OMTP

Open Mobile Terminal Platform

TECHNOLOGY AGNOSTIC CORE SOFTWARE PLATFORM

APPLICATION FRAMEWORK CONCEPT PAPER

VERSION:	OMTP APPLICATION FRAMEWORK CONCEPT PAPER 1_0, RELEASE 1
STATUS:	APPROVED BY BOARD 6 JULY 2005
DATE OF LAST EDIT:	22 JUNE 2005
OWNER:	P3: TECHNOLOGY AGNOSTIC CORE SOFTWARE PLATFORM

CONTENTS

The information contained in this document represents the current view held by OMTP Ltd. on the issues discussed as of the date of publication.

This document is provided “as is” with no warranties whatsoever including any warranty of merchantability, noninfringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein.

This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based on this document.

"OMTP" is a registered trademark. Other product or company names mentioned herein may be the trademarks of their respective owners.

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

This document aims to provide a high-level description of the concept of the OMTP Application Framework.

This description includes:

- A definition of what an OMTP application framework is.
- A description of the modules that comprise such a framework and the functionality (in general terms) provided by each module.
- The most important standards and standardisation organisations relevant to the application framework.

1.2 INTENDED AUDIENCE

This document is intended for external distribution.

2 APPLICATION FRAMEWORK DESCRIPTION

This chapter describes the OMTP application framework. It provides a definition of its functional elements and a description of the concepts used to define it.

2.1 MOTIVATION

The Application Framework is the OMTP approach to ensuring the fulfilment of the functional requirements defined within OMTP Functional Specifications. These functional requirements are translated into technical requirements in terms of the functionalities and capabilities of the Application Framework.

The technical requirements and the Application Framework specification are applied in a technology agnostic way in order to provide an abstracted interface between the client presentation layer and the device’s hardware layer.

The Application Framework implementation provides a homogeneous set of functions to the applications. This eases porting of client applications across different OMTP terminals, reducing costs and time to market while still providing an environment for service innovation and differentiation via application development. It also aids customisation of OMTP compliant terminals.

2.1.1 SCOPE

The Application Framework consists of a set of layered and structured modules. Each module has well defined technology agnostic high-level descriptions of its functionalities. Achieving interface compatibility or even binary compatibility between different products is not a target of the OMTP Application Framework. Such compatibilities would cause conflicts with the technology agnostic goal by relying on a compatibility technology.

All the functions offered by Application Framework follow defined security rules that restrict access to resources, personal data and other sensitive content or functionality. Although these security rules are not covered within this document, they are considered part of the Application Framework. Details of the security related services and mechanisms are handled in a separate OMTP document.

The core relationships between the different application framework modules are consistent across OMTP. This ensures a consistent functionality and user experience across different implementations of the Application Framework in OMTP compliant terminals.

2.2 HIGH LEVEL ARCHITECTURE

This section describes conceptually the architecture of the OMTP Application Framework. Figure 1 identifies and positions the different elements that comprise the framework concept.

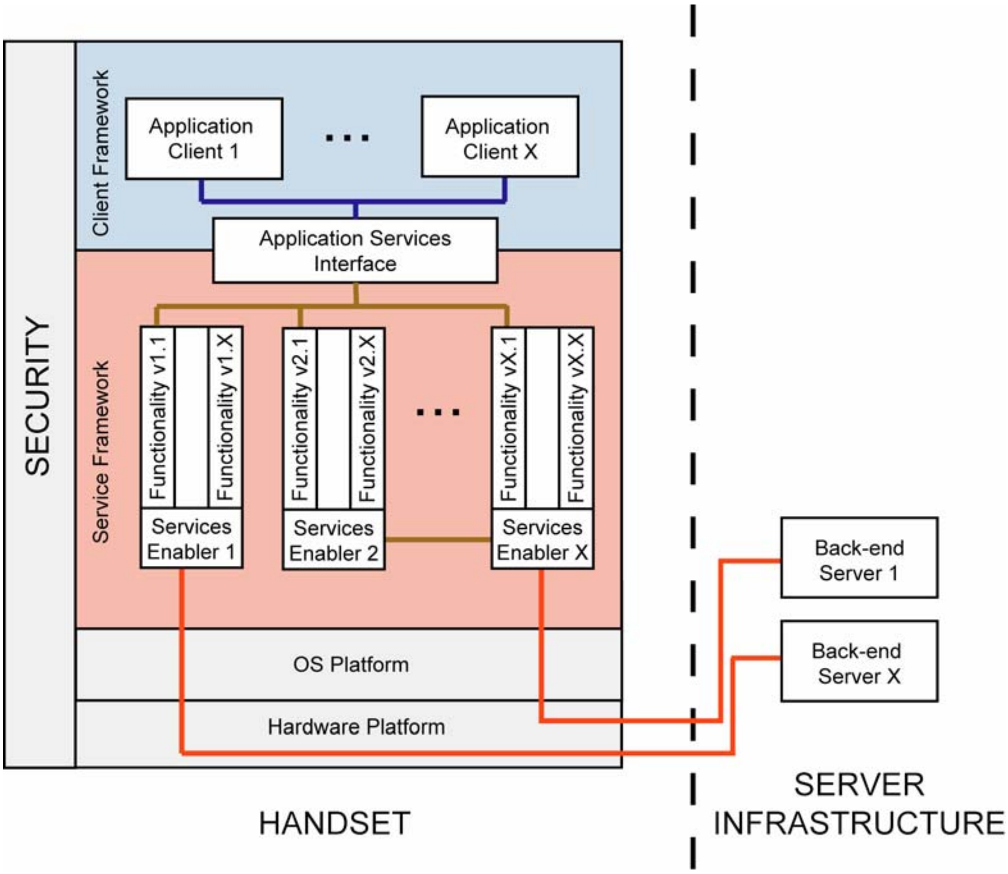

Figure 1: OMTP Application Framework high-level overview.

An OMTP Application Framework comprises a set of service enablers (also called enablers) and application clients that run on top of the OS Platform. This framework can be split into Client and Service Frameworks.

The client framework comprises all the application clients that offer access to the different terminals' functionalities. The Service Framework comprises all the enablers that are responsible for providing these functionalities. These enablers' functionalities can be also used by other enablers within the Service Framework.

Access to these functionalities is via an Application Services Interface (ASI). This interface can be exposed via different means (e.g. APIs). This will allow the development of application clients that can access the enablers' functionalities. The ASI is effectively a superset of the interfaces used to allow Client applications access to the enablers running in the Service Framework.

Enablers in the Service Framework typically depend on the functionality offered by remote back-end servers. Therefore, enablers' requirements will be defined in order to guarantee the interoperability with those remote servers. However, the definition of requirements or details of how these services are implemented in back-end servers are beyond the scope of the OMTP Application Framework specification.

The Application Framework will adopt a set of security rules that will be applied to the functions offered, but the definition of these security rules will be part of a separate specification.

Each module is described in detail in the next sections.

2.3 APPLICATION FRAMEWORK DEFINITION

The Application Framework defines the complete environment for applications and relationships across applications. This includes:

- The relationships between enablers.
- The relationships between clients.
- The relationships between enablers and clients.

This includes the functions offered by the enablers and the identification of Service Enablers that MUST be supported by an OMTP compliant implementation of the Application Framework.

However, it is not the intention to precisely define APIs, so the Application Framework will not be a physical Application Framework. It will be a conceptual framework defined by functional requirements that are mapped to different modules and interfaces.

This Framework can be divided in three main elements:

- Client Framework.
- Service Framework.
- Application Services Interface.

2.3.1 CLIENT FRAMEWORK

The Client Framework includes a collection of client applications and infrastructure modules that support those client applications. Client applications are usually focused around interaction with the user. They get functional services from the Service Framework by using the Applications Services Interface e.g. the SMS client application enables the user to compose new messages, view existing messages, and issue commands to send SMS.

The UI flow of the system often requires cross-activation between client applications in order to create a seamless user experience for the end user e.g. an email client application may be integrated with the PhoneBook client application in a way that allows the end user to compose the distribution list for a new email by cross-activating part of the PhoneBook client application.

The Client Framework Infrastructure software that provides the environment for client applications to execute and communicate is beyond the scope of OMTP specifications. The primary objectives of the Client Framework are:

- To identify client applications of interest to OMTP and create a common reference for their scope.
- To define the minimum relationships between client applications e.g. the email client application should be able to cross-activate the WAP Browser client application when the user selects a URL in the email's body.

2.3.2 SERVICE FRAMEWORK

The Service Framework includes a collection of enablers and infrastructure modules that support these enablers. Enablers are usually focused around high-level functionalities, also known as application servers or application engines. Enablers do not control or contain user interface elements.

The Service Framework infrastructure modules that provide the environment for enablers are beyond of the scope of OMTP specifications.

The primary objectives of the Service Framework are:

- To distinguish between enablers and create a common reference for their scope.
- To define the functions that each enabler must provide in an abstract way in order to allow different implementations. These functions, while technology agnostic, must be detailed enough to describe clearly the functionality expected from each enabler.

2.3.3 APPLICATION SERVICES INTERFACE

The Application Services Interface offers mechanisms to access to a set of the functions provided by service enablers. By means of this interface, applications using the enablers' functions can be developed. Requirements for this interface must be defined in a technology agnostic manner, allowing different interface implementations based on different technologies.

2.4 MAIN ENABLERS AND RELATIONSHIPS

2.4.1 ENABLERS

This section provides an overview of the different enablers that make up the Service Framework. This enabler list is not exhaustive: it includes only the main enablers comprising an OMTP Application Framework. The mandatory enablers and their requirements will be defined in the Application Framework Specification.

For each enabler, a description, a short explanation of the functionality offered, the services exposed and the relevant standards are provided. The standards listed are only intended as examples. Requirements and standards used will be defined in the Application Framework Specification.

Compliancy validation criteria are also included for those enablers where that information is relevant. In general terms, the vendor of each enabler component will create suitable documentation describing to what degree the component fulfils the requirements specified for that enabler. That documentation may include test results achieved using a vendor-specific validation tool and shall be described in detail in the enabler specification.

2.4.1.1 Customisation

Description

The customisation enabler will define the functions needed for the customisation of the 'look&feel' of mobile terminals.

Functionality

The customisation enabler definition will include:

- A common file format to be used for expressing the customisation element contents (e.g. bitmaps, colour maps, UI texts) and policies (e.g. the operator possibly wishing to prevent the user personalising an element).
- All the possible customisation scenarios: message sequence charts and message contents.
- The mechanisms for customisation policies for each element e.g. who can change it, is the user asked for permission, is the user notified about the customisation, applicability to the different SIM/subscription types (prepaid, post-paid, no GPRS plan, etc).

References

The functionality of the customisation enabler in the OMTP Release 1 will outline the solution to the functional requirements defined in

"User Experience Customisation Functional Requirements: Look & Feel, Menu Customisation and Application Integration" [12].

Services exposed to the application clients and to the other enablers

This enabler must offer an interface that will provide services to configure the different aspects of the terminal's 'look&feel' (e.g. modify the background or the colour scheme). Some of these services may be available for applications (e.g. a camera application could be able to modify the background image) either to retrieve the customisation configuration of the terminal or to modify it.

Related standards

Existing standards shall be used wherever applicable. These include (but are not limited to) the following:

- OMA Device Management (including OMA Client Provisioning) is a candidate for performing customisation operations over-the-air. If OMA Device Management is used, the new management objects should be specified together with OMA Device Management Working Group. The nature and process of cooperation between OMA and OMTP needs to be clarified.
- Extensible Markup Language (XML) is a simple, very flexible text format that is playing an increasingly important role in the exchange of a wide variety of data. This format could be a candidate for expressing the customisation elements. XML is a World Wide Web Consortium (W3C) recommendation for media-independent communications across systems allowing the definition of platform-independent protocols for data exchange.

2.4.1.2 Device Management

Description:

Device Management refers to the process of managing device settings and applications resident in the terminal. Device Management facilitates the configuration of the user's terminal with minimal user intervention to enable easy usage of mobile services. As terminals offer more and more services, Device Management provides a mechanism for the operator to remotely configure a terminal's parameters while the terminal is in the hands of a user. The OMA Device Management Working Group is currently specifying standards for Device Management for mobile terminals. OMTP will utilise these specifications to fulfil its user requirements.

Functionality:

OMA Device Management includes the following:

- Setting the initial configuration information in the terminals.
- Installation and updating of persistent stored information in the terminals.
- Processing events and alarms generated by the terminals.
- Retrieval of management information from the terminals.
- Capabilities to remotely update terminal firmware.
- Capabilities to remotely manage terminal applications (install, uninstall, update…).

References

Device Management should address all the requirements defined in "Functional Requirements for Remote Service Provisioning" [13]. Remote Service Provisioning provides initial service provisioning configuration and later updates through standard over-the-air (OTA) mechanisms. Addressing these requirements will mean specifying the parameters that will be configured OTA and the Device Management Device Definition Framework (DDF) for these elements. MMS, WAP, Web Browser and email are examples of the services whose configuration parameters will be specified. This will enable the operators to configure these services in a consistent manner across the range of terminals.

Compliancy validation

Existing testing procedures and arrangements in the area of Device Management shall be used wherever possible.

2.4.1.3 Browsing

Description:

This enabler provides access to contents through the browsing capabilities defined by OMTP.

Functionality:

This module provides a transport framework for the relevant protocols required to provide browsing services (HTTP, WAP Stack etc.). It will also provide the means for handling and parsing the browsing content, including markup languages (XHTML, WML etc.) and style capabilities (CSS etc.).

Services exposed to the application clients and to the other enablers

The services offered by this enabler must be targeted to provide the capabilities required by browser clients (WAP browser, Internet browser, etc.). Additional services will be required by other enablers and some clients will require the parsing capabilities and protocol stacks (e.g. Messaging and WAP push).

Related standards

- Transport Protocols, OMA Browser Protocol Stack [16]: WAP Stack (WSP, WTP, WTLS, WDP etc.), http etc.
- OMA Browsing Enabler [1]: This specification defines the application level protocols, semantics, syntax, content formats, UA behaviour and the use of hypermedia transfer protocols to achieve consistent and interoperable browsing.

Compliancy validation

Compliance criteria must verify that the requirements defined in the enabler's specification are fulfilled. The test criteria will demonstrate that the services offered by the enabler provide the ability to build new applications with browsing capabilities.

2.4.1.4 Shortlink Communications

Description

This enabler provides a framework for using point-to-point communications with other terminals and peripherals.

Functionality:

This enabler abstracts the functionalities offered by the protocols used in short-link communications (Bluetooth, IrDA etc.) in order to:

- Discover terminals and peripherals.
- Exchange information with remote terminals.

2.4.1.5 Downloading

Description

This enabler provides a general download framework.

Functionality

The main functionality that this enabler offers is the ability to download content from a remote server to the terminal's memory.

The functionality related to the basic transport protocols (WAP Stack, HTTP etc.) is beyond the scope of this enabler.

Services exposed to the application clients and to the other enablers

The services offered by this enabler are intended to be used by browser clients and enablers and, in general, by all the applications and enablers that need to download any type of content from a remote server (e.g. a Device Management enabler). The primary objective of the services offered by this enabler is therefore the provision of simple tools to retrieve remote content.

Related standards

- OMA Download (OMA DL) [2]: The OMA Download model leverages the HTTP download mechanisms and adds a number of features, such as
    - ❑ Additional tools for content negotiation.
    - ❑ Well-formalised (protocol independent) META-data presentation.
    - ❑ Application layer confirmation of installation.

2.4.1.6 Data Synchronisation

Description

The Data Synchronisation enabler provides the ability to synchronise a data collection stored in the terminal with a remote server, or vice versa.

Functionality

The main functions that this enabler will offer are:

- Management of synchronisation sessions (starting a session from the client, notification of a session initiated from the server).
- Synchronisation operations (add, delete, change).
- Registration of applications for synchronisation (to enable notification).

Information that is capable of being synchronised is typically personal information (contacts, calendar entries etc.) but all applications could use this enabler in order to synchronise any information.

Services exposed to the application clients and to the other enablers

Services offered by this enabler must be used by enablers and applications that manage the personal information in order to synchronise it.

Related standards

- OMA Data Synchronisation [3]: This standard is based on the use of SyncML protocol to synchronise data between a terminal and a server.

2.4.1.7 PIM Database

Description

The PIM Database enabler provides access to personal information stored in the terminal, SIM cards and memory cards.

Functionality

The functionality of the PIM database enabler is to provide access to different types of information:

- Contact information: This enabler will be responsible for managing the contacts stored in the terminal's memory. It will also provide access to the contacts stored either on the SIM card or in a memory card.
- Agenda: This enabler provides a way to manage the calendar entries (appointments, events etc.).
- To-do: The PIM database will also manage a list of tasks to-do.

Services exposed to the application clients and to the other enablers

This enabler must guarantee that all the relevant applications (agenda client, calendar client etc.) and enablers have access to the personal information required.

Related standards

- vCard [4] [5]: This standard is responsibility of the Internet Mail Consortium (IMC). It defines a format for an electronic business card independent of the particular method used to transport it.
- vCalendar [6] [7] [8]: This standard is responsibility of the Internet Mail Consortium (IMC). It defines a format for calendaring and scheduling information independent of the particular method used to transport it.

2.4.1.8 Messaging

The messaging enabler will support all the messaging services required in OMTP platforms.

Functionality

The most important functionalities offered by this enabler are:

- Create messages.
- Send messages.
- Receive messages.
- Manage messages and mailboxes.

All the functionality related to the basic transport protocols (WAP Stack, HTTP etc.) is beyond the scope of this enabler.

Services exposed to the application clients and to the other enablers

Services exposed to this enabler are mainly intended to allow messaging clients to create, send and receive messages.

Related standards

- SMS [14]: The Short Message Service (SMS) specifies a standard to allow sending and receiving text messages to and from mobile terminals.
- MMS [15]: The Multimedia Messaging Service (MMS) is a technology specified by 3GPP that specifies a standard to allow sending and receiving multimedia messages (text, images, audios, videos etc.) to and from mobile terminals.
- email:
  - ❑ POP3 [9]: Post Office Protocol 3 (POP3) allows a client to retrieve email from a remote server account.
  - ❑ IMAP [10]: The Internet Message Access Protocol (IMAP) allows a client to access and manipulate email messages on a server.
  - ❑ SMTP [11]: The Simple Mail Transfer Protocol (SMTP) allows the transfer of emails reliably and efficiently.

2.4.1.9 Widget Library

The widget library provides functionality to render different elements of a graphical user interface and allows a user to interact with the graphical user interface. Widgets may represent a single user interface element, such as a button, or may combine with other widgets to create more complex elements that can be easily used by clients, such as list of text fields.

Functionality

This enabler must offer a consistent set of widgets across all application environments implemented on the terminal.

It must offer flexible mechanisms to configure the visual appearance of the widgets through the Customisation and Device

Management enablers. In this way, different visual representations (views/themes) of several widgets will be supported to ensure the customisation of the terminal's 'look&feel'.

It will also provide simple mechanisms to extend or replace the set of widgets to support new application clients or new 'look&feel'.

Services exposed to the application clients and to the other enablers

The Widget Library should expose a set of services through the 'services application interface'. The main objective of this interface is to allow application clients to build graphical user interfaces. The services will therefore allow application clients to modify certain parameters of the different widgets in a consistent way e.g. colours, position, size etc.

These services should focus on the functionality offered by this library, rather than on the widgets themselves. In this way, the enabler will allow a higher degree of customisation, e.g. different widgets may implement the same interface while customisation files could control which actual widget is going to be used in a particular screen.

In addition to this, the services must abstract user input in order to allow different user input paradigms (e.g. touch screen, keypad, cirque) to be used with a particular widget implementation, and vice versa.

3 DEFINITION OF TERMS

The table below contains the definition of terms used in this document. Cross-referenced words are printed in *italics*.

TERM	DESCRIPTION
LOOK&FEEL (L&F)	The visual representation and appearance of the user interface (Look) combined with trigger assignment (keys, key sequences, gestures, etc.), trigger structure (main item order), as well as navigation through functionality and system feedback for the user.
APPLICATION CLIENT	An executable program (application) that is part of the Client Framework and utilises the *Application Services Interface*.
CLIENT FRAMEWORK	The environment in which terminal *Application Clients* reside and execute.
APPLICATION SERVICES INTERFACE	The interface used by *Application Clients* to access the functions that the *Service Framework* offers to the *Application Clients*.
ENABLER, SERVICE ENABLER	A component of the *Service Framework* that provides a set of functions or services.
SERVICES	Common functionality or capability offered by *Service Enablers.*
SERVICE FRAMEWORK	The environment comprising all the enablers offering services to the *Application Clients*.
APPLICATION FRAMEWORK	The environment comprising all the clients and enablers available in a terminal and the interfaces across them.
SERVICE FRAMEWORK INFRASTRUCTURE	The basic system of elements that guarantees the functioning of enablers within the *Service Framework*. This system does not provide any functionality to the *Client Framework* infrastructure.
CLIENT FRAMEWORK INFRASTRUCTURE	The basic system of elements that guarantees the proper execution of application clients within the *Client Framework*. This system does not provide any graphical user interface.

4 ABBREVIATIONS

ABBREVIATION	DESCRIPTION
API	Application Programming Interface
ASI	Application Services Interface
CHTML	Compact HypertText Markup Language
CSS	Cascade Style Sheet
DDF	Device Definition Framework
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IMC	Internet Mail Consortium
IMAP	Internet Message Access Protocol
IP	Internet Protocol
L&F	Look&Feel
MMS	Multimedia Messaging Services
OMA	Open Mobile Alliance
OMA DL	Open Mobile Alliance DownLoad
OMA DS	Open Mobile Alliance Data Synchronisation
OMTP	Open Mobile Terminal Platform
OMTP TPC	OMTP Terminal Platform Committee
OTA	Over The Air
PIM	Personal Information Manager
POP3	Post Office Protocol 3
SIM	Subscriber Identity Module
SMS	Short Message Service

ABBREVIATION	DESCRIPTION
SMTP	Simple Mail Transfer Protocol
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol
WML	Wireless Markup Language
WSP	Wireless Session Protocol
WTLS	Wireless Transport Layer Security
WTP	Wireless Transport Protocol
XHTML	eXtensible HyperText Markup Language

5 REFERENCED DOCUMENTS

NO.	DOCUMENT	AUTHOR	DATE
1	Enabler Release Definition for Browsing V2.3 (http://www.openmobilealliance.org/release_program/docs/Browsing/V2_3-20050614-C/OMA-ERELD-Browsing-V2_3-20050614-C.pdf)	Open Mobile Alliance	18 Jan 2005
2	Enabler Release Definition for Download (http://www.openmobilealliance.org/release_program/docs/Download/v1.0-20040625/OMA-ERELD-DL-V1_0-20040625-A.pdf)	Open Mobile Alliance	25 June 2004
3	Enabler Release Definition for Data Synchronization (http://www.openmobilealliance.org/release_program/docs/DS/V1_2_C/OMA-ERELD-SyncML-DS-V1_2-20040601-C.pdf)	Open Mobile Alliance	01 June 2004
4	A MIME Content-Type for Directory Information (http://www.ietf.org/rfc/rfc2425.txt)	IETF	Sept 1998
5	vCard MIME Directory Profile (http://www.ietf.org/rfc/rfc2426.txt)	IETF	Sept 1998
6	Internet Calendaring and Scheduling Core Object Specification (iCalendar) (http://www.ietf.org/rfc/rfc2445.txt)	IETF	Nov 1998
7	iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-dos and Journal Entries (http://www.ietf.org/rfc/rfc2446.txt)	IETF	Nov 1998
8	iCalendar Message-Based Interoperability Protocol (iMIP) (http://www.ietf.org/rfc/rfc2447.txt)	IETF	Nov 1998
9	Post Office Protocol - Version 3 (http://www.ietf.org/rfc/rfc1939.txt)	IETF	May 1996
10	–Internet Message Access Protocol Version 4rev1) (http://www.ietf.org/rfc/rfc3501.txt)	IETF	March 2003

NO.	DOCUMENT	AUTHOR	DATE
11	Simple Mail Transfer Protocol (http://www.ietf.org/rfc/rfc821.txt)	IETF	Aug 1982
12	User Experience Customisation Functional Requirements: Look And Feel, Menu Customisation and Application Integration	OMTP	July 2005
13	Functional Requirements for Remote Service Provisioning	OMTP	July 2005
14	3GPP TS 23.040 version 4.9.0 Release 4 (2004-10) ”Technical realization of the Short Message Service (SMS)”	3GPP	Oct 2004
15	3GPP TS 23.140 version 4.10.0 Release 4 (2003-06) ”Multimedia Messaging Service (MMS); Functional Description)	3GPP	June 2003
16	Enabler Release Definition for Browser Protocol Stack 2.1 (http://www.openmobilealliance.org/release_program/docs/Browser_Protocol_Stack/V2_1-20050204/OMA-ERELD-Browser_Protocol_Stack-V2_1-20050204-C.pdf)	Open Mobile Alliance	4 Feb 2005