WAP Architecture

Version 30-Apr-1998

Wireless Application Protocol Architecture Specification

Disclaimer:

This document is subject to change without notice.

Contents

1. Scope

The Wireless Application Protocol (WAP) is a result of the WAP Forum's efforts to promote industry-wide specifications for technology useful in developing applications and services that operate over wireless communication networks. WAP specifies an application framework and network protocols for wireless devices such as mobile telephones, pagers, and personal digital assistants (PDAs). The specifications extend and leverage mobile networking technologies (such as digital data networking standards) and Internet technologies (such as XML, URLs, scripting, and various content formats). The effort is aimed at enabling operators, manufacturers, and content developers to meet the challenges in building advanced differentiated services and implementations in a fast and flexible manner.

The objectives of the WAP Forum are:

- To bring Internet content and advanced data services to digital cellular phones and other wireless terminals.
- To create a global wireless protocol specification that will work across differing wireless network technologies.
- To enable the creation of content and applications that scale across a very wide range of bearer networks and device types.
- To embrace and extend existing standards and technology wherever appropriate.

The WAP Architecture Specification is intended to present the system and protocol architectures essential to achieving the objectives of the WAP Forum. The WAP Architecture Specification acts as the starting point for understanding the WAP technologies and resulting specifications. As such, it provides an overview of the different technologies and references the appropriate specifications for further details.

2. Document Status

This document is available online in the following formats:

- PDF format at http://www.wapforum.org/.

2.1 Copyright Notice

2.2 Errata

Known problems associated with this document are published at http://www.wapforum.org/.

2.3 Comments

Comments regarding this document can be submitted to the WAP Forum in the manner published at http://www.wapforum.org/.

3. References

3.1 Normative References

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. URL: ftp://ftp.isi.edu/in-notes/rfc2119.txt

[WAEoview] "Wireless Application Environment Overview", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WAE] "Wireless Application Environment Specification", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, April 30, 1998 URL: http://www.wapforum.org/

[WAPConf] "Wireless Application Protocol Conformance Statement, Compliance Profile, and Release List", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WDP] "Wireless Datagram Protocol Specification", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WML] "Wireless Markup Language", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WMLScript] "Wireless Markup Language Script", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WMLSStdLib] "Wireless Markup Language Script Standard Libraries", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WSP] "Wireless Session Protocol", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WTA] "Wireless Telephony Application Specification", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WTAI] "Wireless Telephony Application Interface", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WTLS] "Wireless Transport Layer Security Protocol", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

[WTP] "Wireless Transaction Protocol Specification", WAP Forum, April 30, 1998. URL: http://www.wapforum.org/

3.2 Informative References

[ECMAScript] Standard ECMA-262: "ECMAScript Language Specification", ECMA, June 1997

[HTML4] "HTML 4.0 Specification, W3C Recommendation 18-December-1997, REC-HTML40-971218", D. Raggett, et al., September 17, 1997. URL: http://www.w3.org/TR/REC-html40

[JavaScript] "JavaScript: The Definitive Guide", David Flanagan. O'Reilly & Associates, Inc. 1997

[RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee, et al., December 1994. URL: ftp://ftp.isi.edu/in-notes/rfc1738.txt

[RFC1808] "Relative Uniform Resource Locators", R. Fielding, June 1995. URL: ftp://ftp.isi.edu/in-notes/rfc1808.txt

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed, et al., November 1996. URL: ftp://ftp.isi.edu/in-notes/rfc2045.txt

[RFC2048] "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", N. Freed, et al., November 1996. URL: ftp://ftp.isi.edu/in-notes/rfc2048.txt

[RFC2068] "Hypertext Transfer Protocol - HTTP/1.1", R. Fielding, et al., January 1997. URL: ftp://ftp.isi.edu/in-notes/rfc2068.txt

4. Definitions and Abbreviations

4.1 Definitions

The following are terms and conventions used throughout this specification.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Author – an author is a person or program that writes or generates WML, WMLScript or other content.

Client – a device (or application) that initiates a request for a connection with a server.

Content – subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request.

Content Encoding – when used as a verb, content encoding indicates the act of converting content from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process.

Content Format – actual representation of content.

Device – a network entity that is capable of sending and receiving packets of information and has a unique device address. A device can act as both a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

JavaScript – a *de facto* standard language that can be used to add dynamic behaviour to HTML documents. JavaScript is one of the originating technologies of ECMAScript.

Man-Machine Interface – a synonym for user interface.

Origin Server – the server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

Resource – a network data object or service that can be identified by a URL. Resources may be available in multiple representations (eg, multiple languages, data formats, size and resolutions) or vary in other ways.

Server – a device (or application) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client.

Terminal – a device providing the user with user agent capabilities, including the ability to request and receive information. Also called a mobile terminal or mobile station.

User – a user is a person who interacts with a user agent to view, hear, or otherwise use a resource.

User Agent – a user agent is any software or device that interprets WML, WMLScript, WTAI or other resources. This may include textual browsers, voice browsers, search engines, etc.

WMLScript – a scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript™ scripting language.

4.2 Abbreviations

For the purposes of this specification, the following abbreviations apply.

HTML HyperText Markup Language [HTML4]

HTTP	HyperText Transfer Protocol [RFC2068]
MMI	Man-Machine Interface
PDA	Personal Digital Assistant
PICS	Protocol Implementation Conformance Statement
RFC	Request For Comments
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URL	Uniform Resource Locator [RFC1738]
W3C	World Wide Web Consortium
WAE	Wireless Application Environment [WAE]
WAP	Wireless Application Protocol [WAP]
WDP	Wireless Datagram Protocol [WDP]
WML	Wireless Markup Language [WML]
WSP	Wireless Session Protocol [WSP]
WTA	Wireless Telephony Application [WTA]
WTLS	Wireless Transport Layer Security [WTLS]
WTP	Wireless Transaction Protocol [WTP]
WWW	World-Wide Web

5. Background

5.1 Motivation

WAP is positioned at the convergence of two rapidly evolving network technologies, wireless data and the Internet. Both the wireless data market and the Internet are growing very quickly and are continuously reaching new customers. The explosive growth of the Internet has fuelled the creation of new and exciting information services.

Most of the technology developed for the Internet has been designed for desktop and larger computers and medium to high bandwidth, generally reliable data networks. Mass-market, hand-held wireless devices present a more constrained computing environment compared to desktop computers. Because of fundamental limitations of power and form-factor, mass-market handheld devices tend to have:

- Less powerful CPUs,
- Less memory (ROM and RAM),
- Restricted power consumption,
- Smaller displays, and
- Different input devices (eg, a phone keypad).

Similarly, wireless data networks present a more constrained communication environment compared to wired networks. Because of fundamental limitations of power, available spectrum, and mobility, wireless data networks tend to have:

- Less bandwidth,
- More latency,
- Less connection stability, and
- Less predictable availability.

Mobile networks are growing in complexity and the cost of all aspects for provisioning of more value added services is increasing. In order to meet the requirements of mobile network operators, solutions must be:

- Interoperable – terminals from different manufacturers communicate with services in the mobile network;
- Scaleable – mobile network operators are able to scale services to customer needs;
- Efficient – provides quality of service suited to the behaviour and characteristics of the mobile network;
- Reliable – provides a consistent and predictable platform for deploying services; and
- Secure – enables services to be extended over potentially unprotected mobile networks while still preserving the integrity of user data; protects the devices and services from security problems such as denial of service.

Many of the current mobile networks include advanced services that can be offered to end-users. Mobile network operators strive to provide advanced services in a useable and attractive way in order to promote increased usage of the mobile network services and to decrease the turnover rate of subscribers. Standard features, like call control, can be enhanced by using WAP technology to provide customised user interfaces. For example, services such as call forwarding may provide a user interface that prompts the user to make a choice between accepting a call, forwarding to another person, forwarding it to voice mail, etc.

The WAP specifications address mobile network characteristics and operator needs by adapting existing network technology to the special requirements of mass-market, hand-held wireless data devices and by introducing new technology where appropriate.

5.2 Requirements

The requirements of the WAP Forum architecture are to:

- Leverage existing standards where possible;

- Define a layered, scaleable and extensible architecture;
- Support as many wireless networks as possible;
- Optimise for narrow-band bearers with potentially high latency;
- Optimise for efficient use of device resources (low memory/CPU usage/power consumption);
- Provide support for secure applications and communication;
- Enable the creation of Man Machine Interfaces (MMIs) with maximum flexibility and vendor control;
- Provide access to local handset functionality, such as logical indication for incoming call;
- Facilitate network-operator and third party service provisioning;
- Support multi-vendor interoperability by defining the optional and mandatory components of the specifications; and
- Provide a programming model for telephony services and integration.

6. Architecture Overview

6.1 The World-Wide Web Model

The Internet World-Wide Web (WWW) architecture provides a very flexible and powerful programming model (Figure 1). Applications and content are presented in standard data formats, and are *browsed* by applications known as *web browsers*. The web browser is a networked application, ie, it sends requests for named data objects to a network server and the network server responds with the data encoded using the standard formats.

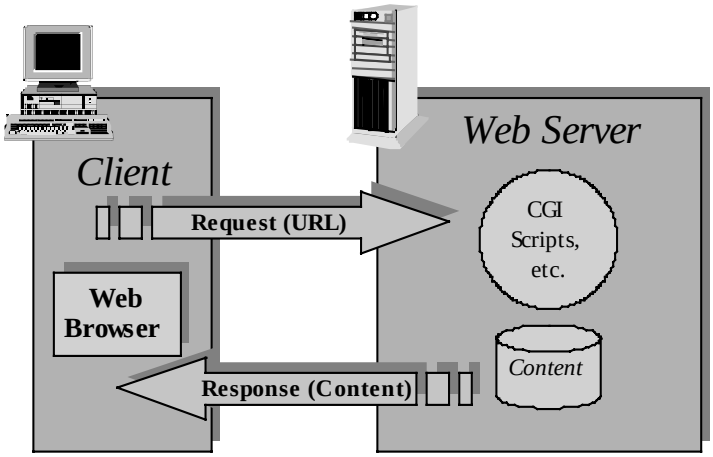

Figure 1. World-Wide Web Programming Model

The WWW standards specify many of the mechanisms necessary to build a general-purpose application environment, including:

- Standard naming model – All servers and content on the WWW are named with an Internet-standard *Uniform Resource Locator* (URL) [RFC1738, RFC1808].
- Content typing – All content on the WWW is given a specific type thereby allowing web browsers to correctly process the content based on its type [RFC2045, RFC2048].
- Standard content formats – All web browsers support a set of standard content formats. These include the HyperText Markup Language (HTML) [HTML4], the JavaScript scripting language [ECMAScript, JavaScript], and a large number of other formats.
- Standard Protocols – Standard networking protocols allow any web browser to communicate with any web server. The most commonly used protocol on the WWW is the HyperText Transport Protocol (HTTP) [RFC2068].

This infrastructure allows users to easily reach a large number of third-party applications and content services. It also allows application developers to easily create applications and content services for a large community of clients.

The WWW protocols define three classes of servers:

- Origin server – The server on which a given resource (content) resides or is to be created.
- Proxy – An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. The proxy typically resides between clients and servers that have no means of direct communication, eg across a firewall. Requests are either serviced by the proxy program or passed on, with possible translation, to other servers. A proxy must implement both the client and server requirements of the WWW specifications.

- Gateway – A server which acts as an intermediary for some other server. Unlike a proxy, a gateway receives requests as if it were the origin server for the requested resource.The requesting client may not be aware that it is communicating with a gateway.

6.2 The WAP Model

The WAP programming model (Figure 2) is similar to the WWW programming model. This provides several benefits to the application developer community, including a familiar programming model, a proven architecture, and the ability to leverage existing tools (eg, Web servers, XML tools, etc.). Optimisations and extensions have been made in order to match the characteristics of the wireless environment. Wherever possible, existing standards have been adopted or have been used as the starting point for the WAP technology.

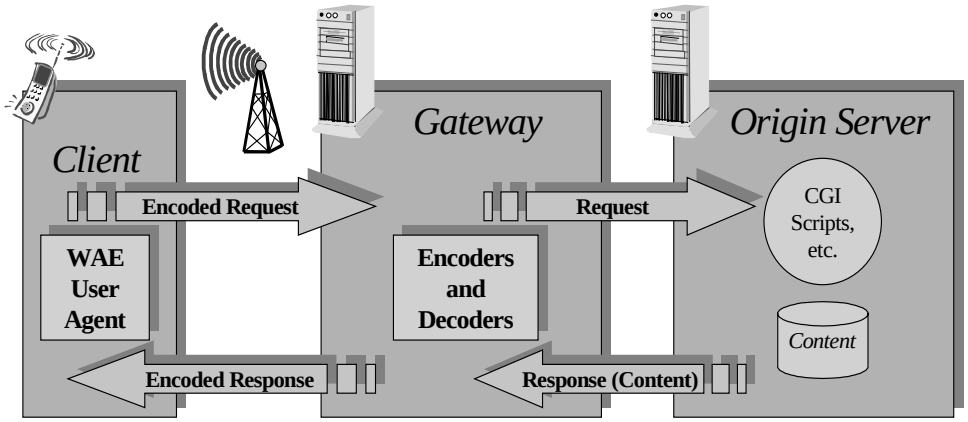

Figure 2. WAP Programming Model

WAP content and applications are specified in a set of well-known content formats based on the familiar WWW content formats. Content is transported using a set of standard communication protocols based on the WWW communication protocols. A *micro browser* in the wireless terminal co-ordinates the user interface and is analogous to a standard web browser.

WAP defines a set of standard components that enable communication between mobile terminals and network servers, including:

- Standard naming model – WWW-standard URLs are used to identify WAP content on origin servers. WWW-standard URIs are used to identify local resources in a device, eg call control functions.
- Content typing – All WAP content is given a specific type consistent with WWW typing. This allows WAP user agents to correctly process the content based on its type.
- Standard content formats – WAP content formats are based on WWW technology and include display markup, calendar information, electronic business card objects, images and scripting language.
- Standard communication protocols – WAP communication protocols enable the communication of browser requests from the mobile terminal to the network web server.

The WAP content types and protocols have been optimised for mass market, hand-held wireless devices. WAP utilises proxy technology to connect between the wireless domain and the WWW. The WAP proxy typically is comprised of the following functionality:

- Protocol Gateway – The protocol gateway translates requests from the WAP protocol stack (WSP, WTP, WTLS, and WDP) to the WWW protocol stack (HTTP and TCP/IP).
- Content Encoders and Decoders – The content encoders translate WAP content into compact encoded formats to reduce the size of data over the network.

This infrastructure ensures that mobile terminal users can browse a wide variety of WAP content and applications, and that the application author is able to build content services and applications that run on a large base of mobile

terminals. The WAP proxy allows content and applications to be hosted on standard WWW servers and to be developed using proven WWW technologies such as CGI scripting.

While the nominal use of WAP will include a web server, WAP proxy and WAP client, the WAP architecture can quite easily support other configurations. It is possible to create an origin server that includes the WAP proxy functionality. Such a server might be used to facilitate end-to-end security solutions, or applications that require better access control or a guarantee of responsiveness, eg, WTA.

6.3 Example WAP Network

The following is for illustrative purposes only. An example WAP network is shown in Figure 3.

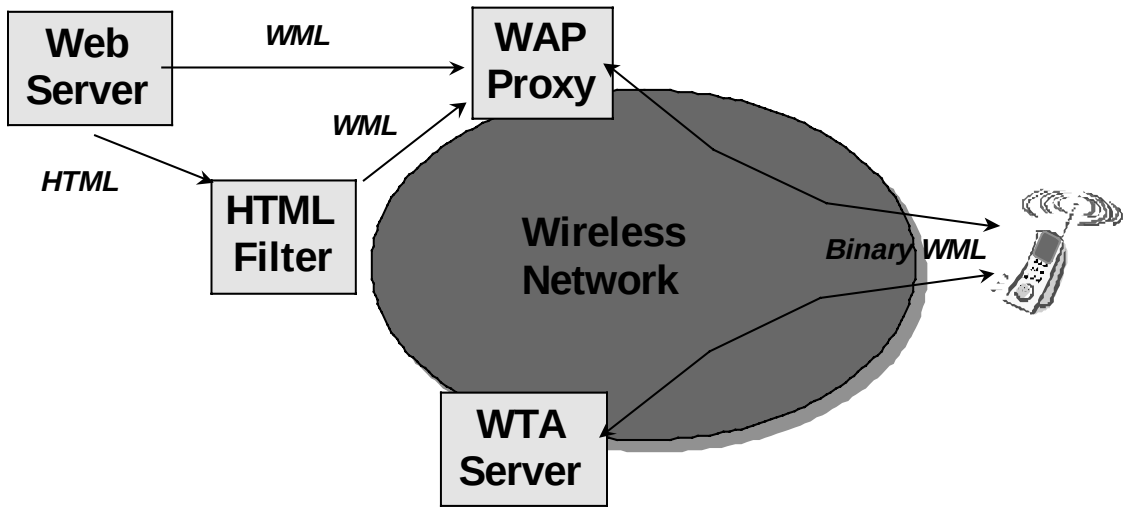

Figure 3. Example WAP Network

In the example, the WAP client communicates with two servers in the wireless network. The WAP proxy translates WAP requests to WWW requests thereby allowing the WAP client to submit requests to the web server. The proxy also encodes the responses from the web server into the compact binary format understood by the client.

If the web server provides WAP content (e.g., WML), the WAP proxy retrieves it directly from the web server. However, if the web server provides WWW content (such as HTML), a filter is used to translate the WWW content into WAP content. For example, the HTML filter would translate HTML into WML.

The Wireless Telephony Application (WTA) server is an example origin or gateway server that responds to requests from the WAP client directly. The WTA server is used to provide WAP access to features of the wireless network provider's telecommunications infrastructure.

6.4 Security Model

WAP enables a flexible security infrastructure that focuses on providing connection security between a WAP client and server.

WAP can provide end-to-end security between WAP protocol endpoints. If a browser and origin server desire end-to-end security, they must communicate directly using the WAP protocols. End-to-end security may also be

achieved if the WAP proxy is trusted or, for example, located at the same physically secure place as the origin server.

7. Components of the WAP Architecture

The WAP architecture provides a scaleable and extensible environment for application development for mobile communication devices. This is achieved through a layered design of the entire protocol stack (Figure 4). Each of the layers of the architecture is accessible by the layers above, as well as by other services and applications.

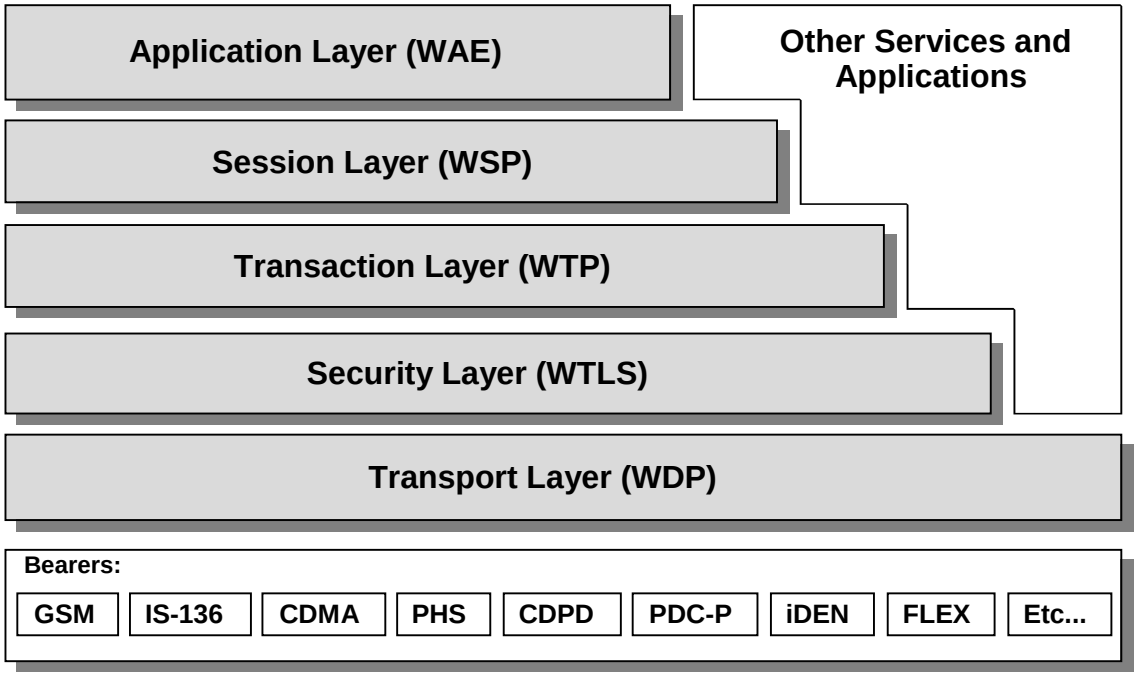

Figure 4. WAP Architecture

The WAP layered architecture enables other services and applications to utilise the features of the WAP stack through a set of well-defined interfaces. External applications may access the session, transaction, security and transport layers directly. The following sections provide a description of the various elements of the protocol stack architecture.

7.1 Wireless Application Environment (WAE)

The Wireless Application Environment (WAE) is a general-purpose application environment based on a combination of World Wide Web (WWW) and Mobile Telephony technologies. The primary objective of the WAE effort is to establish an interoperable environment that will allow operators and service providers to build applications and services that can reach a wide variety of different wireless platforms in an efficient and useful manner. WAE includes a micro-browser environment containing the following functionality:

- Wireless Markup Language (WML) – a lightweight markup language, similar to HTML, but optimised for use in hand-held mobile terminals;
- WMLScript – a lightweight scripting language, similar to JavaScript™;
- Wireless Telephony Application (WTA, WTAI) – telephony services and programming interfaces; and
- Content Formats – a set of well-defined data formats, including images, phone book records and calendar information.

A much more detailed description of the WAE architecture is provided in [WAEoview].

7.2 Wireless Session Protocol (WSP)

The Wireless Session Protocol (WSP) provides the application layer of WAP with a consistent interface for two session services. The first is a connection-oriented service that operates above the transaction layer protocol WTP. The second is a connectionless service that operates above a secure or non-secure datagram service (WDP).

The Wireless Session Protocols currently consist of services suited for browsing applications (WSP/B). WSP/B provides the following functionality:

- HTTP/1.1 functionality and semantics in a compact over-the-air encoding,
- Long-lived session state,
- Session suspend and resume with session migration,
- A common facility for reliable and unreliable data push, and
- Protocol feature negotiation.

The protocols in the WSP family are optimised for low-bandwidth bearer networks with relatively long latency. WSP/B is designed to allow a WAP proxy to connect a WSP/B client to a standard HTTP server. See [WSP] for more information.

7.3 Wireless Transaction Protocol (WTP)

The Wireless Transaction Protocol (WTP) runs on top of a datagram service and provides as a light-weight transaction-oriented protocol that is suitable for implementation in “thin” clients (mobile stations). WTP operates efficiently over secure or non-secure wireless datagram networks and provides the following features:

- Three classes of transaction service:
  - Unreliable one-way requests,
  - Reliable one-way requests, and
  - Reliable two-way request-reply transactions;
- Optional user-to-user reliability - WTP user triggers the confirmation of each received message;
- Optional out-of-band data on acknowledgements;
- PDU concatenation and delayed acknowledgement to reduce the number of messages sent; and
- Asynchronous transactions.

See [WTP] for more information.

7.4 Wireless Transport Layer Security (WTLS)

WTLS is a security protocol based upon the industry-standard Transport Layer Security (TLS) protocol, formerly known as Secure Sockets Layer (SSL). WTLS is intended for use with the WAP transport protocols and has been optimised for use over narrow-band communication channels. WTLS provides the following features:

- Data integrity – WTLS contains facilities to ensure that data sent between the terminal and an application server is unchanged and uncorrupted.
- Privacy – WTLS contains facilities to ensures that data transmitted between the terminal and an application server is private and cannot be understood by any intermediate parties that may have intercepted the data stream.
- Authentication – WTLS contains facilities to establish the authenticity of the terminal and application server.
- Denial-of-service protection – WTLS contains facilities for detecting and rejecting data that is replayed or not successfully verified. WTLS makes many typical denial-of-service attacks harder to accomplish and protects the upper protocol layers.

WTLS may also be used for secure communication between terminals, eg, for authentication of electronic business card exchange.

Applications are able to selectively enable or disable WTLS features depending on their security requirements and the characteristics of the underlying network (eg, privacy may be disabled on networks already providing this service at a lower layer).

See [WTLS] for more information.

7.5 Wireless Datagram Protocol (WDP)

The Transport layer protocol in the WAP architecture is referred to as the Wireless Datagram Protocol (WDP). The WDP layer operates above the data capable bearer services supported by the various network types. As a general transport service, WDP offers a consistent service to the upper layer protocols of WAP and communicate transparently over one of the available bearer services.

Since the WDP protocols provide a common interface to the upper layer protocols the Security, Session and Application layers are able to function independently of the underlying wireless network. This is accomplished by adapting the transport layer to specific features of the underlying bearer. By keeping the transport layer interface and the basic features consistent, global interoperability can be achieved using mediating gateways.

See [WDP] for more information.

7.6 Bearers

The WAP protocols are designed to operate over a variety of different bearer services, including short message, circuit-switched data, and packet data. The bearers offer differing levels of quality of service with respect to throughput, error rate, and delays. The WAP protocols are designed to compensate for or tolerate these varying level of service.

Since the WDP layer provides the convergence between the bearer service and the rest of the WAP stack, the WDP specification [WDP] lists the bearers that are supported and the techniques used to allow WAP protocols to run over each bearer. The list of supported bearers will change over time with new bearers being added as the wireless market evolves.

7.7 Other Services and Applications

The WAP layered architecture enables other services and applications to utilise the features of the WAP stack through a set of well-defined interfaces. External applications may access the session, transaction, security and transport layers directly. This allows the WAP stack to be used for applications and services not currently specified by WAP, but deemed to be valuable for the wireless market. For example, applications, such as electronic mail, calendar, phone book, notepad, and electronic commerce, or services, such as white and yellow pages, may be developed to use the WAP protocols.

7.8 Sample Configurations of WAP Technology

WAP technology is expected to be useful for applications and services beyond those specified by the WAP Forum. Figure 5 depicts several possible protocol stacks using WAP technology. These are for illustrative purposes only and do not constitute a statement of conformance or interoperability.

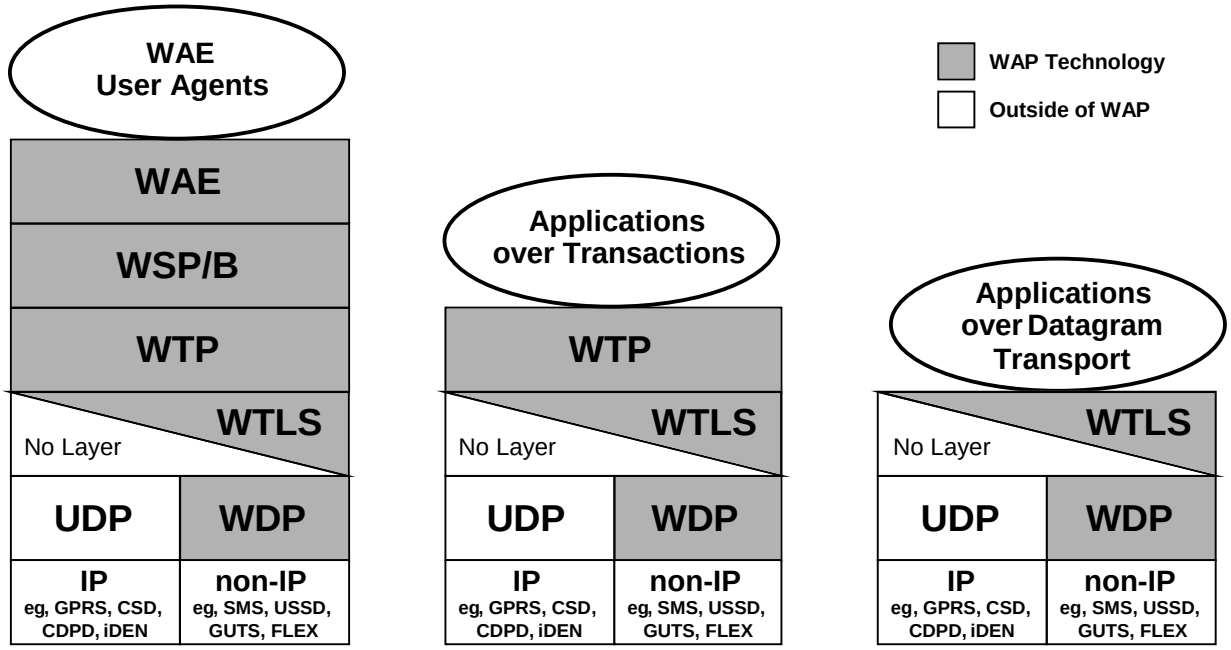

Figure 5. Sample WAP Stacks

The leftmost stack represents a typical example of a WAP application, ie, WAE user agent, running over the complete portfolio of WAP technology. The middle stack is intended for applications and services that require transaction services with or without security. The rightmost stack is intended for applications and services that only require datagram transport with or without security.

8. Compliance and Interoperability

The WAP Forum views multi-vendor interoperability as an important element to the success of WAP products. In order to provide as high a probability as is technically possible that two WAP products developed independently by two different vendors will successfully interoperate, a rigorous definition of conformance, compliance, and testing must be developed. To this end the WAP Forum has created a WAP Conformance Specification [WAPConf] and is working to maintain current information relating to all issues of WAP interoperability.

Successful interoperability can only be achieved by testing products. Testing can be divided into two broad categories of static testing and dynamic testing. Static testing is a manufacturer's statement of the capabilities and functions of a product. Static testing will identify obvious areas of incompatibility between two products, ie, where one implements a feature which the other does not support. All WAP specifications will provide a means for static testing in the form of a Protocol Implementation Conformance Statement (PICS), see [WAPConf] for more details on static testing.

Dynamic testing is the real form of testing which leads to a high degree of confidence that two products will successfully interoperate. Dynamic testing involves the execution or exercising of a product in a live environment, ultimately proving that the product meets the stated claims given in the static test, ie, PICS. There are three general approaches to dynamic testing: pair-wise testing or bake-offs; use of a reference implementation against which all products are measured; and definition of formal test suites containing test cases to be run against a product in a testing laboratory. Each of these approaches to dynamic testing has cost trade-offs and some technical pluses and minuses. The cost of each approach is related to the total number of products that need to be tested. The WAP Forum will promote the most cost effective method that leads to the greatest degree of confidence for successful interoperability given the total number of WAP products available in the market at a given time. This is an evolutionary approach that will change over time as the WAP industry matures. As a starting point the WAP Forum is promoting pair-wise testing in a laboratory environment for new WAP products. As the WAP industry evolves reference implementations may be identified, followed by the definition of formal test suites for WAP specifications.

9. Future Work Items

The Future Work Items list is a collection of areas that warrant further consideration in order to determine whether or not any working groups should be chartered with developing recommendations or specifications. The list is neither prescriptive nor exhaustive. This list is not prioritised in any way, ie, no importance can be attached to the numbering scheme. Areas for consideration can be added or deleted at any time. The list currently contains the following items:

1. Connection-oriented data transport
2. Integration of SIM Toolkit, smartcard and WAP
3. Integration of MExE (ETSI) and WAP
4. Additional integration with the telephony network
5. Downloadable WMLScript libraries
6. Compression (WTLS or other layers)
7. Application levels security, eg, crypto scripting libraries
8. Wider scope of security architecture, including smartcard support, improved handling of end-to-end security, certificate authority hierarchies, etc.
9. Support for streaming multimedia content for higher bandwidth bearers, eg, GPRS
10. Support for multicast data
11. Support for location dependent mobile services, eg, positioning functions and features
12. Downloadable applications
13. Speech API
14. Management Entity Definitions for each layer and across all layers of the WAP
15. Quality of Service for the WAP stack with respect to each bearer service
16. Application Programming Interfaces for each layer of the WAP stack
17. Interoperability Testing (see previous section on compliance and interoperability testing)