Mobile Email Architecture

Draft Version 1.0 – 23 Oct 2005

Open Mobile Alliance

OMA-AD-Mobile_Email-V1_0-20051023-D

Contents

Figures

Tables

1. Scope (Informative)

<< Briefly describe the scope of this document – how it presents the architecture of this particular enabler. Include an explanation of how this architecture relates to Open Mobile Alliance activity. If it adds clarity, also describe what is not in the scope of this architecture. DELETE THIS COMMENT >>

This document describes the logical architecture of the OMA mobile email enabler to guide the technical specification work.

While mobile email is defined in the requirement document [Mobile email RD] as access to email from a mobile device, the focus of this document is to provide an mproved user experience over alternate means of access to email like browsing, email notification or message / voice based access. The goal is rather to provide quasi-instantaneous and secure updates of the MEM client with new emails and server changes, optimized online and off-line usage and capability to securely send email from the appropriate server.

2. References

The policy and guidelines for references, particularly to material from other organizations, is available at http://member.openmobilealliance.org/ftp/tp/gen_info/Reference.shtml, the following is a brief summary:

1. OMA documents listed should have at least one approved version – draft-only docs should not be referenced. Exception exists for reference use in documents that will be approved with or after a referenced doc is approved (may be part of same enabler package). In short – approved docs should not reference unapproved docs.
2. When a reference is made to an OMA specification, then Open Mobile Alliance with the TM symbol (™) should be used in the description.
3. The name + version (no date) for OMA specifications are generally sufficient – dates should be used only if there is a specific reason to limit the usage.
4. For references to WAP Forum docs, dates should not be included as DID's for the old WAP Forum specifications are enough and the reference description should refer to WAP Forum™.
5. References to other docs should similarly provide sufficient information to uniquely determine the needed document and should provide the appropriate source information.
6. The URL for OMA material (new OMA and affiliate) should always be http://www.openmobilealliance.org (an exception is OMNA that is reached through http://www.openmobilealliance.org/tech/omna)

Models to use

[REFLABEL] *<General Model>* "Ref Title", Ref information (source, date, id), URL:http//<ref-source>/

[OMADOC] *<OMA Model>* "OMA Document Title", Open Mobile Alliance™, OMA-<docname>{-<version>}, URL:http//www.openmobilealliance.org/

If there are no entries in the table – enter 'none' to be clear.

DELETE THIS COMMENT

2.1 Normative References

Editor's notes: To be done later.

[OSE]	"OMA Service Environment" URL: http://www.openmobilealliance.org/
[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt
[Mobile email RD]	"Mobile Email Requirements", Open Mobile Alliance, OMA-RD_@@@-Vx_y, URL:http://www.openmobilealliance.org/

<< Add/Remove reference rows as needed! >>

2.2 Informative References

Editor's notes: To be done later.

[ARCH-PRINC]	"OMA Architecture Principles", <*doc ref*>, URL:http://www.openmobilealliance.org/
[ARCH-REVIEW]	"OMA Architecture Review Process", <*doc ref*>, URL:http://www.openmobilealliance.org/
[OMA-DICT]	"OMA Dictionary", <*doc ref*>,URL:http://www.openmobilealliance.org/

<< Add/Remove reference rows as needed! >>

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 and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

<<The Architecture Document is assumed to contain normative material and is expected to use the previous two paragraphs, if not (is it really an AD?), replace it with the following paragraph. DELETE THIS COMMENT >>

This is an informative document, which is not intended to provide testable requirements to implementations.

<<If needed, describe or declare using appropriate normative references the additional conventions that are used. DELETE THIS COMMENT >>

3.2 Definitions

Editor’s notes: To be done later.

<< Add definitions in new rows of the following table as needed. Delete all definitions that are not used in the document.
DELETE THIS COMMENT >>

Interface See [OMA-DICT].

3.3 Abbreviations

<< Add abbreviations as needed to the following table. DELETE THIS COMMENT >>

Editor’s notes: To be done later.

OMA Open Mobile Alliance

MEM Mobile Email Enabler

4. Introduction (Informative)

<< Describe the high level architecture in greater detail than provided in section 1. From a market perspective, this section should answer the following questions (in prose):

- What is the purpose of this architecture?
- What problems does this architecture solve?

DELETE THIS COMMENT >>

The mobile email enabler aims at supporting access to email from a mobile device. Email may be personal email provided by an email service provider or corporate email.

4.1 Use Cases

<< Identify the Use Cases that covered by the architecture. Add all references to Use Case documents to section 2. This section should also identify the major Actors in the architecture. DELETE THIS COMMENT >>

The mobile email enabler is designed to support the use cases identified in [Mobile email RD]. Across these use cases, the following actors can be identified:

- User who is able to access his or her email or email server updates, manipulate or edit his or her email and have it reflected on the email server, and compose or send new email that is then sent from its email server.
- Operators of the mobile network who provide the network features that can support the mobile email enablement
- Mobile email enabler providers who enable mobile email features. They may be:
  - Operators
  - Third party service providers
  - Email service providers
- Email service providers who provide email servers to the user.
  - Service providers (e.g. Operators, other email server providers)
  - Enterprises

As discussed in [Mobile email RD], multiple email service providers may be involved for a same user: the mobile email enabler needs to be able to suppot mobilization of multiple email accounts for a same user.

Eventually, consistent with the use cases described in [Mobile email RD], the user may be able to access his or her email on another device (e.g. laptop), possibly using different service providers and networks. The present AD and enabler provides recommendations on how the MEM client may sometimes synchronize with such a different email repository (e.g. email client on laptop) using its own synchronization mechanism instead of solely relying on the mobile email enabler mechanism while maintaining consistency for the mobile email enabler.

4.2 Requirements

<< This section MUST include:

1. identification of the Requirements Document(s) on which this architecture is based. The referenced RDs MUST be included in section 2.

2. a clear statement about the requirements that are met or satisified and those that are NOT met or satisified. There are many ways to provide this information including (but not limited to): stating "All requirements in RD XXX are met", stating "All requirements in RD XXX are met except Y.Y and Z.Z", stating "only requirements Y.Y and Z.Z in RD XXX are met", etc.

The editor may use a table to specify the above information.

DELETE THIS COMMENT >>

Editor's notes: To be done after progress on the Technical report (TR) and technology analysis.

4.3 Planned Phases

<< Specify where this architecture is within the projected phases (e.g. phase 1.0, phase 2.0, etc.). If the current phase is greater than phase 1.0, briefly describe how this version of the architecture differs from the previous version. It may be appropriate to include a separate sub-section for the various phases.

If no additional phases are planned beyond this architecture then state so.

DELETE THIS COMMENT >>

Editor's notes: To be done after progress on the Technical report (TR) and technology analysis.

5. Architectural Model

<< This section defines the enabler's architectural model. The model identifies: a) all internal functional components of this enabler, and b) all of the communication relationships between the components of this enabler and with other enablers and applications (including those specifications not defined by OMA).

This section SHOULD contain a diagram of the architecture. Diagrams in this section should contain logical entities only and not conflate logical entities with physical entities. However, mobile terminals and networks may be shown because of their potential relevance in the design of the architecture. Figure 1 is an illustrative example of an architectural diagram and should be modified to reflect this architecture.

Working Groups SHOULD re-use functions specified by other enablers. Working Groups should consult other Architecture Documents and Specifications to identify any of this architecture's functionality (e.g. its systems, subsystems, interfaces, etc) that is already specified.

This section MAY include an explanation and/or diagram to show how this architecture relates to the various views (i.e. the reference point view) defined in "Inventory of Architectures and Services". This diagram and explanation, however, are *optional*.

DELETE THIS COMMENT >>

5.1 Dependencies

<< This section MUST enumerate all of the dependencies this architecture has. Dependencies in this context include other enablers, specifications, etc. this enabler calls (i.e. re-uses). Each dependency MUST include a reference to the document(s) that specifies the depdency. All of these references MUST also be included in Section 2.1.

The enumeration would be along the lines of a list with entries such as

- IMAP binary extension [RFC3516]

where the reference (e.g. RFC3516 in this example) would link to the fully qualifed reference in section 2.1 table.

If this architecture has no dependencies, then this section only needs to contain a statement as such.

DELETE THIS COMMENT >>

Dependencies include:

- A MEM protocol.

Editor's note: Multiple technology solutions may exist. Dependencies for this item are to be done after progress on the Technical report (TR) and technology analysis.

- OMA CP support for Mobile Email enabler parameters
- OMA CP support to bootstrap installation of mobile email enabler over the air
- OMA DM for life cycle management of MEM client and parameters + revocation of the MEM client

- Some support of generic notification mechanisms (e.g. SIP event notify, UDP, …) which may lead to a generic cross enabler push / notification enabler. This is a loose dependency as existing enablers can be used for some form of outband notification (e.g. [OMA-EMN]).
- Non-intrinsic P parameters required to support policy enforcement on mobile email exchanges (e.g. charging, privacy/ spam protection, …)
- OMA STI to support transcoding when desired via external server(s).

5.2 Architectural Diagram

The mobile email enabler logical architecture is illustrated in Figure 1.

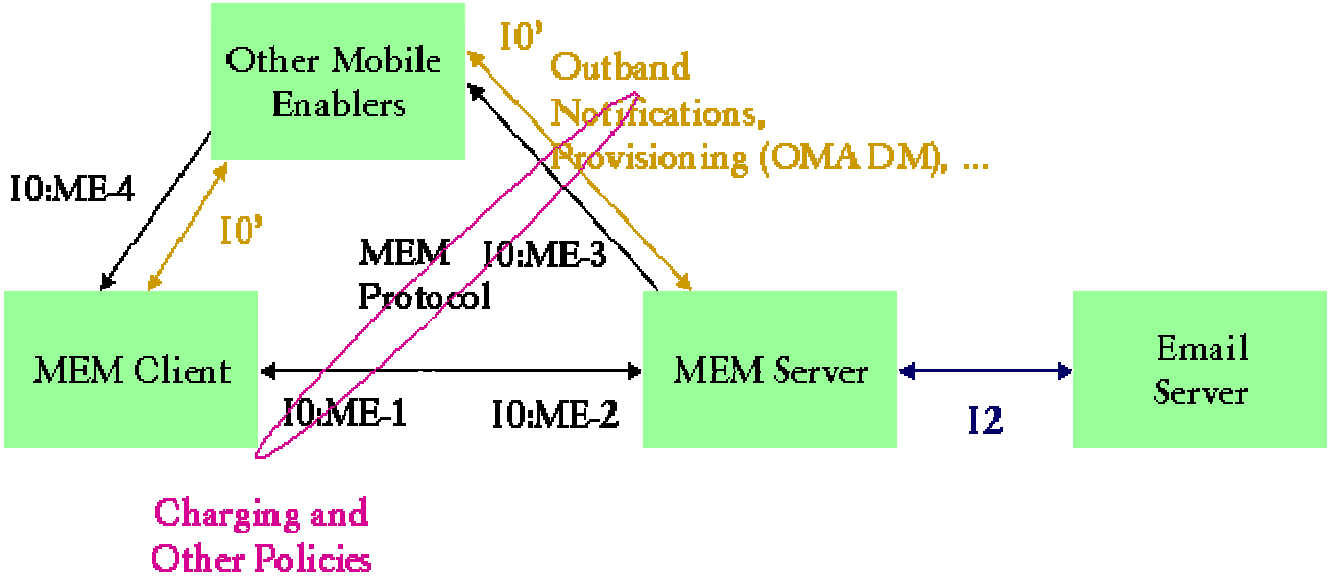

Figure 1 – Logical architecture for OMA Mobile Email enabler.

5.3 Functional Components and Interfaces

<< This section describes all of the architecture's functional components and interfaces. Each of the components should be described in a separate subsection and MUST contain at least the following information:

- Name
- Description
- Responsibility (e.g. what does the component do/perform)

Each component SHOULD have at least one interface that can be used by some other functional component, enabler, application, etc.

All of the interfaces should be described in this section. The interfaces MUST be described in a language-independent way as required by [ARCH-PRINC].

Each interface description MUST include at least the following information:

- Name

- Description
- Entities in this enabler that will use the interface

Interface naming convention: The name of an interface consists of one, two, or three characters, followed by a dash, followed by a running number (starting at "1" and counting upwards in steps of 1 for each new interface). Each work group decides about the character(s) for their interfaces as long as there is no duplication with already existing names (work groups can consult ARCH to confirm). Interface names should be chosen in an intuitive way to allow easy recognition of the interface (e.g. based on what functionality is communicated over the interface). Some examples are:

B-1 B stands for "Browsing"

POC-5 POC stands for "Push to Talk over Cellular"

MMS-7 MMS stands for "Multimedia Messaging"

DELETE THIS COMMENT >>

The following main enabler components are identified:

- The MEM client which implements the client-side functionality of the OMA Mobile Email Enabler. It is also responsible for providing the mobile email user experience and interface to the user and storing the email and data to be sent to the MEM server when not connected.
- The MEM server which implements the server-side functionality of the OMA Mobile Email Enabler (MEM).
- The MEM protocol between the MEM Client and MEM Server.

  It is responsible for all the data exchanges other than server to client event notifications that take place between the MEM client and server in order to update the MEM client with email server changes, the email server with changes in the MEM client and to send new email from the email server.

  Server to client notifications of email server events can be transported via the MEM protocol. We then speak of inband notifications.

Other OMA enablers are needed to directly support the mobile email enabler:

- OMA DM and CP to support over the air installation of the MEM client on the device, provisioning of its settings and revocation
- Messaging enablers for outband notification, where outband notifications that are server to client event exchanges not transported by the MEM protocol but via other channels. Such channels may involve:
  - SMS including GSMSMS or WAP WDP a la EMN
  - MMS
  - WAP Push
  - Additional outband notifications like SIP push (SIP event notify) or UDP might also be used considered.

  It should be noted that these notification mechanisms could be seen as part of a generic new OMA push / notification enabler.

The different interfaces that are identified are:

- ME-1: MEM client I0 interface to interact via the MEM protocol with the MEM server

- ME-2: Corresponding I0 interface of the MEM server
- ME-3: Outband MEM server I0 interfaces (e.g. to support generation of server to client notifications).
- ME-4: Outband MEM client I0 interfaces (e.g. to receive server to client notifications).

In addition, the MEM server and client may interfaces to the I0 interfaces of other mobile enablers (DM, CP, messaging). ME-3 and ME-4 may be bound on such interfaces.

According to the OSE [OSE], non-intrinsic functions can be provided by other enablers to enforce service providers policies like:

- Charging of the traffic
- Privacy and spam protection

These are not discussed in the present document.

The MEM server enables an email server. In a particular implementation, the email server may be packaged within (internal to it) the MEM server or be in a separate component. In such cases, interfaces to the email server are provided via an I2 type of interface (out of scope of this work).

5.3.1 MEM Server

The MEM Server is responsible for the following features of mobile email:

- Maintaining a high level of security of the message contents and the interchanges between the MEM Client and the email server.
    - Resolution of address for recipient of events
    - Authentication and authorization of the MEM client retrieving message content (i.e. headers, body, and attachments)
    - Authentication of the email server
    - Authentication of the MEM client
    - Authentication and authorization of originator of submitted messages
- Applying user preferences/filters/settings to the email information obtained from email server
    - Event and message filtering – based on header information, recipient's location (e.g., roaming), and folder information
    - Content screening – based on spam/virus-prevention information
    - Content adaptation (of attachments) – based on client capabilities
- Sending of events to the client/server when requested
- Support of extended mailing services
    - Forward without download – while editing different header fields or attached content of the original message.
    - Reply without download – including attachments in reply message, editing of the distribution list.
- Maintain connectivity to email server session even when client's connectivity may be intermittent.
    - Maintain state of session and update client when session reconnected
- Identify the source email server & account for each message/event, to allow client to handle the messages/events according to source, e.g., different "logical folders" for different accounts, different "icons" for different accounts.
- Collect metering information for per-unit metering schemes

Due to the wide range of deployment models, types of clients, usage profiles, and email servers that are being accessed – the MEES needs to be configurable to support different feature sets, levels of security, and logical flows. The configuration should take into account the various characteristics of the installation.

5.3.2 Technology feature requirements for the mobile email enabler

Editor's note: The changes agreed when disposing OMA-MEM-2005-0054R01-Comments_MEM_Technology_Features and OMA-MEM-2005-0056R01-Comments_0054 are awaiting minutes of October 18, 2005 MEM meeting (or a revision 0054R02)…

- Mechanisms to align email messages between the MEM client and the email server via the MEM server. The mobile email enabler focuses solely on the interaction between the MEM client and MEM server.
- Mechanisms for event-based server to client alignment:
    - Defines the relationship between notification mechanisms and MEM protocol
        - To minimize the latency observed for email events on the email server to be reflected in the MEM client.
        - To avoid unnecessary polling and requests from the MEM clients:
            - To reduce the total amount of data to be exchanged between MEM server and client, e.g. by allowing the MEM client to select which messages to align.
            - To reduce the amount of transactions.
    - Needs to cope with possible lost or delayed notifications
    - Support in-band (ME-1/ME-2 exchanges) and out-band notifications (Exchanged via ME-3/ME-4 via other enablers).
        - Specified in ways that are network transport independent but may contain some bindings to particular notification channels (e.g. SMS binary, WAP Push, SIP Notification, …)
        - When the MEM client is connected to the MEM server, only inband notifications shall take place
    - Defines notification payload for inband and outband mechanisms.
- Server-side filtering to decide which messages will be accessible by the MEM client.
    - Filtering results into the following logical types:
        - Type A: Messages filtered out and not accessible by the MEM client (not notification, no header access, no access)
        - Type B: Messages that are accessible by the MEM client but no outband notification takes place. Inband notification might however take place if MEM client is already connected to MEM server.
        - Type C: Messages that are accessible by the MEM client for which notifications (outband or inband) are always sent to the MEM client.
    - Notions of Filters:
        - View filters: Filters that determine which email messages are of type B and C or A
        - Notification filters: Filters that determine which email messages are of type C or B
        - Event filters: Filters that determines what events are to be notified to the client
    - Mechanisms to allow the user to update the filters from the MEM client
- Client-side download and storage preferences:
    - Manage which of the accessible messages are maintained on MEM client

  - Manage which parts are maintained on MEM client
  - Configurable by user
  - MEM client may support encrypting and password protecting the messages.
- Client-side event filtering:
  - Local delete: ability to delete email message from the MEM client view while retaining the message on the email server. Some information may be passed to the MEM server.
  - Attachment local delete: Ability to delete from the MEM client the attachment while maintaining the view that an attachment is available for download from the email server.
  - Remote delete: ability to delete email messages both on the MEM client and on the email server.
- Mechanisms for media conversion
  - Allows the MEM client to request conversion – including transcoding - of a body part or attachment from the MEM server when the email message part is fetched from the server.
    - The client may request conversion to a specific format/size, or
    - The client may request conversion to a server-selected format/size - where the server decides the format/size credentials based on any knowledge (e.g. client capabilities, user preferences) it may have.

Editor's note: need to detail at spec level what happens if the server has no knowledge of the client or user preferences.

  - Conversion does not alter the messages in the email server.
- Mechanisms for MEM client to submit email to the MEM server.
  - Mechanism to support remote message assembly on the MEM server based on email parts (body, address fields and attachments) that may not have been downloaded and others that may have been locally created or may have been downloaded and edited.
    - It may be desirable to support just uploading the differences of the body parts (e.g. address fields)
- Mechanisms to allow configuration and exchange of settings between the client and the server in band or outband:
  - Server to client: e.g. server ID, account name, policies, …
  - Client to server: e.g. rules filters vacation notices, notification channel, …
- Mechanisms to optimize bandwidth and/or delays on any data exchanges:
- Mechanisms for encryption of the email data exchanged between the email server and the MEM client.
  - It is critical that the email data remains encrypted at all time even if the MEM server is deployed outside the email server domain.
  - The mechanism should also be applicable to notifications if they carry information worth protecting
- Mechanisms for the MEM client to determine the capabilities of the server.
- Mechanisms to manage sessions:
  - Handling connectivity issues
    - E.g. dealing with IP address changes
    - E.g. re-establish secure connection

      E.g. suspend and resume minimizing data exchange duplication
- Mechanisms to support the different deployment models in appendix
  - Mobile email must be usable in the presence of firewalls
- Mechanisms to ensure integrity of the email data exchanged between the email server and the MEM client.

- Mechanisms for mutual authentication of the MEM client and the MEM server
- Mechanism to allow the MEM client to send recall request to the email server via the MEM server.
- Mechanisms to sign data exchanged between MEM client and MEM server.
- Mechanisms to allow the MEM client to work off line or in intermittent connectivity:
    - Store email and client email event
    - Detects network availability
    - Sends emails and email client events when network connectivity is available

5.4 Flows

<< The objective of this section is to describe the high-level logical flows between the architectural entities.

DELETE THIS COMMENT >>

The high level logical flows associated to the mobile email enabler are described below:

- Server to client notification:
    - An event (new email, change of state of email) takes place in the email server.
    - The MEM server generates a notification, if prescribed by enabler settings and filtering rules (as set by administrator or user based on the type of event). Notifications may additionally provide information about events generated or received in the server (e.g. the subject of the new email, the name of the attachments, etc).
        - If outband notifications are used, the notification is sent in the appropriate channel:
            - E.g. as separate message through ME-3 (e.g. SIP event notify) or bound to another OMA messaging enabler (e.g. as a WAP Push message).
        - If inband notification is used (and therefore a ME-1/ME2 session is established), the notification is sent via ME-2 interface.
    - The MEM client receives the notification:
        - Respectively via ME-4 (possibly bound to the I0' of another enabler) or ME-1 I0'
    - Based on its settings, the MEM client:
        - Updates its state (e.g. delete a local email)
        - Queues the notification for the next time it retrives information from the server
        - Goes back to the MEM server via ME-1 to act on the notification by retrieving appropriate data.
    - If no data connection is established between the MEM client and the MEM server, the MEM client establishes a connection (including authentication etc…). If a connection exists (e.g. when using inband notification), this steps is not repeated.
    - Through ME-1 the MEM client requests data from the MEM server to act on the notification. This is received by the MEM server through its ME-2 interface.

- The MEM server provides (via ME-2) the requested data to the MEM client (via ME-1).
- The MEM server may provide additional events and data for:
  - Notifications that it has previously sent to the client but to which the MEM client never reacted (e.g. may have been lost) or that it had queued (e.g. because the MEM client was not reachable and there was little value to continue to send server to client notifications.
  - Notifications for new server events that occurred since
  - Additional data and information needed by the server as described above.

  This robustizes the behaviour of the enabler to intermittent connectivity and ureliable connectivity.

- Client events (deleted mail, readunread changes etc…) are sent (via ME-1) to the MEM server (received via ME-2) if appropriate based on settings / filtering rules:
  - Depending if a connection exists or not, it is first established as described above when accessing additional data and events
  - After sending the data, the MEM server may reply with data analogous to the notifications and data as descried above in answer to a request for more data.

  The MEM server then updates the appropriate email server.

  If the MEM client can not connect to the MEM server, the events are queued and stored in the client and sent when connection is eventually established.

- New emails are sent from the MEM client (via ME-1) to the MEM server (received via (ME-2) as for the previous case. The MEM server then sends the email from the emails server. If the MEM client can not connect to the MEM server, the new emails are queued and stored in the client and sent to the MEM server when connection is eventually established.

- The MEM server and MEM client can directly interact to support usage of the enabler:
  - Request more data from MEM server (e.g. access more of an email partially on the MEM client).
  - Save of email draft to MEM server
  - Attachment manipulation:
    - Download an attachment
    - Convert an attachment (mime type / sub-type conversion or transcoding)
    - Server-side (partial) composition for reply / forward:
      - Forward without download
      - Partial edit and partial forward without download of:
        - Address fields,
        - Body
        - Attachments

Appendix A. Change History (Informative)

<< The following is a model of a revision table. DELETE THIS COMMENT >>

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA
OMA-xxyyz-V1_0-20021001-A	01 Oct 2002	Initial document to address the basic starting point Ref TP Doc# OMA-TP-2002-1234-xxyyzForApproval
OMA-xxyyz-V1_1-20030405-A	05 Apr 2003	description of changed Ref TP Doc# OMA-TP-2003-0321-xxyyzV1_1forApproval

A.2 Draft/Candidate Version 1.0 History

<< This section is available in pre-approved versions – it should be removed in the actual approved versions. DELETE THIS COMMENT >>

Document Identifier	Date	Sections	Description
Draft Versions OMA-AD-Mobile_Email-V1_0	21 Jun 2005	All	Incorporates input OMA-MWG-2005-0057_Mobile_Email_AD_population
	29 Sep 2005	All	CRs and input agreed at the London face to face meeting: - OMA-MEM-2005-0009R03-IC-AD-Mobile_email_V1_0-20050621-Deployment-Model - OMA-MEM-2005-0038R01-CR-MEES-Responsibilities.zip - OMA-MEM-2005-0040-Mobile-email-CR-AD - OMA-MEM-2005-0048-AD_CR_interfaces - Global e-mail to email changes.
	23 Oct 2005	All	Updates based on agreements after London FTF meeting (R&A) and at Sydney FTF: - OMA-MEM-2005-0041R01-Needed_Technology_Features - OMA-MEM-2005-0052-AD_CR_Lemonade_model_mapping - OMA-MEM-2005-0055-CR-AD-New-Names-for-Functional-Components - OMA-MEM-2005-0058R01-follow_up_0054_and_0056 - OMA-MEM-2005-0062-positioning_STI - OMA-MEM-2005-0064R01-General-contents-of-Notification

Appendix B. Implementation considerations

If needed, add annex to provide additional information to support the document. In general, this information should be informative, as normative material should be contained in the primary body of the document.

Note that the styles for the headers in the appendix (App1, App2, App3) are different than the main body. The use below is intended to validate the styles to be used. Remove if not needed.

DELETE THIS COMMENT

B.1 Implementation of the mobile email enabler server

Mobile email enabler implementation may wish to delegate transcoding to the OMA STI enabler [OMA-STI]. It is an implementation choice and it may not be appropriate for certain deployments.

Editor's note: References to be added.

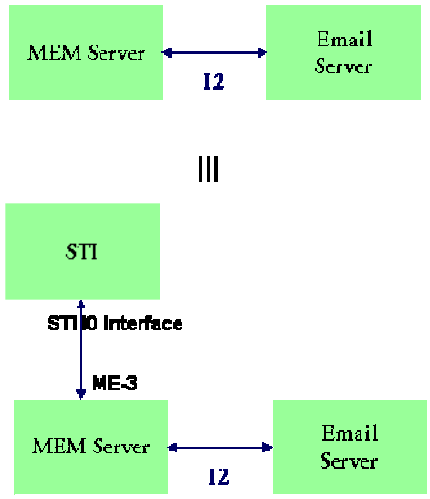

Figure 2 - Particular implementation case where MEM server relies on OMA STI enabler for transcoding.

B.2 Proxies and firewalls

[Mobile email RD] requires that the mobile email enabler must be compatible with firewalls. Figure 3 illustrates where firewalls may be present.

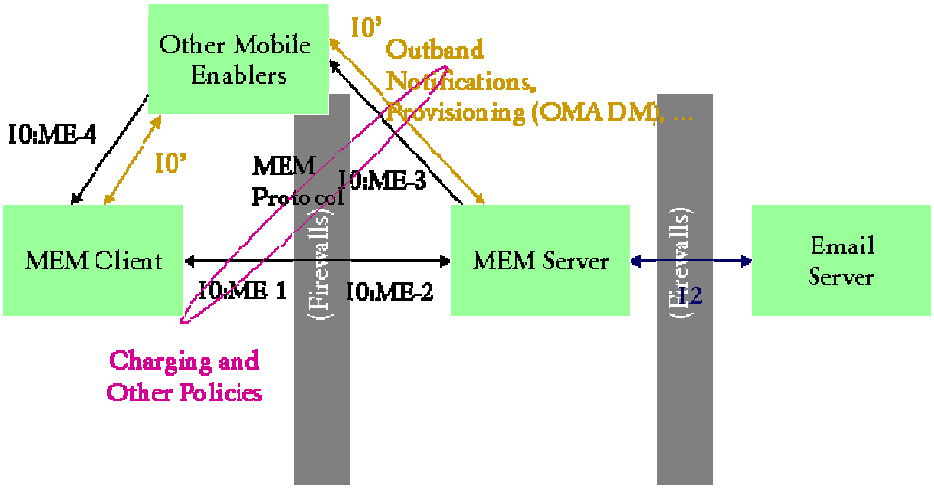

Figure 3 – Mobile email enabler logical architecture and possible firewalls

To facilitate crossing the firewalls and support of different deployment models, it should be noted that the mobile email enabler can be deployed via a mobile email enabler proxy between ME-1 and ME-2. The proxy channels all ME-1/ME-2 communications to and from the MEM server.

The proxy allows the MEM protocol through the firewalls in front of the MEM server. The role of such a proxy is to allow the mobile email enabler to be located in the same domain as the email server in some deployment models and therefore alleviate the confidentiality and other security constraints that may be imposed on MEM server implementations.

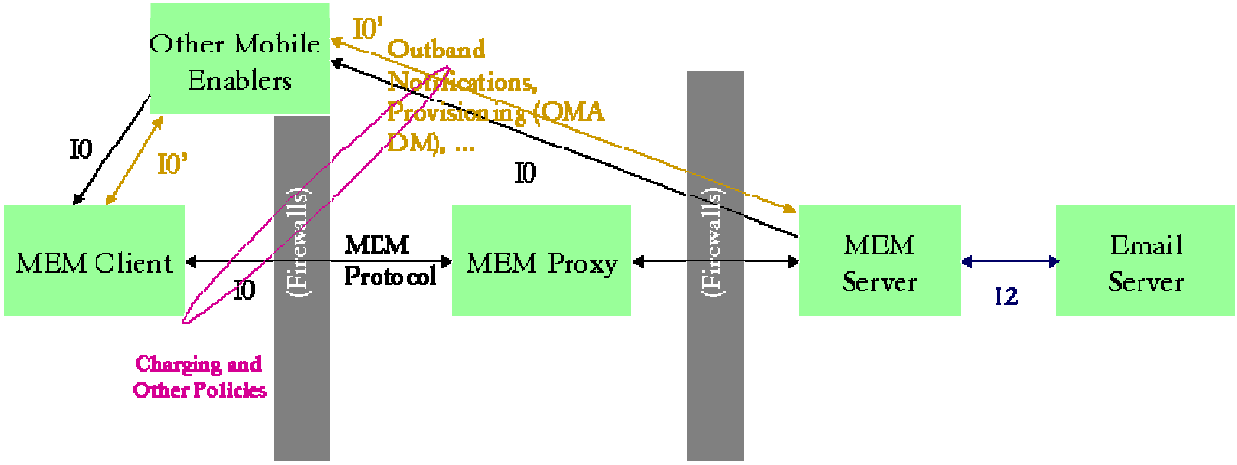

Figure 4 – MEM protocol proxy

If the out-band notifications and provisioning are used, the following deployment model may also be needed, otherwise the other mobile enablers may require more resources. In addition, if a more secure connection is used between the proxy and MEM server, this model shall be more reliable. In such case the proxy function channels all ME-1/ME-2 and ME-3 (I0') exchanges to and from the MEM server.

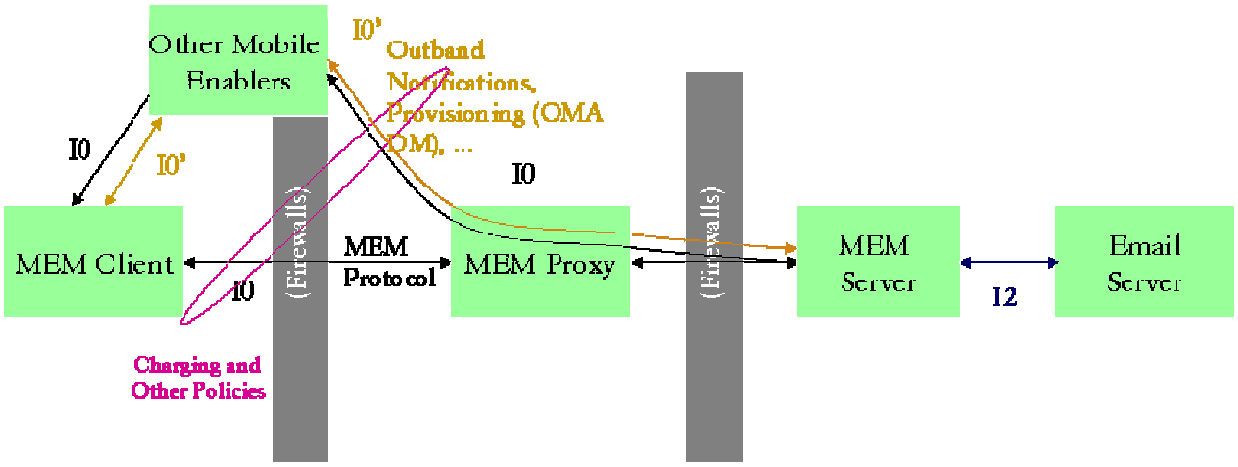

Figure 5- Logical Usage Model with Proxy also for other enablers

B.3 Deployment cases

The proposed logical architecture for the mobile email enabler supports a rich set of deployment models illustrated in the following figures. This covers all the deployment case that have been envisaged to date. Note that the cases of Figure 10 and Figure 11 imply that mobile email enabler must be compatible with end to end encryption between the email server and MEM client. Such schemes must be introduced as part of the MEM server implementation / I2 / email server implementation when considering these deployments.

All the cases with proxy can exist with both variations described in Figure 4 and Figure 5.

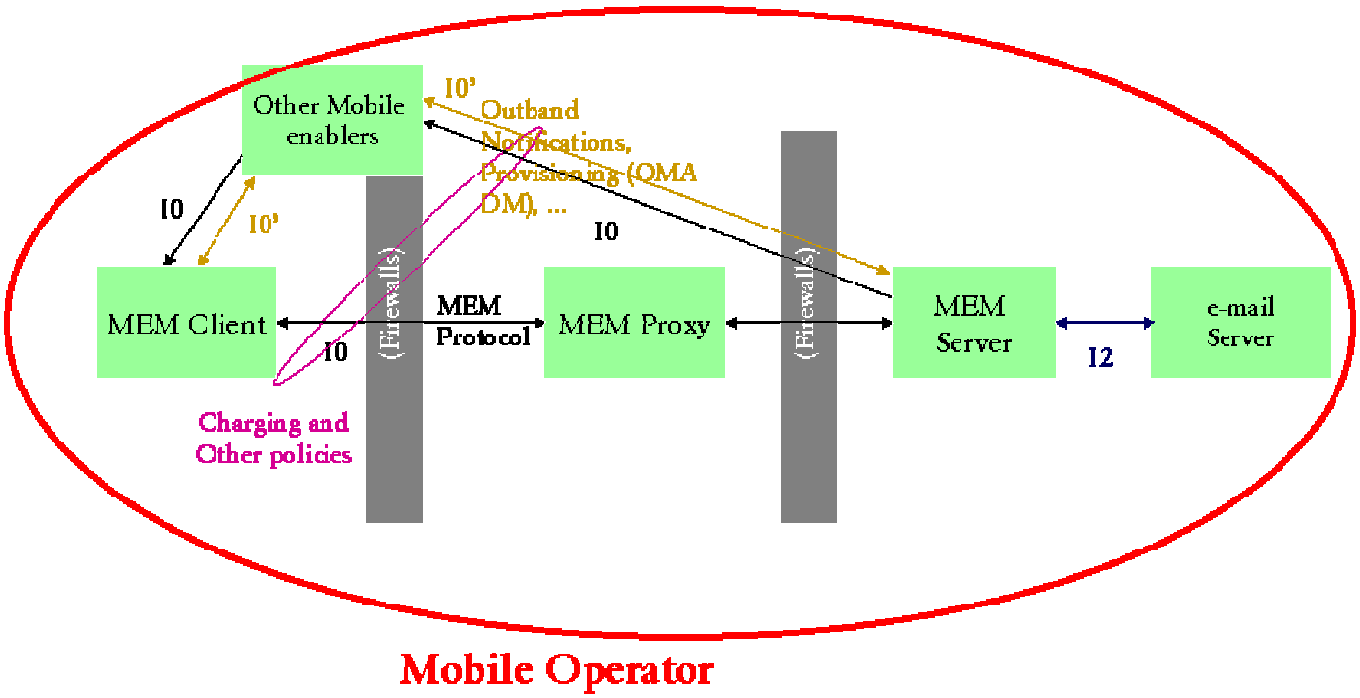

Figure 6 – Deployment within a mobile operator domain for an operator hosted email service. The use of the proxy is optional.

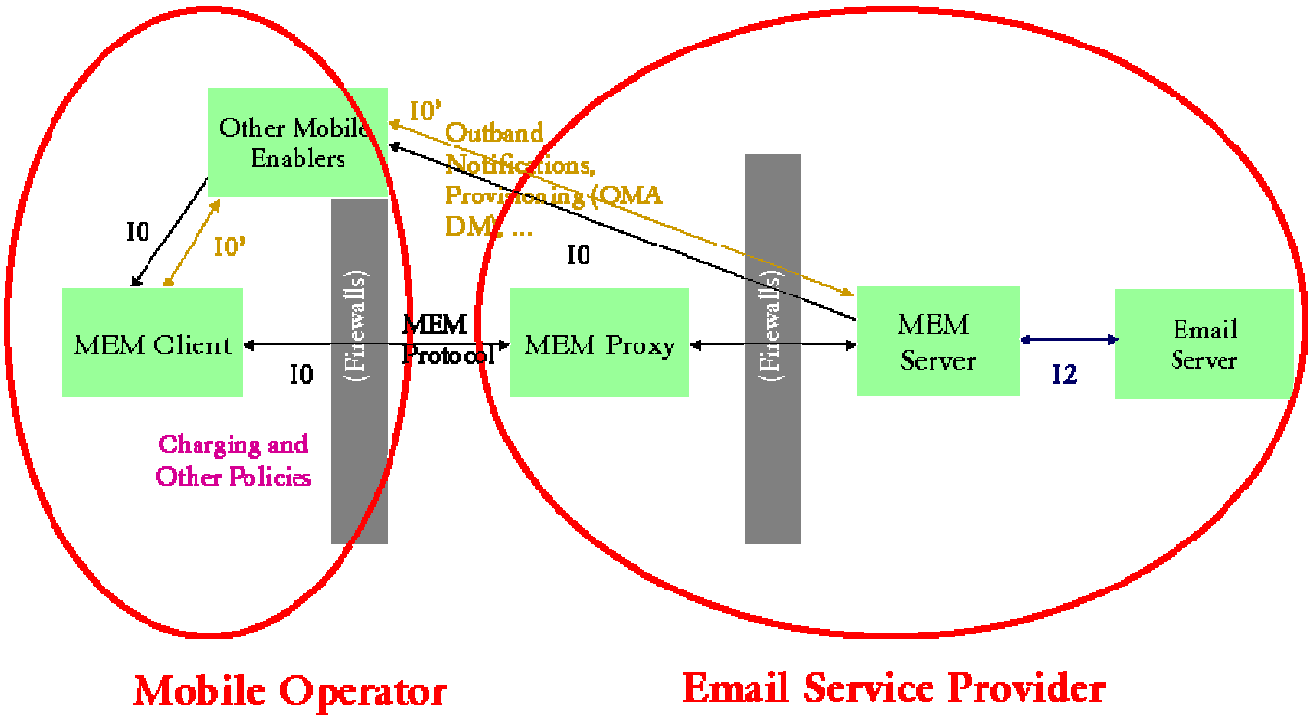

Figure 7 – Deployment by an email service provider (enterprise or ISP (e.g. personal email provider)). The proxy is deployed by the email service provider in the DMZ. Other mobile enablement and transport are provided by the mobile service operator.

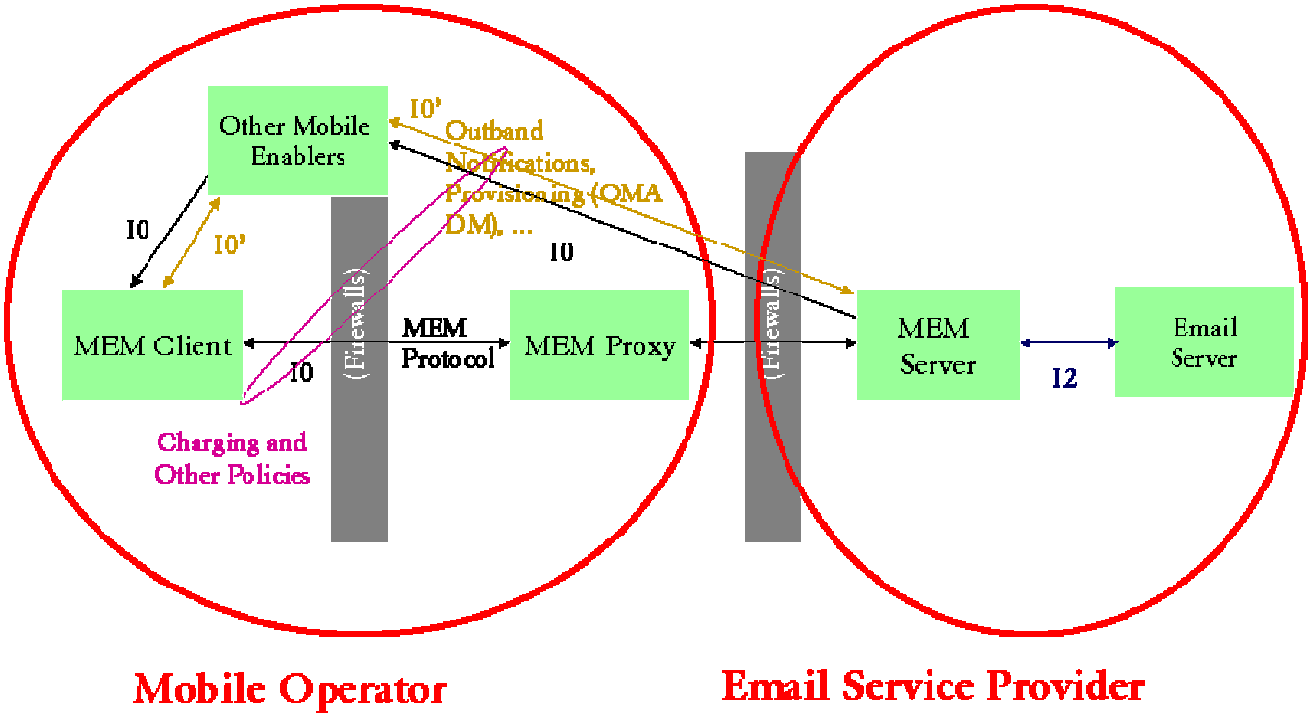

Figure 8 - Deployment by an email service provider (enterprise or ISP (e.g. personal email provider)). The proxy is provided as a service by the mobile operator. Other mobile enablement and transport are provided by the mobile service operator.

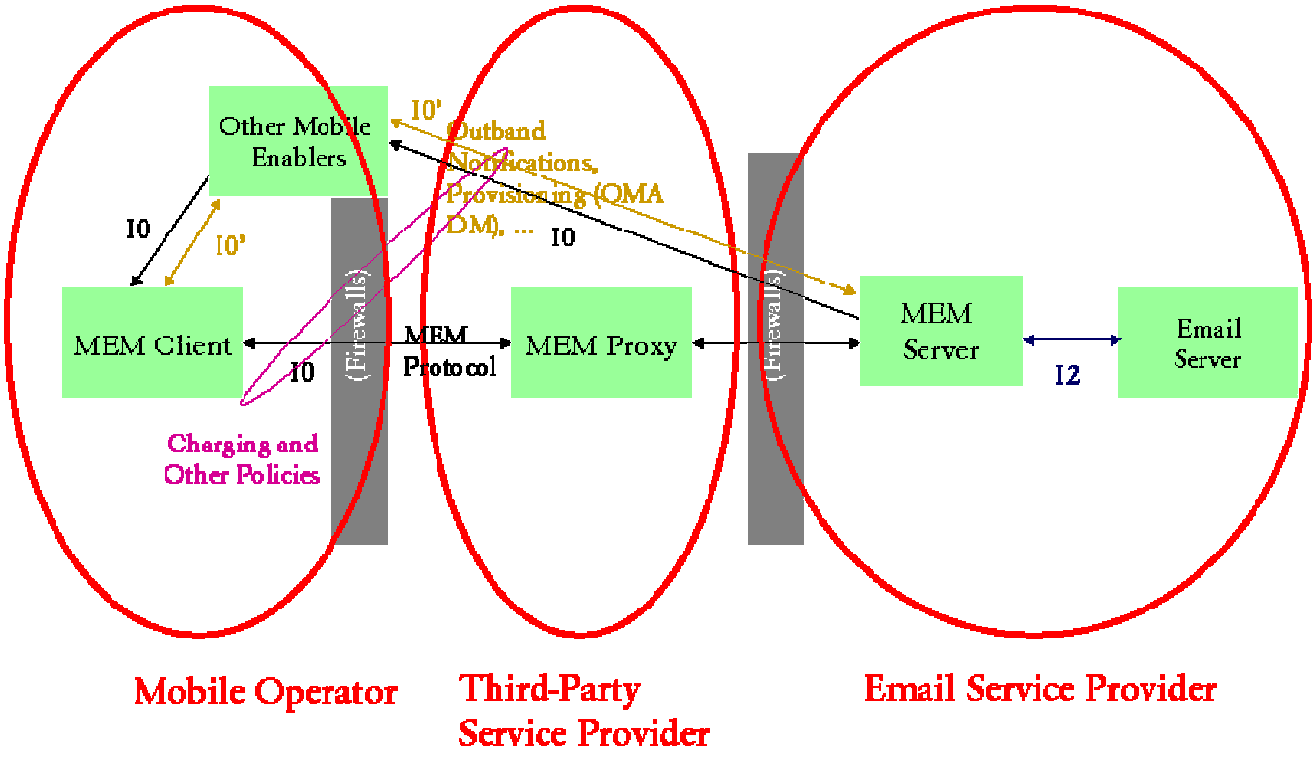

Figure 9 - Deployment by an email service provider (enterprise or ISP (e.g. personal email provider)). The proxy is provided as a service by a third party. Other mobile enablement and transport are provided by the mobile service operator.

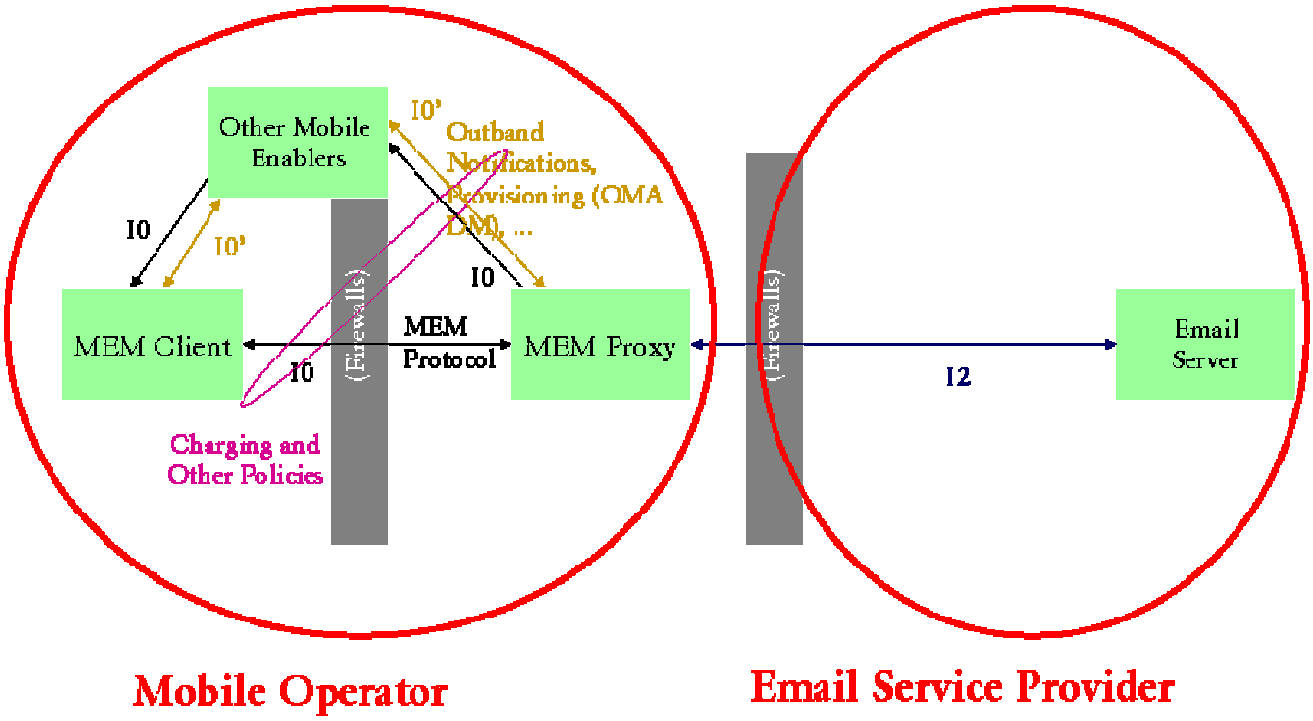

Figure 10 - Deployment by a mobile operator of a mobile email enablement service offered to email service provider (enterprise or ISP (e.g. personal email provider)). All mobile enablement and transport are provided by the mobile service operator. All data must remain end-to-end secure, including at the level of the MEM server implementation.

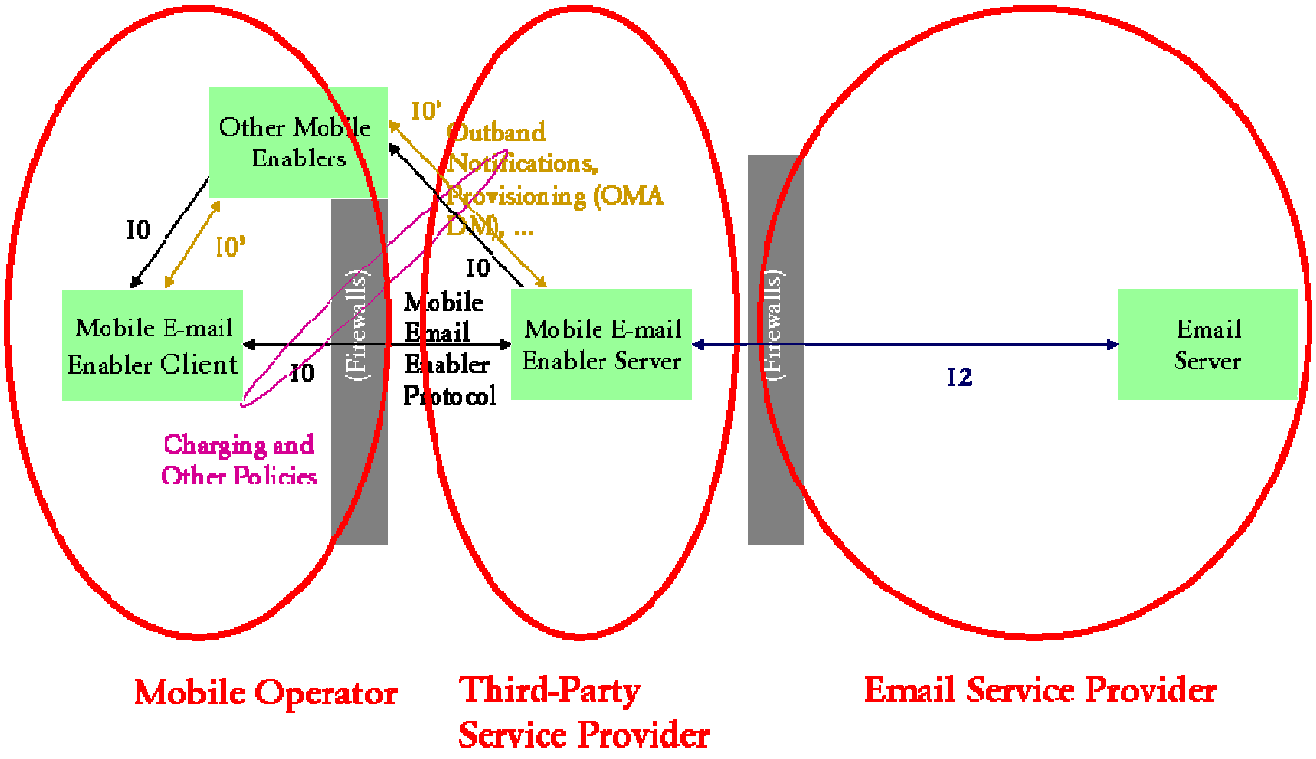

Figure 11 - Deployment by a third party service provider of a mobile email enablement service offered to email service provider (enterprise or ISP (e.g. personal email provider)). Other mobile enablement and transport are provided by the mobile service operator. All data must remain end-to-end secure, including at the level of the MEM server implementation.

Appendix C. IETF Lemonade Realization

The IETF Lemonade WG defines IMAP and Submit extensions that can support the mobile email requirements and use cases addressable within the scope of IETF. These specifications are captured in the Lemonade profile.

Editor's note: References and definitions to be added to appropriate AD sections.

The IETF Lemonade addresses standard message stores and submit servers.

Realization of the OMA mobile email enabler based on the Lemonade profile and IETF internet email stack is represented in Figure 12.

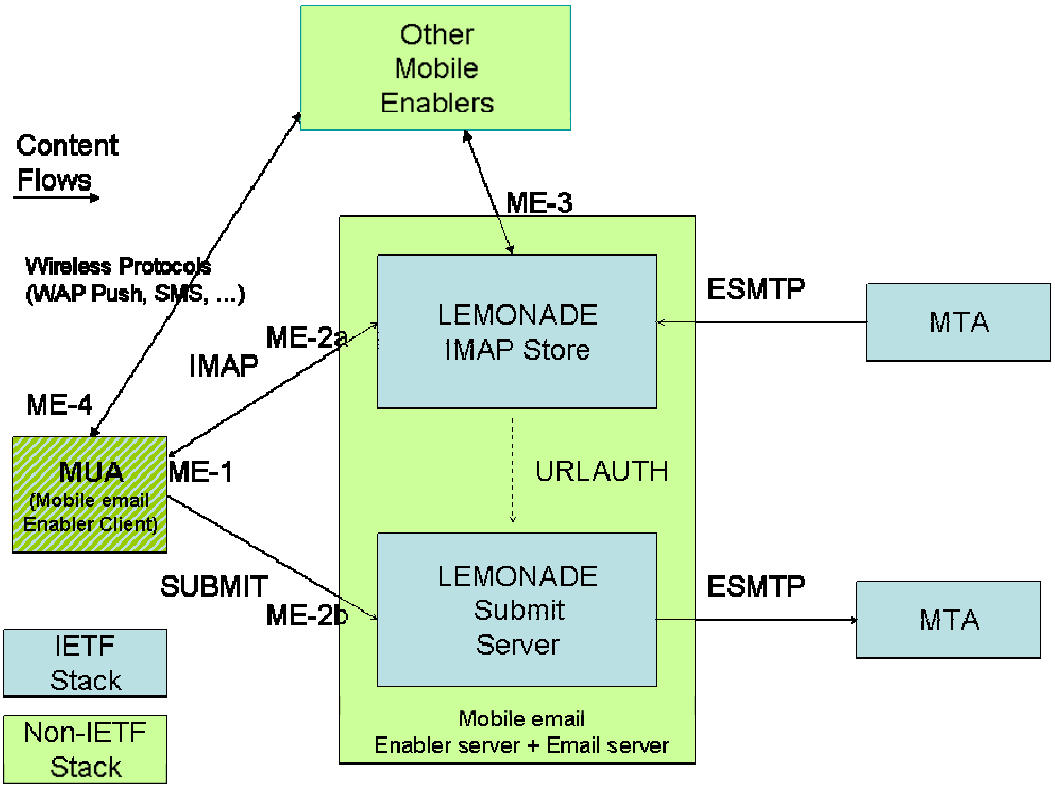

Figure 12 - Lemonade realization of mobile e-mail enabler using Lemonade IMAP and submit servers. IETF and non IETF stacks are colour coded.

In Figure 12, mobile email enabler server and email server components have collapsed into just two components specified by IETF Lemonade: the Lemonade IMAP store and submit server plus mechanisms to support out band data exchanges.

As discussed in section B.2, proxies may be involved.

In general, messages stores and bindings may not be limited to the IETF stack. Still the enabler may consist of an Lemonade realization of the mobile email enabler. The resulting architecture is represented in Figure 13. Again, proxies may be involved.

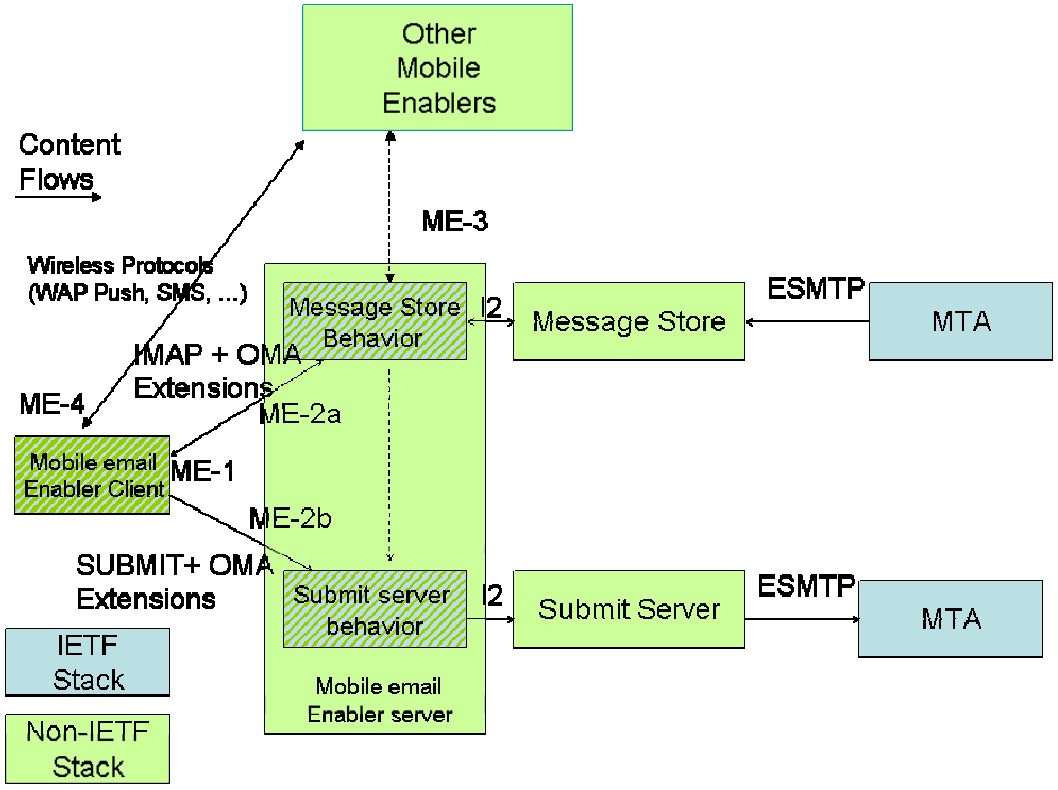

Figure 13 - OMA Mobile email enabler realized based on IETf Lemonade stack