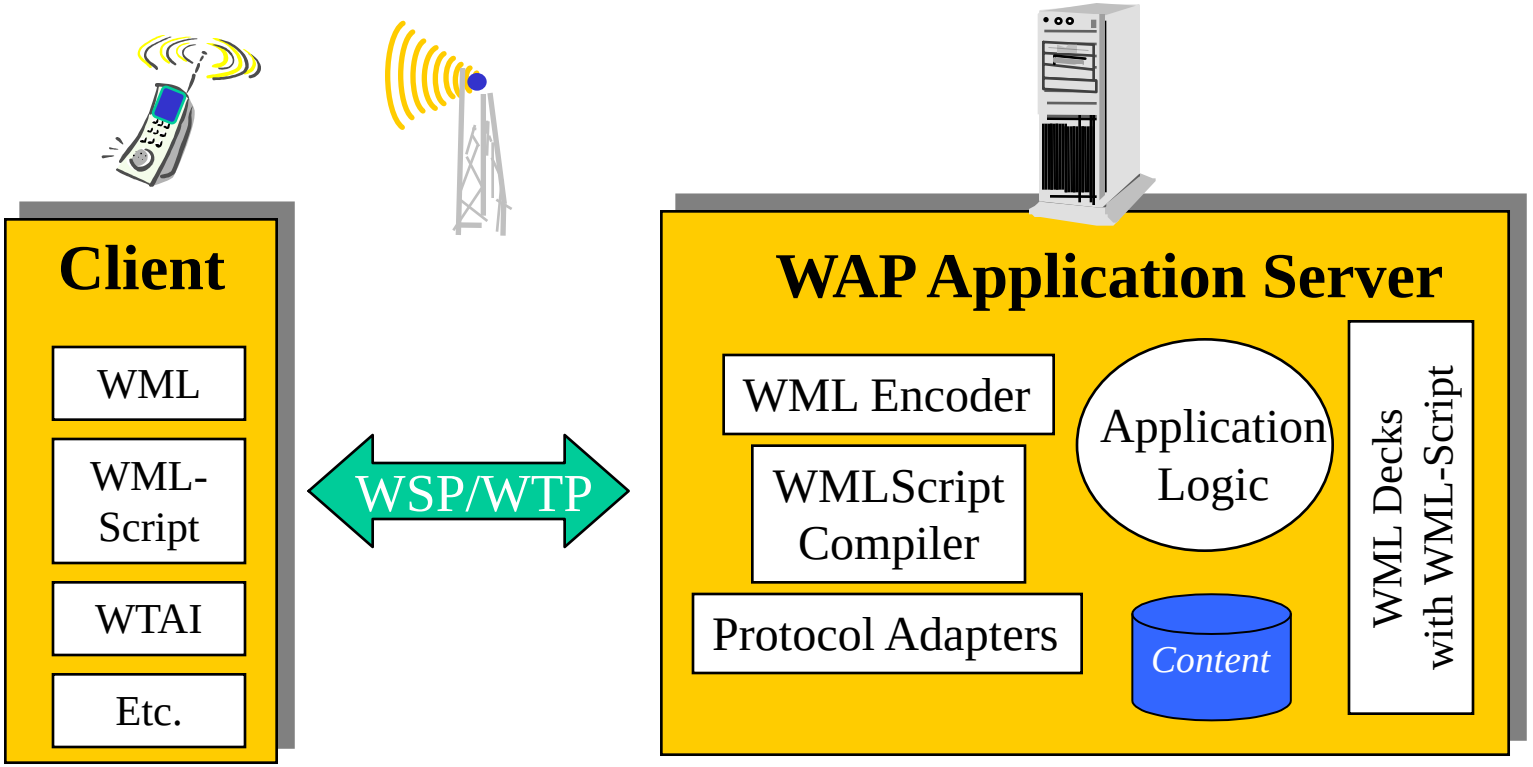
HTML
HTTP
TLS/SSL
TCP/IP

WAP Architecture



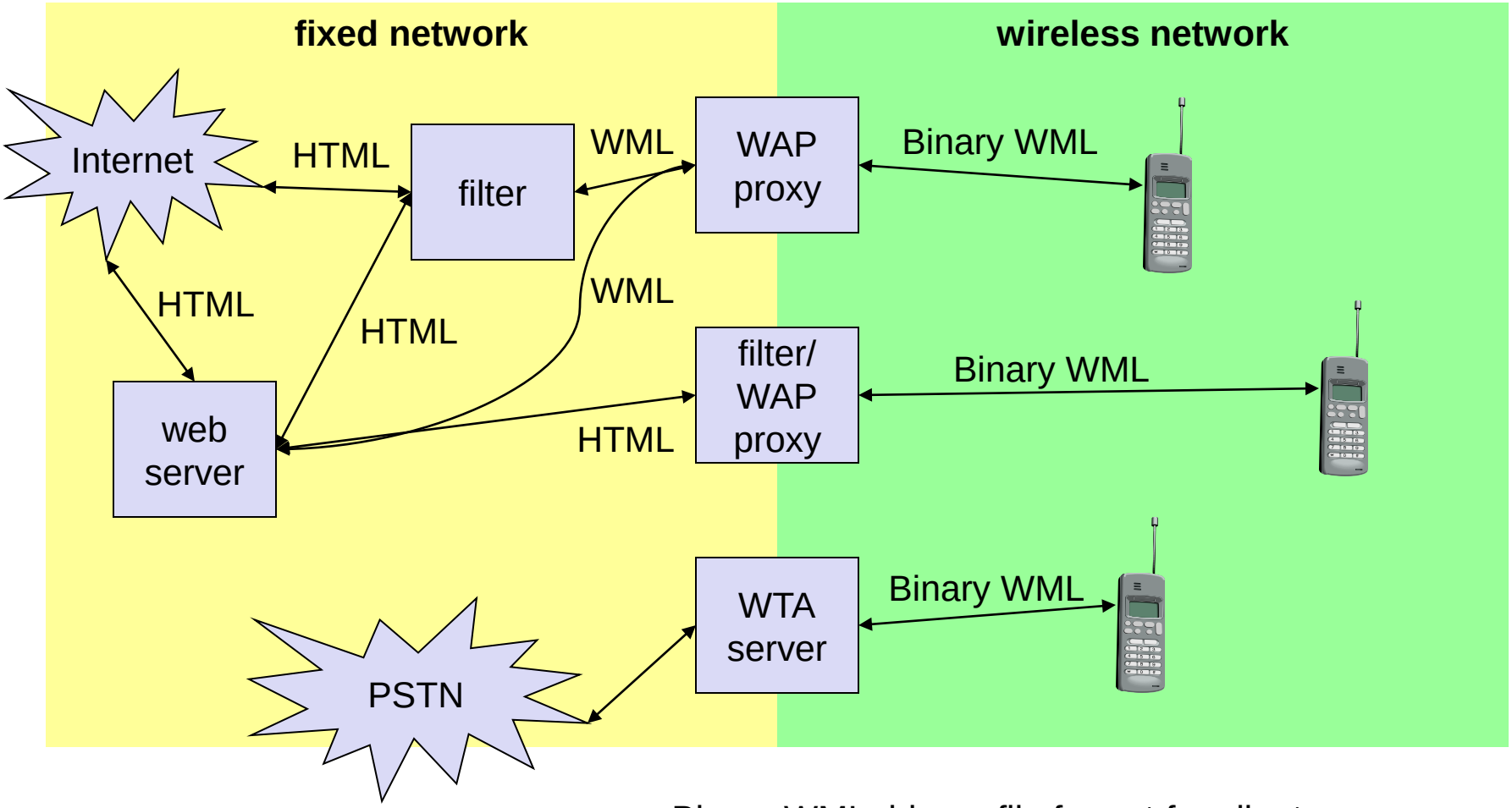
Source: WAP Forum

WAP Application Server



Source: WAP Forum

WAP: Network Elements



Binary WML: binary file format for clients

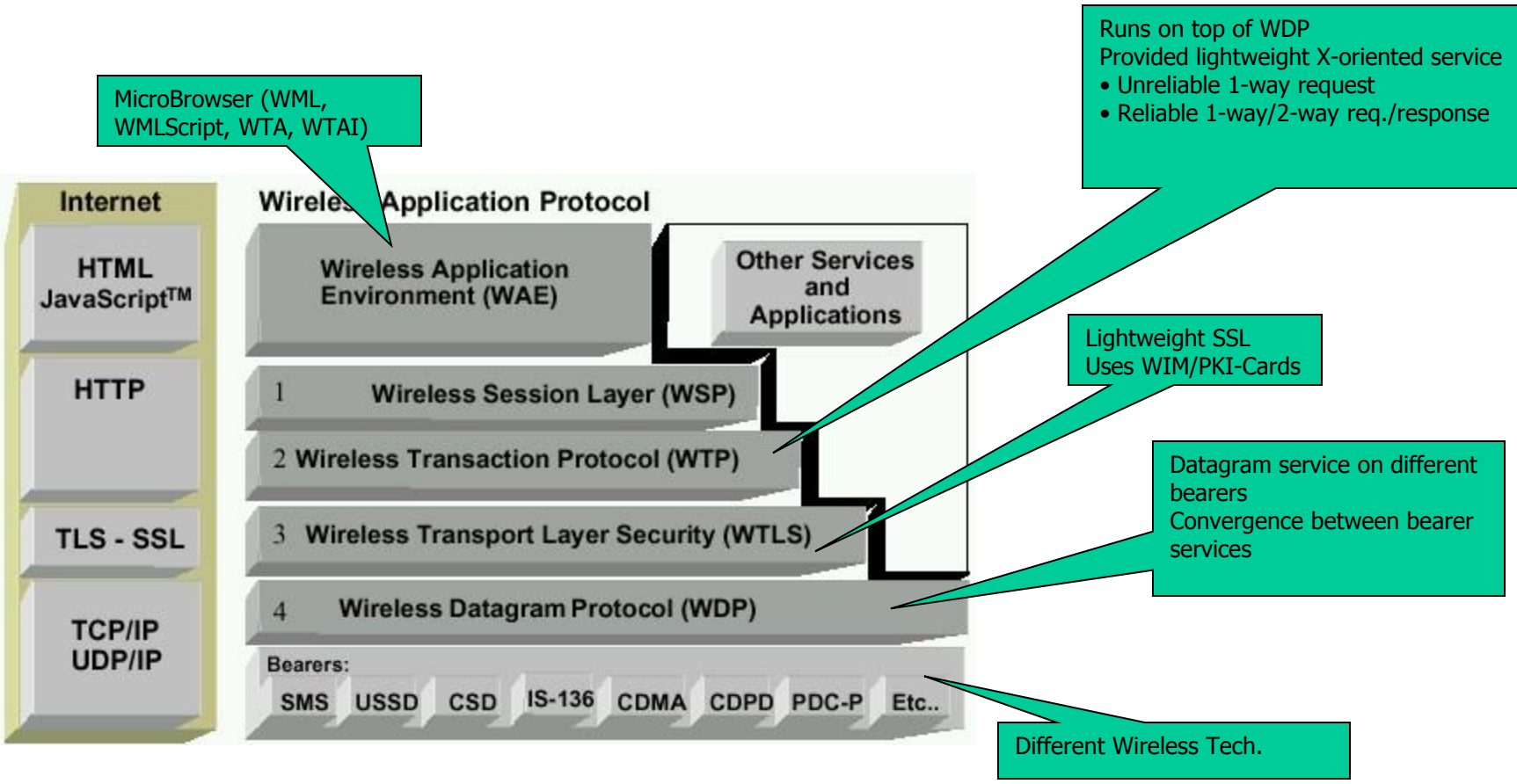
Source: Schiller

WAP Specifies

- Wireless Application Environment
  - WML Microbrowser
  - WMLScript Virtual Machine
  - WMLScript Standard Library
  - Wireless Telephony Application Interface (WTAI)
  - WAP content types

- Wireless Protocol Stack
  - Wireless Session Protocol (WSP)
  - Wireless Transport Layer Security (WTLS)
  - Wireless Transaction Protocol (WTP)
  - Wireless Datagram Protocol (WDP)
  - Wireless network interface definitions

WAP Stack



Source: WAP Forum

WAP Stack

- WAE (Wireless Application Environment):
  - Architecture: application model, browser, gateway, server
  - WML: XML-Syntax, based on card stacks, variables, ...
  - WTA: telephone services, such as call control, phone book etc.
- WSP (Wireless Session Protocol):
  - Provides HTTP 1.1 functionality
  - Supports session management, security, etc.
- WTP (Wireless Transaction Protocol):
  - Provides reliable message transfer mechanisms
  - Based on ideas from TCP/RPC
- WTLS (Wireless Transport Layer Security):
  - Provides data integrity, privacy, authentication functions
  - Based on ideas from TLS/SSL
- WDP (Wireless Datagram Protocol):
  - Provides transport layer functions
  - Based on ideas from UDP

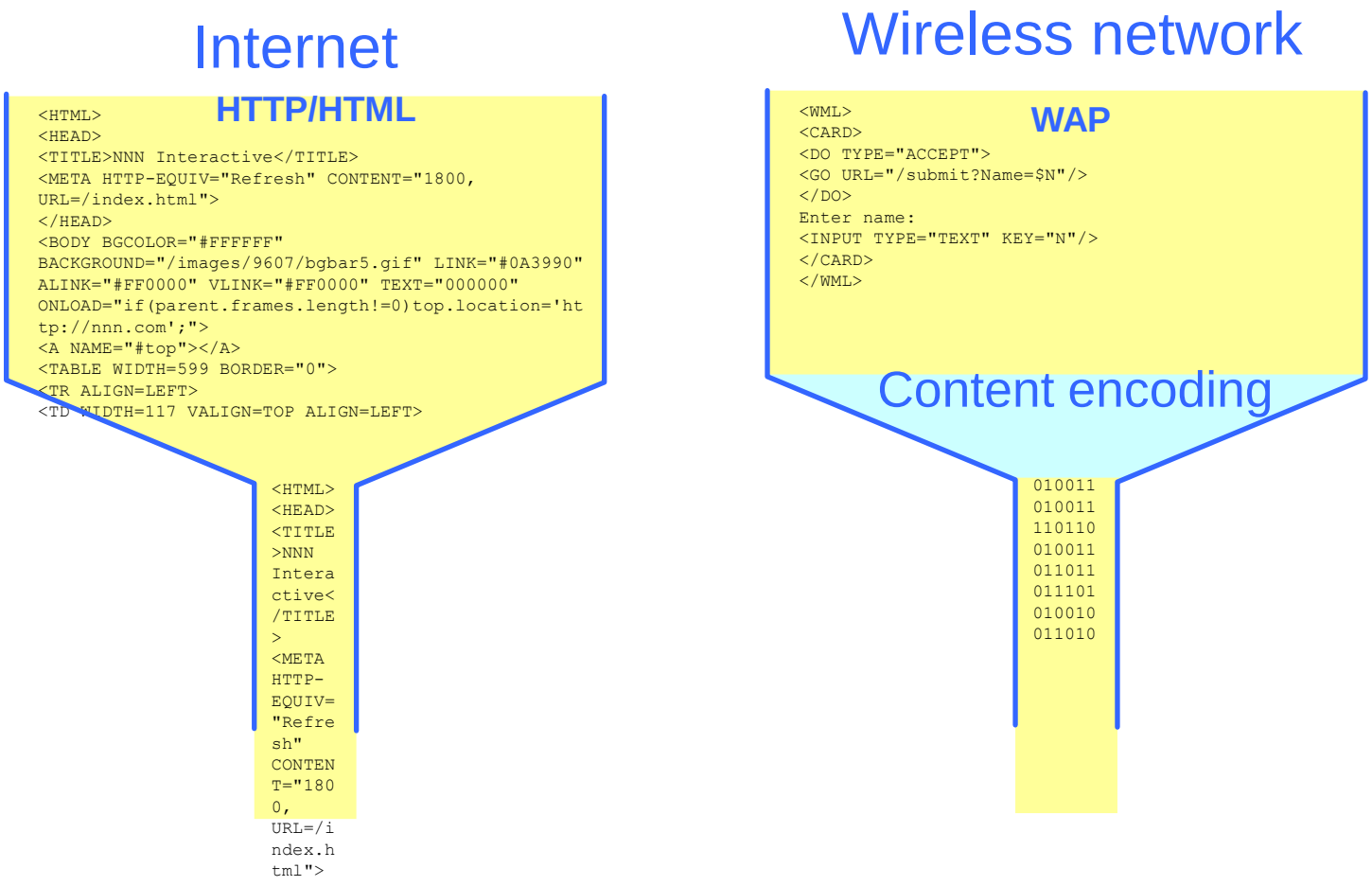
Content encoding, optimized for low-bandwidth channels, simple devices

WHY WAP?

- Wireless networks and phones
  - have specific needs and requirements (low-BW, small displays, low CPU, low RAM, connection instability, etc)
  - not addressed by existing Internet technologies (??)
- WAP
  - Enables any data transport
    - TCP/IP, UDP/IP, GUTS (IS-135/6), SMS, or USSD.
  - Optimizes the content and air-link protocols
  - Utilizes plain Web HTTP 1.1 servers
    - leverages existing development methodologies
    - utilizes standard Internet markup language technology (XML)
    - all WML content is accessed via HTTP 1.1 requests
  - WML UI components map well onto existing mobile phone user interfaces
    - no re-education of the end-users
    - leveraging market penetration of mobile devices
  - Several modular entities together form a fully compliant Internet entity

Why is HTTP/HTML not enough?

Big pipe - small pipe syndrome



Source: WAP Forum

Wireless Application Environment (WAE)

- Goals
  - device and network independent application environment
  - for low-bandwidth, wireless devices
  - considerations of slow links, limited memory, low computing power, small display, simple user interface (compared to desktops)
  - integrated Internet/WWW programming model
  - high interoperability

WAE Components

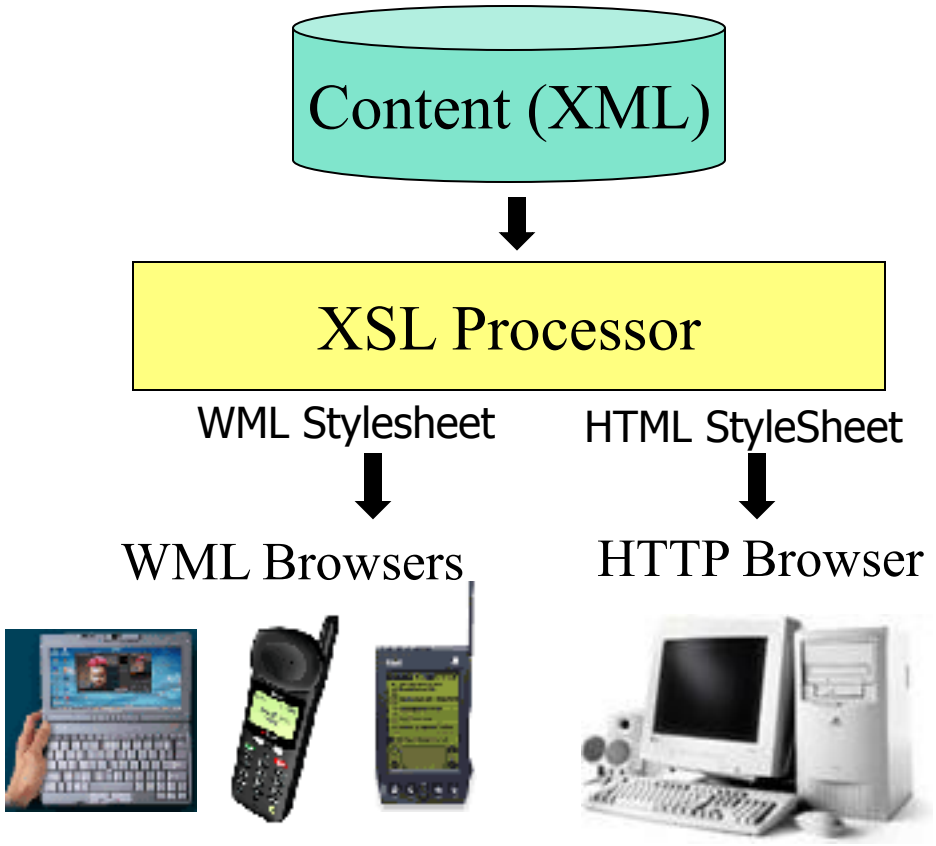
- Architecture
  - Application model, Microbrowser, Gateway, Server
- User Agents
  - WML/WTA/Others
  - content formats: vCard, vCalendar, Wireless Bitmap, WML, ...
- WML
  - XML-Syntax, based on card stacks, variables, ...
- WMLScript
  - procedural, loops, conditions, ... (similar to JavaScript)
- WTA
  - telephone services, such as call control, text messages, phone book, ... (accessible from WML/WMLScript)
- Proxy (Method/Push)

WAP Microbrowser

- Optimized for wireless devices
- Minimal RAM, ROM, Display, CPU and keys
- Provides consistent service UI across devices
- Provides Internet compatibility
- Enables wide array of available content and applications

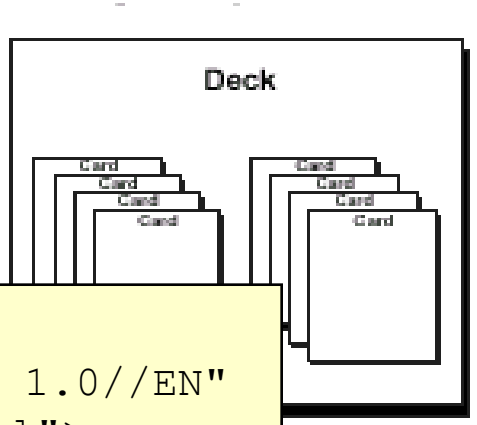
WML: Wireless Markup Language

- Tag-based browsing language:
  - Screen management (text, images)
  - Data input (text, selection lists, etc.)
  - Hyperlinks & navigation support
- Takes into account limited display, navigation capabilities of devices
- XML-based language
  - describes only intent of interaction in an abstract manner
  - presentation depends upon device capabilities
- Cards and Decks
  - document consists of many cards
  - User interactions are split into cards
  - Explicit navigation between cards
  - cards are grouped to decks
  - deck is similar to HTML page, unit of content transmission
- Events, variables and state mgmt



WML

- The basic unit is a **card**. Cards are grouped together into **Decks** Document ~ Deck (unit of transfer)
- All decks must contain
  - Document prologue
    - XML & document type declaration
  - <WML> element
    - Must contain one or more cards

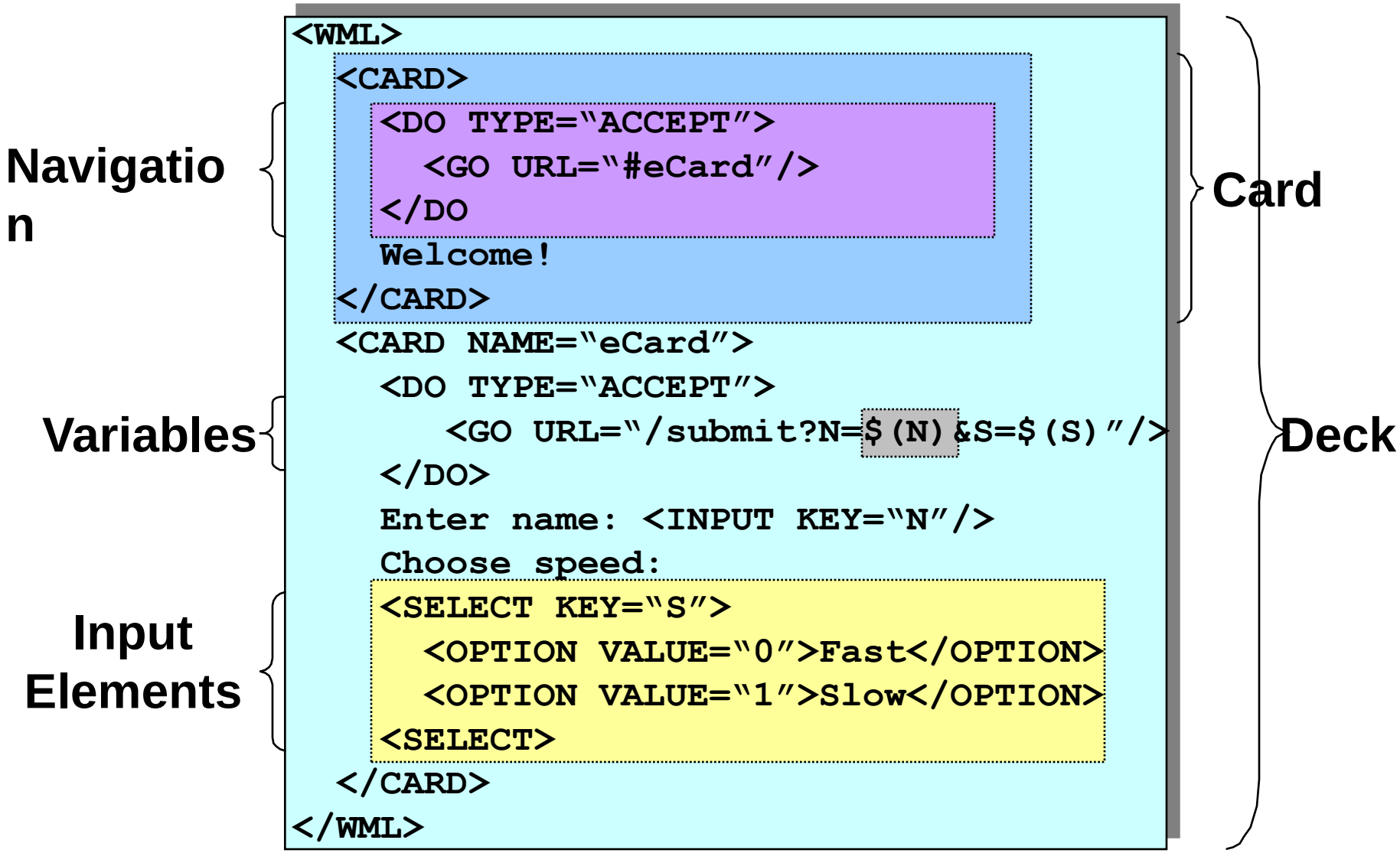


WML File Structure

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

<WML>
    ...
</WML>
```

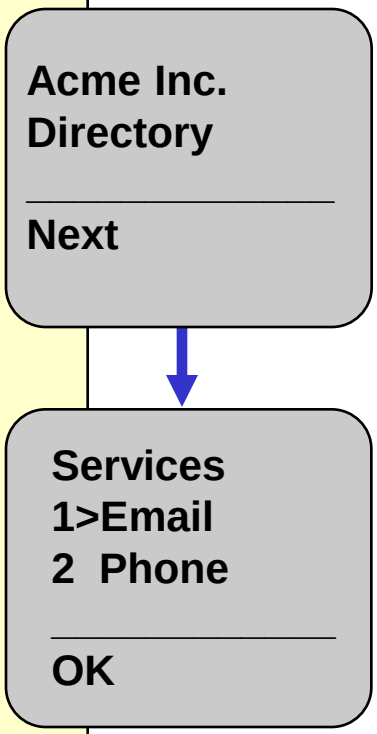
WML Example


Navigation
Variables
Input Elements
Card
Deck
<WML>
<CARD>
<DO TYPE="ACCEPT">
<GO URL="#eCard"/>
</DO
Welcome!
</CARD>
<CARD NAME="eCard">
<DO TYPE="ACCEPT">
<GO URL="/submit?N=$(N)&S=$(S)"/>
</DO>
Enter name: <INPUT KEY="N"/>
Choose speed:
<SELECT KEY="S">
<OPTION VALUE="0">Fast</OPTION>
<OPTION VALUE="1">Slow</OPTION>
<SELECT>
</CARD>
</WML>

A Deck of Cards

```
<WML>
    <CARD>
        <DO TYPE="ACCEPT" LABEL="Next">
            <GO URL="#card2"/>
        </DO>
        Acme Inc.<BR/>Directory
    </CARD>

    <CARD NAME="card2">
        <DO TYPE="ACCEPT">
            <GO URL="?send=$type"/>
        </DO>
        Services
        <SELECT KEY="type">
            <OPTION VALUE="em">Email</OPTION>
            <OPTION VALUE="ph">Phone</OPTION>
            <OPTION VALUE="fx">Fax</OPTION>
        </SELECT>
    </CARD>
</WML>
```



Source: WAP Forum

WMLScript

- Complement to WML
  - Derived from JavaScript™
- Provides general scripting capabilities
  - Procedural logic, loops, conditionals, etc.
  - Optimized for small-memory, small-cpu devices

- Features
  - local user interaction, validity check of user input
  - access to device facilities (phone call, address book etc.)
  - extensions to the device software
    - configure device, download new functionality after deployment

- Bytecode-based virtual machine
  - Stack-oriented design, ROM-able
  - Designed for simple, low-impact implementation
- WMLScript compiler resides in the network

WAE Summary

- WML
  - analogous to HTML (optimized for wireless)
  - event based, microbrowser user agent
- WMLScript
  - analogous to JavaScript
  - features of compiler in the network
- WTA
  - WTAI: different access rights for different applications/agents
  - WTA User Agent (analogy with operating systems)
    - Context – Activation Record
    - Channel – Interrupt Handler
    - Resource – Shared routines invoked by interrupt handlers
    - Repository – Library of interrupt handlers
  - feature of dynamically pushing the interrupt handler before the event
- Push
  - no analogy in Internet

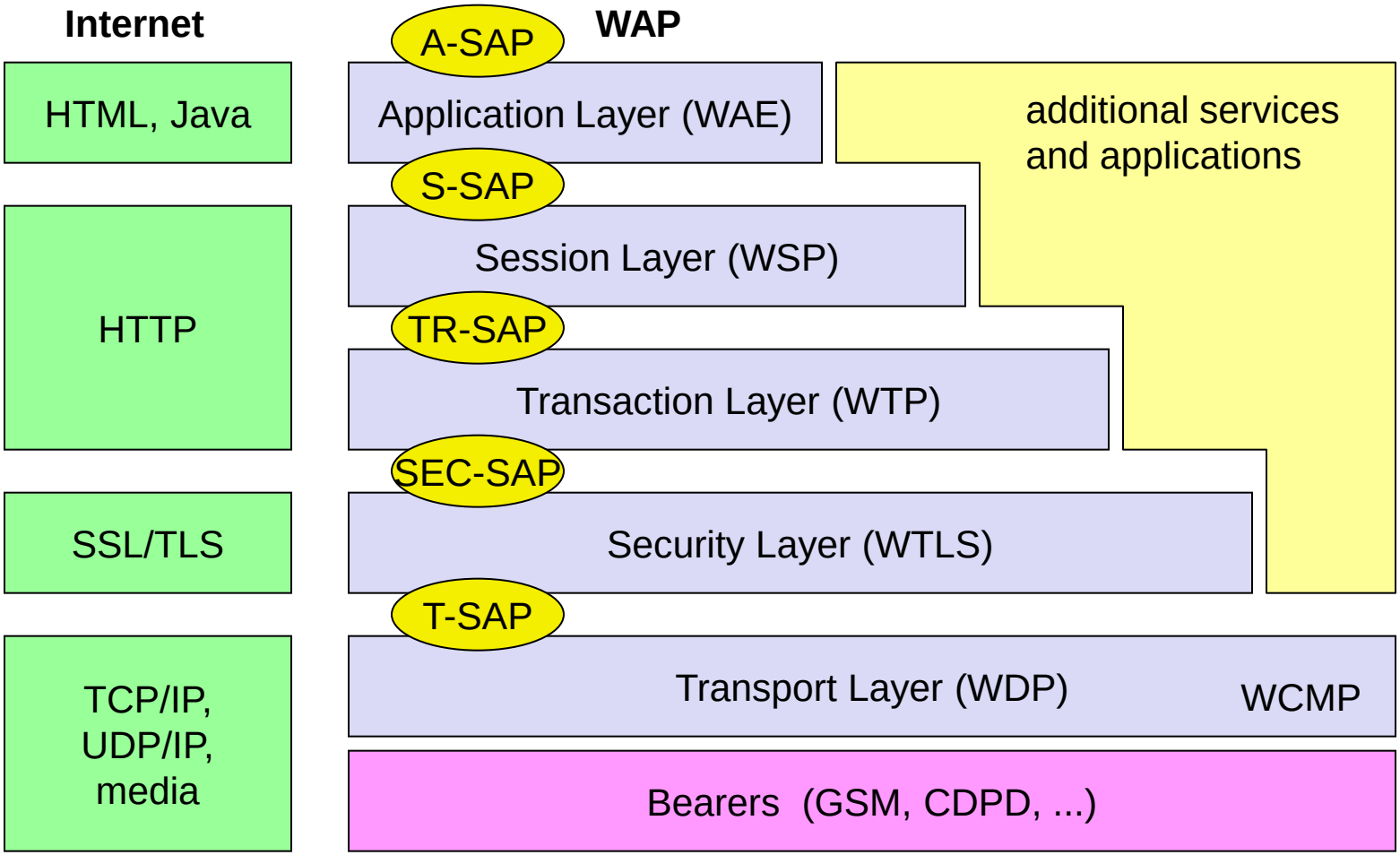
WAP Gateway Summary

- Encoders
  - translate between binary (WML) and text (HTML/WML)
- Filters
  - transcoding between WML (wireless) and HTML (wired)
- Method Proxy
  - similar to standard proxy services
  - WAP stack on wireless interface and TCP/IP stack on Internet interface
- Push Proxy
  - Push Access Protocol with Internet Push Initiator (Web Server)
  - Over the Air Protocol with mobile device (and WAP Push Initiator)
  - Performs necessary filtering, translation etc.

WAP Servers Summary

- Origin Server
  - Web server with HTML/WML contents
  - Runs TCP/IP stack, needs PAP protocol for push, no end-to-end security
- WAP Server
  - Serves WML content
  - Runs WAP stack, uses OTA protocol for push, end-to-end security possible
- WTA Server
  - Specialized for telephony applications (runs WAP stack, uses push extensively)
  - Client initiated (make call “hyperlink” from a Yellow pages service)
  - Server intiated (incoming call from a Voice mail service)

WAP: Protocol Stack



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

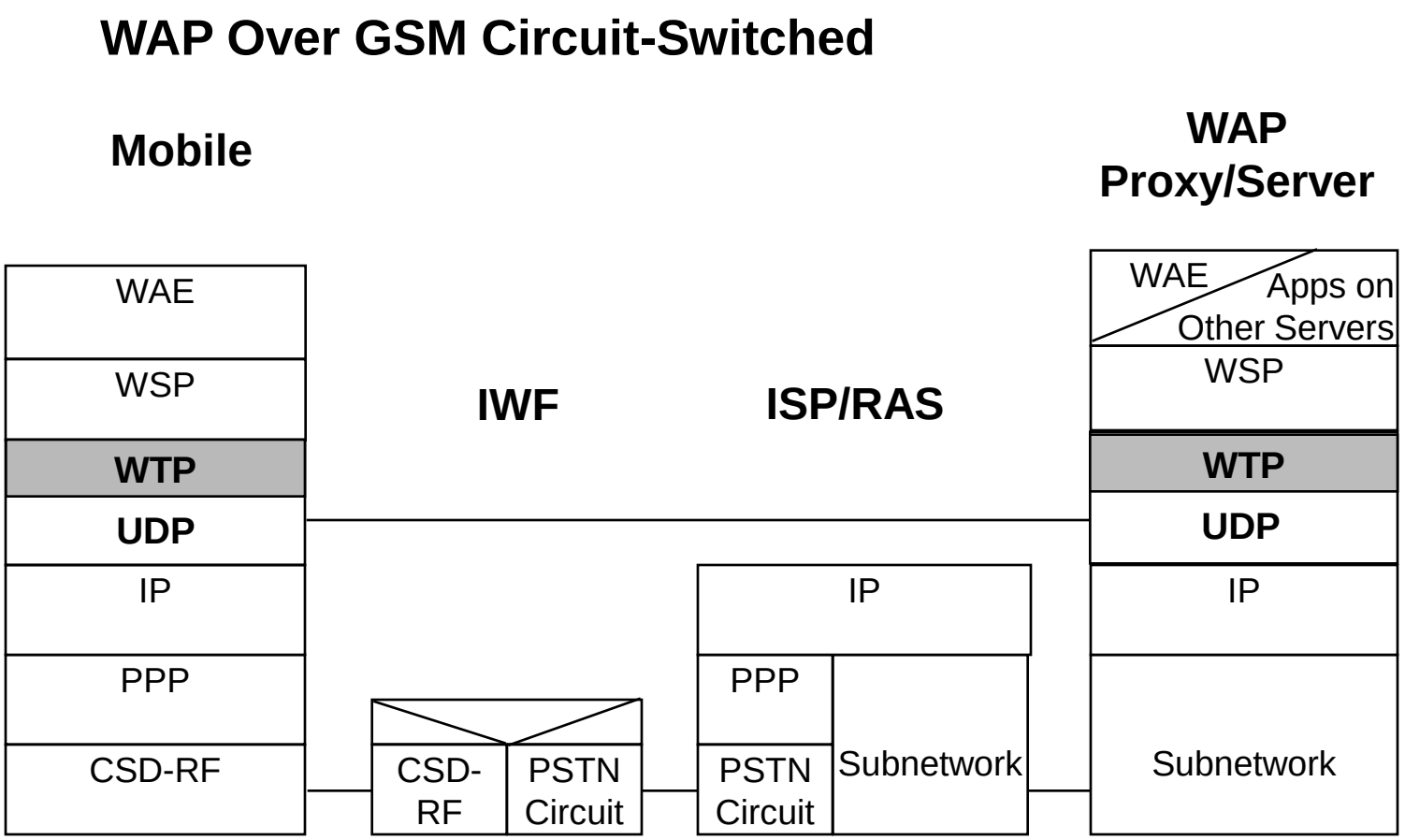
Source: Schiller

WDP: Wireless Datagram Protocol

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

- WDP
  - Transport layer protocol within the WAP architecture
  - uses the Service Primitive
    - T-UnitData.req .ind
  - uses transport mechanisms of different bearer technologies
  - offers a common interface for higher layer protocols
  - allows for transparent communication despite different technologies
  - addressing uses port numbers
  - WDP over IP is UDP/IP

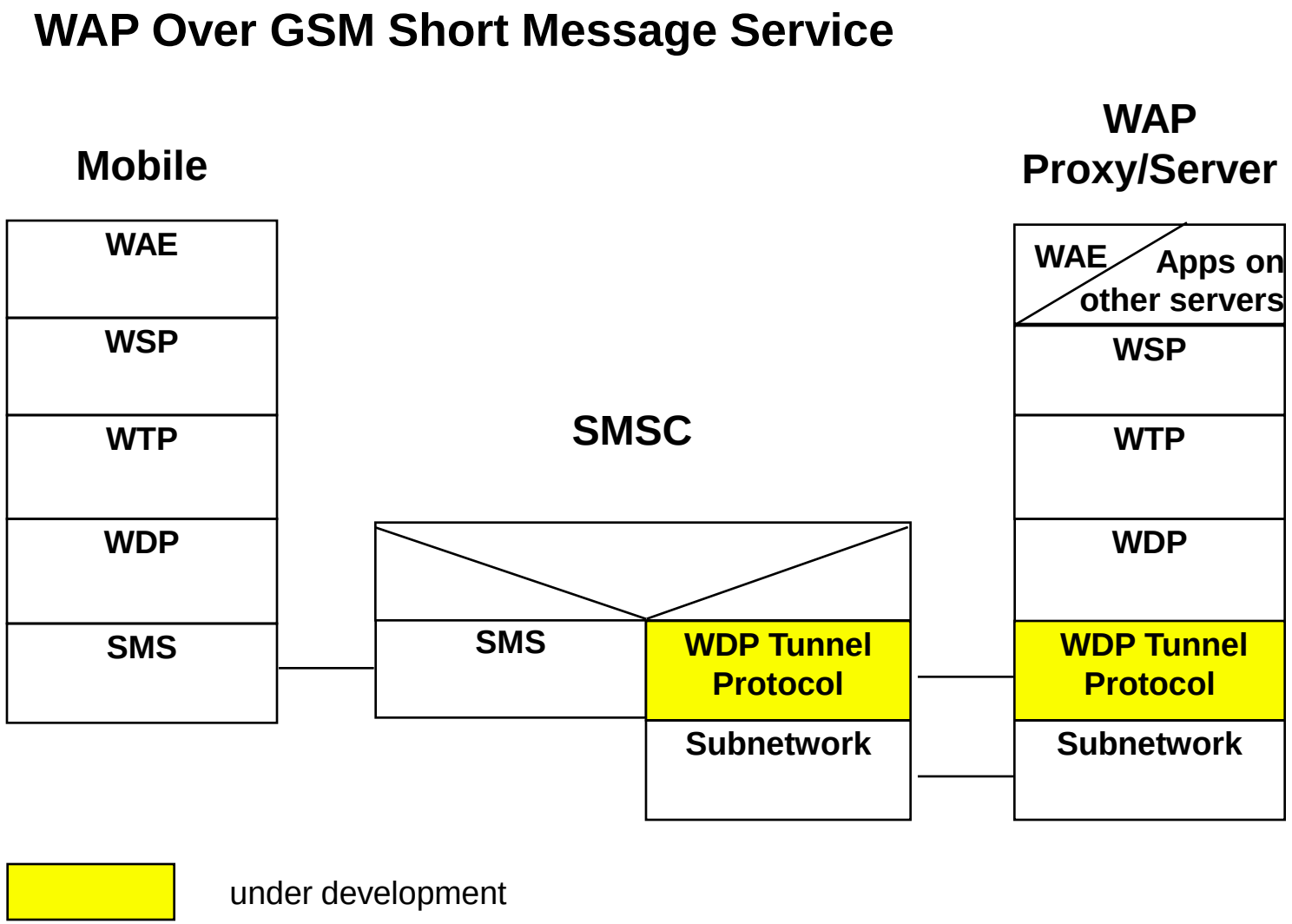
Service, Protocol, and Bearer Example



RAS - Remote Access Server
IWF - InterWorking Function

Source: WAP Forum

Service, Protocol, and Bearer Example



Source: WAP Forum

WTP: Wireless Transaction Protocol

- Goals
  - different transaction services that enable applications to select reliability, efficiency levels
  - low memory requirements, suited to simple devices (< 10kbyte)
  - efficiency for wireless transmission
- WTP
  - supports peer-to-peer, client/server and multicast applications
  - efficient for wireless transmission
  - support for different communication scenarios
  - class 0*:* unreliable message transfer
    - unconfirmed Invoke message with no Result message
    - a datagram that can be sent within the context of an existing Session
  - class 1*:* reliable message transfer without result message
    - confirmed Invoke message with no Result message
    - used for data push, where no response from the destination is expected
  - class 2*:* reliable message transfer with exactly one reliable result message
    - confirmed Invoke message with one confirmed Result message
    - a single request produces a single reply

WTP Services and Protocols

- WTP (Transaction)
  - provides reliable data transfer based on request/reply paradigm
    - no explicit connection setup or tear down
    - optimized setup (data carried in first packet of protocol exchange)
    - seeks to reduce 3-way handshake on initial request
  - supports
    - header compression
    - segmentation /re-assembly
    - retransmission of lost packets
    - selective-retransmission
    - port number addressing (UDP ports numbers)
    - flow control
  - message oriented (not stream)
  - supports an Abort function for outstanding requests
  - supports concatenation of PDUs
  - supports User acknowledgement or Stack acknowledgement option
    - acks may be forced from the WTP user (upper layer)
    - default is stack ack

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

- WSP Services
  - provides shared state between client and server, optimizes content transfer
  - session management (establish, release, suspend, resume)
  - efficient capability negotiation
  - content encoding
  - push

- WSP/B (Browsing)
  - HTTP/1.1 functionality - but binary encoded
  - exchange of session headers
  - push and pull data transfer
  - asynchronous requests

WSP Overview

- Header Encoding
  - compact binary encoding of headers, content type identifiers and other well-known textual or structured values
  - reduces the data actually sent over the network
- 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
- Suspend and Resume
  - server knows when client can accept a push
  - multi-bearer devices
  - dynamic addressing
  - allows the release of underlying bearer resources

WAP Stack Summary

- WDP
  - functionality similar to UDP in IP networks
- WTLS
  - functionality similar to SSL/TLS (optimized for wireless)
- WTP
  - Class 0: analogous to UDP
  - Class 1: analogous to TCP (without connection setup overheads)
  - Class 2: analogous to RPC (optimized for wireless)
  - features of “user acknowledgement”, “hold on”
- WSP
  - WSP/B: analogous to http 1.1 (add features of suspend/resume)
  - method: analogous to RPC/RMI
  - features of asynchronous invocations, push (confirmed/unconfirmed)

WAP: Hype vs Reality

- Low-bandwidth wireless links
  - TCP/IP over wireless can also address these problems
  - encoding in http can also reduce data transfer on wireless links
- Limited device capabilities
  - Microbrowser is appropriate to address this problem
  - WTAI features are not present in TCP/IP domain

- Challenges in WAP
  - adapting to applications rich in content and interaction
  - service guarantees
  - interface design and usability
  - WAP website (wap.yahoo.com, m.google.com)

- Other approaches for WWW access through mobiles
  - i-Mode (from NTT DoCoMo)
  - WAP is a TRAP (http://www.freeprotocols.org/wapTrap)
- Modern smartphones have larger screens and full browsers, so WAP future is bleak.