Mobile Application Environment Specification

Approved Version 2.4 – 29 Mar 2011

Open Mobile Alliance

OMA-TS-MAESpec-V2_4-20110329-A

Contents

Figures

Tables

1. Scope

The Mobile Application Environment (MAE) is part of the Open Mobile Alliance's effort to specify an application framework for mobile terminals such as mobile phones, pagers, and PDAs. The primary focus is on the mobile browser. The framework integrates, extends and leverages other OMA and non-OMA technologies, including session, transport and security protocols, as well as other Internet technologies such as XML, URIs, scripting, and various media types. This effort enables operators, manufacturers, and content developers to meet the challenges in building advanced and differentiated services and implementations in a fast and flexible manner

This document provides a general overview of the overall MAE architecture and builds on the convergence with the Internet achieved in the previous WAE convergence releases, WAE versions 2.0 and above. This specification also represents the root document of the Mobile Application Environment (MAE) version 2.4 specification hierarchy, i.e., the normative document hierarchy for both the required and optional features which mostly are specified in other documents. For additional information on the WAP architecture, refer to "*Wireless Application Protocol Architecture Specification*" [WAPArch]

Note： MAE is formerly known as WAE (Wireless Application Environment). MAE, as a term, may be used interchangeably with WAE, otherwise terms are clearly distinguished.

2. References

2.1 Normative References

[Browsing24RD] “Browsing Requirements, V2.4”, Open Mobile Alliance™, OMA-RD-BrowsingV2_4, URL:http://www.openmobilealliance.org/

[CacheMod] “User Agent Caching Model, V1.1”, Open Mobile Alliance™. OMA-TS-UACACHE-V1_1. URL:http://www.openmobilealliance.org/

[ESMP11] “ECMAScript Mobile Profile, V1.1”, Open Mobile Alliance™. OMA-TS-ESMP-V1_1. URL:http://www.openmobilealliance.org/

[ESMPCrypto] “Crypto Object for ECMA Script Mobile Profile”, Open Mobile Alliance™. -OMA-WAP-ECMACR-V1_1. URL:http://www.openmobilealliance.org/

[HTTP11] “Hypertext Transfer Protocol -- HTTP/1.1”, RFC2616, R. Fielding et al., June 1999. URL:http://www.ietf.org/rfc/rfc2616.txt

[HTTPBasicAuth] “HTTP Authentication: Basic and Digest Access Authentication”, J. Franks et al., June 1999. URL:http://www.ietf.org/rfc/rfc2617.txt

[HTTPSM] “HTTP State Management Specification, V1.1”, Open Mobile Alliance™. OMA-TS-HTTPSM-V1_1. URL:http://www. openmobilealliance.org/

[ISO10646] “Information Technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane”, ISO/IEC 10646-1:2000.

[PICT] “WAP Pictogram Specification, V1.1”, Open Mobile Alliance™. OMA-WAP-TS-Pictogram-V1_1. URL:http://www.openmobilealliance.org/

[PSTOR] “Persistent Storage Specification”, Open Mobile Alliance™. WAP-227-PSTOR-20011220-a http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2387] The MIME Multipart/Related Content-type”, E. Levinson, August 1998. URL:http://www.ietf.org/rfc/rfc2387.txt

[RFC2388] “Returning Values from Forms: multipart/form-data”, L. Masinter, August 1998. URL:http://www.ietf.org/rfc/rfc2388.txt

[RFC3023] “XML Media Types”, M. Murata et al., January 2001. URL:http://www.ietf.org/rfc/rfc3023.txt

[RFC3236] “The 'application/xhtml+xml' Media Type. M. Baker, P. Stark.” RFC3236. January 2002. URL:http://www.ietf.org/rfc/rfc3236.txt

[RME] “Rich Media Environment, V1.0”, Open Mobile Alliance™. OMA-TS-RME-V1_0, URL:http://www.openmobilealliance.org/

[SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[SMIL21] “Synchronized Multimedia Integration Language (SMIL 2.1)”, W3C Recommendation, Dick Bulterman et al, 13 December 2005. URL: http://www.w3.org/TR/2005/REC-SMIL2-20051213/

[SVG] “Scalable Vector Graphic, V1.0”, ™. OMA-TS-SVG-V1_0, URL:http://www.openmobilealliance.org/

[UAPROF] “WAG UAProf”, Open Mobile Alliance^{TM}. WAP-248-UAPROF. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[URIScheme] “URI Schemes for the Mobile Applications Environment, V1.0”, Open Mobile Alliance™. OMA-TS-URI_Schemes-V1_0, URL:http://www.openmobilealliance.org/

[vBookmark] “Specifications for Ir Mobile Communications (IrMC) Erranta 2000 07 18”, URL: http://www.irda.org/

[vObject] “vObject Minimum Interoperability Profile, V1.0”, Open Mobile Alliance™., OMA-TS-vObjectOMAProfile-V1_0, URL:http://www.openmobilealliance.org/

[WAE23] “Wireless Application Environment Specification , V2.3”, Open Mobile Alliance™. OMA-WAP-TS-MAESpec-V2_3. URL:http://www.openmobilealliance.org/

[WBXML] “WAP Binary XML Content Format”, Open Mobile Alliance™. WAP-192-WBXML, http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WCSS12] “Wireless CSS, V1.2”, Open Mobile Alliance™. OMA-TS-WCSS-V1_2. URL:http://www.openmobilealliance.org/

[WML1] “Wireless Markup Language Version 1.3”, Open Mobile Alliance™. WAP-191-WML. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WML2] “Wireless Markup Language”, Open Mobile Alliance™. WAP-238-WML. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WTAI] “Wireless Telephony Application Interface Specification”, Open Mobile Alliance™. WAP-268-WTAI. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WTLS] “Wireless Transport Layer Security”, Open Mobile Alliance™. WAP-261-WTLS. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[XML] “Extensible Markup Language (XML) 1.0 (Fourth Edition)”, W3C Recommendation 16 August 2006, Tim Bray et al. URL: http://www.w3.org/TR/xml/

[XHTMLBasic11] “XHTML^{TM} Basi c 1.1”, W3C Candidate Recommendation, Shane McCarron, et al., 13 July 2007. URL: http://www.w3.org/TR/2007/CR-xhtml-basic-20070713.

[XHTMLMP13] “XHTML Mobile Profile, V1.3”, Open Mobile Alliance™. OMA-TS-XHTMLMP-V1_3. URL:http://www.openmobilealliance.org/

2.2 Informative References

[CC/PP] “Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation”, W3C Note, Franklin Reynolds, Johan Hjelm, Spencer Dawkins, and Sandeep Singhal, 27 July 1999. URL;http://www.w3.org/1999/07/NOTE-CCPP-19990727/

[CCPPex] “CC/PP exchange protocol based on HTTP Extension Framework”, W3C Note, Hidetaka Ohto and Johan Hjelm, 24 June 1999. URL:http://www.w3.org/1999/06/NOTE-CCPPexchange-19990624.

[CSS21] “Cascading Style Sheets, level 2 (CSS2) Revision 1 Specification”, W3C Candidate Recommendation 19 July 2007, Bert Bos et al..
URL: http://www.w3.org/TR/css21/

[CSSMP20] “CSS Mobile Profile 2.0”, W3C Candidate Recommendation 2007. URL: http://www.w3.org/TR/css-mobile

[E2ESec] “WAP Transport Layer End-to-end Security Specification”, Open Mobile Alliance™. WAP-187-TLE2E. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[ECMA327] Standard ECMA-327, “ECMAScript 3 rd Edition Compact Profile”, ECMA, June 2001, URL:

ftp://ftp.ecma.ch/ecma-st/Ecma-327.pdf

[ECMAScript] Standard ECMA-262: “ECMAScript Language Specification – Edition 3”, ECMA, December 1999. URL: ftp://ftp.ecma.ch/ecma-st/Ecma-262.pdf

[HTML4] “HTML 4.01 Specification”, W3C Recommendation, Dave Raggett, Arnaud Le Hors, and Ian Jacobs, 24 December 1999. http://www.w3.org/TR/html401/

[JavaScript] “JavaScript: The Definitive Guide”, David Flanagan, O’Reilly & Associates, Inc., 1997

[RDF] “Resource Description Framework (RDF) Model and Syntax Specification,” W3C Recommendation, O. Lassila, R Swick, February 1999. URL:http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[RFC2045] “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, N. Freed, et al., November 1996. URL:http://www.ietf.org/rfc/rfc2045.txt

[RFC2046] “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, N. Freed et al., November 1996. URL:http://www.ietf.org/rfc/rfc2046.txt

[RFC2047] “MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text”, K. Moore, November 1996. URL:http://www.ietf.org/rfc/rfc2047.txt

[RFC2318] “The text/css Media Type”, H.Lie et al., March 1998. URL://www.ietf.org/rfc/rfc2318.txt

[RFC2965] “HTTP State Management Mechanism”, D. Kristol et al., October 2000. URL:http://www.ietf.org/rfc/rfc2965.txt

[RFC3986] “Uniform Resource Identifiers (URI): Generic Syntax”, T. Berners-Lee, et al., January 2005. URL:http://www.ietf.org/rfc/rfc3986.txt

[RFC3987] “Internationalized Resource Identifiers”, M. Duerst, M Suignard, January 2005. URL:http://www.ietf.org/rfc/rfc3987.txt/

[RFC4329] “Scripting Media Types”, B. Hoehrmann, April 2006, http://www.ietf.org/rfc/rfc4329.txt

[TLS] “The TLS Protocol Version 1.0”, T. Dirks and C. Allen, January 1999, http://www.ietf.org/rfc/rfc2246.txt

[VCAL] "vCalendar - the Electronic Calendaring and Scheduling Format", version 1.0, The Internet Mail Consortium (IMC), September 18, 1996, URL:http://www.imc.org/pdi/vcal-10.doc

[VCARD] “vCard - The Electronic Business Card”, version 2.1,The Internet Mail Consortium (IMC), September 18, 1996, URL:http://www.imc.org/pdi/vcard-21.doc

[WAE20] “Wireless Application Environment Specification – version 2.0”, Open Mobile Alliance™. WAP-236-WAESpec. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WAE21] “Wireless Application Environment Specification , V2.1”, Open Mobile Alliance™. OMA-WAP-WAESpec-V2_1. URL:http://www.openmobilealliance.org/

[WAEMedia] “WAE Defined Media Type”, Open Mobile Alliance™. WAP-237-WAEMT. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WAEOverview1] “Wireless Application Environment Overview”, Open Mobile Alliance™. WAP-195-WAEOverview. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WAESpec1] “Wireless Application Environment Specification”, Open Mobile Alliance™. WAP-190-WAESpec. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WAP1] “WAP June 2000 (WAP 1.2.1) Conformance Release”, Open Mobile Alliance™. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WAPArch] “WAP Architecture Specification”, Open Mobile Alliance™. WAP-210-WAPArch. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WMLScript] “WMLScript Language Specification”, Open Mobile Alliance™. WAP-193-WMLS. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WMLStdLib] “WMLScript Standard Libraries Specification”, Open Mobile Alliance™. WAP-194-WMLSL. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WTA] “Wireless Telephony Application Specification”, Open Mobile Alliance™. WAP-266-WTA. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[WTP] “Wireless Transaction Protocol Specification”, Open Mobile Alliance™. WAP-224-WTP. http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[XHTML] “XHTML™ 1.1 - Module-based XHTML”, W3C Recommendation, 31 May 2001. URL:http://www.w3.org/TR/2001/REC-xhtml11-20010531

[XHTMLMP12] “XHTML Mobile Profile, V1.2”, Open Mobile Alliance™. OMA-TS-XHTMLMP-V1_2. URL:http://www.openmobilealliance.org/

[XHTMLMod] “Modularization of XHTML™”, W3C Recommendation, Murray Altheim et al., 10 April 2001. URL:http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410

3. Terminology and Conventions

3.1 Conventions

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]. All sections, except “Scope”, “Terminology and Conventions” and “MAE Version 2.4 Introduction” are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

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

Bytecode Content encoding where the content is typically a set of low-level opcodes (i.e., instructions) and operands for a targeted hardware (or virtual) machine.

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

Client Server Communication Communication between a client and a server. Typically the server performs a task (such as generating content) on behalf of the client. Results of the task are usually sent back to the client (e.g., generated content).

Content Synonym for data objects.

Deprecated A deprecated feature (e.g. specification, element or attribute) is one that has been outdated by a newer feature. Deprecated features are defined in the specification and are clearly marked as deprecated. Deprecated features may become obsolete in a future version.

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 and 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.

ECMAScript A scripting language produced and managed by the European Computer Manufacturers Association (ECMA) which provides a common scripting language for the computer industry. Early version was called as “Javascript” [JavaScript]

Gateway (or WAP Gateway) A server which acts as an intermediary for some other server. A gateway performs protocol transformation as well as encoding/decoding content.

Host Object ECMAScript objects provided by the user agent for the purpose of interaction with the loaded document.

Hypermedia Transfer The hypermedia transfer services provides for the transfer of self-describing hypermedia resources. The HTTP (Hypertext Transfer Protocol) [HTTP11] provides the hypermedia transfer service over secure and non-secure connection-oriented transports over connection-oriented protocol stack.

MAE Version The version of the MAE User Agent. MAE Version starts from 2.4, following WAE version 2.3.

Media type A MIME media type or an identifier for a given data type.

Obsolete The feature is outdated and removed from the specification.

For user agent developers, properties that have been designated as obsolete should not be built into new implementations. Older implementations may still support these properties.

Note: Content providers are strongly discouraged from using obsolete properties.

Note: All implementations must be able to parse, and at a minimum ignore obsoleted syntax.

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 URI. Resources may be available in multiple representations (e.g., 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 typically used by a user 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 access a resource.

MAE User Agent (or User Agent) A User Agent is any software or device that interprets markup and scripting languages or other content. This may include textual browsers, voice browsers, search engines, etc. MAE User agent is a uer agent to interpret contents defined in MAE.

WAE version The version of the WAE User Agent. The version of the WAE User Agent may be uniquely identifiable by the WAP version, e.g. WAP version 1.1 contains WAE version 1.1, or it may be a feature of the WAP version in which case the WAE versioning mechanisms are used to determine the WAE User Agent version.

WAP1 WAP Version 1, nominally the latest point release, e.g. WAP V1.2.1, unless otherwise noted.

WML The Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrowband device, e.g., a phone.

WMLScript A scripting language used to program the mobile device. WMLScript is an extended subset of the ECMAScript scripting language.

XML The Extensible Markup Language is a World Wide Web Consortium (W3C) standard for Internet markup languages, of which WML is one such language. XML is a restricted subset of SGML.

vCalendar Internet Mail Consortium (IMC) electronic calendar record.

vCard Internet Mail Consortium (IMC) electronic business card.

3.3 Abbreviations

API	Application Programming Interface
CPI	Capability and Preference Information
CSS	Cascading Style Sheets
ECMA	European Computer Manufacturer Association
EFI	External Functionality Interface
ESMP	ECMAScript Mobile Profile
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol [HTTP11]
IMC	Internet Mail Consortium
ISO	International Standards Organisation
MAE	Mobile Application Environment
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
OTA	Over The Air
PDA	Personal Digital Assistant
RFC	Request For Comments
RME	Rich Media Environment
SIM	Subscriber Identification Module
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics

UAProf	User Agent Profile
URI	Uniform Resource Identifier [RFC3986]
URL	Uniform Resource Locator [RFC3986]
W3C	World Wide Web Consortium
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WBMP	Wireless BitMaP
WCSS	Wireless Cascading Style Sheets [WCSS12](formerly known as WAP CSS)
WDP	Wireless Datagram Protocol
WML	Wireless Markup Language (WML1 or WML2)
WML1	Wireless Markup Language Version 1.3
WML2	Wireless Markup Language Version 2.0
WSP	Wireless Session Protocol
WTA	Wireless Telephony Application
WTAI	Wireless Telephony Application Interface
WTP	Wireless Transaction Protocol
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language

4. MAE Version 2.4 Introduction

This section is informative.

The Mobile Application Environment (MAE) version 2.4 is a general-purpose application environment, whose core is the mobile browser, where operators and service providers can build their applications and services for a wide variety of mobile platforms.

4.1 Background

This section is informative.

WAE, since WAP version 1[WAP1], has been based on the World Wide Web (WWW) technologies with additional optimisations for wireless devices and networks, taking account of the constraints wireless devices and networks present such as smaller displays, limited user-input facilities, narrower bandwidth, higher latency, limited memory, limited CPU, etc.

The range of devices accessing the Internet continues to evolve with enabling the new mobile use-cases.

The developments in base Internet technology and the device and wireless network technologies allows WAE version 2 and above to converge with, or adopt, more standard Internet technologies than was possible in WAP version 1. This ongoing convergence will maximise the ease with which WAP-enabled systems can use existing Internet/WWW applications and content. It will also simplify OMA's adoption of new evolving services.

The version history from WAE 2.0 to MAE 2.4 is illustrated in Figure 1.

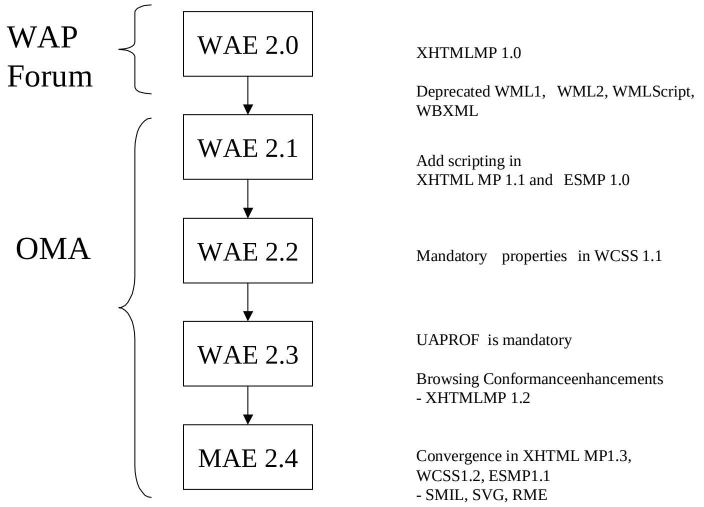
WAP Forum
OMA
WAE 2.0
WAE 2.1
WAE 2.2
WAE 2.3
MAE 2.4
XHTMLMP 1.0
Deprecated WML1, WML2, WMLScript, WBXML
Add scripting in XHTML MP 1.1 and ESMP 1.0
Mandatory properties in WCSS 1.1
UAPROF is mandatory
Browsing Conformanceenhancements - XHTMLMP 1.2
Convergence in XHTML MP1.3, WCSS1.2, ESMP1.1 - SMIL, SVG, RME

Figure 1: Version History from WAE 2.0 to MAE 2.4

This release, MAE 2.4 provides markup convergence to corresponding W3C standards (XHTMLBasic 1.1[XHTMLBasic11], CSS Mobile Profile 2.0[CSSMP20]) and some commonly adopted scripting enhancements.

4.2 Goals and Requirements for MAE version 2.4

This section is informative.

The high-level goals and requirements for MAE version 2.4 are stated in this section. The formal requirements are specified in [Browsing24RD]. The specifications associated with the goals and requirements will be delivered as part of MAE version 2.4.

4.2.1 Goals

MAE version 2.4 is an evolution of WAE version 2.3. The goals are:

- To take advantage of the work of other standards bodies and to converge with those technologies.
  Furthermore, work done in the OMA has been fed back into other standards bodies to achieve successful convergence. Specific standards bodies with which MAE version 2.4 has converged with, are:
  - ➢ W3C
  - ➢ ECMA (introduced in WAE version 2.1)
- To enable optimised access to applications and services from the current and emerging generations of mobile devices. These devices are generally assumed to be small, battery operated, have relatively limited memory and CPU power. Their small form factor results in limited display space and restricted input facilities.
  For these reasons, the application environments and delivery are optimised for efficient use of device resources. Furthermore, WAP architecture and protocols maximise the user experience given the limited display space and restricted input facilities.
- To create system and application services that are sensitive to the limited bandwidth requirements of mobile devices, including but not limited to response time, round trip service cost, and cost of bandwidth resources.
- To enhance and extend the web-centric application model to support wider varieties of mobile devices.
- To allow the creation of Man Machine Interfaces (MMIs) with maximum flexibility and ability for the vendor to enhance the user experience. In this way vendors can provide distinct user interfaces.
- To have security that is consistent and interoperable with Internet security models.
- To ensure the privacy of the user’s personal information such as user location, credit card information, address and phone numbers. Many types of Internet content, including the mobile internet content, are more exposed to user privacy issues during exchanges between origin servers and clients. MAE provides the means to protect the transfer of personal information and ensures an adequate degree of service authentication.
- To target common character codes for international use. This includes international symbols and pictogram sets for end users, and local-use character encoding for content developers.
- To enable the personalisation and customisation of the device, the content delivered to it, and the presentation of the content. It is possible for such personalisation and customisation to be performed by a variety of parties, including end-users, operators, service providers, infrastructure vendors and device vendors.

4.2.2 Requirements

- Backward compatibility – Special attention has been paid to ensure a smooth transition of the standard. Potential discontinuities have been identified, and handled by creating a feasible migration path, allowing preservation of the investment in services built on WAE version 2.
- Convergence with W3C Specifications
  - ➢ XHTML[XHTML] defined by W3C is a reformulation of HTML in XML [XML]. By including XHTML in the OMA specifications, OMA will easily incorporate new technologies from the W3C.

- CSS [CSS21] is a style sheet language to control the presentation of HTML/XML documents. By specifying a suitable subset of CSS for mobile terminals, authors can be guaranteed a set of the same presentation mechanisms as is used in the internet in general.

- Convergence with ECMAScript - the established scripting technology used in today's Internet applications. WAE versions 2.1 and above achieve convergence through the adoption of a profile of ECMAScript [ECMAScript] Edition 3. By adopting an appropriate profile of ECMAScript, content written for MAE user agents will be renderable by common Internet browsers while the MAE user agents will be able to process significant amounts of Internet content which utilises ECMAScript.
- Prescription of User Agent Behaviour/Processing model will be specified to ensure interoperability.
- The caching model for User Agents will be defined to ensure interoperability.
- Description of specific Media types will be described.
- OMNA Database must be properly maintained.

Legacy technologies not defined here are available with references to [WAE23].

4.3 MAE Specification Organisation and How to Read it

This section is informative.

MAE ultilises building blocks, such as markup and script languages and combinations thereof, to achieve its goals. Further, it is organised to provide building blocks other enablers may use. To help with this the following building block terminology is adopted.

> **Technology component**: any technical specification about a building block which is orthogonal to other technology components, e.g. a markup language such as XHTMLMP that is transparent to any scripting languages and stylesheet languages. It is intended to be reusable in other OMA Enablers (e.g. to put color in text, Enablers can use WCSS). A combination of technology components can itself be a technology component.
>
> **Integration**: any conformance to enable multiple building blocks (e.g. technology components and underlying executive environment components which MAE assumes; the latter includes image decoder for example.) in an integrated way on a client.
>
> **User Agent conformance description**: any conformance statements that define a specific environment which uses one or multiple technical components (e.g. integrated BACK behavior in WML1.3 and XHTML MP co-existant environment, guaranteed minimum functional support of a feature in a certain environment, how to use a secure connection in a browser environment) .

The following guideline can be applied to the readers:

Enabler designers: For those who would like to implement or reuse the technical components in the browsing enabler, please refer directly to the appropriate specification. This is an inventory to find the appropriate technical component in Section 5 MAE - the technology components for Content.

User Agent Implementors: For those who would like to implement a browsing user agent, they need both component specs and MAEspec to ensure complete conformance.

Security Designers: For those who would like to reuse browser-level security, the browsing security and access control subsection needs to be considered.

4.4 MAE Version 2.4 Components

This section is informative.

Major components of MAE include: XHTML Mobile Profile, WCSS, ECMAScript Mobile Profile (introduced in WAE version 2.1). WML, WMLScript, WBMP and WBXML are deprecated components. See [WAE23]. Section 5.3 “WAE Version 2.3 Components”. The complete list is defined by the following table.

Component		Reference	Reference in MAEspec
XHTML Mobile Profile		[XHTMLMP13]	5.1.1
WML1		WML1	5.1.3
	WBXML	See [WAE23] Section 6.5	5.1.3, [WAE23] Section 6.5
WML2		WML2	5.1.2
ECMAScript Mobile Profile		[ESMP11]	5.3.2
WCSS		[WCSS12]	5.2.1
vObject		[vObject]	5.5
	vBookmark	[vObject] Section 6	5.5.1
	vCard	[vObject] Section 7	5.5.1
	vCalendar	[vObject] Section 8	5.5.1
SVG		SVG	5.4.2
SMIL		[SMIL21]	5.4.3
RME		[RME]	5.4.4
Pictogram		[PICT]	5.4.1
WBMP		[WAEMedia]	5.4.5, [WAE23] Section 6.6.1

Table 1: MAEspec Technology Components for Content

The following table describes the integration components.

Component		Reference	Reference in MAEspec
Transport Service Integration	HTTP	[XHTMLMP13]	6.2.4.1
	WAP-PUSH	See [WAE23] 7.5	See [WAE23] 7.5
	HTTP SM	HTTPSM	6.2.4.3
	Cache	CacheMod	6.2.4.2
	WDP	See [WAE23] Section 6.7	See [WAE23] 7.8.1
	WSP	See [WAE23] Section 7.1.1	[WAE23] Section 7.1.1
	WTP	WTP	
User Agent Integration	URIscheme	[URIScheme]	6.1.4
	PSTOR	PSTOR	See [WAE23] Section 7.14
	EFI	See [WAE23] Section 7.10	6.1.5
	WTA	WTA	6.1.6
Application Layer Client-Server Negotiation	UAProf	UAPROF	6.4.2

Table 2: MAEspec Technology Components for Integration

The following figure shows the full set of components defined by MAE, and notes those that are deprecated.

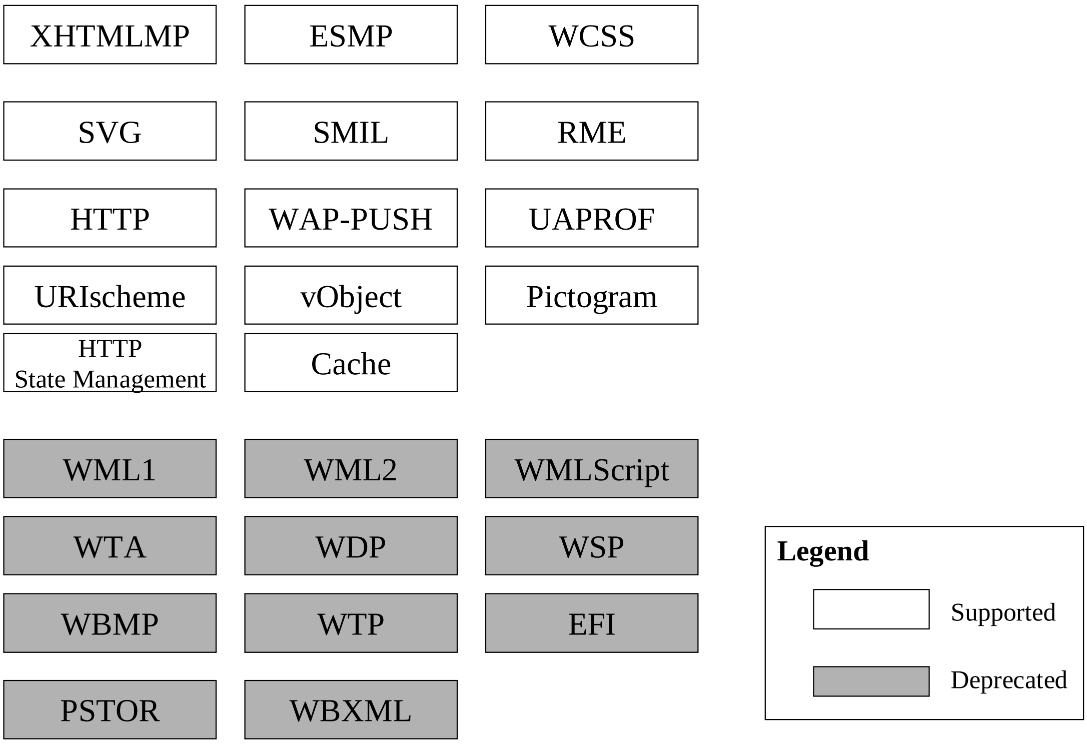

Figure 2: MAE Components in relation to the MAE version 2.4 architecture

5. MAE - the technology components for Content

The MAE components include:

- XHTML Mobile Profile version 1.3 [XHTMLMP13]

  XHTMLMP is mandatory for MAE user agents and provides the content authoring language for MAE with a feature set intended to meet the requirements of the mobile domain, XHTMLMP 1.3 is a converged release and share the DTD with W3C XHTML Basic 1.1.

- Wireless Cascading Style Sheets 1.2 [WCSS12]

  WCSS is mandatory for MAE user agents. WCSS assures content developers and service providers of a minimum level of supported function. All the mandatory properites are identical to W3C CSS Mobile Profile 2.0 [CSSMP20].

- ECMAScript Mobile Profile 1.1 [ESMP11]

  ESMP is mandatory for MAE user agents, even Script is optional in XHTMLMP with a recommendation for script to be [ESMP11] if supported.

- Pictogram [PICT]

  A pictogram is an icon-like image that is rendered within the text, and shows more intuitive information than text Pictogram is optional for MAE user agents.

Details of the languages and and the media types associated with them are described in the following subsections and the associated referenced specifications.

5.1 Markup Languages

5.1.1 XHTML Mobile Profile

This section is normative.

XHTML Mobile Profile version 1.3 [XHTMLMP13] enhances XHTML Mobile Profile version 1.2[XHTMLMP12] by adding additional support for forms including file upload,, the definition and use of the object element, and an extention to allow the setting of input mode for text input. MAE Version 2.4 user agents must support XHTML Mobile Profile version 1.3. [XHTMLMP13]. See section 5.3 for details of related scripting language support.

Conformance: A MAE User Agent MUST support the textual form of XHTML Mobile Profile [XHTMLMP13] (MAESpec-ML-C-005).

5.1.2 WML2

WML 2.0 [WML2] may be provided for those environments that chose this option rather than separate XHTML Mobile Profile [XHTMLMP13] and WML 1.3 as the basis of a hybrid MAE 2.4 environment. For WML2 conformance, see [WAE23] Section 6.2.2 “WML2”. WML2 is deprecated.

5.1.3 WML 1

WML 1.3 [WML1] defines the latest pre-convergence WML version providing the basis for backwards compatibility with legacy content. WML 1.3 is recommended to fulfull this required WML backwards compatibility functionality. For WML1 conformance, see [WAE23] Section 6.2.3 “WML1”. WML1 is deprecated.

WBXML [WBXML] is a defined XML compression technology and is used by some components asspociated with legacy WAE. It is deprecated in MAE version 2.4. Refer to [WAE23] Section 6.5 “WBXML”.

5.2 Style Sheet Language

5.2.1 Wireless CSS

This section is normative.

Wireless CSS [WCSS12] is an extended subset of the *Cascading Style Sheet* (CSS) [CSS21], which is used to specify document presentation in conjunction with WML2 and XHTML Mobile Profile. The only extension to CSS 2.1 is the support W3C-defined Marquee functionality [WCSS12].

The **`text/css`** MIME media type[RFC2318] is used to represent Wireless CSS style sheets, thus no specific MIME media type for Wireless CSS has been defined.

The use of style, and in particular [WCSS12], to achieve presentational control is required for all devices capable of supporting it. The intent is that devices will support style unless it is impractical for that class of device, i.e. the device has insufficient user interface capabilities which prevents its meaningful use such as a fixed format, non UI programmable low-end device.

Some features in WAP version 1 can be achieved by both the WML proprietary elements/attributes and the XHTML elements/attributes with CSS properties. Preference should be given to the XHTML expression and thus content authors are encouraged to use the latter if possible, to leverage the convergence of WAP and the standard Internet.

See [WCSS12] for more information.

Conformance: The User Agent MUST support Wireless CSS as defined in [WCSS12] (MAESpec-STY-C-002).

5.3 Scripting languages

This section is normative

WAE version 2.1 introduced the ECMAScript Mobile Profile scripting environment which is the preferred scripting environment, this preference being indicated by its “required” status for all conformant WAE user agents of version 2.1 and above. The WMLScript and its associated WMLScript Standard Libraries scripting environment remains part of WAE version 2.1 and above though its status is deprecated to be consistent line with WML. The inclusion of WMLScript and WMLScript Standard Libraries provides a recognised backwards compatibility path.

ECMAScript Mobile Profile is the scripting mechanism used for all content and services aimed at MAE version 2.4 user agents.

5.3.1 WMLScript

WMLScript is a scripting language for WML. For the scripting language and associated library functions, see [WAE23] Section “6.4.1 WMLScript”.

For details of WMLScript and its use in MAE, including conformance and managing access to WMLScript, see [WAE23] Section 6.4.3 “Managing access to WMLScript content in WAE”. WMLScript is deprecated.

5.3.2 ECMAScript Mobile Profile

This section is normative.

ECMAScript Mobile Profile [ESMP11] provides the general scripting capability for the WAE version 2.1 user agent and is consistent with the embracing of standard Internet technologies in the WAP version 2 architecture [WAPArch].

ESMP is based on a profile of ECMAScript [ECMAScript] targeted at mobile devices. ESMP inherits mobile constraints from the ECMA327 [ECMA327] profile of ECMAScript Edition 3 [ECMAScript] but there are small differences in functionality between ECMA327 and the core functionality of ESMP, e.g. version property. ESMP also adopts well-known and appropriate host objects to provide a common, interoperable programming environment to application writers.

ESMP complements the XHTML markup language [XHTMLMP13] and style capabilities of WCSS in MAE. The following list contains some capabilities that are not supported by XHTMLMP:

- The ability to check the validity of user input (forms validation).
- The ability to apply mathematic and procedural logic locally to document data.
- Providing access to facilities of the device. For example, on a phone, allow the programmer to make phone calls, send messages, add phone numbers to the address book, access the SIM card etc.
- The ability to generate messages and dialogs locally, reducing the need for expensive round-trip for alerts, error messages, confirmations etc.
- The ability to handle events.
- The ability to allow the dynamic creation and/or modification of documents on the client.
- The ability to manage asynchronous interactions with a server.

ESMP is designed to overcome these limitations and to provide programmable functionality that can be used over narrowband communication links and in clients with limited capabilities.

ESMP is intended to provide interoperability between MAE capable devices implementing ESMP and applications or content which use the feature combination from ECMAScript and additional host objects that represents ESMP.

Mechanisms for extensibility and version control are provided.

Support for cryptographic functions is handled by [ESMPCrypto]. See Section 6.5.1.5 of this document.

MAE version 2.4 user agents must support ESMP to provide the scripting capability for XHTML Mobile Profile version 1.3 [XHTMLMP13].

Conformance: A User Agent MUST support ECMAScript Mobile Profile [ESMP11] for any content using an XHTMLMP MIME type (MAESpec-ESMP-C-001) .

For the processing of XHTMLMP and ESMP through WAP proxies, refer to WAESpec 2.3[WAE23].

5.4 Graphics Objects and Applications

5.4.1 Pictogram

This section is normative.

"WAP Pictogram Specification" [PICT] defines the common pictogram set and the format of its use within content. The common pictogram set is a set of pictograms that the User Agents recognise. Content authors may use these images to promote efficiency of communication, data transfer and network traffic. Manufactures may install images of pictograms that are proprietary to a device (e.g., size, colour, image format, etc.).

To meet the requirements of the worldwide market, the common pictogram set is divided into several classes. For example, some pictograms have operational, cultural and time independent semantics and are intended for general use. Other pictograms are included in a glossary of symbols representing certain embodiments.

Conformance: A MAE User Agent MAY support pictograms [PICT] (MAESpec-PICT-C-001).

5.4.2 SVG

SVG is a language for describing two-dimensional graphics and graphical applications in XML [XML].

Conformance: A User Agent MAY support SVG [SVG](MAESpec-SVG-C-001).

5.4.3 SMIL

SMIL is a language that enables simple authoring of interactive audiovisual presentations [SMIL21]. MAE user agents may support Mobile Profile and Extended Mobile Profile of [SMIL21].

Conformance: A User Agent MAY support SMIL Mobile Profile and Extended Mobile Profile [SMIL21] (MAESpec-SMIL-C-001) .

5.4.4 RME

RME is an enabler to provide enhanced rich media services. RME [RME] is an environment to facilitate rich media user experience.

Conformance: A User Agent MAY support RME [RME] (MAESpec-RME-C-001).

5.4.5 WBMP

WBMP is a graphical image format for WML. For WBMP conformance, see [WAE23] Section 6.6 “Graphical Images”.

5.5 The MAE application specific components

The MAE application specific components are considered to be a component which has application specific semantics in MAE.

The MAE application specific components include:

- vObjects

  Profiles of electronic business cards, Internet calendar and scheduling core objects, and bookmark are defined in [vObject].

- Other content types:

  MAE does not require support for other content types though provides the means to support them through extensibility mechanisms adopted through MAE itself or the components of it.

Details of the languages and and the media types associated with them are described in the following subsections and the associated referenced specifications.

5.5.1 vObject

This section is normative.

[vObject] defines the electronic business card, Internet calendar and bookmark object formats.

For WDP or WSP based exchange of electronic business cards, see [WAE23] Section 6.7 “The Electronic Business Card Format (vCard)”.

For WSP and WDP based transfer of Internet caldenar, see [WAE23] Section 6.8 “Internet Calendaring and Scheduling Core Object (vCalendar)”.

The conformances for user agents supporting electronic business cards are specified in [vObject] Section 6 “vCard 2.1 Minimum Interoperability Profile and Implementation Guidelines”.

The conformances for user agents supporting Internet calendaring and scheduling core object are specified in [vObject] Section 7 “vCalendar 1.0 Minimum Interoperability Profile and Implementation Guideline”.

Conformances for user agents supporting bookmark are specified in [vObject] Section 8 “vBookmark1.0 Minimum Interoperability Profile and Implementation Guideline”.

Conformance: WAE User Agents supporting vCard[VCARD] exchange through HTTP MUST support the **`text/x-vCard`** MIME type. WAE User Agents supporting vCard exchange MAY use the .vcf file extension name in addition to the MIME type, or if the correct MIME type is not provided, when determining the data type (MAESpec-VCAL-C-002).

Conformance: WAE User Agents supporting vCalendar**[VCAL]** exchange through HTTP MUST support the **`text/x-vCalendar`** MIME type. WAE User Agents supporting vCalendar exchange MAY use the .vcs file extension name in addition to the MIME type, or if the correct MIME type is not provided, when determining the data type (MAESpec-VCAL-C-002) .

Conformance: MAE User Agents supporting electronic business card MUST support vCard conformances in [vObject](MAESpec-VCARD-C-006).

Conformance: MAE User Agents supporting Internet calendaring and scheduling core object MUST support vCalendar conformances in [vObject] (MAESpec-VCAL-C-005).

Conformance: MAE User Agents supporting Internet bookmark object MUST support bookmark conformances in [vObject] (MAESpec-VBOOKC-001).

6. MAE Integration

This section is normative.

This section describes integration with underlying layers and conformance among multiple component specifications.

6.1 User Agent Integration

6.1.1 Navigation History

This section is normative.

The MAE User Agent must include a navigational history model that allows the author to manage backward navigation in a convenient and efficient manner. The user agent history is modelled as a stack of entries that represent the resources in the navigational path the user traversed to arrive at the current location. The stack is configured temporally, such that the newest entry is at the top of the stack and the oldest entry is at the bottom of the stack. The following operations must be performed on the history stack:

- Push - a new entry is pushed onto the history stack as an effect of forward navigation.
- Pop - the current entry (top of the stack) is popped as a result of backward navigation.

Additional operations may be defined for the processing of specific document types.

As each document or document fragment is accessed via an explicitly specified URI, (e.g., via the `href` attribute in `<a>`) an entry for the resource is added to the history stack even if it is identical to the most recent entry. At a minimum, each entry must record the resource request information that comprises the absolute URI of the resource, the method (get or post) used to access the resource. The exact composition of the stack entry is document type specific.

Conformance: The User Agent MUST implement the navigation history model and support all the operations on it defined in [WML2] Section 5.2.2 “Navigation History” (MAESpec-UAB-C-001).

6.1.2 The BACK Key

This section is normative.

The BACK key is a user interface object (physical key, soft key, or rendered user interface control) provided by the user agent to return to the previously viewed document. The BACK key is always available to the end user. The BACK key is commonly known as the "back button" on an HTML browser.

Given the OMA’s goal to provide a language with W3C convergence and the ability to navigate to a previously viewed document is a de-facto standard in desktop HTML browsers, the MAE 2.4 User Agent must provide the end user access to a BACK key at all times. This BACK key may be represented as a physical key, a soft key, or any rendered user interface control as long as it is available to the end user at all times.

Conformance: The MAE User Agent MUST provide the end user access to a BACK key at all times (MAESpec-UAB-C-002).

Conformance: When the BACK key is activated the User Agent MUST perform a pop operation as defined in [WML2] Section 5.2.2 “Navigation History” (MAESpec-UAB-C-003).

6.1.2.1 The Back Key in WML1

This section is normative.

For MAE user agent conformance for BACK key in WML, see [WAE23] section 7.12.3 “The BACK Key”.

6.1.3 Minimum User Interface Processing Capacity for HTTP User Agent

This section is normative.

The user agent should support the minimum number of bytes in a textbox (e.g. used for the HTTP authentication dialog).

Conformance: A mobile web browser MUST support at least 80 bytes in a text box , either in plain text box or password text box (e.g. used for the HTTP authentication) (MAESpec-UAB-C-006).

6.1.4 URI Schemes

This section is normative.

Conformance: MAE defines a number of URI schemes to identify resources. The MAE user agent SHALL support the OMA URIscheme specification [URIScheme] (MAESpec-URI-C-007).

MAE User Agents and WAP proxies must support a URI with a minimum length. of 1024 characters (See Note for definition).

Conformance: MAE User Agents and WAP Proxy MUST support a URI with a minimum length of 1024 Unicode/ISO 10646 encoded characters as defined in ISO10646 [RFC3987] (MAESpec-URI-C-001).

NOTE: The length of the URI is not specified in terms of octets as this leads to varying sizes of URIs depending on the internal implementation of a User Agent. As an example, a URI [RFC3986] would have all characters UTF-8 encoded into one to four octets (bytes) per character. Any unreserved octets would then be percent encoded, yielding up to 12 octets for a single character, or a worst-case encoding of 12kbyte to store a 1024 character URI. A URI stored as an IRI [RFC3987] would have an encoding of two bytes per character or 2kbyte.

For WAP Proxy conformance, see [WAE23] Section 7.2 "URIScheme".

6.1.4.1 Local URI Schemes

This section is informative.

There are several URI schemes that are local to the MAE environment. For example:

- `wtai` – Access to some of the Wireless Telephony Application Interface (WTAI) function libraries can be handled through URI "calls" using the dedicated WTAI URI encoding scheme. See [WTAI] for more information. This URIscheme is deprecated in MAE version 2.4.
- `pict` – A pictogram representation can be specified using PICT URI encoding scheme. See **[PICT]** for more information.
- `wps` – The base level persistent storage system's naming scheme. See [PSTOR] for more information. This URIscheme is deprecated in MAE version 2.4.
- `efi` – This scheme identifies the name as belonging to the EFI namespace. This URIscheme is deprecated in MAE version 2.4. See [WAE23] Section 7.2.3 "Other URISchemes" and Section 7.10 "External Functionality Interface (EFI)" for more information.

6.1.5 External Functionality Interface (EFI)

This section is normative.

EFI is deprecated.

For EFI, see [WAE23] Section 7.10 "External Functional Interface (EFI)".

6.1.6 WTA

This section is normative.

The Wireless Telephony Application [WTA] is a deprecated application framework for telephony services. WTA extends the legacy WAE user agent by adding the capability to interface with mobile network services available to a mobile telephony device, e.g. setting up and receiving phone calls. Preferred URIscheme for accessing telephony services in MAE user agents is “tel:” in [URIScheme].

For normative conformance of WTA, see [WAE23] Section 7.14 “WTA”.

6.2 Browser User Agents and Transport Layer Integration

6.2.1 The MAE Protocol Framework

The MAE Protocol Framework is the HyperMedia Transfer Service as defined in [WAPArch].

6.2.2 Media Types Supported by MAE

This section is normative.

The following tables Show the list of MAE Media Types. A MAE User Agent must support the MIME Media Types defined in Table 3 for the supported device features. A MAE User Agent may support any other media types. A MAE User Agent should advertise a list of data types that it supports by sending MIME Media Types with the HTTP Accept header. A server may send back content with a MIME media type different from the one the User Agent advertised.

Note: File Extension is only a hint as to the application media type.

There are three MAE media types supported by MAE:

- The media types supported by the current MAE.
- The media types supported by the deprecated component by MAE.
- The media types supported by the other OMA enablers, not closely related to MAE.

Data Type	MIME Media Type	File Extension
XHTML Basic	application/xhtml+xml [RFC3236] text/html[1]	.xhtml, .xht
XHTML Mobile Profile	application/vnd.wap.xhtml+xml application/xhtml+xml; profile=”http://www.wapforum.org/xhtml” [RFC3236] text/html	.xhtml, .xht
Wireless CSS (formerly WAP CSS)	text/css	.css

[1] "text/html" MIME media type can be used for the XHTML Mobile Profile content or XHTML Basic content to be delivered to a PC browser.

ECMAScript Mobile Profile	text/ecmascript text/javascript application/ecmascript [RFC4329][2] application/javascript [RFC4329]	.js
WBXML	application/vnd.wap.wbxml	
WBMP	image/vnd.wap.wbmp	.wbmp
vCard	text/x-vCard	.vcf
vCalendar	text/x-vCalendar	.vcs
vBookmark	text/x-vBookmark	
Textual form of Multipart Messages that are used when the body parts are independent and need to be bundled in a particular order	multipart/mixed	
Textual form of Multipart Message representing objects that are aggregates of related MIME body parts	multipart/related	
Textual form of Multipart Message that is used when each of the body parts is alternative version of the same information	multipart/alternative	
Textual form of Multipart Message for returning values from form	multipart/form-data	
Binary form of Multipart Message for returning values from form	application/vnd.wap.multipart.form-data	
Service Indication in binary form	application/vnd.wap.sic	
Service Loading in textual format	text/vnd.wap.sl	
Service Loading in binary format	application/vnd.wap.slc	
Cache Operation in textual form	text/vnd.wap.co	
SVG	image/svg+xml	
SMIL	application/smil+xml	

[2] Note: A user agent that does not conform to [RFC4329] should advertise text/ecmascript. The inclusion of application/ecmascript is forward-looking to show an intent to align with [RFC4329] in the future. The future version may obsolete text/ecmascript and text/javascript. The user agent is not required to adhere to to the processing requirements in [RFC4329] in this version.

RME	video/richmedia+xml video/3gp application/richmediacommand+xml application/richhmedia+xml	 .3gp .scg (for sceneCommandGroup)

Table 3: MIME Media Types Supported by MAE

Data Type	MIME Media Type	File Extension
WML1 textual form	text/vnd.wap.wml	.wml
WML1 binary form	application/vnd.wap.wml	.wmlc
WML2	application/vnd.wap.wml+xml (The registration of application/wml+xml is on-going)	.wml
WMLScript textual form	text/vnd.wap.wmlscript	.wmls
WMLScript binary form	application/vnd.wap.wmlscriptc	.wmlsc
WBXML	application/vnd.wap.wbxml	
WBMP	image/vnd.wap.wbmp	.wbmp
WTA-WML textual form	text/x-wap-wta-wml	
WTA-WML binary from	application/x-wap-wta-wmlc	
Binary form of Multipart Messages that are used when the body parts are independent and need to be bundled in a particular order	application/vnd.wap.multipart.mixed	
Binary form of Multipart Message representing objects that are aggregates of related MIME body parts	application/vnd.wap.multipart.related	
Binary form of Multipart Message that is used when each of the body parts is alternative version of the same information	application/vnd.wap.multipart.alternative	
Binary form of Multipart Message for returning values from form	application/vnd.wap.multipart.form-data	
Channels in textual form	text/vnd.wap.channel	
Channels in binary form	application/vnd.wap.channelc	
Service Indication in binary form	application/vnd.wap.sic	
Service Loading in binary format	application/vnd.wap.slc	
Cache Operation in binary form	application/vnd.wap.coc	

Provisioning Document in binary form	application/vnd.wap.connectivity-wbxml	

Table 4: Legacy MIME Media Types Supported by MAE

Data Type	MIME Media Type	File Extension
Service Indication in textual form	text/vnd.wap.si	
Service Loading in textual format	text/vnd.wap.sl	
Cache Operation in textual form	text/vnd.wap.co	
Provisioning Document in textual form	text/vnd.wap.connectivity-xml	

Table 5: Other Application MIME Media Types Supported by MAE

Conformance: A User Agent MUST use the MIME media type as one component when determining the content data type of a document (MAESpec-MT-C-001).

Conformance: A User Agent MUST support the MAE MIME Media types defined in Table 3 for all supported features (MAESpec-MT-C-002).

For the WAP Proxy conformance regarding MIME types, see in [WAE23] 6.1 “Media Types Supported by WAE”.

6.2.3 Multipart Messages

This section is informative

MAE includes a multipart encoding specification, suitable for exchanging multiple typed entities over Hypermedia Transfer Service as well as Push.

The **`multipart/mixed`** messages can be used to transfer multiple related and/or unrelated resources at once. The **`multipart/related`** can be used to transfer related resources together, e.g., a document with its related images using hyperlinks. The **`multipart/alternative`** can be used to transfer the alternative resources of the same information, e.g., the same information in different languages.

For binary form of the MIME multipart entity by WSP, see [WAE23] Secion 6.9 “Multipart Messages”.

6.2.3.1 multipart/mixed

This section is normative.

Upon receipt of the **`multipart/mixed`** (or equivalent **`application/vnd.wap.multipart.mixed`**) content, the User Agent should decompose the message into its elements. The elements should be processed in the order in which they are in the multipart message. The processing for each element is the same as it would be when the element is received individually.

Conformance: A User Agent that supports HTTP SHOULD support **`multipart/mixed`**. (MAESpec-MUL-C-002)

For conformance for multipart/mixed in WSP, refer to [WAE23] Section 6.9.2 “multipart/mixed”.

6.2.3.2 multipart/related

This section is normative.

Upon receipt of the **`multipart/related`** (or equivalent **`application/vnd.wap.multipart.related`**) content, a User Agent should process the content according to [RFC2387] . The multipart is retained as a single compound entity. The root entity is processed as if only that entity was received. The processing of the root entity might imply that other entities in the multipart are processed.

Conformance: A User Agent that supports HTTP SHOULD support **`multipart/related`** (MAESpec-MUL-C-004).

For comformance of multipart/related in WSP, see [WAE23] Section 6.9.3 "multipart/related".

6.2.3.3 multipart/alternative

This section is normative.

Upon receipt of the **`multipart/alternative`** (or equivalent **`application/vnd.wap.multipart.alternative`**) content, a User Agent should choose the "best" content type based on the order of the entities within the multipart message and the local environment, e.g. device constraints and user preferences. The entities should be ordered from least preferable to most preferable representation.

Conformance: A User Agent that supports HTTP MAY support **`multipart/alternative`** (MAESpec-MUL-C-006).

For multipart/alternative conformance in WSP, see [WAE23] Section 6.9.4 "multipart/related".

6.2.3.4 multipart/form-data

This section is normative.

The **`multipart/form-data`**[RFC2388] returns a set of values from a form filled out by the user.

Conformance: A User Agent MUST support multipart messages returning a set of values from a form (MAESpec-MUL-C-007).

Conformance: A User Agent that supports HTTP MUST be able to submit form data in **`multipart/form-data`** (MAESpec-MUL-C-009).

For **`multipart/form-data`** conformance in WSP and WAP proxy, see [WAE23] Section 6.9.5 "multipart/form-data".

6.2.4 Hypermedia Transfer Service Interface

6.2.4.1 Transport Protocol

This section is normative.

MAE User Agents do not depend on any particular transport protocol, although MAE only defines a browser model of User Agent. User Agents are required to provide a hypermedia transfer service [WAPArch]. The HTTP (Hypertext Transfer Protocol) [HTTP11] provides the hypermedia transfer service over secure and non-secure connection-oriented transports. MAE User Agents must, at a minimum, implement HTTP. The network communication takes place in the form of HTTP 1.1 headers and content.

For WSP and WTP based hypermedia transfer service, see [WAE23] Section 7.1 "Transport Protocol".

For network communication with WSP header, see [WAE23] Section 7.1 "Transport Protocol".

For comformance for WAP proxy and WSP in Transport Protocol, [WAE23] see Section 7.1.1 "Transport Protocol".

Conformance: MAE User Agents MUST support the Hypermedia Transfer Service (MAESpec-HTS-C-001).

Conformance: A MAE User Agent MUST support HTTP[HTTP11] (MAESpec-HTS-C-003).

6.2.4.2 Cache Model

This section is normative.

MAE defines the caching model for User Agents. The User Agent caching model is an implementation of the HTTP/1.1 caching model used on the World Wide Web. The design is intended to allow the adoption of the HTTP/1.1 caching model with no loss of semantics or function while providing support for low-end WAP terminals.

The User Agent Caching Model Specification [CacheMod] addresses the following issues:

- User agent caching of resources fetched from network services. This may include XTHML Mobile Profile documents, or any other resources named with a URI and fetched over the hypermedia transfer service.

- Support infrastructure needed to implement HTTP/1.1 caching on a User Agent.

For WML, WMLScript and WAP proxy behavior, see [WAE23] Section 7.1.2 “Cache Model”.

Conformance: A User Agent MUST support caching model as specified in [CacheMod] (MAESpec-HTS-C-004).

6.2.4.3 HTTP State Management (Cookie)

This section is normative.

MAE provides the HTTP state management model, also known as “cookie management”, defined in [RFC2965]. Some User Agents may be capable of storing and managing cookies locally. This functionality follows precisely the current World Wide Web model, where cookies are typically stored and managed by regular web browsers. In addition, MAE defines an additional alternative mechanism to enable WAP proxies to store and manage cookies on behalf of the User Agent.

The HTTP State Management specification defines a minimum level of support for “cookies” in low functionality devices.

See [HTTPSM] for more information.

Conformance: A User Agent MUST support HTTP State Management as specified in [HTTPSM] (MAESpec-HTS-C-006).

6.2.4.4 Legacy Client Header Handling

This section is normative.

For client header handling in WSP layer and WAP proxies, see [WAE23] Section 7.1.4 “Client Header Handling”.

6.2.5 HTTP User Agent Behaviour

This section is normative.

Conformance: MAE user agents MUST follow the user agent behavior defined in HTTP11 when it uses HTTP[[HTTP11]] (MAESpec-HTS-C-007).

6.2.5.1 HTTP Redirection 3XX status codes

This section is normative.

HTTP [HTTP11] provides a redirection mechanism that allows further action to be taken by a User Agent when an origin server has returned a Redirection status code in response to a request from that User Agent for an identified resource.

Conformance: A User Agent MUST support the Redirection status codes and the associated actions to be taken by the User Agent to fulfil a request as defined by [HTTP11] (MAESpec-HTS-C-005).

Conformance: If a User Agent receives a 301 Moved Permanently status code in response to a POST request then that User Agent MUST NOT change the condition under which the request was issued, i.e. change POST request to a GET request[HTTP11]] (MAESpec-HTS-C-005).

Conformance: If a User Agent receives a 303 See Other status code in response to a request then that User Agent MUST initiate a GET request[HTTP11]] (MAESpec-HTS-C-005).

6.2.6 WAP Push

For user agent conformance related to WAP Push, see [WAE23] Section 7.5 “Push”.

6.2.6.1 Push-related Cache Operation

See [WAE23] Section 6.13 “Cache Operation” for Push-related Cache operation.

6.2.7 vObject Data Exchange

For WDP, see [WAE23] Section 7.8.1 “WDP Datagram Data Exchange”.

See [WAE23] Section 7.8.2 “WSP/HTTP Data Exchange”.

Transfer mechanisms for vObject are not defined for UDP, however, WDP is defined as UDP when the underlying bearer is IP. The conformance statements for this case, [WAE23] Section 7.8.1 “WDP Datagram Data Exchange”..

6.2.7.1 Calendar and Phone Book

This section is informative.

MAE includes support for the exchange of calendar and phone book data objects. There are currently two available methods for exchanging vCard and vCalendar data:

- Using WDP datagrams enables clients to transfer vCard and vCalendar data without the use of a WAP proxy or other network proxy.

- Issuing HTTP-based requests to a network server or receiving data via WAP Push enables clients and servers transfer vCard and vCalendar data.

For WSP request via WAP Push, see [WAE23] Section 6.8 “Internet Calendaring and Scheduling Core Object (vCalendar)”.

WAE User Agents may exchange calendar, scheduling and bookmark information in both client-server communication and peer-to-peer communication.

6.3 User Agent and Presentation Layer Integration

6.3.1 Character Set and Character Encoding

This section is normative.

The document character set for XHTML Mobile Profile, ESMP, WML and WMLScript is the Universal Character Set, defined jointly by the Unicode Consortium and ISO as ISO/IEC 10646 [ISO10646]. A conformant MAE User Agent must support character encodings UTF-8 and UTF-16 as required by XML [XML]. Additional support may be available in User Agents using compatible character sets and encodings from which the characters can be transcoded into the Universal Character Set (e.g., US-ASCII, ISO-8859-1).

Although listed as an optional parameter, the use of the charset parameter in the Content-Type field is strongly recommended when the content (e.g., XHTML Mobile Profile document, WML document or WMLScript) is conveyed via the hypermedia transfer service, unless otherwise this information is reasonably embedded in the content (e.g., by a charset field in WBXML). A User Agent uses this information to determine the character encoding of the content. The charset parameter can also be used to provide protocol-specific operations, such as charset-based content negotiation in the hypermedia transfer service.

A User Agent should inform the origin server, which language and character encoding it supports using Accept-Language and Accept-Charset HTTP header.

Conformance: User Agents MUST support both UTF-8 and UTF-16; i.e., they MUST be able to process text encoded in UTF-8 and UTF-16 (MAESpec-I18N-C-001, MAESpec-I18N-C-002).

Conformance: When processing XML documents, it is REQUIRED to treat character encoding of the document as specified in [RFC3023]. User Agents MUST inform the user of the error if the XML document includes unknown characters (MAESpec-I18N-C-003).

For conformance for WML, WSP, and WAP proxies, see [WAE23] Section 7.6.1 “Character Set and Character Encoding”.

6.4 User Agent Integration for Application Layer Server-Client Negotiation

6.4.1 Advertising of User Agent Characteristics

This section is informative.

In order to optimise the MAE client-server model, a number of characteristics are sent from the user agent to the origin server. These characteristics allow the origin server to avoid sending inappropriate content to the user agent. They also provide the server with a means of customising the response for a particular user agent.

The general mechanisms designed to provide this functionality are described in detail in [UAPROF]. A MAE User Agent that wants to convey Accept header information must do so through standard hypermedia transfer service (i.e., HTTP or WSP) headers such as Accept, Accept-Charset and Accept-Language. To avoid ambiguities between advertised characteristics information contained in these Accept headers must be consistent with the information provided by UAProf.

The general mechanisms provided by UAProf and the use of Accept headers for conveying information provide an accurate means of identifying the supported capabilities of the MAE User Agent.

The utilization of Accept headers and mechanisms provided by UAProf ensures interoperability between a MAE User Agent and origin server, especially for those legacy server implementations that do not provide support for x-wap-profile and x-wap-profile-diff headers [UAPROF].

For legacy implementations the use of Accept header information is required. For improved service and network performance, especially Over-The-Air (OTA), efforts continue in avoiding verbose Accept header information from being advertised by MAE User Agent.

For WAP proxy and WSP, see [WAE23] Section 7.7 “Advertisement of User Agent Characteristics”.

6.4.1.1 HTTP Accept headers

This section is normative.

HTTP includes the following request-header fields for describing user agent capabilities and user preferences to the server:

- *Accept* – The Accept request-header field can be used to specify certain media types which are acceptable for the response (e.g., text/vnd.wap.wml).
- *Accept-Charset* – The Accept-Charset request-header field can be used to indicate what character sets are acceptable for the response (e.g., iso-8859-1).
- *Accept-Encoding* – The Accept-Encoding request-header field restricts the content-codings that are acceptable in the response (e.g., compress, gzip).
- *Accept-Language* – The Accept-Language request-header field restricts the set of natural languages that are preferred as a response to the request (e.g., en, dk).

However, an origin server is not limited to these accept-headers and may vary the response based on any aspect of the request, including information outside the request –header fields.

A MAE User Agent using the legitimate HTTP media-range */* [RFC2616] to declare its supported content types to avoid verbose message size (i.e. declaration for support of all media types) may lead to a MAE User Agent being served with incompatible content. In this situation an origin server or proxy should use information provided by UAProf to determine User Agents support capabilities.

See Sections 14.1 and 10.4.7 of [HTTP11] for guidance.

Conformance: A MAE User Agent that wants to advertise its characteristics MUST at least use HTTP Accept, Accept-Charset, Accept-Encoding and Accept-Language request headers regardless of the UAProf support (MAESpec-UAC-C-001, MAESpec-UAC-C-002, MAESpec-UAC-C-003, MAESpec-UAC-C-004).

Conformance: A MAE User Agent SHOULD declare a HTTP media-range of */* [HTTP11] with a “q” value less than 1 and key media types to indicate a mobile device (MAESpec-UAC-C-005).

Conformance: Capabilities and Preference Information (CPI) referenced by the URI contained in a MAE User Agent’s Profile header [UAPROF] MUST be a strict superset of the characteristics advertised by that MAE User Agent (MAESpec-UAC-C-006).

For WAP Proxy conformance, see [WAE23] Section 7.7.1 “HTTP/WSP Accept-Header”.

6.4.2 UAProf

This section is normative.

UAProf [UAPROF] captures classes of device information for content formatting by origin servers, WAP proxies and other interim servers. This device information is referred to as Capability and Preference Information (CPI). It includes information about:

- Hardware Platform – screen size, colour capabilities, image capabilities, manufacturer, etc.
- Software Platform – operating system vendor and version, list of audio and video encoders, etc.
- Browser UA – browser manufacturer and version, mark-up language and versions supported, scripting languages supported, etc.
- Network Characteristics – bearer characteristics such as latency and reliability, etc.

For supported WapCharacteristics, see [WAE23] Section 7.7.2 “UAProf”.

The user agent profile is defined to comply with the W3C standards for Composite Capabilities/Preferences Profile (CC/PP) [CC/PP, CCPPex] distribution over the Internet. The user agent profile uses an RDF schema and vocabulary [RDF], as defined in the CC/PP model, to define a robust, extensible framework for describing and transmitting CPI.

The CPI is transmitted and maintained using designated hypermedia transfer service (i.e., HTTP or WSP) headers.

The hypermedia transfer service uses Profile and Profile-Diff headers to convey the CPI. A Profile header contains URI(s), where each URI is referencing an externally accessible CPI document. The Profile-Diff header contains a CPI document. Multiple Profile and Profile-Diff headers may be cached by a WAP proxy and/or included with a request.

For WSP and WAP proxy behaviors, see [WAE23] Section 7.7.2 “UAProf “.

It is expected that a class of proxy services will be deployed to support the conversion of content based on the information available in the UAProf. Those adapting proxies will extend the request profile (e.g., by adding a new Profile-Diff) to advertise their capabilities. If WAP proxies or other interim servers wish to add to or modify the Accept* header information, they must also add an equivalent Profile-diff segment in order to reflect the same information in the CPI.

See [UAPROF] for more information.

Conformance: A MAE User Agent SHALL support the Profile (*x-wap-profile*) header as defined in [UAPROF] (MAESpec-UAC-C-007).

NOTE: As stated in [UAPROF] the URI referencing the CPI is expected to resolve to a specific URL.

6.5 Browsing Security and Access Control

6.5.1.1 Basic Authentication Scheme

This section is normative.

HTTP provides a simple challenge-response authentication mechanism. HTTP may be used by a server to challenge a client request and used by a client to provide authentication information. The “basic” authentication scheme is based on the model that the client must authenticate itself with a user-ID and a password. This basic authentication scheme is specified in [HTTP11] and [HTTPBasicAuth].

This Basic authentication scheme is not a secure method of user authentication, nor does it in any way protect the cleartext entity transmitted across the physical network.

Conformance: User Agents MUST implement HTTP/1.1 Basic Authentication as specified in [HTTP11] and [HTTPBasicAuth] (MAESpec-SEC-C-001).

6.5.1.2 Access Control Pragma in WML and WMLScript

For access control pragma in WML and WMLScript, see [WAE23] Section 7.12.2 “Access Control Pragma in WML and WMLScript”.

6.5.1.3 Persistent Storage in WML and WMLScript

For Persistent Storage in WML and WMLScript, see section [WAE23] Section 7.12.3 “Persistent Storage”.

6.5.1.4 WMLScript Crypto Library

For WMLScript Crypto Libarary, see [WAE23] Section 7.12.4 “WMLScript Crypto Library”.

6.5.1.5 ECMAScript Mobile Profile Crypto Library

This section is normative.

[ESMPCrypto] specifies the application layer security object interface for ECMAScript Mobile Profile [ESMP11] that provides cryptographic functionality to User Agents and complements the use of WMLScript Crypto Library. [ESMPCrypto] also specifies a signed content format to be used to convey signed data to/from User Agents.

Conformance: A MAE user agent MAY support ECMAScript Mobile Profile Crypto Library [ESMPCrypto] (MAESpec-SEC-C-003).

6.5.1.6 Secure Transport

This section is informative.

At the Transport Service layer protocols are defined for secure transport over datagrams and connections. WTLS[WTLS] is defined for secure transport over datagrams. WTLS and TLS[TLS] are defined for secure transport over connections.

TLS is the preferred method, when the User Agent directly communicates with the origin server over TCP.

For secure transport with WAP proxy, see [WAE23] Section 7.12.6 “Secure Transport”.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-MAESpec-V2_4-20110329-A	29 Mar 2011	Status changed to Approved by TP: OMA-TP-2011-0097-INP_Browsing_V2_4_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 MAE User Agent

B.1.1 MAE Media Types

Item	Function	Reference	Status	Requirement
MAESpec-MT-C-001	User agent use of MIME Media type	6.2.2	M	
MAESpec-MT-C-002	Support for MIME Media Types	6.2.2	M	

Item	Function	Reference	Status	Requirement
MAESpec-ML-C-003	Support for textual form of WML2	[WAE23] 6.2	O	WML2:MCF AND MAESpec-ML-C-004
MAESpec-ML-C-004	Maintains WML context	[WAE23] 6.2	O	
MAESpec-ML-C-005	Support for textual form of XHTMLMP	[WAE23] 6.2	M	XHTMLMP:MCF
MAESpec-ML-C-006	Support for the textual form of WML1	[WAE23] 6.2	O	WML1:MCF AND MAESpec-ML-C-004 AND MAESpec-UAB-C-004 AND MAESpec-UAB-C-005
MAESpec-ML-C-007	Support for the binary form of WML1	[WAE23] 6.2	O	WML1:MCF AND WBXML:MCF AND MAESpec-ML-C-004 AND MAESpec-UAB-C-004 AND MAESpec-UAB-C-005

Item	Function	Reference	Status	Requirement
MAESpec-STY-C-002	Support for Wireless CSS	5.2	M	WCSS:MCF

Item	Function	Reference	Status	Requirement
MAESpec-WMLS-C-001	Support for WMLScript execution	[WAE23] 6.4	O	WMLScript:MCF AND MAESpec-WMLS-C-002
MAESpec-WMLS-C-002	Support for WMLScript Standard Library	[WAE23] 6.4	O	WMLScriptLibs:MCF

Item	Function	Reference	Status	Requirement
MAESpec-ESMP-C-001	Support for ECMAScript Mobile Profile	[WAE23] 6.4.4	M	ESMP:MCF AND XHTMLMP-SCRIPT-C-001

Item	Function	Reference	Status	Requirement
MAESpec-IMG-C-001	Support for graphical images	[WAE23] 6.6	O	AESpec-IMG-C-002
MAESpec-IMG-C-002	Support for WBMP	[WAE23] 6.6	O	WAEMT:MCF

Item	Function	Reference	Status	Requirement
WAESpec-VCARD-C-002	Support of text/x-vCard MIME Type	5.5.1	O	
WAESpec-VCARD-C-003	vCard data port	[WAE23] 6.7	O	WDP:MCF
MAESpec-VCARD-C-006	Support for vCard in vObject	5.5.1	O	

Item	Function	Reference	Status	Requirement
WAESpec-VCAL-C-002	Support of text/x-vCalendar MIME Type	5.5.1	O	
WAESpec-VCAL-C-003	vCalendar data port	[WAE23] 6.8	O	WDP:MCF
MAESpec-VCAL-C-005	Support for vCalendar in vObject	5.5.1	O	

Item	Function	Reference	Status	Requirement
MAESpec-VBOOK-C-001	Support for vBookmark in vObject	5.5.1	O	

Item	Function	Reference	Status	Requirement
MAESpec-MUL-C-001	application/vnd.wap.multipart. mixed	6.2.3.1	O	
MAESpec-MUL-C-002	multipart/mixed	6.2.3.1	O	
MAESpec-MUL-C-003	application/vnd.wap.multipart. related	6.2.3.2	O	
MAESpec-MUL-C-004	multipart/related	6.2.3.2	O	
MAESpec-MUL-C-005	application/vnd.wap.multipart. alternative	6.2.3.3	O	
MAESpec-MUL-C-006	multipart/alternative	6.2.3.3	O	
MAESpec-MUL-C-007	Support for the multipart message returning a set of values from a form	6.2.3.4	M	MAESpec-MUL-C-008 OR MAESpec-MUL-C-009
MAESpec-MUL-C-008	application/vnd.wap.multipart. form-data	6.2.3.4	O	

MAESpec-MUL-C-009	multipart/form-data	6.2.3.4	O	

B.1.2 MAE Features

Item	Function	Reference	Status	Requirement
MAESpec-HTS-C-001	Support for Hypermedia Transfer Service	6.2.4.1	M	
MAESpec-HTS-C-002	Support for WSP	[WAE23] 7.1.1	O	WSP:MCF AND MAESpec-MUL-C-008
MAESpec-HTS-C-003	Support for HTTP 1.1	6.2.4.1	M	HTTP11:MCF AND MAESpec-MUL-C-009
MAESpec-HTS-C-004	Support for Caching Model	6.2.4.2	M	CacheMod:MCF
MAESpec-HTS-C-005	HTTP Redirection 3xx status codes	6.2.5	M	
MAESpec-HTS-C-006	Support for HTTP State Management	6.2.4.3	M	HTTPSM:MCF
MAESpec-HTS-C-007	Support for HTTP User Agent Behaviour	6.2.5	M	HTTP11:MCF

Item	Function	Reference	Status	Requirement
MAESpec-URI-C-001	Minimum URI length	6.1.4	M	
MAESpec-URI-C-007	Support for URIScheme	6.1.4	M	

Item	Function	Reference	Status	Requirement
MAESpec-PUSH-C-001	Support for Push	[WAE23] 7.5.2	O	MAESpec-PUSH-C-009 AND MAESpec-PUSH-C-010 AND MAESpec-PUSH-C-011 AND (MAESpec-PUSH-C-002 OR MAESpec-PUSH-C-003) AND (MAESpec-PUSH-C-004 OR MAESpec-PUSH-C-005) AND (MAESpec-PUSH-C-006 OR MAESpec-PUSH-C-007)
MAESpec-PUSH-C-002	application/vnd.wap.multipart.mixed	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-003	multipart/mixed	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-004	application/vnd.wap.multipart.related	[WAE23] 7.5.2	O	

MAESpec-PUSH-C-005	multipart/related	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-006	application/vnd.wap.multipart. alternative	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-007	multipart/alternative	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-008	undefined push behaviour	[WAE23] 7.5.2	O	
MAESpec-PUSH-C-009	Support for Push Message	[WAE23] 7.5.2	O	PushMessage:MCF
MAESpec-PUSH-C-010	Support for Push OTA	[WAE23] 7.5.2	O	PushOTA:MCF
MAESpec-PUSH-C-011	Support for Service Indication	[WAE23] 7.5.2	O	ServiceInd:MCF

Item	Function	Reference	Status	Requirement
MAESpec-I18N-C-001	Support for UTF-8	6.3	M	
MAESpec-I18N-C-002	Support for UTF-16	6.3	M	
MAESpec-I18N-C-003	Treat the character encoding of an XML document as defined in [RFC3023].	6.3	M	

Item	Function	Reference	Status	Requirement
MAESpec-PICT-C-001	Support for Pictograms	5.4.1	O	WAPInterPic:MCF

Item	Function	Reference	Status	Requirement
MAESpec-UAC-C-001	Informs supported media type using Accept header	6.4.1.1	O	
MAESpec-UAC-C-002	Informs supported character encoding using Accept-Charset header	6.4.1.1	O	
MAESpec-UAC-C-003	Informs supported content-codings using Accept-Encoding header	6.4.1.1	O	
MAESpec-UAC-C-004	Informs supported language using Accept-Language header	6.4.1.1	O	
MAESpec-UAC-C-005	Declaration of HTTP media-range of */* with a “q” value less than 1 and key media types	6.4.1.1	O	
MAESpec-UAC-C-006	UAProf being a strict superset of explicitly listed Accept header information	6.4.1.1	M	
MAESpec-UAC-C-007	MAE UA support of UAProf	6.4.2	M	

Item	Function	Reference	Status	Requirement
MAESpec-EFI-C-001	Support for EFI	6.1.5	O	EFI:MCF

Item	Function	Reference	Status	Requirement
MAESpec-PSTOR-C-001	Support for Persistent Storage	[WAE23] 7.12.3	O	
MAESpec-PSTOR-C-002	Managing access to stored objects	[WAE23] 7.12.3	O	

Item	Function	Reference	Status	Requirement
MAESpec-SEC-C-001	Support for HTTP/1.1 Basic Authentication	6.5.1.1	M	
MAESpec-SEC-C-002	Support for WMLScript Crypto Library	6.5.1.3	O	Crypto:MCF AND MAESpec-WMLS-C-001
MAESpec-SEC-C-003	Support for ESMP Crypto	6.5.1.5	O	MAESpec-ESMP-C-001 AND ESMPCR:MCF

B.1.3 User Agent Behaviour

Item	Function	Reference	Status	Requirement
MAESpec-UAB-C-001	Navigation History	6.1.1	M	
MAESpec-UAB-C-002	Access to Back key at all times	6.1.2	M	
MAESpec-UAB-C-003	User Agent performs pop operation on BACK	6.1.2	M	
MAESpec-UAB-C-004	WML1 User Agent executes `prev` task on BACK	[WAE23] 7.13.2	O	
MAESpec-UAB-C-005	WML1 `do type=prev` behaviour	[WAE23] 7.13.2	O	
MAESpec-UAB-C-006	Byte size of textbox	6.1.3	M	

Item	Function	Reference	Status	Requirement
MAESpec-WTA-C-001	Support for WTAI Public URI functions	[WAE23] 7.14.1	O	WTAI-PU-C-001 AND WTAI-PU-C-002 AND WTAI-PU-C-003
MAESpec-WTA-C-002	Support for WTAI Public WMLScript functions	[WAE23] 7.14.1	O	WTAI-PS-C-001 AND WTAI-PS-C-002 AND WTAI-PS-C-003 AND WMLScript:MCF

B.2 WAP proxy and Server

B.2.1 MAE Media Types

Item	Function	Reference	Status	Requirement

MAESpec-MT-S-001	Support for appropriate MAE MIME Media types	[WAE23] 6.1	M	
MAESpec-MT-S-002	Proxy transparency	[WAE23] 6.1	M	

Item	Function	Reference	Status	Requirement
MAESpec-WMLS-S-001	Ability to transform WMLScript into bytecode	[WAE23] 6.4	M	MAESpec-SEC-S-001 AND MAESpec-EFI-S-001
MAESpec-WMLS-S-002	Informs WAE User Agents of compilation errors using the HTTP status code 502	[WAE23] 6.4	O	

Item	Function	Reference	Status	Requirement
MAESpec-ESMP-S-001	Support for ECMAScript Mobile Profile	[WAE23] 6.4.4	M	

Item	Function	Reference	Status	Requirement
MAESpec-MUL-S-001	convert application/vnd.wap.multipart.form-data into multipart/form-data	[WAE23] 6.9.5	O	
MAESpec-MUL-S-002	convert application/vnd.wap.multipart.mixed into multipart/mixed	[WAE23] 6.9.2	O	
MAESpec-MUL-S-003	convert application/vnd.wap.multipart.related into multipart/related	[WAE23] 6.9.3	O	
MAESpec-MUL-S-004	convert application/vnd.wap.multipart.alternative into multipart/alternative	[WAE23] 6.9.4	O	

B.2.2 MAE Features

Item	Function	Reference	Status	Requirement
MAESpec-HTS-S-001	Support for Hypermedia Transfer Service	[WAE23] 7.1.1	M	MAESpec-HTS-S-002 OR MAESpec-HTS-S-003
MAESpec-HTS-S-002	Support for WSP	[WAE23] 7.1.1	O	WSP:MSF AND MAESpec-HTS-S-004ANDMAESPEC-MUL-S-002
MAESpec-HTS-S-003	Support for HTTP	[WAE23] 7.1.1	O	HTTP11:MSF

MAESpec-HTS-S-004	Client header handling in the connect service primitive	[WAE23] 7.1.3	O	
MAESpec-HTS-S-005	Client header handling in the resume service primitive	[WAE23] 7.1.3	O	

Item	Function	Reference	Status	Requirement
MAESpec-URI-S-001	Minimum URI length	[WAE23] 7.2	M	
MAESpec-URI-S-002	HTTP URI Scheme	[WAE23] 7.2.1	M	
MAESpec-URI-S-003	HTTPS URI Scheme	[WAE23] 7.2.2	M	MAESpec-URI-S-004 OR MAESpec-URI-S-005 OR MAESpec-URI-S-006
MAESpec-URI-S-004	HTTPS URI Scheme over HTTP	[WAE23] 7.2.2	O	TLS:MSF
MAESpec-URI-S-005	HTTPS URI Scheme over WSP	[WAE23] 7.2.2	O	WTLS:MSF
MAESpec-URI-S-006	Report an error when no TLS or WTLS support is available	[WAE23] 7.2.2	M	

Item	Function	Reference	Status	Requirement
MAESpec-MMS-S-001	Support for MMS	[WAE23] 7.4	O	MMSCTR:MSF

Item	Function	Reference	Status	Requirement
MAESpec-I18N-S-001	Ability to transform character encoding	[WAE23] 7.6.1	O	
MAESpec-I18N-S-002	Support for UTF-8	[WAE23] 7.6.1	M	
MAESpec-I18N-S-003	Support for UTF-16	[WAE23] 7.6.1	M	
MAESpec-I18N-S-004	Treat the character encoding of an XML document	[WAE23] 7.6.1	M	

Item	Function	Reference	Status	Requirement
MAESpec-UAC-S-001	Support for Accept header	[WAE23] 7.7.1	M	
MAESpec-UAC-S-002	Support for Accept-Charset header	[WAE23] 7.7.1	M	
MAESpec-UAC-S-003	Support for Accept-Encoding header	[WAE23] 7.7.1	M	
MAESpec-UAC-S-004	Support for Accept-Language header	[WAE23] 7.7.1	M	

MAESpec-UAC-S-005	Support for the behaviour defined in HTTP/1.1	[WAE23] 7.7.1	O	
MAESpec-UAC-S-006	Honouring the User Agent’s preferences	[WAE23] 7.7.1	O	
MAESpec-UAC-S-007	Avoidance of sending content that the User Agent does not accept	[WAE23] 7.7.1	O	
MAESpec-UAC-S-008	Avoidance of unnecessary transcoding	[WAE23] 7.7.1	M	

Item	Function	Reference	Status	Requirement
MAESpec-SEC-S-001	Ability to compile a script compilation unit that refers to the Crypto Library	[WAE23] 7.13.3	M	WMLScriptCrypto:MSF

Item	Function	Reference	Status	Requirement
MAESpec-EFI-S-001	Ability to compile a script compilation unit that refers to the EFI Library	[WAE23] 7.10	M	EFI:MSF