# Enhanced Visual Voice Mail

# Architecture and Technical Specification

Approved Version 1.0 – 15 Sep 2015

**Open Mobile Alliance**

OMA-ER-EVVM-V1_0-20150915-A

# Contents

# Figures

# Tables

# 1. Scope

The scope of this EVVM Enabler Release is to fulfil all those requirements items that have been outlined in the Enhanced Visual Voicemail requirements document [OMA EVVM RD] and marked as EVVM 1.0. The two main parts of this document are as follows:

- The architecture part of this document comprises the definition of the functional components and the interfaces used or exposed by these functional components. It also describes the relationship between the EVVM Enabler and the external resources that are utilized by the EVVM Enabler.
- The technical specifications part specifies all basic procedures performed by the EVVM functional entities and through the interactions between them, i.e., client-server and server-server. It also specifies the data formats for the interactions.

In addition, this document contains a selective set of detailed call flows for demonstrating procedures in the EVVM Enabler.

# 2. References

## 2.1 Normative References

**[3GPP TS23.140]** “Multimedia Messaging Service (MMS); Stage 2”, 3GPP, TS 23.140, URL:http://www.3gpp.org/

**[3GPP TS26.103]** "Speech codec list for GSM and UMTS" URL:http://www.3gpp.org/ftp/Specs/html-info/26103.htm

**[GSMA VVM]** “Visual Voice Mail Interface Specifications”, Version 1.3, Open Mobile Terminal Platform, OMTP URL:http://www.gsmworld.com/documents/

**[NIST FIPS-197]** “Advanced Encryption Standard (AES)”, URL:http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

**[OMA CPM AD]** “Converged IP Messaging Architecture”, Open Mobile Alliance™, OMA-AD-CPM-V1_0, URL:http://www.openmobilealliance.org

**[OMA CPM TS MS]** “OMA Converged IP Messaging Message Storage”, Open Mobile Alliance™, OMA-TS-CPM-MessageStorage-V1_0, URL:http://www.openmobilealliance.org/

**[OMA CPM TS SD]** “OMA Converged IP Messaging System Description”, Open Mobile Alliance™, OMA-TS-CPM-System_Description-V1_0, URL:http://www.openmobilealliance.org/

**[OMA DM]** “OMA Device Management”, Open Mobile Alliance™, OMA-ERP-DM-V1_3, URL:http://www.openmobilealliance.org/

**[OMA EVVM MO]** “Enhanced Visual Voice Mail Management Object”, Open Mobile Alliance™, OMA-TS-EVVM_MO-V1_0, URL:http://www.openmobilealliance.org

**[OMA EVVM RD]** “Enhanced Visual Voice Mail Requirements”, Open Mobile Alliance™, OMA-RD-EVVM-V1_0, URL:http://www.openmobilealliance.org

**[OMA EVVM XDM]** “EVVM XDM Specification”, Open Mobile Alliance™, OMA-TS-EVVM-XDM-V1_0, URL:http://www.openmobilealliance.org

**[OMA Push AD]** “Push Architecture”, Open Mobile Alliance™, OMA-AD-Push-V2_3, URL:http://www.openmobilealliance.org

**[OMA Push]** “OMA Push”, Open Mobile Alliance™, OMA-ERP-Push-V2_3, URL:http://www.openmobilealliance.org/

**[OMA SEC CF]** “Security Common Functions Architecture”, Open Mobile Alliance™, OMA-AD-SEC_CF-V1_1, URL:http://www.openmobilealliance.org/

**[OMA STI]** “OMA Standard Transcoding Interface”, Open Mobile Alliance™, OMA-ERP-STI-V1_0, URL:http://www.openmobilealliance.org/

**[OMA XDM]** “XML Document Management”, Open Mobile Alliance™, OMA-ERP-XDM-V2_1, URL:http://www.openmobilealliance.org/

**[RFC1847]** “Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted”, J. Galvin, October 1995, URL:http://tools.ietf.org/html/rfc1847

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

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

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

**[RFC2141]** "URN Syntax", R. Moats, May 1997, URL:http://tools.ietf.org/html/rfc2141

**[RFC2177]** "IMAP4 IDLE command", Leiba, B., June 1997, URL:http://tools.ietf.org/html/rfc2177

**[RFC2595]** “Using TLS with IMAP, POP3 and ACAP”, C. Newman, June 1999, URL:http://tools.ietf.org/html/rfc2595

**[RFC2822]** “Internet Message Format”, P. Resnick, April 2001, URL:http://tools.ietf.org/html/rfc2822

**[RFC2852]** “Deliver By SMTP Service Extension”, D. Newman, June 2000, URL:http://tools.ietf.org/html/rfc2852

**[RFC2971]** “IMAP4 ID extension”, T. Showalter, October 2000, URL:http://tools.ietf.org/html/rfc2971

**[RFC3030]** "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", G. Vaudreuil, December 2000, URL:http://tools.ietf.org/html/rfc3030

**[RFC3156]** “MIME Security with OpenPGP”, M. Elkins et al, August 2001, URL:http://tools.ietf.org/html/rfc3156

**[RFC3207]** “SMTP Service Extension for Secure SMTP over Transport Layer Security”, P. Hoffman, February 2002, URL:http://tools.ietf.org/html/rfc3207

**[RFC3261]** "SIP: Session Initiation Protocol", U. Rosenberg et al, June 2002, URL:http://tools.ietf.org/html/rfc3261

**[RFC3461]** “Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)”, K. Moore, January 2003, URL:http://tools.ietf.org/html/rfc3461

**[RFC3464]** “An Extensible Message Format for Delivery Status Notifications”, K. Moore et al, January 2003, URL:http://tools.ietf.org/html/rfc3464

**[RFC3501]** “Internet Message Access Protocol - Version 4rev1”, M. Crispin, March 2003, URL:http://tools.ietf.org/html/rfc3501

**[RFC3516]** "IMAP4 Binary Content Extension", L. Nerenberg, December 2000, URL:http://tools.ietf.org/html/rfc3516

**[RFC3798]** “Message Disposition Notification”, T. Hansen, Ed. et al, May 2004, URL:http://tools.ietf.org/html/rfc3798

**[RFC3801]** “Voice Profile for Internet Mail - version 2 (VPIMv2)”, G. Vaudreuil et al, June 2004, URL:http://tools.ietf.org/html/rfc3801

**[RFC3966]** "The tel URI for Telephone Numbers", H. Schulzrinne, December 2004, URL:http://tools.ietf.org/html/rfc3966

**[RFC4021]** “Registration of Mail and MIME Header Fields”, G. Klyne et al, March 2005, URL:http://tools.ietf.org/html/rfc4021

**[RFC4122]** "A Universally Unique IDentifier (UUID) URN Namespace", P. Leach et al, July 2005, URL:http://tools.ietf.org/html/rfc4122

**[RFC4467]** “Internet Message Access Protocol (IMAP) - URLAUTH Extension”, M. Crispin, May 2006, URL:http://tools.ietf.org/html/rfc4467

**[RFC4536]** “The application/smil and application/smil+xml Media Types”, P. Hoschka, May 2006, URL:http://tools.ietf.org/html/rfc4536

**[RFC4648]** “The Base16, Base32, and Base64 Data Encodings”, S. Josefsson, October 2006, URL:http://tools.ietf.org/html/rfc4648

**[RFC4865]** “SMTP Submission Service Extension for Future Message Release”, G. White, May 2007, URL:http://tools.ietf.org/html/rfc4865

**[RFC4880]** “OpenPGP Message Format”, J. Callas et al, November 2007, URL:http://tools.ietf.org/html/rfc4880

**[RFC5162]** “IMAP4 Extensions for Quick Mailbox Resynchronization”, A. Melnikov, D. Cridland, C. Wilson, March 2008, URL:http://tools.ietf.org/html/rfc5162

**[RFC5234]** “Augmented BNF for Syntax Specifications: ABNF”, D. Crocker & P. Overell, January 2008, URL:http://tools.ietf.org/html/rfc5234

**[RFC5321]** “Simple Mail Transfer Protocol”, J. Klensin, October 2008, URL:http://tools.ietf.org/html/rfc5321

**[RFC5322]** “Internet Message Format”, P. Resnick, October 2008, URL:http://tools.ietf.org/html/rfc5322

**[RFC5465]** "The IMAP NOTIFY Extension", Gulbrandsen, A., King, C., and A. Melnikov, February 2009, URL:http://tools.ietf.org/html/rfc5465

**[RFC5550]** “The Internet Email to Support Diverse Service Environments (Lemonade) Profile”, D. Cridland, Ed. et al, August 2009, URL:http://tools.ietf.org/html/rfc5550

**[RFC5598]** “Internet Mail Architecture”, D. Crocker, July 2009, URL:http://tools.ietf.org/html/rfc5598

**[RFC5738bis]** “IMAP Support for UTF-8”, P. Resnick, C. Newman, S. Shen, December 2011, URL:http://tools.ietf.org/html/draft-ietf-eai-5738bis

Note: this document is a draft specification. The reference is to be updated when the RFC number becomes available or at the latest, before the final approval of the Enabler package.

**[RFC5751]** Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification, B. Ramsdell, January 2010, URL:http://tools.ietf.org/html/rfc5751

**[RFC6531]** “SMTP Extension for Internationalized Email Addresses”, J. Yao & W. Mao, February 2012, URL:http://tools.ietf.org/html/rfc6531

**[RFC6532]** “Internationalized Email Headers”, A. Yang, S. Steele, N. Freed, February 2012, URL:http://tools.ietf.org/html/rfc6532

**[RFC6533]** “Internationalized Delivery Status and Disposition Notifications”, T. Hansen, .C. Newman & A. Melnikov, February 2012, URL:http://tools.ietf.org/html/rfc6533

**[RSA PKCS]** “PKCS #1 v2.1: RSA Cryptography Standard”, URL:ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf

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

**[SECG SEC1]** “Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography version 2.0”, URL:http://www.secg.org/download/aid-780/sec1-v2.pdf

**[SMPP]** “SMPP: Short Message Peer-to-Peer Protocol Specification version 3.4”, URL:http://www.smsforum.net

**[XSD-DEAC]** “XML Schema Definition: EVVM Deactivation Notification”, Open Mobile Alliance™, OMA-SUP-XSD_EVVM_DEAC-V1_0, URL:http://www.openmobilealliance.org

**[XSD-PREF]** “XML Schema Definition: EVVM Preferences Notification”, Open Mobile Alliance™, OMA-SUP-XSD_EVVM_PREF-V1_0, URL:http://www.openmobilealliance.org

**[XSD-SYNC]** “XML Schema Definition: EVVM Synchronization Notification”, Open Mobile Alliance™, OMA-SUP-XSD_EVVM_SYNC-V1_0, URL:http://www.openmobilealliance.org

**[XSD-TRAN]** “XML Schema Definition: EVVM Transcription Notification”, Open Mobile Alliance™, OMA-SUP-XSD_EVVM_TRAN-V1_0, URL:http://www.openmobilealliance.org

## 2.2 Informative References

**[OMADICT]** “Dictionary for OMA Specifications”, Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org

**[OMA-OSE]** “OMA Service Environment”, Open Mobile Alliance™, OMA-AD-Service-Environment-V1_0_5-20091008-A, URL:http://www.openmobilealliance.org

**[RFC6530]** “Overview and Framework for Internationalized Email”, J. Klensin & Y. Ko, February 2012, URL:http://tools.ietf.org/html/rfc6530

# 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.

All figures in the normative sections follow the conventions depicted in Figure 1.

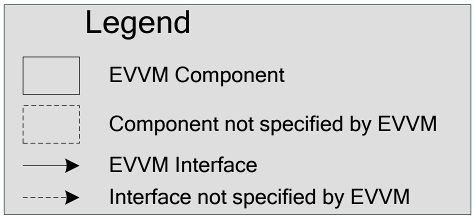
Legend
EVVM Component
Component not specified by EVVM
EVVM Interface
Interface not specified by EVVM

**Figure 1: Conventions for figures in the normative sections**

While describing various mechanisms across this document, this specification refers to individual preferences using a concise expression; the expression is defined using the ABNF notation [RFC5234] as follows:

```
reference-to-a-preference       = usage-expression path-expression
usage-expression                = "serviceprefs-global" \        ; Global service preferences.
                                / "serviceprefs-subscription" \   ; Subscription-specific service pref.
                                / "serviceprefs-final" \         ; Final preference set, see Figure 4.
                                / "clientprefs" \                ; Client preferences.
                                / "userprefs"                    ; User preferences.
path-expression                 = <XPath expression>
```

Examples:

- While describing the global password policies that apply to all users, the following expression would be used:
  `serviceprefs-global/prefs/bounds`
- While describing the password policies of an individual subscriber, the following expression would be used:
  `serviceprefs-subscription/prefs/bounds`
- While describing the overall password policies that apply to a subscriber, the following expression would be used:
  `serviceprefs-final/prefs/bounds`

## 3.2 Definitions

| | |
|---|---|
| XDM Agent | see definition in [OMA XDM] |
| XDM Client | see definition in [OMA XDM] |

## 3.3 Abbreviations

| | |
|---|---|
| **ASCII** | American Standard Code for Information Interchange |
| **BOM** | Byte Order Mark |
| **CPM** | Converged IP Messaging |
| **DM** | Device Management |
| **DSN** | Delivery Status Notification |

| | |
|---|---|
| **ESME** | External Short Message Entity |
| **EVVM** | Enhanced Visual VoiceMail |
| **GSMA** | GSM (Groupe Spéciale Mobile) Association |
| **IANA** | Internet Assigned Number Authority |
| **IMAP4** | Internet Message Access Protocol 4 |
| **IP** | Internet Protocol |
| **MDN** | Message Disposition Notification |
| **MIME** | Multipurpose Internet Mail Extension |
| **MMS** | Multimedia Messaging Service |
| **MSS** | Message Storage Server |
| **NUT** | New User Tutorial |
| **OMA** | Open Mobile Alliance |
| **OMNA** | Open Mobile Naming Authority |
| **OMTP** | Open Mobile Terminal Platform |
| **OSE** | OMA Service Environment |
| **PDA** | Personal Digital Assistant |
| **RFC** | Request For Comments |
| **SIP** | Session Initiation Protocol |
| **SMPP** | Short Message Peer-to-Peer Protocol |
| **SMS** | Short Message Service |
| **SMTP** | Simple Mail Transfer Protocol |
| **STI** | Standard Transcoding Interface |
| **TUI** | Telephony User Interface |
| **URI** | Uniform Resource Identifier |
| **UTF** | Unicode Translation Format |
| **VM** | Voice Mail |
| **VPIM** | Voice Profile for Internet Mail |
| **VVM** | Visual VoiceMail |
| **XDM** | XML Document Management |
| **XML** | Extensible Markup Language |

# 4. Introduction

The purpose of this specification is to provide the logical architecture model, the technical specification and static conformance requirements for the OMA Enhanced Visual Voice Mail (EVVM) Enabler.

The logical architecture model is described in section 6 "*Architectural Model*". The purpose of the logical architecture model is to identify functional components and interfaces and describe their high level roles, functionalities and relationships.

The technical specification is described in section 7 "*Technical Specification*". The purpose of the technical specification is to describe the identified interfaces in detail: the associated protocols, the means of identification, the data formats and, in general, the procedures and behaviors of the individual functional components with regards to utilizing these interfaces.

The static conformance requirements are described in Appendix B "*Static Conformance Requirements*". The purpose of these static conformance requirements is to describe conformance definitions and their dependencies for this Enabler, ultimately allowing validating compliance of individual client and server implementations to this specification.

## 4.1 EVVM Version 1.0

An EVVM user is capable of managing and handling his/her voicemails by exploiting the following enhanced functionalities compared to GSMA/OMTP VVM 1.3 [GSMA VVM]:

An EVVM user can create an EVVM voicemail locally at his/her EVVM Client and send it to the recipient's voicemail box via the interactions between the EVVM Client and EVVM Server. The EVVM Client includes the EVVM user identifier of the recipient when sending a voicemail. If the recipient is located in a remote EVVM environment, this voicemail will be sent first to the home EVVM server, then routed to the remote EVVM server and finally deposited to the recipient's voicemail box.

An EVVM user can access EVVM-based services with multiple devices (e.g., a fixed/mobile phone, a desktop/laptop computer, a PDA). Multiple identifiers of various types (e.g., phone number, email address, SIP URI) can be associated with his/her EVVM VM box. He/she shall be authenticated once for EVVM-based services with any one of the identifiers associated with his/her VM box. After a successful authentication, he/she can choose to identify himself/herself as the sender with any of the associated identifiers to send/forward voicemails. Other users can send voicemails using any of the identifiers.

An EVVM voicemail can contain several discrete media items (e.g., audio clips, text, images, video clips). An EVVM user can add a background audio item and send it together with his/her voicemail. The EVVM Enabler can set attribute 'title=' of the background media to" BackgroundSound" in the SMIL to differentiate the background media from a normal voice media. The recipient's EVVM Client can play simultaneously a voicemail and its associated background media with a lower volume, if present, for the recipient.

An EVVM user can request a voicemail to be delivered at a future time. The EVVM Client can include a delivery time in the voicemail. The EVVM Server then delivers the voicemail at the designated delivery time, if present.

Upon request of an EVVM user, the EVVM Client includes in a voicemail a Reply-to Indication containing the EVVM user/client identifier by which the voicemail recipient can reply instead of replying by the EVVM user identifier that the voicemail was sent from. With the EVVM client identifier, the recipient can reply to a certain device with which the sender expects to be reached. The EVVM Client can also include in a voicemail a Callback-Number field containing a phone number by which the recipient can make a call to the sender. The EVVM Server is allowed to add prefixes, if needed, to the phone number in the Callback-Number field in case the callback needs to traverse different network domains; it is out of scope how this will be implemented.

An EVVM user can send a stored EVVM voicemail without having to download it to the device. The EVVM Client can include a reference to a stored voicemail in the EVVM voicemail to be sent out.

Upon receiving an EVVM voicemail containing a reference to a stored voicemail, the EVVM Server fetches the voicemail referred to by the reference and replaces the reference with the stored voicemail in the received voicemail.

An EVVM user can request forwarding VMs via MMS/email as voice, or via SMS/MMS/email as text after voice-to-text conversion (transcription), if needed. A VM can be converted into a SMS/MMS/email message and forwarded to a

SMS/MMS/email system by an EVVM Forwarding Gateway, which is also responsible for converting a returned SMS/MMS/email delivery/read report into an EVVM delivery/read report and forwarding it to the EVVM Server.

An EVVM user can report a voicemail as spam to the EVVM Server. The processing of spam reports (e.g. human intervention to determine a spam and adding the spammer's address to the system black list) is out of scope.

# 5. Requirements (Normative)

The requirements driving this work have been captured in the Enhanced Visual Voice Mail Requirements specification, [OMA EVVM RD].

# 6. Architectural Model

The EVVM Enabler is realized using a client-server architecture based on a similar architecture from the GSMA VVM 1.3 Specifications. The OMA EVVM Enabler leverages the GSMA VVM 1.3 Specifications [GSMA VVM] by improving the existing voicemail service with market driven requirements and extending it with several new features and functions. It interacts with other network elements and re-uses concepts and technologies specified by other OMA Enablers and in Standard non-OMA specifications.

## 6.1 Dependencies

The EVVM Enabler utilizes several OMA Enablers and other non-OMA specifications (IETF, 3GPP, GSMA).

IETF dependencies:

- Internet Message Access Protocol version 4 (IMAP4) family of protocols and extensions
- Simple Mail Transfer Protocol (SMTP) family of protocols  and extensions

OMA dependencies:

- Message Storage: optional use of CPM Enabler’s Message Storage Server (MSS) as described in [OMA CPM AD] and [OMA CPM TS SD]
- XML Document Management: XDM Enabler as described in  [OMA XDM]
- Device Management: Device Management Enabler as described in [OMA DM]
- Notification delivery: Push Enabler as described in [OMA Push]
- Transcoding: Standard Transcoding Interface Enabler as described in [OMA STI]

## 6.2 Architectural Diagram

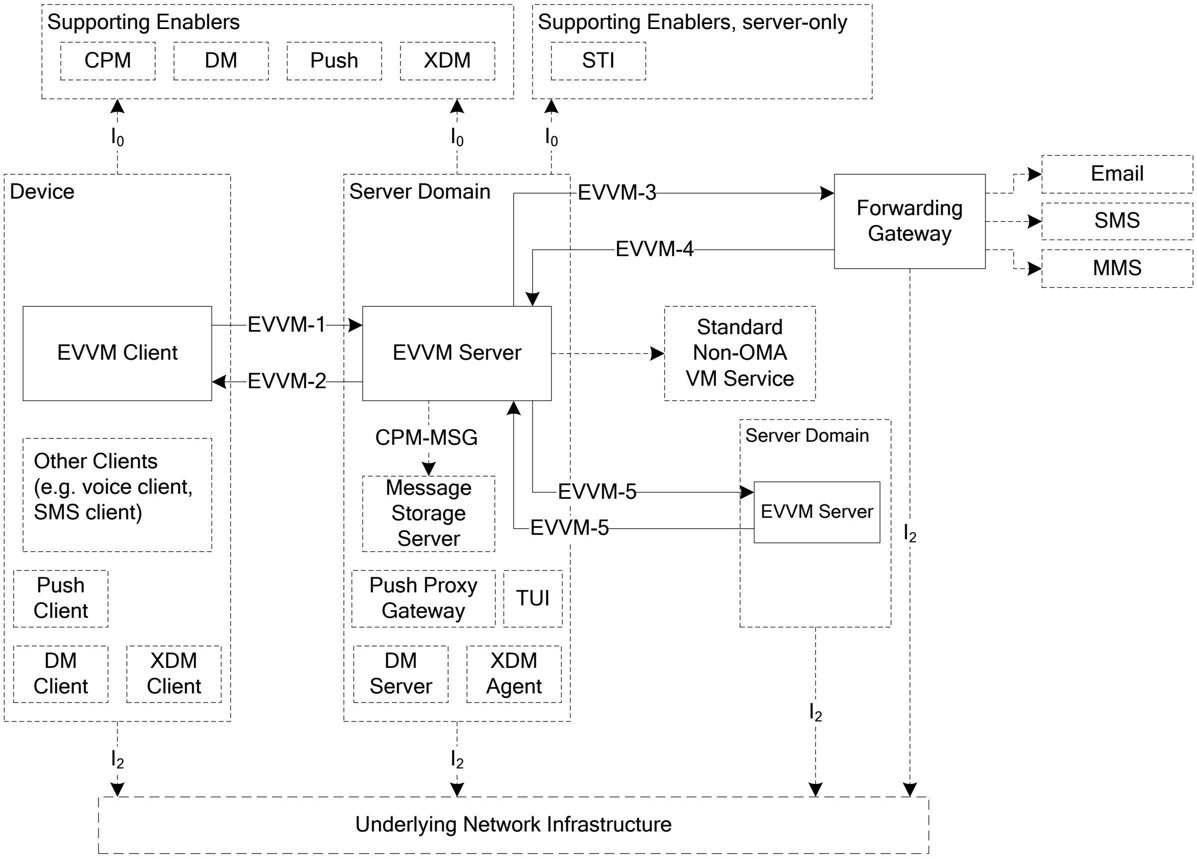

**Figure 2: OMA EVVM logical architecture**

Figure 2 presents the EVVM logical architecture model, depicting the EVVM Enabler’s functional components and their interactions with each other and with external entities such as the Supporting Enablers and the remote EVVM Server.

The EVVM Enabler consists of the following EVVM functional components:

- The EVVM Client, which resides in the user’s Device and allows the EVVM User to use the EVVM-based services by interacting with the EVVM Server and the components in the Device.
- The EVVM Server, which resides in the network’s Server Domain and interacts with the EVVM Client and other network components such as the Forwarding Gateway.
- The Message Storage Server (MSS), based on a component of the OMA CPM Enabler specified in [OMA CPM TS MS], provides storage functions for EVVM user’s voicemails, delivery/read reports, greetings and voice signatures including their attachments (e.g., audio, display of logos, advertisements).
    - Note: When the EVVM Server provides storage only internally, the Message Storage Server and the CPM-MSG interface exposed by the MSS are omitted.
- The Forwarding Gateway, which forwards voicemails to external non-VM services such as Email, MMS and SMS.
- The Telephony User Interface (TUI), which allows the user to access his/her EVVM VM box via traditional telephony systems.

The EVVM functional components, if and when needed, interact with the following external functional components:

- The following supporting Enablers, which are other OMA Enablers used to support the EVVM Enabler:

- The CPM Enabler (MSS), which provides external message storage capabilities to support EVVM voicemail storage functionalities.
- The XDM Enabler (XDM Client and Agent), which provides XML document management capabilities to support creating, storing and managing EVVM user preferences, group lists for black-lists to block incoming voicemails.
- The Push Enabler (Push Client and Push Proxy Gateway), which provides push notifications to support informing multiple devices about server-side events.
- The DM Enabler (DM Client and DM Server), which provides device management to support remote management of EVVM-specific device parameters.

- The remote EVVM Server, which is another EVVM server residing in another (remote) server domain.
- The Underlying Network Infrastructure, which provides Telephony-based and IP-based functionalities that are needed for the EVVM Enabler to provide the required services.

Figure 3 shows the details of the interactions between the functional components of the EVVM Enabler and the supporting Enablers. A more detailed informational figure is available in section D.1.1 "*Detailed Enabler Interactions*".

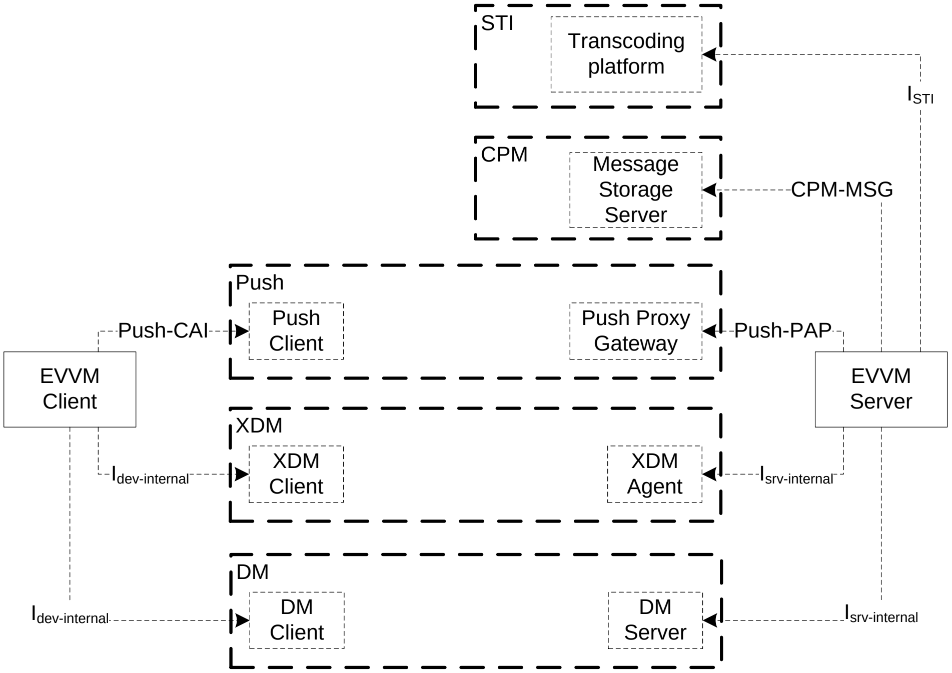

**Figure 3 - Interactions between the EVVM Enabler functional components and the supporting Enablers.**

Interfaces labeled as $I_{srv-internal}$ are implementation-specific interfaces that are internal to the EVVM Server and the relevant functional component. All $I_{srv-internal}$ interfaces are out of the scope of EVVM 1.0.

Interfaces labeled as $I_{dev-internal}$ are implementation-specific interfaces that are internal to the EVVM Client and the relevant functional component. All $I_{dev-internal}$ are out of the scope of EVVM 1.0.

# 6.3 Functional Component and Interface definitions

## 6.3.1 Functional Components

### 6.3.1.1 EVVM Client

The EVVM Client resides in a Device. It is used to access network-based EVVM functional components and to manage voicemails, greetings and user preferences. User can use multiple EVVM Clients (which may reside on multiple devices) to access EVVM-based services.

The EVVM Client is responsible for:

- Establishing connections as necessary while the user provisioning status is active, authenticating the user, maintaining and closing established connections as necessary:
    - Providing information for authentication between the EVVM User and the EVVM Server prior to accessing the EVVM-based service.
    - Coping with intermittent connectivity or bearer-switching-caused interruptions (including suspending and resuming).
- Client capability, service and user preferences, management via the XDM Enabler [6.3.1.9.4 "*XML Document Management Enabler*"]
- Folder management:
    - Selecting the current folder.
- Voicemail handling:
    - Creating a voicemail from:
        - voice,
        - text,
        - references to voicemails stored in the EVVM Voice Mail Storage Server.
    - Requesting to send a new or a stored voicemail to the recipient(s)
    - Requesting future voicemail release
    - Adding emotion indication (e.g., happy, sad, angry) to outgoing voicemails
    - Requesting to forward a voicemail or part(s) of it to the recipient(s)
    - Requesting to forward a transcription of the voicemail or parts of it to the recipient(s)
    - Requesting to forward voicemail to SMS, MMS and/or Email recipient(s) , optionally including a request for delivery or read report
    - Supporting international email addresses.
    - Requesting delivery and read report
- Voicemail management:
    - Creating a voicemail from voice or text.
    - Obtaining a list of voicemail(s) from a folder such as inbox or spambox.
    - Fetching the blocked information and informing EVVM Server to unblock a previously blocked user.
    - Deleting a voicemail.
    - Retrieving voicemail with emotion indication
    - Retrieving a voicemail as

        - voicemail only
        - transcribed text only
        - voicemail and transcribed text together
- Greeting and voice signature management, based on [GSMA VVM]:
    - Creating greetings and voice signatures
    - Uploading greetings and voice signatures.
    - Obtaining list of greetings and voice signatures.
    - Retrieving greetings and voice signatures.
    - Deleting greetings and voice signatures.
    - Replacing existing greetings and voice signatures.
- Content adaptation:
    - Requesting on-demand text to speech conversion.
    - Requesting on-demand voicemail transcription.
    - Starting/stopping automatic voicemail transcription.
- Notifications:
    - Supporting VVM 1.3 notifications for backward compatibility
    - Receiving and consuming notifications including the emotion indication if present.
    - Receiving network-initiated deactivation notification.
- Support for multiple identifiers
    - Authenticating using any one of the identifiers
    - Performing operations using selected identifiers
- Spam Management:
    - Reporting voicemail as a spam.
    - Informing EVVM Server that a previously reported spam is no longer a spam.
    - Managing a set of parameters that will be used to identify a voicemail as spam.

#### 6.3.1.2 EVVM Server

The EVVM Server resides in a Server Domain. It stores resources and provides access to these resources to other EVVM functional components.

A remote EVVM Server is an EVVM Server residing in another (remote) EVVM Server Domain and supports the full set or a subset of the EVVM functions described in this document. The local EVVM Server interacts with the remote EVVM Servers to allow EVVM Users of the local EVVM Server Domain to interact with EVVM Users of the other (remote) EVVM Server Domain.

The EVVM Server SHALL provide storage either internally, externally or both, see section D.3 “*Deployment Options*”. The interface between the EVVM Server and its internal storage is an implementation issue, and as such, it is out of the scope of this specification. The interface between the EVVM Server and its external storage SHALL be CPM-MSG, as defined in [OMA CPM TS MS], see section 7.7.9 “*Managing the external storage*”.

The EVVM Server SHALL be capable of transcoding content using the OMA STI Enabler (see section 6.3.1.9.5 “*STI (Standard Transcoding Interface)*”).

The EVVM Server is responsible for:

- Serving connection requests, authenticating the EVVM user, maintaining and closing established connections as requested.
- Providing internal storage for voicemails, delivery/read reports, greetings and voice signatures
- Interacting with external storage for voicemails, delivery/read reports, greetings and voice signatures.
- Supporting the storage of the following user preferences (either in external or in internal storage, not both):
    - Preferred media formats.
- Processing and validation of requests originated from the EVVM Client and the TUI.
- Applying the device/client capabilities, user preferences and service provider policies to requests from the EVVM Client and the TUI, as applicable.
- Disclosing the server capabilities to the client:
    - List of supported media formats
- Performing requested operations, if permitted:
    - Client and server capabilities, service and user preferences management via the XDM Enabler [6.3.1.9.4 "*XML Document Management Enabler*"]
    - Voicemail handling:
        - Fetching, from the EVVM Voice Mail Storage Server, the referenced media and replacing the references included in the voicemail.
        - Sending voicemail to the recipient(s) upon receiving a request
            - If present, including emotion indication.
        - Sending voicemail at the future time specified by the EVVM Client.
        - Delivering voicemail with appropriate codec
            - If needed, performing on-the-fly transcription to the preferred media format(s).
            - If roaming, respecting delivery with roaming preference.
        - Receiving voicemail:
            - Rejecting with respect to the white and black lists
            - Accepting with respect to the white and black lists
            - Recording an incoming voicemail
            - Storing the recorded voicemail to the Voice Mail Storage Server
        - Storing incoming voicemail(s)
        - Inserting images as attachments to a voicemail (e.g., display of logos, advertisements) based on service provider's policy and with prior consent of the VM receiving user.
        - Forwarding an already stored voicemail to the recipient(s) respecting white list/black list
        - Sending Delivery/Read Status information according to user preference.
        - Supporting international email addresses.
    - Support for multiple identifiers
        - Allowing associating multiple identifiers with one specific user
        - Authenticating the user using his/her selected identifier
        - Performing operations using selected EVVM user identifier
        - Allowing using all associated identifiers during an authenticated/authorized session

        - Delivery and/or read report handling:
            - Receiving a read report, indicating when the corresponding voicemail, or the voicemail forwarded as MMS/Email, was read by the recipient.
            - Receiving a delivery report, indicating when the corresponding voicemail, or the voicemail forwarded as SMS/MMS/Email, was properly delivered to its destination.
            - Notifying the EVVM Client of a received read and/or delivery report, according to user preferences.
        - Greeting and voice signature management:
            - Storing greetings and voice signatures.
            - Playing the appropriate greetings.
        - Content adaptation:
            - Realizing voicemail transcription according to user preference.
            - Providing transcribed text to the client.
            - Realizing text to speech conversion according to user preference.
            - Providing converted voice to the client.
        - Notifications:
            - Supporting VVM 1.3 notifications
            - Dispatching event notifications or not (according to user preferences)
                - If roaming, respecting notification setting while roaming preference and preferred roaming notification type.
- Spam Management:
    - Identifying incoming voicemails as spam and storing them in the spam folder.
    - Processing spam report received from EVVM Client.
    - Handling the spam voicemail according to the spam report, user preference and service provider policies
    - Receiving and processing information from the EVVM Client that a previously reported spam is no longer a spam
- Reporting the result of the requested operation to the EVVM Client and the TUI, as applicable.
- Responding to the EVVM Client's request to retrieve the blocked information.
- Receiving request from the EVVM Client to unblock a user that was previously blocked from leaving a voicemail.
- Forwarding transcribed text to the recipient(s).
- Sending notification to the appropriate EVVM Client(s) that the EVVM User is no longer provisioned for EVVM services i.e. network initiated deactivation.
- Supporting EVVM operations in multi-device environment.
- Inter-domain communication with remote EVVM Servers, to transfer:
    - Voicemails.
    - Reports (delivery reports, read reports).
- Interworking with Standard non-OMA VM Service according to service provider's policies:
    - Identifying voicemail's destination as a Standard non-OMA VM service and forwarding the voicemail to its destination.

    - Identifying destination of report (e.g., delivery report, read report) as a Standard non-OMA VM service and forwarding the report to its destination.
    - Receiving voicemails from Standard non-OMA VM Service;
    - Receiving reports (delivery reports, read reports) from Standard non-OMA VM Service

#### 6.3.1.3 TUI

The TUI resides in the Server Domain. It provides a voice-based interface to allow the user to access his/her voicemails via traditional telephony systems.

The TUI is responsible for:

- Serving voice call requests, authenticating the user, maintaining and closing established voice calls.
- TUI-specific features:
    - Changing user password.
    - Changing user language.
    - NUT.
- Voicemail management:
    - Obtaining list of voicemail(s).
    - Listening to voicemail(s).
    - Deleting voicemail(s).
- Greeting and voice signature management:
    - Recording greetings and voice signatures.
    - Obtaining the list of greetings and voice signatures.
    - Listening to greetings and voice signatures.
    - Deleting greetings and voice signatures.
- Voice call handling:
    - Playing the appropriate greeting and voice signature.
    - Recording voicemail(s).

#### 6.3.1.4 Message Storage Server

Providing the same functionality as the CPM Message Storage Server (MSS), the EVVM Enabler’s external voicemail storage, if present, is a stand-alone network-based server, which provides all functionalities required for the storage of the EVVM user’s voicemails, delivery/read reports, greetings and voice signatures and their attachments.

The detailed functionalities of the MSS are as described for the OMA CPM MSS in [OMA-CPM-TS SD].

#### 6.3.1.5 Forwarding Gateway

The Forwarding Gateway provides capabilities to forward voicemails as Email, MMS and SMS.

The Forwarding Gateway is responsible for:

- Handling a request to forward a voicemail into a MMS, Email or SMS:
    - Converting the voicemail into SMS/MMS/Email:
        - Copying and adapting the content parts to be forwarded into a new MMS, Email or SMS,

        - Setting the originator address as specified by user preferences or service provider policies (an address resolution might be needed but is not specified in this specification),
        - If requested, including a request for a read or delivery report,
    - Sending the MMS, Email, or SMS to the entity providing MMS, Email, or SMS functionality respectively.
- Delivery and read report handling:
    - Receiving the read or delivery report from the entity providing MMS, Email, or SMS functionality,
    - Identifying the voicemail corresponding to the received read or delivery report,
    - Sending a read or delivery report to the EVVM Server.

#### 6.3.1.6 Device

The Device is out of the scope of this specification and it is used for illustration purposes only. A general description is available in [OMADICT]. In the context of OMA EVVM, the Device is expected to host a variety of clients, including the EVVM Client. The Device is responsible for providing the means for:

- the EVVM Client to interact with the user,
- the hosted clients to communicate with each other internally,
- the hosted clients to communicate with their server-side counter-parts using the underlying network infrastructure.

#### 6.3.1.7 Other Clients

Other Clients are the clients residing in user device, e.g., voice client, SMS client and Push client. The EVVM Client can interact with those clients to provide EVVM-based service.

#### 6.3.1.8 Server Domain

The Server Domain is a collection of the EVVM network based functionality that supports communicating with the EVVM Client, Forwarding Gateway, Remote EVVM Server and other OMA supporting Enablers.

#### 6.3.1.9 Supporting Enablers

##### 6.3.1.9.1 Converged IP Messaging Enabler

The Converged IP Messaging (CPM) Enabler defines a framework to provide IP based real-time and non-real-time multimedia communications. The OMA CPM Enabler specification defines the Message Storage Server (MSS) in [OMA CPM TS MS]. The MSS allows storing various types of objects, including voicemails. OMA EVVM Enabler supports using the MSS component of OMA CPM Enabler for external storage, as described in section 6.3.1.2 "*EVVM Server*".

##### 6.3.1.9.2 Device management

The Device Management (DM) Enabler [OMA DM] defines how configuration parameters can be delivered to an OMA client from an OMA DM server that is part of the overall system architecture. OMA EVVM Enabler supports using the DM Enabler to deliver configuration parameters to the OMA EVVM Client.

##### 6.3.1.9.3 Push Enabler

The Push Enabler [OMA Push] allows Push Initiators and Application Servers to initiate service-related transactions and content delivery to user devices. OMA EVVM Enabler supports using the Push Enabler to push notifications from the EVVM Server to the EVVM Client.

#### 6.3.1.9.4 XML Document Management Enabler

The XDM Enabler defines a common mechanism that makes service-related information accessible to the Enablers that need it. The XDM Enabler specifies how such information is represented in XML documents. It uses a common protocol for access and manipulation (e.g. create, modify, delete, etc.) of such XML documents.

The XDM Enabler allows:

- the EVVM Clients to create, store and manage client-specific service-related information.
- the EVVM servers to create, store and manage server-specific service-related information.
- the EVVM users to create, store and manage user-specific service-related information.

It provides XDM management operations for:

- Client preference management:
  - Obtaining list of client preferences including:
    - Client type.
    - Host device type.
    - List of supported media formats (supplementary and optional only).
    - List of supported notification encryption algorithms.
    - List of supported protocol version(s).
    - List of local supporting application capabilities (e.g., whether supporting SMS, PUSH, Device Management).
    - Notification security settings.
  - Retrieving client preferences.
  - Deleting/clearing client preferences (EVVM Client only).
  - Setting/updating client preferences (EVVM Client only).
- Service preference management:
  - Obtaining list of service preferences including:
    - Service parameters:
      - Server access parameters (host name, port).
      - Service name.
      - Subscriber User-ID.
  - Server capability management:
    - Mailbox name map (mapping storage-level names to logical names)
    - List of supported media formats (supplementary and optional only).
    - List of supported encryption methods.
    - List of supported notification encryption algorithms.
    - List of supported protocol version(s).
    - List of supported notification types
    - List of supported forwarding capabilities
    - List of identifiers associated with this voicemail box.
    - Content adaptation

            - Text to speech available
            - Voice to text available
        - Maximum VM size (including headers) allowed.
        - Tel-URI for TUI
    - Retrieving service preferences.
    - Deleting/clearing service preferences (EVVM Server only).
    - Setting/updating service preferences (EVVM Server only).
- User preference management:
    - Obtaining list of user preferences including:
        - Mailbox name map (mapping display names to storage-level names)
        - Message archiving (sending and receiving)
        - Preferred media format
        - Black list(s)
        - White list(s)
        - Notification on/off while in home network, roaming or roaming internationally
        - Delivery status information (sending and receiving)
        - Content adaptation
            - Text to speech
            - Voice to text
            - Include original media
        - EVVM password
        - TUI password
        - TUI language
    - Retrieving user preferences.
    - Deleting/clearing user preferences (EVVM Client only).
    - Setting/updating user preferences (EVVM Client only).

#### 6.3.1.9.5 STI (Standard Transcoding Interface)

The STI Enabler supports the procedure for transcoding different media types and describes protocols between application platform and server how to do this.

The EVVM Server MAY use the STI Enabler to transcode different media types.

### 6.3.1.10 Standard non-OMA VM Service

A Standard non-OMA VM Service is a voicemail service. The EVVM Enabler interacts with the Standard non-OMA VM Service to allow EVVM Users to interact with users of Standard non-OMA VM Service.

### 6.3.1.11 Email

The Email component provides capabilities to send and receive emails, and deliver read and/or delivery reports.

### 6.3.1.12 MMS

The MMS component provides capabilities to send and receive MMS, and deliver read and/or delivery reports.

#### 6.3.1.13 SMS

The SMS component provides capabilities to send and receive SMS, and deliver delivery reports.

#### 6.3.1.14 Underlying Infrastructure

The Underlying Network Infrastructure provides IP-based functionalities that are essential for the EVVM Enabler's functional entities to perform.   It provides functions such as routing, IP signalling and address resolution.

Any interface exposed by the Underlying Network Infrastructure and used by the EVVM functional entities is viewed as an I2 interface in the context of the OMA Service Environment (OSE) as described in [OMA-OSE].

## 6.3.2 Interfaces

### 6.3.2.1 EVVM-1

The EVVM Server is exposing the EVVM-1 interface to the EVVM Client, allowing it to access the resources on the EVVM Server.

Supported functionalities include:

- VVM 1.3 client originated notifications [GSMA VVM]
- EVVM User authentication.
  - Supporting multiple devices environment
- Client identification
- Disclosing capabilities of the EVVM Client:
  - Supported media formats.
  - Supported encryption methods.
- Sending, requesting forwarding, listing, retrieving, and deleting voicemails while providing support for:
  - Requesting voicemail release at a specific point of time in the future.
  - International email addresses.
  - Emotion indication in voicemails.
  - Indications on how voicemails should be received (see section 6.3.1.1 "*EVVM Client*").
  - Transcribed text which the EVVM Client sends for forwarding purposes.
  - One or more recipient's addresses to forward the transcribed text
  - Type of message (i.e., SMS, MMS or Email) to forward a voicemail as.
  - Request a previously blocked user to be unblocked.
  - Request to retrieve the blocked information.
- Uploading, listing, retrieving, and deleting greetings and voice signatures, based on [GSMA VVM].
- Requesting voicemail transcription:
  - On demand.
  - Automatically with every voicemail.
- Spam Management:
  - Reporting voicemail as spam.
  - Supporting to communicate a set of parameters that will be used to identify a voicemail as spam.

  - Informing that a previously reported spam is no longer a spam

#### 6.3.2.2 EVVM-2

The EVVM Client is exposing the EVVM-2 interface to the EVVM Server, allowing it to send notifications to the EVVM Client.

Supported functionalities include:

- VVM 1.3 system originated notifications [GSMA VVM]
- Sending notifications:
  - Including emotion indication, if available.
  - Encrypted as applicable
  - Sending network-initiated deactivation notification.

#### 6.3.2.3 EVVM-3

The EVVM-3 interface is exposed by the Forwarding Gateway to allow the EVVM Server to access the functions provided by Forwarding Gateway.

Supported functionalities include:

- Requesting to send voicemail as SMS, MMS or Email to the recipient(s), with or without delivery and/or read report request(s)

#### 6.3.2.4 EVVM-4

The EVVM-4 interface is exposed by the EVVM Server to allow the Forwarding Gateway to access the functions provided by EVVM Server.

Supported functionalities include:

- Request sending delivery or read reports corresponding to a previously forwarded voicemail.

#### 6.3.2.5 EVVM-5

The EVVM Server exposes the EVVM-5 interface to the remote EVVM Servers, allowing the remote EVVM Servers to interact with it. With interface EVVM-5, EVVM Users belonging to different Server Domains can interact with each other.

Supported functionalities of EVVM-5 include relaying:

- Voicemails.
- Report (delivery reports, read reports).

#### 6.3.2.6 CPM-MSG

The CPM-MSG interface is defined in [OMA-CPM].

## 6.4 Security Considerations

The EVVM Enabler provides the following security aspects:

- Authentication,
- Security of voicemail contents,
- Security of notification content,
- Security of voicemail interworking.

### 6.4.1 Authentication

EVVM supports associating multiple identifiers with the user’s voicemail box. The EVVM user SHALL be authenticated by EVVM Server using any one of these identifiers. Once one of his/her identifiers is authenticated, all the other associated identifiers SHALL be considered as authenticated.

### 6.4.2 Security of Voice Mail Contents

The EVVM Enabler ensures the security of EVVM voicemail contents as a supplement to the security mechanism of VVM 1.3.

Regarding voicemail contents, the EVVM Enabler SHALL support the following security-related functionalities:

- Encrypting all voicemail contents or individual contents for confidentiality,
- Preserving the integrity of the voicemail,
- Ensuring the non-repudiation for the voicemail.

### 6.4.3 Security of notification content

For security of notification content, the EVVM Client is responsible for:

- Requesting using encrypted notifications or clear text notifications;
- Exchanging private keys, public keys, etc as required for encrypting notifications;
- Decrypting encrypted notifications;

and the EVVM Server is responsible for:

- Selecting a suitable encryption method while taking into account client capabilities.
- Exchanging private keys, public keys, etc as required for encrypting notifications.
- If requested, encrypting notifications to be sent to the EVVM Client.

### 6.4.4 Security of voicemail interworking

While forwarding a voicemail as a MMS/EMAIL/SMS message, the EVVM Enabler SHOULD transfer the data with confidentiality and/or integrity, if present, to the non-EVVM services if the non-EVVM services support the data confidentiality and/or integrity also. See section 7.8 “*Procedures at the Forwarding Gateway*” for more details.

# 7. Technical Specification

This section contains the detailed technical specification of the OMA EVVM Enabler. This version of the EVVM specifications is backward compatible with VVM 1.3 Specifications [GSMA VVM] with the clarifications given in section 7.1 “*Versioning*”.

## 7.1 Versioning

The VVM 1.3 specification [GSMA VVM] was prepared to ensure that the standard functionality of voicemail servers can be accessed using a wide variety of voicemail clients via a well-defined interface. By focusing on the client-server interface, the scope of the service in VVM 1.3 was left open for operators and vendors, allowing differentiating their products.

The EVVM 1.0 specification focuses on the behavior of the EVVM Client, EVVM Server and other EVVM functional entities, while maintaining the possibility for differentiation of the final EVVM-based services. For this purpose, this specification version addresses several notions and procedures as follows:

- EVVM data models for various data elements used during the composition, exchange and management of a voicemail and its attachments, e.g., identifiers, user preferences, voicemails, greetings, notification messages.
- EVVM user/Client Registration (session login) with the EVVM Server including authentication and handling and exchanges of information such as client and server capabilities.
- EVVM Client and EVVM Server procedures either specific to each or jointly carried out:
    - User authentication,
    - Client/Server session establishment,
    - Voicemail deposit and retrieval procedures or originating (outgoing) and terminating (incoming) voicemails, respectively,
    - Voicemail spam handling,
    - Decision on forwarding to or interworking with Non-EVVM Services,
    - Notifications for newly arrived voicemails, sent voicemail delivery and read report, etc.
- EVVM gateway procedures specifying support for interworking with non-EVVM services (e.g., SMS, MMS & email).
- EVVM stand-alone VM storage procedures, e.g., storing voicemails, creating new folders, managing stored objects and folders.
- Procedures related to the use of XDM for user’s preferences profiles for various lists, e.g., white, black.

Not included in this version are specifications for interactions between the EVVM Client and other clients within the device and fulfilling requirements tagged as “Future” in the EVVM Requirements document [OMA EVVM RD].

The scope of the specified backward compatibility with VVM 1.3 is limited to:

- A VVM 1.3 Client can interact with an EVVM 1.0 Server.
- An EVVM 1.0 Client with SMS capability can interact with a VVM 1.3 Server.

## 7.2 Protocol transport bindings

This section determines and describes the protocols for each EVVM interface and their transport bindings.

This specification supports clients based on the GSMA VVM [GSMA VVM] specification. EVVM Servers SHALL support the GSMA VVM specification. All EVVM Clients and EVVM Servers SHALL support the transport bindings defined in this section.

### 7.2.1 IMAP

The IMAP4 protocol, as defined in [RFC3501], is bound to the EVVM-1 interface. The EVVM Client SHALL establish IMAP connections using the host name and the TCP port specified by the service provider in the service preferences (serviceprefs-final/prefs/serviceparameters/imap). The EVVM Server SHALL accept IMAP connections via TCP port specified by the service provider in the service preferences (serviceprefs-final/prefs/serviceparameters/imap).

The EVVM Client can establish a single IMAP4 connection with the EVVM Server for all associated identifier(s), and based on this IMAP4 connection, the EVVM Client SHALL perform the authentication procedures with the EVVM Server, as specified in section 7.6.1 "*Authentication*".

To secure the IMAP communications channel, EVVM Servers SHALL support TLS 1.0, as specified in [RFC2595].

This specification does not rely on the capability command defined in [RFC3501] to indicate server-side support for OMA EVVM.

### 7.2.2 SMTP

The SMTP protocol, as defined in [RFC5321], is bound to the EVVM-1, EVVM-3, EVVM-4 and EVVM-5 interfaces.

Used for the EVVM voicemail submission, the protocol for the EVVM-1 interface is SMTP. To secure the SMTP communications channel, for EVVM-1 interface, the EVVM Server SHALL support TLS 1.0, as specified in [RFC3207].

SMTP is the protocol for voicemail submission over EVVM-3 for forwarding to the Forwarding Gateway, and the protocol for report (including read-report and delivery-report) submission over EVVM-4 for forwarding to EVVM Server.

SMTP is the protocol for voicemail submission and report (including read-report and delivery-report) submission over EVVM-5 interface among the EVVM Servers in different Server Domains.

The EVVM Client SHALL establish EVVM-1 SMTP connections using the host name and the TCP port specified by the service provider in the service preferences (serviceprefs-final/prefs/serviceparameters/smtp).

The EVVM Server SHALL accept EVVM-1 SMTP connections via TCP port specified by the service provider in the service preferences (serviceprefs-final/prefs/serviceparameters/smtp).

The Forwarding Gateway SHALL accept SMTP connections via a configurated TCP port.

The EVVM Server SHALL accept SMTP connections via a configurated TCP port.

### 7.2.3 SMS

Supporting the EVVM Enabler's backward compatibility to VVM [GSMA VVM], the SMS messaging is bound to both the EVVM-1 and EVVM-2 interfaces. The OMA EVVM Enabler relies on the OMA Push Enabler [OMA Push] for notifications. See section 7.4 "*Introduction to notifications*".

## 7.3 Data Model

This section describes the header/data/payload formats and encodings that EVVM uses:

- Identifiers in section 7.3.1 "*Identifiers*".

- Preferences in section 7.3.2 "*Preferences*".
- Media Format in section 7.3.3 "*Media Format*".
- Media types in section 7.3.4 "*Media types*".
- Mailboxes in section 7.3.5 "*Mailboxes*".
- Notifications in section 7.3.6 "*Notifications*".
- Extended Server Capabilities in section 7.3.7 "*Extended Server Capabilities*".
- EVVM Specific Headers in section 7.3.8 "*EVVM Specific Headers*".

## 7.3.1 Identifiers

This section describes the formats of various identifiers used for EVVM-based services.

### 7.3.1.1 User Identifier

An OMA EVVM user's EVVM voicemail (VM) box SHALL be identified by a user identifier. Each user identifier SHALL be globally unique. Each EVVM user identifier SHALL belong to a unique EVVM VM box. An EVVM user MAY have multiple user identifiers associated with his/her EVVM VM box.

A user identifier SHALL be either

- An email address, or
- A TEL URI as specified in [RFC3966], or
- A SIP URI as specified in [RFC3261].

The EVVM Server MAY support associating any of an EVVM user's identifiers (as defined in serviceprefs-subscription/prefs/associations) with a voicemail box. Each identifier can only be associated with one and only voicemail box. Association of identifiers is performed by the service provider. How an EVVM user can request the association of his/her identifiers is out of the scope of this specification. However, the EVVM Server SHALL expose the user identifiers associated with the EVVM voicemail subscription in the subscription-specific service preferences (serviceprefs-subscription/prefs/associations).

### 7.3.1.2 Client Identifier

An EVVM user may possess several devices and each of these devices may host multiple EVVM Clients from the same or different vendors. The EVVM Client identifier (Client-ID) identifies an individual EVVM Client within an individual device, therefore the Client-ID SHALL be globally unique.

In general, a Client-ID is a freeform, case insensitive string; however it SHALL NOT include characters that are not permitted in SIP URIs [RFC3241].

EVVM Clients without a factory-assigned Client-ID SHALL generate a globally unique Client-ID before their first login attempt as the Client-ID is required []. It is RECOMMENDED to generate the Client-ID to be a Universally Unique Identifier (UUID), as defined in [RFC4122].

The EVVM Client SHALL store its Client-ID persistently, to overcome power loss, device failure, firmware update, software update, etc.

## 7.3.2 Preferences

Preferences are the common name for EVVM-related information, such as settings, capabilities and policies. The OMA EVVM Enabler organizes all preferences into XML documents and uses the XML Document Management Enabler [OMA XDM] to manage these XML documents. Therefore, the EVVM Client shall have access to an XDM Client residing in the device. Similarly, the EVVM Server shall have access to an XDM Agent residing in the Server Domain. The interactions between the EVVM Client and the XDM Client in the Device as well as the interactions between the EVVM Server and the XDM Agent in the Server Domain are out of the scope of this specification.

OMA EVVM defines three sets of preferences, based on whose preferences we refer to:

- Service preferences, in section 7.3.2.1 “*Service preferences*”.
- Client preferences, in section 7.3.2.2 “*Client preferences*”.
- User preferences, in section 7.3.2.3 “*User preferences*”.

The corresponding document structures, authorization policies, validation constraints are defined in [OMA EVVM XDM]. Consequently, all EVVM Clients and EVVM Servers SHALL adhere to the requirements defined for the corresponding application usages in [OMA EVVM XDM].

When the EVVM Server is started, it SHOULD retrieve a fresh copy of client and user preferences from the XDMS rather than relying on possibly incorrect cached information. The EVVM Server SHALL register with the XDM Agent residing in the Service Domain to ensure that it receives notifications from the XDMS about changes in XML documents that contain client or user preference updates. When the EVVM Server receives a notification from the XDMS about changes in the client or user preferences, it SHALL retrieve the appropriate XML document(s) and apply the new preferences within. Once new user preferences have been applied, the EVVM Server SHALL send a PREF notification to all of the user’s EVVM Clients. Once new client preferences have been applied, the EVVM Server SHALL send a PREF notification only to the EVVM Client whose client preferences have been updated.

When the EVVM Client is started, it SHOULD retrieve a fresh copy of service and user preferences from the XDMS rather than relying on possibly incorrect cached information. The EVVM Client SHALL register with the XDM Client residing in the Device to ensure that it receives notifications from the XDMS about changes in XML documents that contain service or user preference updates. When the EVVM Client receives a notification from the XDMS about changes in service preferences, it SHALL retrieve the appropriate XML document and apply the new preferences immediately. When the EVVM Client receives a notification from XDM Server through XDM Client about changes in user preferences, it SHALL retrieve the appropriate XML document immediately. The EVVM Client SHOULD apply the new user preferences after it has received the PREF notification from the EVVM Server.

#### 7.3.2.1 Service preferences

Service preferences are managed by the EVVM Server. The service preferences are interpreted by the EVVM Client.

There are two types of service preferences: global service preferences and subscription-specific service preferences. Global service preferences include information about the general settings imposed onto all EVVM users and their EVVM Clients by the EVVM Server or the service provider. Subscription-specific service preferences include information about the service settings imposed onto individual subscribers’ and their clients by the EVVM Server or service provider. Existing subscription-specific service preferences override the corresponding global preferences as follows: the EVVM Client and the EVVM Server SHALL follow the global service preferences unless a subscription-specific service preference exists, in which case, the EVVM Client and the EVVM Server SHALL follow the corresponding subscription-specific service preferences instead of the global service preferences in all applicable contexts. Figure 4 expresses the calculation of the final set of preferences from global and subscription-specific preferences using set theory and Venn-diagrams.

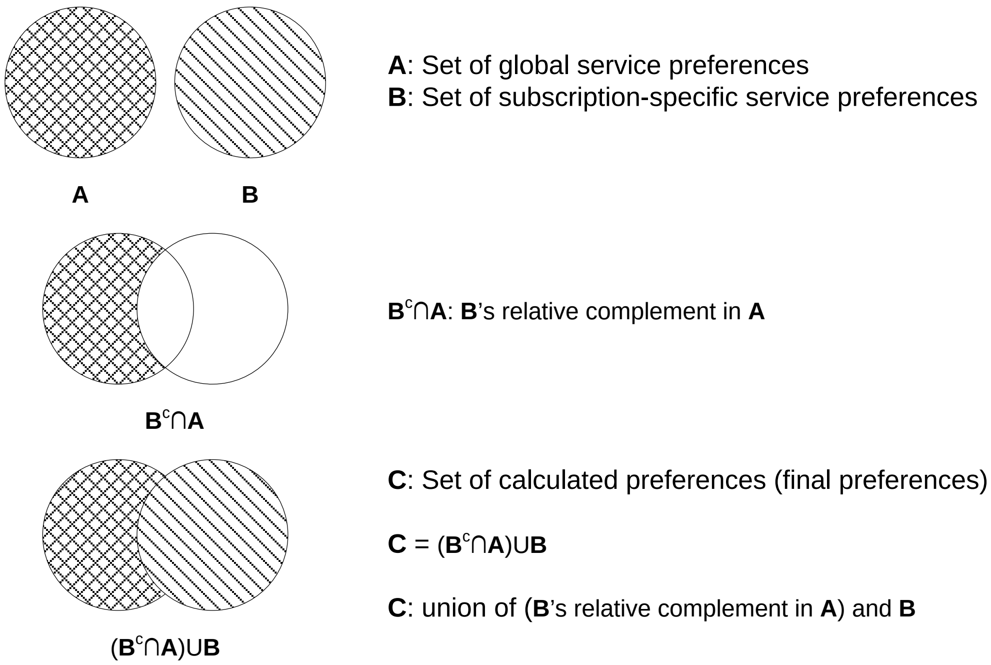
A
B
A: Set of global service preferences
B: Set of subscription-specific service preferences
$B^c \cap A$
$B^c \cap A$: B's relative complement in A
$(B^c \cap A) \cup B$
C: Set of calculated preferences (final preferences)
$C = (B^c \cap A) \cup B$
C: union of (B's relative complement in A) and B

**Figure 4: Calculating the final set of service preferences**

EVVM Servers SHALL maintain the global service preferences ensuring that they are up-to-date in the appropriate XML document. Since global service preferences are universal, one and only one XML document is available, storing all global service preferences. The XML document containing the global service preferences SHALL conform to all requirements that are defined for the “global service preferences” application usage in [OMA EVVM XDM].

EVVM Servers MAY maintain subscription-specific service preferences. The EVVM Server SHALL ensure that all available subscription-specific service preferences are up-to-date in the appropriate XML documents. Since subscription-specific service preferences may differ from user to user, one and only one XML document is available for each user to whom the preferences apply. The XML document containing the subscription-specific service preferences SHALL conform to all requirements that are defined for the “subscription-specific service preferences” application usage in [OMA EVVM XDM].

## 7.3.2.1.1 Global service preferences

This section contains the details and intended use of the global service preferences.

Preferences defined in section 7.3.2.1.1.1 “*Server capabilities*” are used by the EVVM Client during capability negotiation, see section 7.6.6.1 “*Capability agreement*”.

Preferences defined in section 7.3.2.1.1.2 “*Service limitations*” are limitations imposed by the server on users without subscription-specific service preferences (see section 7.3.2.1.2 “*Subscription-specific service preferences*”).

Preferences defined in section 7.3.2.1.1.3 “*Service parameters*” are used in service discovery; see section 7.5 “*Provisioning*”.

Existing subscription-specific service preferences override the global setting(s), as described in section 7.3.2.1 “*Service preferences*”.

### 7.3.2.1.1.1. Server capabilities

### 7.3.2.1.1.1.1. Supported media formats

The supported media formats is the list of media formats that the EVVM Server is capable to transcode and hence accepts as voicemail, greeting or voice signature. The minimum set of media types is REQUIRED, and as such, it is omitted from the list. The support for supplementary and optional media types is indicated either as a whole by setting the appropriate flag(s) to true, or, partially by setting the appropriate flag(s) to false and including the individual media types. The minimum, supplementary and optional media type sets are defined in section 7.3.4 “*Media types*”.

Corresponding preference: serviceprefs-global/prefs/mediatypes

#### 7.3.2.1.1.1.2. Supported character sets

The supported character sets is a list of character sets that the EVVM Server is capable to transcode. The minimum set of character sets is not included in the list. The support for the supplementary set is indicated either as a whole by setting the appropriate Boolean XML attribute to true, or, partially by setting the appropriate XML attribute to false and including the individual character sets. The minimum and supplementary character sets are defined in section 7.3.4.1 “*Character sets*”.

Corresponding preference: serviceprefs-global/prefs/charsets

#### 7.3.2.1.1.1.3. Supported protocol versions

The supported protocol versions are a list of supported protocols that the EVVM Server is capable to communicate with. Each protocol is represented by an empty element with two attributes. The first attribute is a string, containing the abbreviated name of the SDO, followed by a dash (‘-’) and the abbreviated name of the standard. The second attribute is a string, containing a comma-separated list of version number, indicates support for revisions of OMA EVVM.

Corresponding preference: serviceprefs-global/prefs/protocols

#### 7.3.2.1.1.1.4. Mappings

The mappings list of logical mailbox name and physical mailbox name pairs. Each pair associates a logical mailbox name with a physical mailbox name. XML unsafe strings SHALL be encoded using Base64 encoding [RFC4648]. The EVVM Server SHALL ensure that all logical mailbox names defined in section 7.3.5 “*Mailboxes*” are mapped to physical mailbox names.

The EVVM Client and the EVVM Server use this information to determine the location of special mailboxes, as described in section 7.3.5 “*Mailboxes*”.

Corresponding preference: serviceprefs-global/prefs/mappings

#### 7.3.2.1.1.1.5. Supported notification types

The supported notification types are a set of flags indicating which notification types the EVVM Server is capable of sending. Each flag corresponds to a specific notification type. When a flag is ‘true’, it indicates that the EVVM Server supports sending notifications of the corresponding type. When a flag is ‘false’, it indicates that the EVVM Server does not support sending notifications of the corresponding type.

The EVVM Client uses this information to derive notification capabilities of the EVVM Server in order to set up its own notification preferences.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 “*Outband notifications*”).

Corresponding preferences:

- serviceprefs-global/prefs/notifications@deac
- serviceprefs-global/prefs/notifications@pref
- serviceprefs-global/prefs/notifications@sync
- serviceprefs-global/prefs/notifications@tran

###### 7.3.2.1.1.1.6. Supported notification encryption algorithms

The supported notification encryption algorithms are a list of tokens indicating which encryptions the EVVM Server is capable of performing on notifications. Each token corresponds to a specific encryption algorithm. The tokens are defined in section "*6.2 Elementary Data Types*" of [OMA EVVM XDM]. The EVVM Server MAY disclose its requirements regarding minimum encryption strength, maximum encryption strength, and the preferred encryption strength individually, for each encryption algorithm.

The EVVM Client uses this information to derive notification capabilities of the EVVM Server in order to set up its own notification preferences.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 "*Outband notifications*").

Corresponding preference: serviceprefs-global/prefs/notifications/encryption

###### 7.3.2.1.1.1.7. Supported forwarding capabilities

Not covered in this specification.

###### 7.3.2.1.1.1.8. Content adaptation

The supported content adaptation capabilities are flags indicating if the EVVM Server is capable of following adaptations:

a) Voice to Text

b) Text to Speech

When the flag is 'true', it indicates that the EVVM Server supports the corresponding adaptation. When the flag is 'false', it indicates that the EVVM Server does not support the corresponding adaptation.

Corresponding preferences:

- serviceprefs-global/prefs/contentadaptation@v2t
- serviceprefs-global/prefs/contentadaptation@tts

###### 7.3.2.1.1.2. Service limitations

###### 7.3.2.1.1.2.1. Notification encryption

The notification encryption is a token indicating the service provider's policy regarding encrypting notifications. The tokens are defined in section "*6.2 Elementary Data Types*" of [OMA EVVM XDM].When the token is 'always', it indicates that the service provider forces notification encryption. When the token is 'ifneeded', it indicates that the service provider leaves the choice to the user. When the token is 'never', it indicates that the service provider forbids notification encryption as whole.

The EVVM Client uses this information to derive service provider policies in order to set up its own notification preferences.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 "*Outband notifications*").

Corresponding preference: serviceprefs-global/prefs/notifications@encrypt

###### 7.3.2.1.1.2.2. Roaming policies for notifications

The roaming policies for notifications are a set of Boolean flags indicating the service provider's policy regarding sending notifications under various roaming conditions. Each flag corresponds to a specific roaming condition. When a flag is 'true', it indicates that the service provider allows sending notifications under the corresponding roaming condition – according to the user's preferences. When a flag is 'false', it indicates that the service provider forbids sending notifications under the corresponding roaming condition – irrespective of the user's wish.

The EVVM Client uses this information to derive service provider policies in order to set up its own notification preferences.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 "*Outband notifications*").

Corresponding preferences:

- serviceprefs-global/prefs/notifications@home
- serviceprefs-global/prefs/notifications@roaming
- serviceprefs-global/prefs/notifications@intlroaming

### 7.3.2.1.1.2.3. Maximum VM size

The maximum VM size indicates the maximum total size of any voicemail message that the EVVM Server accepts for delivery. The total size includes any and all overhead, such as those imposed by transfer encoding, encryption, etc. The maximum VM size is an attribute of a generic element specifically designed to represent limitations. The size is an integer values, representing the number of bytes. The value SHALL be higher than 0.

Corresponding preference: serviceprefs-global/prefs/bounds@maxvmsize

### 7.3.2.1.1.2.4. Maximum lengths

The maximum lengths indicate the maximum length of greetings and voice signatures that the EVVM Server accepts. The maximum lengths are attributes of a generic element specifically designed to represent limitations. Each length is an integer value, representing the number of seconds. The value SHALL be 0 or higher, where 0 indicates that the EVVM Server does not permit the using the corresponding feature.

The EVVM Client SHALL NOT store greetings and voice signatures that are longer than their corresponding maximum lengths.

It is expected that the EVVM Server will provision any VVM 1.3 clients with the same limitation using a system-originated STATUS SMS, see section 7.3.6.1.1 "*VVM 1.3*".

Corresponding preferences:

- serviceprefs-global/prefs/bounds@maxgreetinglength
- serviceprefs-global/prefs/bouns@maxvoicesignaturelength

### 7.3.2.1.1.2.5. Message Storage quota

Not covered in this specification.

### 7.3.2.1.1.2.6. Password strength requirements

The password strength requirements indicate the constraints for EVVM and TUI passwords, imposed by the service provider. The EVVM Client SHALL validate new passwords versus these constraints when the EVVM user attempts to change his EVVM or TUI password via his/her preferences (see sections 7.3.2.3.2 "*EVVM password*" and 7.3.2.3.3 "*TUI settings*", respectively) and reject the attempt when the new password does not meet the minimum requirements.

Corresponding preferences:

- serviceprefs-global/prefs/bounds/evvm/password
- serviceprefs-global/prefs/bounds/tui/password

### 7.3.2.1.1.2.7. TUI limitations

The TUI limitations indicate the TUI features available to the user: the list of supported languages and a Boolean flag indicating whether the user is allowed to control the TUI status (turn it on/off). The values in the language list are used when the EVVM user attempts to change his TUI language via his/her TUI preferences (see section 7.3.2.3.3 "*TUI settings*"). Each language may have a name associated with it, which may be shown on the user interface to aid the choice of the user.

The syntax defined in OMA EVVM is based on the "Language number" defined in [GSMA VVM].

Corresponding preferences:

- serviceprefs-global/prefs/bounds/tui@usercontrol
- serviceprefs-global/prefs/bounds/tui/language

#### 7.3.2.1.1.3. Service parameters

#### 7.3.2.1.1.3.1. IMAP server

The IMAP server settings are a pair of host name (or IP address) and port number .The EVVM Client uses this information to access the EVVM Server via the EVVM-1 interface, as described in section 7.2.1 "*IMAP*".

The service provider must ensure that the IMAP server settings are identical to the corresponding settings in the DM managed object [7.5.1 "*Provisioning service parameters using OMA DM*"].

Corresponding preference: serviceprefs-global/prefs/serviceparameters/imap

#### 7.3.2.1.1.3.2. SMTP Server

The SMTP server settings are a pair of host name (or IP address) and port number. The EVVM Client uses this information to access the EVVM Server via the EVVM-1 interface, as described in section 7.2.2 "*SMTP*".

The service provider must ensure that the SMTP server settings are identical to the corresponding settings in the DM managed object [7.5.1 "*Provisioning service parameters using OMA DM*"].

Corresponding preference: serviceprefs-global/prefs/serviceparameters/smtp

#### 7.3.2.1.1.3.3. Service name

The service name SHALL be a string. The string itself is informational. It is intended to be a user-friendly hint to the end-user about the service to which these settings belong. By including tracing information, it may also be used for troubleshooting purposes to verify that the user has the correct service preferences document on the device.

The service provider must ensure that the service name is identical to the corresponding setting in the DM managed object [7.5.1 "*Provisioning service parameters using OMA DM*"].

Corresponding preference: serviceprefs-global/prefs/serviceparameters@servicename

### 7.3.2.1.2 Subscription-specific service preferences

The subscription-specific service preferences may include any global service preference (see section 7.3.2.1.1 "*Global service preferences*") as a subscription-specific service preference. This section defines only additional subscription-specific service preferences.

Existing subscription-specific service preferences override the global setting(s), as described in section 7.3.2.1 "*Service preferences*".

#### 7.3.2.1.2.1. Associated identifiers

Associated identifiers are those identifiers that have been associated with a voicemail box, and can be used during authentication (see sections 7.6.1 "*Authentication*" and 7.7.1 "*Authentication*") and message submission via SMTP or IMAP (see section 7.6.2.3.1 "*Voicemail submission via SMTP*" or 7.6.2.3.3 "*Voicemail submission via IMAP*", respectively).

The EVVM Server SHALL ensure that user identifiers follow the definition in section 7.3.1.1 "*User Identifier*".

The EVVM Server MAY choose to store this preference when there is only one identifier in use.

Corresponding preference: serviceprefs-subscription/prefs/associations

#### 7.3.2.1.2.2. TUI address

The TUI-address designates the address that the user can use to access his/her voicemail messages. When the address is not available to the user, it implies that he/she cannot use the TUI as a whole and all other TUI-related settings are to be ignored.

Since TUI is not in the scope of this specification, there are no normative requirements imposed onto the TUI.

Corresponding preference: serviceprefs-subscription/prefs/bounds/tui/address

#### 7.3.2.1.2.3. Subscriber User-ID

The subscriber User-ID designates the user identifier that corresponds to the user’s EVVM subscription and the password in the user preferences (userprefs/prefs/evvm@password). The subscriber User-ID SHALL be a string, conforming to the format defined in section 7.3.1.1 “*User Identifier*”.

When there are more than one user identifiers associated with a subscription, the EVVM Server SHALL include the subscriber User-ID in the list of associated identifiers (serviceprefs-subscription/prefs/associations).

The EVVM Server SHALL ensure that no User-ID is associated with more than one voice mailbox at a time. Therefore, when the current subscriber User-ID is associated with another voice mailbox, the EVVM Server SHALL update the subscriber User-ID to contain a User-ID that is not yet associated with any other voice mailboxes. Note: when this happens, the EVVM Server may need to update the EVVM and TUI passwords or link the voice mailbox to another set of user preferences as a whole, however such procedures are out of scope of the specification.

Corresponding preference: serviceprefs-subscription/prefs/serviceparameters@subscriber

## 7.3.2.2 Client preferences

Client preferences are managed by EVVM Clients. The client preferences are interpreted by the EVVM Server.

EVVM Clients SHALL maintain their client preferences. The EVVM Client SHALL ensure that all available client preferences are up-to-date in the appropriate XML document. Since client preferences may differ from client to client, one and only one XML document is available for each EVVM Client to which these preferences apply. Each of those XML documents SHALL correspond to a single EVVM Client identified by its Client-ID (defined in section 7.3.1.2 “*Client Identifier*”). The XML document containing the client preferences SHALL conform to all requirements that are defined for the “client preferences” application usage in [OMA EVVM XDM].

The EVVM Client SHALL use the same set of client preferences throughout its lifetime. An administrative entity MAY remove those client preferences documents from the XDMS that have not been used or managed (retrieved or stored by the EVVM Server or the EVVM Client) for an operator administered day-range such as 180 consecutive days.

### 7.3.2.2.1 Client type

The client type is a string, indicating the type of the client that stored the client preferences.

There are no normative requirements derived from this client preference. The EVVM Server uses this information to determine that the client is compliant with the service. The EVVM Client uses this information to determine the software version of the client that stored the client preferences, so it can amend the setting correctly – if needed – after a software upgrade.

Corresponding preference: clientprefs/client

### 7.3.2.2.2 Host device type

The host device type is a string, indicating the type of the device that hosts the EVVM Client.

There are no normative requirements derived from this client preference. The EVVM Server uses this information to determine basic capabilities of the device.

Corresponding preference: clientprefs/device

#### 7.3.2.2.3 List of supported media formats

The supported media formats is the list of media formats that the EVVM Client is capable to process and present. The EVVM Server uses this information to determine the capabilities of the EVVM Client (see section 7.7.6.1 “*Capability agreement*”).

The format is defined in section 7.3.2.1.1.1.1 “*Supported media formats*”.

Corresponding preference: clientprefs/mediatypes

#### 7.3.2.2.4 List of supported character sets

The supported character sets is a list of character sets that the EVVM Client is capable to process and present.

The format is defined in section 7.3.2.1.1.1.2 “*Supported character sets*”.

Corresponding preference: clientprefs/charsets

#### 7.3.2.2.5 List of supported protocol versions

The supported protocol versions is a list of supported protocols that the EVVM Client is capable to communicate with.

The format is defined in section 7.3.2.1.1.1.3 “*Supported protocol versions*”.

Corresponding preference: clientprefs/protocols

#### 7.3.2.2.6 Device address

The device address is an address indicating the destination for all outband notifications. The address SHALL be a string according to the definition of ‘device address’ in section “5.3.2.1.3.1 Push Client Addressing” of [OMA Push AD]. The address SHALL refer to the network location of the device hosting the EVVM Client that stored the client preferences. The EVVM Client SHALL be reachable using this address and the EVVM Push application ID defined in section 8.2.1 “*OMNA PUSH Application ID*” via the OMA Push Enabler. If the EVVM Client cannot produce this address, it cannot receive notifications and therefore the EVVM Client SHALL NOT request any notifications.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 “*Outband notifications*”). Without it, the EVVM Server cannot send notifications to the EVVM Client.

Corresponding preference: clientprefs/prefs/notifications@destination

#### 7.3.2.2.7 List of supported notification types

The supported notification types are a set of flags indicating which notification types the EVVM Client is supporting. Each flag corresponds to a specific notification type. When a flag is ‘true’, it indicates that the EVVM Client supports receiving notifications of the corresponding type. When a flag is ‘false’, it indicates that the EVVM Client does not support receiving notifications of the corresponding type.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 “*Outband notifications*”).

Corresponding preferences:

- clientprefs/prefs/notifications@deac
- clientprefs/prefs/notifications@pref
- clientprefs/prefs/notifications@sync
- clientprefs/prefs/notifications@tran

#### 7.3.2.2.8 Notification encryption

The notification encryption is a token indicating the client's choice regarding encrypting notifications. The tokens are defined in section "*6.2 Elementary Data Types*" of [OMA EVVM XDM]. When the token is 'always', it indicates that the client forces notification encryption. When the token is 'ifneeded', it indicates that the client leaves the choice to the EVVM Server; in this case, the EVVM Server SHALL encrypt the content when it is unable to determine the delivery path, or, it is able to determine the delivery path but it is deemed to be unsecure. When the token is 'never', it indicates that the client forbids notification encryption as whole.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 "*Outband notifications*").

Corresponding preference: clientprefs/prefs/notifications@encrypt

#### 7.3.2.2.9 Notification encryption algorithms

The notification encryption algorithms are a list of index, token, strength pairs indicating which encryption algorithms and strengths the EVVM Client is capable of decrypting. The purpose of the index is to provide an additional level of obfuscation to the content. The index is an integer number and it SHALL be unique within the scope of the preferences of a single EVVM Client. Each token corresponds to a specific encryption algorithm. The tokens are defined in section "*6.2 Elementary Data Types*" of [OMA EVVM XDM]. Depending on the characteristics of the chosen encryption mechanism (asymmetric or symmetric), the EVVM Client SHALL include a public key or a shared secret along with each encryption setting,

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 "*Outband notifications*").

Corresponding preference: clientprefs/prefs/notifications/encryption

#### 7.3.2.2.10 Filtering of large media

The filtering of large media is a flag indicating whether the media items larger than a certain size should be filtered in the voicemails to be fetched by this client. When the flag is 'true', it indicates that, while fetching a voicemail, this client will receive a voicemail where any media item larger than the size given in the preference element 'largerthan' has been removed by the EVVM Server.

The EVVM Server uses this information to adapt the voicemail content before delivering it to the EVVM Client:

Corresponding preferences:

- clientprefs/prefs/filterlargemedia@active
- clientprefs/prefs/filterlargemedia/largerthan

#### 7.3.2.2.11 List of local supporting application capabilities

The preference of the local supporting application capabilities is the list of the local supporting capabilities by the local applications in the user's device as supporting EVVM Client accessing the EVVM-based service, including SMS Capabilities, DM Capabilities, Push Capabilities, CPM Storage Capabilities, etc. The EVVM Server uses those information to determine the capabilities of the user's device, e.g., whether supporting SMS capabilities.

### 7.3.2.3 User preferences

User preferences are managed by the EVVM Clients on behalf of their EVVM users, respectively. The user preferences are interpreted by the EVVM Server.

EVVM Clients SHOULD maintain user preferences according to the wishes of their EVVM users. The EVVM Client SHALL ensure that all available user preferences are up-to-date in the appropriate XML document. Since user preferences are universal, one and only one XML document is available for each EVVM user to whom these preferences apply.

The XML document containing the user preferences SHALL conform to all requirements that are defined for the "user preferences" application usage in [OMA EVVM XDM].

#### 7.3.2.3.1 Subscriber status

The subscriber status is a Boolean flag. The flag indicates the user's desire regarding the voicemail service as a whole. When the flag is true, the EVVM Server SHALL provide the service normally. When the flag is false, the EVVM Server MAY provide access to the voicemail storage but it SHALL reject any attempts to deposit voicemails from any source.

Corresponding preference: userprefs/prefs/evvm@active

#### 7.3.2.3.2 EVVM password

The EVVM password is a string. The string contains the user's password that he/she is using to access the EVVM Server (via SMTP or IMAP).

The EVVM Client SHALL ensure that the EVVM password meets the requirements imposed by the service provider, see password strength requirements in section 7.3.2.1 "*Service preferences*".

Corresponding preference: userprefs/prefs/evvm@password

#### 7.3.2.3.3 TUI settings

The TUI settings are comprised of various settings: a password, a language indicator and a status flag. The password is used to access the TUI. The language indicator contains the user's choice of language for the TUI. The status flag indicates whether the user accepts voicemails via the TUI or not (true means yes, false means no).

Since TUI is not in the scope of this specification, there are no normative requirements imposed onto the TUI.

The EVVM Client SHALL ensure that the language choice is one of the TUI languages disclosed in the service preferences (see section 7.3.2.1 "*Service preferences*").

The EVVM Client SHALL NOT change the status flag when the user is not allowed to control it, see TUI limitations in section 7.3.2.1 "*Service preferences*".

The EVVM Client SHALL ensure that the TUI password meets the requirements imposed by the service provider, see password strength requirements in section 7.3.2.1 "*Service preferences*".

Corresponding preference: clientprefs/prefs/tui

#### 7.3.2.3.4 Message archiving

The message archiving setting is a Boolean flag. The flag indicates the user's desire to save a copy of each voicemail that has been sent via the EVVM Server. While the flag is true, the EVVM Server SHALL store a copy of each voicemail that has been sent; the voicemails SHALL be stored in sentbox. While the flag is false, the EVVM Server SHALL NOT store copies of sent voicemails.

Corresponding preference: userprefs/prefs/evvm@storesentvm

#### 7.3.2.3.5 Preferred media format

Not covered in this specification.

#### 7.3.2.3.6 Groups

Once defined, groups can be used for black list (see section 7.3.2.3.7 "*Black list*") and white list (see section 7.3.2.3.8 "*White list(s)*"). Groups are comprised of a name, an identifier, and a list of users. The group's name is merely cosmetic, intended to be shown on the user interface. The group's identifier is used to refer to the group; it was not meant to be shown on the user interface. The list of users contains user names and user identifiers. The user names are merely cosmetic, intended to be shown on the user interface. The user identifiers refer to the users.

The EVVM Client SHALL ensure that the identifier of the group is unique within the scope of the user.

The EVVM Client SHALL ensure that user identifiers follow the definition in section 7.3.1.1 "*User Identifier*".

Corresponding preference: userprefs/prefs/groups

#### 7.3.2.3.7 Black list

The black list is comprised of a list of group identifiers.

The EVVM Client SHALL ensure that the identified group(s) exist, see section 7.3.2.3.6 “*Groups*”.

The EVVM Client SHALL omit the name of the group.

Corresponding preference: userprefs/prefs/blacklists

#### 7.3.2.3.8 White list(s)

The white list is comprised of a list of group identifiers.

The EVVM Client SHALL ensure that the identified group(s) exist, see section 7.3.2.3.6 “*Groups*”.

The EVVM Client SHALL omit the name of the group.

Corresponding preference: userprefs/prefs/whitelists

#### 7.3.2.3.9 Notification settings

The notification settings are a set of flags indicating the user’s wish regarding sending notifications under various roaming conditions. Each flag corresponds to a specific roaming condition. When a flag is ‘true’, it indicates that the user requests sending notifications under the corresponding roaming condition. When a flag is ‘false’, it indicates that the user forbids sending notifications under the corresponding roaming condition.

The EVVM Server uses this information during the notification procedures (see section 7.7.5.2 “*Outband notifications*”).

Corresponding preferences:

- userprefs/prefs/notifications@home
- userprefs/prefs/notifications@roaming
- userprefs/prefs/notifications@intlroaming

#### 7.3.2.3.10 Mappings

The mappings list of physical mailbox name and diplay name pairs. Each pair associates a display name with a physical mailbox name. XML unsafe strings SHALL be encoded using Base64 encoding [RFC4648]. The EVVM Client SHALL ensure that no physical mailbox name have more than one display name assigned.

The EVVM Client uses this information for cosmetic purposes only, as described in section 7.3.5.1 “*Mailbox naming conventions*”.

Corresponding preference: userprefs/prefs/mappings

#### 7.3.2.3.11 Delivery status information (sending and receiving)

Not covered in this specification.

#### 7.3.2.3.12 Text to speech conversion

The demand for text to speech conversion can be set by the user and will be stored in EVVM user preferences. The text to speech conversion is used for generating audio parts (synthesized speech) for text based greetings or messages.

The following settings define the user’s request to perform text to speech conversion.

When ‘greetings’ flag is set to ’true’, the text based greeting stored in the Greetings ‘folder’ will be converted into synthesized speech.

When ’voicemails’ flag is set to ’true’, the text based message stored in the Outbox or Draft ‘folder’ will be converted into synthesized speech.

Corresponding preferences:

- userprefs/prefs/tts@greetings
- userprefs/prefs/tts@voicemails

#### 7.3.2.3.13 Voice to text conversion

The preferences for voice to text conversion are set by the user and will be stored in EVVM user preferences.

The following settings define the user’s request to perform voice to text conversion for received voice messages

When ‘active’ flag is set to ‘true’ received voicemail messages will be converted into text, which will be delivered to the EVVM Client. If it is set to ‘false’ only the original voicemail will be delivered to the EVVM Client.

When ‘includeoriginalvm’ flag is set to ‘true’ the original voicemail will be sent to the EVVM Client in addition to the resulting text. The ‘includeoriginalvm’ flag has no effect if ‘active’ flag is set to ‘false’.

Corresponding preferences:

- userprefs/prefs/v2t@active
- userprefs/prefs/v2t@includeoriginalvm

#### 7.3.2.3.14 Preferred reception mechanism

The preferred reception mechanism indicates whether the EVVM user prefers to receive his/her voicemails as SMS, MMS, or e-mail messages.

Corresponding preference: userprefs/prefs/nonvmreception

## 7.3.3 Media Format

### 7.3.3.1 Voicemails

An EVVM voicemail is an internet message based on the format already defined in VVM 1.3 [GSMA VVM].

This specification introduces a number of improvements in the following areas:

- New message format specification, see section 7.3.3.1.1 “*New message format specification*”.
- Header fields, see section 7.3.3.1.2 “*Header fields*”.
- Internationalization support, see section 7.3.3.1.3 “*Internationalization support*”.
- Transfer encoding, see section 7.3.3.1.4 “*Transfer encoding*”.

#### 7.3.3.1.1 New message format specification

EVVM Clients SHALL send voicemail messages using the message format described in [RFC5322].

EVVM Servers SHALL support sending and receiving voicemail messages using the message format described in [RFC5322].

While the preferred voicemail message format is that defined in [RFC5322], EVVM Clients and EVVM Servers may receive voicemails in the legacy format from several sources. For example, from: VVM 1.3 client, TUI, Message Storage Server, Standard Non-OMA VM service. Therefore, for backwards compatibility reasons, all EVVM Clients SHALL support receiving voicemails in the message format defined in [RFC2822]. Similarly, all EVVM Servers SHALL support both sending and receiving voicemails in the message format defined in [RFC2822].

#### 7.3.3.1.2 Header fields

The usage of header fields SHALL conform to [RFC5322] and [GSMA VVM]. In addition, when a voicemail is being composed at the EVVM Client,

- If there are multiple identifiers associated with the user, the EVVM Client SHALL allow the user to choose one of his/her identifiers as the originator's identifier and include the chosen identifier in the From header field.
- If the user requests a read report, the EVVM Client SHALL include a Disposition-Notification-To header field as specified in [RFC3798].
- If the EVVM user desires a reply voicemail to be sent to an identifier different from the identifier included in the From header field, the EVVM Client SHALL include that identifier to be used for reply in the Reply-To header field as specified in [RFC5322].
- If the EVVM user desires a callback, the EVVM Client SHALL include a phone number to be used for call-back in the Callback-Number header field as specified in section 7.3.8.2 "*Callback-Number Header*".
- If the EVVM user desires to include emotion indication to the voice mail, the EVVM Client SHALL include the header field as specified in Section 7.3.8.1 "*Emotion Indication Header*".

#### 7.3.3.1.3 Internationalization support

The OMA EVVM Enabler supports including international characters in headers, delivery status notification and disposition notifications. A general overview of this framework is available in [RFC6530]. OMA EVVM requires internationalization support for the IMAP and the SMTP protocols; therefore all EVVM Clients and EVVM Servers SHALL support the following specifications:

- Internationalized email headers, [RFC6532],
- SMTP extension for internationalized email addresses, [RFC6531],
- Internationalized delivery status and disposition notifications, [RFC6533],
- IMAP support for UTF-8, [RFC5738bis].

#### 7.3.3.1.4 Transfer encoding

The SMTP and IMAP protocols have been designed to transmit text-based content. Transmission of binary content is achieved by encoding the binary content using the BASE64 encoding scheme. All EVVM Clients and EVVM Servers SHALL support the BASE64 encoding scheme defined in [RFC4648].

The BASE64 encoding introduces an approximately 33% size increase for the encoded binary content. Considering that the content in voicemails is always binary, this overhead is significant. In order to omit the need for the BASE64 encoding, all EVVM Clients and EVVM Servers SHOULD support the following specifications:

- SMTP service extensions for transmission of large and binary MIME messages, [RFC3030].
- IMAP4 binary content extension, [RFC3516].

### 7.3.3.2 Greetings

This section describes only the format of greetings; see section 7.6.3 "*Greeting and Voice Signature management*" and section 7.7.3 "*Greeting and Voice Signature management*" for using greetings.

In general, the format of a greeting SHALL follow the voicemail format established in section 7.3.3.1 "*Voicemails*". This section describes only the differences from voicemails.

In addition to limiting the supported content types, the EVVM Server MAY impose a limitations on the maximum length of greetings as well; see section 7.3.2.1 "*Service preferences*".

While the voicemail envelope allows including any type of content in greetings, considering the purpose of greetings, greetings SHOULD include audio.

#### 7.3.3.3 Voice signatures

This section describes only the format of voice signatures; see section 7.6.3 "*Greeting and Voice Signature management*" and section 7.7.3 "*Greeting and Voice Signature management*" for using voice signatures.

In general, the format of a voice signature SHALL follow the voicemail format established in section 7.3.3.1 "*Voicemails*". This section describes only the differences from voicemails.

In addition to limiting the supported content types, the EVVM Server MAY impose a limitation on the maximum length of voice signatures as well; see section 7.3.2.1 "*Service preferences*".

While the voicemail envelope allows including any type of content in voice signatures, considering the purpose of voice signatures, voice signatures SHOULD include only audio.

### 7.3.4 Media types

While the basic container format allows inserting any type of media into voicemails, greetings and voice signatures, it is necessary to identify a minimum set, a supplementary set and an optional set of media types to establish conformance requirements. All text-based media types SHALL conform to the character sets defined in section 7.3.4.1 "*Character sets*".

EVVM Clients and EVVM Servers SHALL support the following media types:

| Name | MIME | Special |
|---|---|---|
| Adaptive Multi-Rate audio | audio/amr | 12.2 kbit/s (GSM EFR) |
| Plain text | text/plain | |

**Table 1: Minimum set of media types**

EVVM Clients and EVVM Servers SHOULD support the following media types:

| Name | MIME | Special |
|---|---|---|
| Adaptive Multi-Rate Wideband audio | audio/amr-wb | Config-WB-Code 0 as defined in [3GPP TS26.103] |
| HTML | text/html | |

**Table 2: Supplementary set of media types**

EVVM Clients and EVVM Servers MAY support any other media types. The following table includes a set of commonly used media types that are not required by the OMA EVVM Enabler, however device and server vendors should consider implementing these. It is RECOMMENDED that a service based on the OMA EVVM Enabler provides transcoding for these content types.

| Name | MIME | Special |
|---|---|---|
| 3rd Generation Partnership Project video with Adaptive Multi-Rate audio | video/3gpp;codec=h263_amr | |
| G711, A-law | audio/wav;codec=g711a | |
| G711, μ-law | audio/wav;codec=g711u | |
| High Rate Speech Service Option 17 for Wideband Spread Spectrum Communications Systems (QCELP 13K vocoder) | audio/qcelp | |
| Tagged Image File Format | image/tiff | |
| Portable Document Format | application/pdf | |

**Table 3: Optional media types**

#### 7.3.4.1 Character sets

While the basic container format allows inserting text-based content with any character sets into voicemails, greetings and voice signatures, it is necessary to identify a minimum set and a supplementary set to establish conformance requirements.

EVVM Clients and EVVM Servers SHALL support the following character sets:

| Name | Preferred MIME name | Special |
|---|---|---|
| UTF-8 | UTF-8 | |

**Table 4: Minimum set of character sets**

EVVM Clients and EVVM Servers SHOULD support the following character sets:

| Name | MIME | Special |
|---|---|---|
| Latin 1 | ISO-8859-1 | |
| Latin 2 | ISO-8859-2 | |
| Latin 9 | ISO-8859-15 | |
| US ASCII | US-ASCII | |

**Table 5: Supplementary set of character sets**

EVVM Clients and EVVM Servers MAY support any other character sets.

#### 7.3.4.2 Using the SMIL XML content type

EVVM Clients MAY support Synchronized Multimedia Integration Language (SMIL) in voicemail messages and support the “application/smil+xml” media type [RFC4536]. The use of SMIL allows a user to create and consume synchronized multimedia presentations containing multiple discrete media elements or media streams.

### 7.3.5 Mailboxes

The OMA EVVM Enabler reuses the special mailboxes defined in [GSMA VVM], and, in addition, allows assigning names to mailboxes that are otherwise not supported (such as special and international characters).

The [GSMA VVM] specification defines the following special mailboxes:

- Inbox “folder”, referred to as Inbox in EVVM.
- Greetings “folder”, referred to as Greetings in EVVM.

OMA EVVM follows the same principle and defines additional special mailboxes:

- Draftbox, used to store draft voicemails, greetings and/or voice signatures.
- Outbox, used for submitting voicemails for delivery.
- Sentbox, used to store sent voicemails.
- Spambox, used to report voicemails as spam.

All special mailboxes are logical, meaning that there is no actual storage-level mailbox name implied, see section 7.3.5.1 “*Mailbox naming conventions*”.

The EVVM Server SHALL provide Inbox, Greetings and Sentbox at all times.

The EVVM Server MAY provide Draftbox, Outbox and Spambox. All EVVM Clients and EVVM Servers SHALL support all special mailboxes and support management of those that exist in the service preferences (see section 7.3.2.1 “*Service preferences*”).

GSMA VVM clients have no knowledge of the special purpose of the mailboxes introduced to the EVVM Enabler. Consequently, they are going to treat all mailboxes defined in EVVM as user-created mailboxes. Therefore, the EVVM Server SHALL refuse any requests other than deletion from GSMA VVM clients that may result in changes in Sentbox and Spambox.

#### 7.3.5.1 Mailbox naming conventions

As defined in section 7.3.5 “*Mailboxes*”, all special mailboxes have a logical name, which are used throughout this specification to refer to them. Each mailbox – including the special mailboxes – SHALL have a storage-level name and MAY have a display name. The storage-level names are the actual mailbox names on the physical storage, and these are used on the protocol-level while accessing the storage and therefore, it is limited by the characteristics of the physical storage and the protocol. The display names are the mailbox names that may appear on the user interface of the device, replacing the storage-level mailbox names with a user-friendly name. While display names are purely cosmetical, they are extremely important in case of users using their native language script or character sets.

Mailbox storage-level names SHALL follow the storage naming conventions in [RFC3501]. The storage-level names SHALL be used on the protocol level.

Mailbox display names SHALL be UTF-8. The EVVM Client SHALL ensure that each display name is unique within the scope of the EVVM User, and deny any attempts to introduce a display name that is already in use.

The EVVM Server SHALL map the logical name of each special mailbox to storage-level names in the service preferences, as described in section 7.3.2.1 “*Service preferences*”. The EVVM Client SHALL follow this mapping.

### 7.3.6 Notifications

OMA EVVM supports IMAP-based notifications, VVM 1.3 [GSMA VVM] notifications and defines a new set of EVVM notifications, as described in section 7.4 “*Introduction to notifications*”.

In their raw format, EVVM notifications are XML-based, providing structured information. However, EVVM notifications may be transmitted in raw or encrypted formats. These transformations (encryption) are governed by applicable service (serviceprefs-final/prefs/notifications) and client preferences (clientprefs/prefs/notifications). The transformations are applied by the EVVM Server as defined in section 7.7.5.2 “*Outband notifications*”. Consequently, the notifications that the EVVM Client receives may be ‘raw’ XML documents or binary data. The EVVM Client decodes the notification as defined in section 7.6.5.2 “*Outband notifications*”.

#### 7.3.6.1 Subscriber status notification

There are two types of subscriber status notifications: client-originated subscriber status notifications and system-originated subscriber status notifications. The client-originated subscriber status notifications are used to query the current subscriber status, or, to update the subscriber status on the server side. The system-originated subscriber status notifications are either responses to client-originated subscriber status notifications, or, system-generated notifications; both are used to inform the client that the subscriber status on the server side has been updated.

##### 7.3.6.1.1 VVM 1.3

The client-originated STATUS SMS and the system-originated STATUS SMS are defined in [GSMA VVM].

The [GSMA VVM] specification does not include message flows. For illustration purposes, example flows are included in sections C.1.1 “*Client-initiated subscriber status notifications, VVM 1.3*” and C.1.2 “*Server-initiated subscriber status notifications, VVM 1.3*”.

##### 7.3.6.1.2 EVVM 1.0

The OMA EVVM subscriber status is not managed via notifications, rather, by managing the relevant settings directly using the XDM Enabler, as described in section 7.3.2.2 “*Client preferences*”. The notifications about changes to these preferences are also carried out by the XDM Enabler. An example flow is included in section C.1.5 “*XDM-EVVM notification framework*”, detailing the general framework.

#### 7.3.6.2 Sync notification

The sync notification is a system-originated notification, indicating that there has been a change in the message store. The payload of the notification contains information about the changes in the message storage.

#### 7.3.6.2.1 VVM 1.3

The sync notification via SMS (SYNC SMS) is defined in [GSMA VVM].

The [GSMA VVM] specification does not include message flows; for illustration purposes, an example flow is included in section C.1.3 "*Sync notification, VVM 1.3*".

#### 7.3.6.2.2 EVVM 1.0

The OMA EVVM Enabler defines inband sync and outband sync notifications.

##### 7.3.6.2.2.1. Inband sync notifications

The inband sync notification payloads for IMAP IDLE and IMAP NOTIFY are defined in [RFC5465] and [RFC5465], respectively.

Examples flows for inband sync notification using IMAP IDLE and IMAP NOTIFY are included in section C.1.4.1 "*Inband sync notifications*".

It is RECOMMENDED that EVVM Clients that support both IMAP NOTIFY and IMAP IDLE use IMAP NOTIFY only.

##### 7.3.6.2.2.2. Outband sync notification

Notification type: SYNC

The notification content without encryption SHALL be an XML document conforming to the XML Schema defined in [XSD-SYNC] without the BOM character (if present).

Example: see sections C.1.6.4 "*SYNC, minimal*" and C.1.6.5 "*SYNC, complex*"

An example flow for outband sync notification is included in section C.1.4.2 "*Outband sync notification*".

### 7.3.6.3 Deactivation notification

Notification type: DEAC

The notification content without encryption SHALL be an XML document conforming to the XML Schema defined in [XSD-DEAC] without the BOM character (if present).

Example: see sections C.1.6.1 "*DEAC, temporary*" and C.1.6.2 "*DEAC, permanent*"

### 7.3.6.4 Preferences notification

Notification type: PREF

The notification content without encryption SHALL be an XML document conforming to the XML Schema defined in [XSD-PREF] without the BOM character (if present).

Example: see section C.1.6.3 "*PREF*"

### 7.3.6.5 Transcription notification

Notification type: TRAN

The notification content without encryption SHALL be an XML document conforming to the XML Schema defined in [XSD-TRAN] without the BOM character (if present).

Example: see section C.1.6.6 "*TRAN*"

## 7.3.7 Extended Server Capabilities

The EVVM Server SHALL inform the EVVM Client of the capabilities supported by the EVVM Server, following the rules defined in the IMAP protocol [RFC3501].

In addition to the capabilities specified in [RFC3501], OMA EVVM defines several new capability names as follows:

- 'XVM2NONVM', which indicates the EVVM Server supports forwarding voicemails towards non-VM services, i.e., SMS, MMS or e-mail;
- 'XSPAMREPORT', which indicates the EVVM Server supports receiving spam-reporting messages from the EVVM Client.

The EVVM Server SHALL include the above capability names in the CAPABILITY response code or the CAPABILITY response, if the corresponding capabilities are supported, according to [RFC3501].

## 7.3.8 EVVM Specific Headers

This section describes EVVM-specific message header formats.

### 7.3.8.1 Emotion Indication Header

The following additional header field can be added to the message being deposited:

Header: "EVVM-EmotionIndication"

Description: Indicates the emotion associated with the voicemail as specified by the EVVM User at the time of composing the voicemail, or, as determined on the device and/or the server side at the time of submitting the voicemail for delivery. The source of the emotion indication (user vs. device vs. server) SHALL be indicated accurately. The device and/or the server may determine the emotion indication based on any available factors, such as facial expressions, voice tones, etc., however these procedures are not in the scope of this specification.

This header is OPTIONAL.

Legal Values (case insensitive): see ABNF definition of 'emotion-indication' below.

The syntax of the emotion indication header field is defined using the ABNF notation [RFC5234] as follows:

```
emotion-indication             = user-defined-emotion / determined-emotion
user-defined-emotion           = "user" "SP" emotion-list
determined-emotion             = determinant "SP" emotion-list \
                                 *( "SP" "nil" "SP" determinant "SP" emotion-list )
determinant                    = "device" / "server" / "tui"
emotion-list                   = emotion-value *( "SP" emotion-value )
emotion-value                  = string         ; A smiley. It SHALL NOT contain "SP" "nil" "SP".
string                         = 1*( ALPHA / DIGIT / char-special)
char-special                   = <any XML-safe special character>
```

Examples:

- The user determines the emotion indicator:
  `EVVM-EmotionIndication: user :o)`
- The device-side determines the emotion indicator and EVVM Client adds it when the message is submitted. The server-side determines its own interpretation and the EVVM Server adds its own values:
  `EVVM-EmotionIndication: device :D ;-) nil server ;-)`
- When a voicemail is recorded via the TUI, the TUI may also determine the emotion indication:
  `EVVM-EmotionIndication: tui :-o :-(`

### 7.3.8.2 Callback-Number Header

The following additional header field can be added to the message being deposited:

Header: "EVVM-CallbackNumber"

Description: Indicates the callback number by which the recipient can make a call to the sender.

This header is OPTIONAL.

Legal Values: see ABNF definition of ‘callback-number’ below.

The syntax of the Callback-Number header is specified below, in the ABNF format described in [RFC5234]:

```
callback-number-header          = “EVVM-CallbackNumber:” callback-number
callback-number                 = phonenumber
```

Examples:

- `EVVM-CallbackNumber: +12345678`

## 7.4 Introduction to notifications

This section introduces the notification mechanisms in general. Notification formats and payloads are defined in section 7.3.6 “*Notifications*”. Notification procedures are defined in section 7.6.5 “*Notification handling*” and section 7.7.5 “*Notification handling*”.

For backwards compatibility, the OMA EVVM Enabler supports the existing notification mechanisms and payloads already defined in VVM 1.3 [GSMA VVM]. OMA EVVM defines additional types of notifications: inband notifications based on IMAP IDLE [RFC2177] and IMAP NOTIFY [RFC5465] as well as outband notifications based on OMA PUSH [OMA-PUSH]. Inband notifications are used for EVVM Clients which maintain long lived IMAP connection with the EVVM Server, while outband notifications are used for EVVM Clients which are connected to the EVVM Server only for limited time needed for necessary reconciliation.

The mechanisms to dispatch and deliver outband notifications via the OMA PUSH Enabler are defined in [OMA Push]; therefore, this specification defines only the payload in section 7.3.6 “*Notifications*”, a client preference to determine the device address in section 7.3.2.2.6 “*Device address*” and the PUSH application ID in section 8.2 “*OMNA Considerations*”.

The EVVM Clients MAY request delivering notifications from the EVVM Server by storing a preference for the desired types of notifications. The EVVM Client MAY also request encryption for notification payloads that are to be dispatched via the OMA PUSH Enabler. The management of the client-specific notification preferences is defined in section 7.3.2 “*Preferences*”.

The EVVM Server SHOULD send notifications to the EVVM Client according to the notification preferences of the EVVM Client.

The following table denotes all notification types supported by OMA EVVM:

| Name | Uses | Type | Server req. | Client req. | Section |
|---|---|---|---|---|---|
| Client-originated STATUS SMS | SMS | | M | O | 7.3.6.1.1 |
| Deactivation notification | OMA PUSH | DEAC | O | O | 7.3.6.3 |
| Preferences notification | OMA PUSH | PREF | M | M | 7.3.6.4 |
| System-originated STATUS SMS | SMS | | M | O | 7.3.6.1.1 |
| Sync notification | SMS | | M | O | 7.3.6.2.1 |
| Sync notification | OMA PUSH | SYNC | M | M | 7.3.6.2.2.2 |
| Sync notification | IMAP IDLE | | O | O | 7.3.6.2.2.1 |
| Sync notification | IMAP NOTIFY | | O | O | 7.3.6.2.2.1 |
| Transcription notification | OMA PUSH | TRAN | O | O | 7.3.6.5 |

**Table 6: OMA EVVM notification types**

The ‘Uses’ column indicates the delivery mechanism used. VVM 1.3 uses SMS as defined in [GSMA VVM], OMA PUSH is defined in [OMA Push], IMAP IDLE is defined in [RFC2177], IMAP NOTIFY is defined in [RFC5465].

The ‘Type’ column indicates the mnemonic used to manage the corresponding notification preferences. See management of notification preferences in section 7.3.2.2 “*Client preferences*”. A missing mnemonic indicates that the notification setting corresponding to a certain notification type cannot be managed.

The ‘Server req.’ column indicates the support requirement for the EVVM Server.

The ‘Client req.’ column indicates the support requirement for the EVVM Client.

The ‘Section’ column indicates the section number of this document that describes the notification in detail.

### 7.4.1 Notification encryption

The notification content SHALL be encrypted by the EVVM Server or stay in clear text according to the EVVM Client preferences, if allowed by the service provider’s policy. The EVVM Enabler supports the use of asymmetric and symmetric encryption mechanisms. In order to ensure some level of interoperability, all EVVM Clients and EVVM Servers SHOULD support the following encryption mechanisms but SHALL support at least one:

- Elliptic Curve Integrated Encryption Scheme (ECIES) defined in [SECG SEC1]
- Advanced Encryption Standard (AES) defined in [NIST FIPS-197]
- RSA Cryptography Standard defined in [RSA PKCS]

The EVVM Enabler permits the use of encryption mechanism not listed above, using the extension mechanism provided by the XML Schema in the corresponding XML elements and attributes of service and client preferences. However, the use of these encryption mechanisms is not in the scope of this specification.

The EVVM Server stores all of the encryption mechanisms that it supports in the service preferences (serviceprefs-final/prefs/notifications/encryption). EVVM Clients - preferably, taking into account the mechanisms supported by the server – store their own encryption preferences (mechanism, strength, public key/shared secret) individually in their own set of client preferences (clientprefs-final/prefs/notifications/encryption). The EVVM Client SHALL also store a random string called client prefix (clientprefs-final/prefs/notifications@clientprefix). The EVVM Server SHALL include the client prefix in all notifications in order to support verification of the sender.

## 7.5 Provisioning

As most OMA Enablers, OMA EVVM supports provisioning devices with EVVM service parameters using OMA DM.

It is expected that not all devices include an OMA DM stack. Personal computers without mobile network connectivity typically do not include support for OMA DM. In order to mitigate the burden of manual configuration steps for end users, the OMA EVVM Enabler provides an alternative solution.

Each solution is described in the following sub-sections.

### 7.5.1 Provisioning service parameters using OMA DM

This section specifies the parameters that are needed for initiation of EVVM Service by the EVVM client, as well as continuous provisioning by the Service Provider.

The parameters listed in the table below are needed for EVVM Client provisioning:

| Name | Description | Mandatory (M) / Optional (O) |
|---|---|---|
| EVVM-ProviderName | Identity of the EVVM Service Provider | O |
| EVVM-ServiceName | User displayable name for the EVVM service | O |
| EVVM-XDM | Reference to the XDM settings used by this EVVM Service. | M |
| EVVM-IMAP | Hostname and port for the IMAP server. | M |
| EVVM-SMTP | Hostname and port for the SMTP server. | M |

The Management Object for EVVM service SHOULD be used for provisioning of the EVVM Client by the Service Provider.

The Management Object for EVVM service is defined in [OMA EVVM MO].

### 7.5.2 Service discovery without OMA DM

As described in section 7.6.6.1 "*Capability agreement*", the EVVM Client is REQUIRED to obtain a copy of the service preferences documents prior to accessing the EVVM Server. The service preferences contain the service parameters described in section 7.3.2.1.1.3 "*Service parameters*", designating the appropriate network locations to access the EVVM Server using the IMAP and SMTP protocols. As such, the EVVM Client will have access to the service parameters provided that the XDM Client in the device can access the XDM Server and fetch the service preferences documents.

In order to access the XDM Server using the XDM Client, the device needs to be configured with information that allows accessing the documents stored in the XDM Server. When OMA DM is not available, this is achieved by providing the address of the XDM Server and the user credentials to the end-user by other means such as a phone call, internet mail, internet chat, or even on paper. Using the XDM Server address and the user credentials to access the XDM Server, the user can manually configure the XDM Client.

Having configured the XDM Client, the EVVM Client can obtain the service preferences documents, which include the service parameters that are essential to accessing the EVVM Server via the EVVM-1 interface. Extracting the service parameters allows from the service preferences allows the end-user to start using the service immediately, without having to manually configure EVVM Client with SMTP and IMAP settings.

## 7.6 Procedures at the EVVM Client

The EVVM Client provides the following procedures:

- Authentication, as described in section 7.6.1 "Authentication",
- Voicemail handling, as described in section 7.6.2 "Voicemail handling",
- Greeting and Voice Signature management, as described in section 7.6.3 "Greeting and Voice Signature management",
- Delivery/read report handling, as described in section 7.6.4 "Delivery/read report handling",
- Notification handling, as described in section 7.6.5 "Notification handling",
- IMAP4 session management, as described in section 7.6.6 "IMAP4 session management",
- Content adaptation, as described in section 7.6.7 "Content adaptation",
- Spam reporting, as described in section 7.6.8 "*Spam Reporting*".

### 7.6.1 Authentication

The EVVM Client SHALL support the authentication procedures described in VVM 1.3 [GSMA VVM] with the clarifications included in section 6.4.1 "*Authentication*".

The EVVM Client SHALL either use the plain-text username/password method or the SASL [RFC4422] method to perform authentication procedures with the EVVM Server. How the EVVM Client initiates the authentication mechanism SHALL conform to the rules defined in [RFC3501].

The EVVM Client MAY set up a TLS session prior to authenticating with the EVVM Server, according to [RFC3501] and [GSMA VVM].

### 7.6.2 Voicemail handling

#### 7.6.2.1 Voicemail composition

The EVVM Client SHALL support voicemail composition locally and it SHALL follow the voice mail format defined in section 7.3.3.1 "*Voicemails*".

Voicemails are composed locally by creating a blank voicemail and adding various header fields and attachments, or, by opening a locally stored voicemail and adding/removing various header fields and attachments.

The attachments may originate from the device (newly created or stored media), or, from another voicemail (existing media). If a media item to be attached to the voicemail exists in the voicemail storage on the network side as another voicemail or part of it, the EVVM Client MAY send an IMAP GENURLAUTH command to request that media item's URLAUTH URL, according to [RFC4467] and [RFC5550], and use it as a reference to that media item while sending the voicemail (see section 7.6.2.3 "*Voicemail submission*").

The EVVM Client MAY request the EVVM Server to forward a local voicemail, i.e., a voicemail composed or stored on the EVVM Client, towards an SMS, MMS or e-mail system. To request forwarding a local voicemail via SMS, MMS or e-mail, the EVVM Client SHALL include a Non-VM Service Identifier (NSID) parameter in the To header field in the voicemail's MIME header part and set the parameter to either "sms", "mms" or "email", depending on the user's request.

The following ABNF expression defines the Non-VM Service Identifier (NSID) as a parameter of the To header field:

```
NSID                      = "nsid=" ("sms" / "mms" / "email")
```

Example:

```
To: Bob@Example.COM; nsid=mms
```

The EVVM Client MAY also request forwarding a remote voicemail, i.e., a voicemail stored in the network, via SMS, MMS or e-mail. To request forwarding a remote voicemail towards SMS, MMS or e-mail, the EVVM Client SHALL include a Non-VM Service Identifier (NSID) parameter, with its value set to either "sms", "mms" or "email", in the To header field in the MIME header part and deposit this message containing the remote media item's or voicemail's URL(s) into the Draft box, via an IMAP APPEND command with the CATENATE extension, before requesting the URLAUTH URL of the assembled message, according to [RFC5550].

While EVVM Client MAY insert any header fields, the EVVM Client SHALL honor to the header field requirements defined in section 7.3.3.1.2 "*Header fields*".

Upon request from the EVVM user to include emotion indication in the voice mail, the EVVM Client SHALL include the header field and the value provided by the EVVM user (or the device) as specified in Section 7.3.8.1 "*Emotion Indication Header*".

Similarly, while the EVVM Client MAY insert any type of attachments, the EVVM Client SHALL honor to the media type requirements defined in section 7.3.4 "*Media types*".

The EVVM Client MAY add a local or remote media item as a background music to a voice media. If the media item to be added as the background media exists in the voicemail storage on the network side as part of another voicemail, the EVVM Client MAY send an IMAP GENURLAUTH command to request that media item's URLAUTH URL, according to [RFC4467] and [RFC5550], and use it as a reference to that media item while sending the voicemail (see section 7.6.2.3 "*Voicemail submission*"). The EVVM Client SHALL insert a Content-Description header indicating "background media" in the body part corresponding to the background media.

The EVVM Client SHALL ensure that the resulting voicemail does not exceed the maximum voicemail size imposed by the service provider, see section 7.3.2.1 "*Service preferences*".

While NOT REQUIRED, it is RECOMMENDED that the EVVM Client allows the EVVM User to specify the sender and recipient identifiers prior to composing the message.

If the EVVM user requests the user-to-user encryption of one or more media items in the voicemail to ensure the confidentiality of the items between the originator and the recipient, the EVVM Client SHALL encrypt and encapsulate each of those selected media items using any suitable encryption mechanism and encapsulate them in the voicemail according to the procedures specified in [RFC1847]. Considering that the level of support at the endpoints cannot be determined and the end-to-end encryption has no impact on the interoperability, the EVVM Enabler leaves the choice of the encryption mechanism out of scope; OpenPGP, as specified in [RFC3156] and [RFC4880], S/MIME as specified in [RFC5751], or any other suitable mechanism can be used. Implementers should take into account various characteristics of the intended service deployment.

If the EVVM user requests the digital signing of one or more media items in the voicemail to ensure that the integrity of the content can be verified by the recipient, the EVVM Client SHALL generate a digital signature based on each of those selected media items and the sender's private key, and encapsulate each signature together with its corresponding media item, according to the procedures specified in [RFC1847]. Considering that the level of support at the endpoints cannot be determined and the digital signatures have no impact on the interoperability, the EVVM Enabler leaves the choice of the mechanism used for digital signatures out of scope; OpenPGP, as specified in [RFC3156] and [RFC4880], S/MIME as specified in [RFC5751], or any other suitable mechanism can be used. Implementers should take into account various characteristics of the intended service deployment.

## 7.6.2.2 Requesting read reports

While composing a voicemail, the EVVM user MAY request a read report before the voicemail is submitted for delivery. When a read report is requested by the EVVM user, the EVVM Client SHALL request MDNs as defined in [RFC3798].

## 7.6.2.3 Voicemail submission

### 7.6.2.3.1 Voicemail submission via SMTP

Upon composition of voice mail as specified in section 7.6.2.1 “*Voicemail composition*”, the EVVM Client SHALL be able to send voicemails to the EVVM Server via SMTP.

Before sending a voicemail, the EVVM Client SHALL check the service preferences to ensure that the voicemail to be sent does not exceed the predefined message size limit. Corresponding preference: /serviceprefs-final/prefs/bounds@maxvmsize

To submit a voicemail via SMTP, the EVVM Client SHALL perform the SMTP “message submission” procedures as specified in [RFC5321] with the following clarifications:

- If there are multiple identifiers associated with the sender, the EVVM Client SHALL include the user-selected identifier in the MAIL command.

If the user requests a delivery report, the EVVM Client SHALL include in the RCPT command the NOTIFY parameter with its appropriate value(s) (i.e., SUCCESS, FAILURE, or DELAY), according to [RFC3461].

If a URL is to be included in the voicemail as a reference to a media item existing in the voicemail storage on the network side (see section 7.6.2.1 “*Voicemail composition*”), the EVVM Client MAY send the URL with an SMTP BURL command while submitting the voicemail via SMTP, according to [RFC5550].

### 7.6.2.3.2 Voicemail Future Release

The EVVM Client MAY allow an EVVM user (sender) to request to send a voicemail at a future time.

When composing a voicemail as specified in section 7.6.2.1 “*Voicemail composition*”, EVVM user can set one type of parameters (hold-param) for Future Message Release via SMTP submission service extension as defined in [RFC4865], the hold-param includes:

- Future-release-interval, and
- Future-release-date-time.

An EVVM Client preparing to use Future Message Release extension SHALL first verify whether the EVVM Server supports it. An EVVM Client SHALL check the Extended Hello (EHLO) reply, if it includes FUTURERELEASE keyword, that means the EVVM Server supports this extension.

After setting the hold-param, the EVVM Client sends the voicemail to the EVVM Server just as other common voicemails, except containing the hold-param.

### 7.6.2.3.3 Voicemail submission via IMAP

Upon composition of voice mail as specified in section 7.6.2.1 “*Voicemail composition*”, the EVVM Client MAY be able to submit voicemails via IMAP.

Before sending a voicemail, the EVVM Client SHALL check the service preferences to ensure that the voicemail to be sent does not exceed the predefined message size limit and that the EVVM Server does in fact, provide Outbox. Corresponding preferences:

- /serviceprefs-final/prefs/bounds@maxvmsize
- /serviceprefs-final/mappings.

To submit a composed voicemail via IMAP, the EVVM Client SHALL use an IMAP APPEND command to deposit the voicemail message into the EVVM user's Outbox in the EVVM Server, as specified in [RFC3501].

To submit a voicemail that exists on the EVVM Server (i.e. in the Draftbox) via IMAP, the EVVM Client SHALL use the IMAP COPY command to deposit the voicemail message into the Outbox, as specified in [RFC3501]. After the successful COPY operation from Draftbox to Outbox, the EVVM Client SHOULD remove the voicemail from Draftbox.

The EVVM Server MAY choose to request or not to request a delivery report on behalf of the EVVM Client according to the user's preference.

If a URL is to be included in the voicemail as a reference to a media item existing in the voicemail storage on the network side (see section 7.6.2.1 "*Voicemail composition*"), the EVVM Client MAY use the IMAP CATENATE Extension to concatenate the URL with the voicemail while depositing the voicemail into the Outbox via the APPEND command, according to [RFC5550].

The EVVM Client MAY request forwarding a remote voicemail, i.e., a voicemail stored in the network, via SMS, MMS or e-mail. To request forwarding a remote voicemail towards SMS, MMS or e-mail with IMAP, the EVVM Client SHALL send the URL corresponding to the remote voicemail using an APPEND command with the CATENATE extension according to [RFC5550] and include a Non-VM Service Identifier (NSID) parameter in the To header field in the literal argument of the APPEND command.

The following ABNF expression defines the Non-VM Service Identifier as a parameter of the To header field:

```
NSID                       = "nsid=" ("sms" / "mms" / "email")
```

Example:

```
To: Bob@Example.COM; nsid=mms
```

### 7.6.2.4 Voicemail presentation

After fetching a voicemail from the EVVM Server, the EVVM Client SHALL distinguish the main body part (which may contain the voicemail sender's voice) from other attachments. If the voicemail body consists of multipart/related structure [RFC2387] and includes a SMIL part which contains instructions on how the multimedia content should be rendered to the display and speakers etc. on the terminal, the EVVM Client SHALL present the voicemail accordingly.

The EVVM Client SHOULD render the media contained in the main body part to the user automatically when the user opens the voicemail with the background media, if present. After the main body is presented the EVVM Client SHOULD present the other attachments subject to the control/preference of the user and device capability.

If the voicemail fetched from the EVVM Server contains "EVVM-EmotionIndication" header and the EVVM Client supports emotion indication then the EVVM Client SHALL present the message's emotion contained in the header value.

If the voicemail fetched from the EVVM Server contains any encrypted data, the EVVM Client SHALL first decrypt the encrypted data, before presenting the voicemail to the recipient.

If the voicemail fetched from the EVVM Server contains any signed data with digital signature(s), the EVVM Client SHALL first verify the digital signature(s), before presenting the voicemail to the recipient.

While presenting the voicemail, the EVVM Client can allow a user to reply to the presented voicemail with the identifier indicated by the Reply-to header in the voicemail or call back with the phone number indicated by the Callback-Number header in the voicemail.

#### 7.6.2.4.1 Processing unsupported media

If the EVVM Client fetches a voicemail containing unsupported media format(s), it SHALL be able to ignore the unsupported media and present the supported media item(s).

#### 7.6.2.5 Voicemail management

The EVVM Client SHALL support fetch, delete and store via IMAP according to the procedures specified in [RFC3501].

##### 7.6.2.5.1 Storing draft voicemails in the Server Domain

The EVVM Client MAY store a voicemail in the EVVM user's Draftbox using the IMAP APPEND command as specified in [RFC3501].

If the EVVM Client supports storing draft voicemails in the EVVM user's Draftbox, the EVVM Client SHALL also support retrieving a stored draft voicemail using the IMAP FETCH (or UID FETCH) command as specified in [RFC3501].

### 7.6.3 Greeting and Voice Signature management

The EVVM Client SHALL support all functionality described in section "2.7 Guidelines For Greetings And Voice Signature Management" in [GSMA VVM].

### 7.6.4 Delivery/read report handling

#### 7.6.4.1 Delivery report

If a copy of the sent voicemail is stored in the EVVM user's voicemail box, the EVVM Client SHOULD match the Original-Envelope-ID contained in the incoming DSN with the Message-ID contained in the original sent voicemail while displaying the DSN.

#### 7.6.4.2 Read report

If a copy of the sent voicemail is stored in the EVVM user's voicemail box, the EVVM Client SHOULD match the Original-Message-ID contained in the incoming MDN with the Message-ID contained in the original sent voicemail while displaying the MDN.

### 7.6.5 Notification handling

#### 7.6.5.1 Inband notifications

When the EVVM Client supports IMAP IDLE, the EVVM Client SHALL follow the notification procedures defined in [RFC2711].

When the EVVM Client supports IMAP NOTIFY, the EVVM Client SHALL follow the notification procedures defined in [RFC5465].

#### 7.6.5.2 Outband notifications

When the EVVM Client receives a notification directly via SMS, it SHALL follow the appropriate procedures in [GSMA VVM].

When the EVVM Client receives an outband notification via the OMA PUSH Enabler, the EVVM Client SHALL perform the following actions:

- Decrypt the notification if it is encrypted.
- Check that the content is a valid XML document. If the content is a valid XML document, the EVVM Client MAY proceed with parsing it, otherwise the EVVM Client SHALL discard the notification.
- Check the sequence number. If the sequence number is not sequentially increasing (the previous sequence number was 2 while this one is 4), the EVVM Client SHALL assume that a notification has been lost, discard the notification and it SHOULD connect to the EVVM Server and perform reconciliation.

- Check the client prefix vs. the client prefix stored in the notification settings (clientprefs/prefs/notifications). If the client prefix does not match the client prefix stored in the notification settings, the EVVM Client SHALL assume that the notification has been forged and discard the notification.
- Handle the notification based on the type of the notification as described in sections 7.6.5.2.1 "*Deactivation notification*", 7.6.5.2.2 "*Preferences notification*", 7.6.5.2.3 "*Synchronization notification*", and 7.6.5.2.4 "*Transcription notification*".

### 7.6.5.2.1 Deactivation notification

Following the general procedure described in section 7.6.5.2 "*Outband notifications*", the EVVM Client SHALL perform the following actions:

- Check that the identifier included in the notification refers to an administrative entity that is authorized to deactivate the client by comparing the identifier to the value(s) available in the service preferences (serviceprefs-final/prefs/notifications). If the identifier does not match any administrative entities in the service preference, the EVVM Client SHALL discard the notification.
- Check that the key included in the notification matches the key of the administrative entity by comparing the key to the key of the appropriate administrative entity in the service preferences (serviceprefs-final/prefs/notifications). If the key does not match the key of the administrative entity in the service preference, the EVVM Client SHALL discard the notification.
- Check the event type.
    - If the event type indicates permanent deactivation, the EVVM Client SHALL disconnect from the EVVM Server immediately and it SHALL NOT attempt to connect for any reasons.
    - If the event type indicates temporary deactivation, the EVVM Client SHALL disconnect from the EVVM Server immediately and it SHALL NOT attempt to connect until the amount of time indicated in the notification has passed.
- The EVVM Client SHALL check the wipe flag and if the wipe flag is 'true', the EVVM Client SHALL wipe all information associated with the service.

### 7.6.5.2.2 Preferences notification

Following the general procedure described in section 7.6.5.2 "*Outband notifications*", the EVVM Client SHALL perform the following actions:

- Check the identifier included in the notification. If the identifier does not match any application usage namespaces known to the EVVM Client, the notification SHALL be discarded.
- The EVVM Client SHALL retrieve a fresh copy of the referenced preferences document of the indicated application usage from the XDMS via its XDM Client.

### 7.6.5.2.3 Synchronization notification

Following the general procedure described in section 7.6.5.2 "*Outband notifications*", the EVVM Client SHALL perform the following actions:

- Check the amount of information in the notification.
    - If the amount of information is deemed to be sufficient, the EVVM Client SHOULD update its local representation of the voicemail storage using the information in the notification and avoid connecting to the EVVM Server.
    - If the amount of information is deemed to be insufficient, the EVVM Client SHOULD discard the notification, connect to the EVVM Server and perform reconciliation.

### 7.6.5.2.4 Transcription notification

Following the general procedure described in section 7.6.5.2 "*Outband notifications*", the EVVM Client SHALL perform the following actions:

- Check that the identifier included in the notification refers to an existing voicemail.
    - If the identifier refers to a voicemail known to the EVVM Client, the EVVM Client MAY store the transcription aside the voicemail and notify/present the transcription to the EVVM user.
    - If the identifier does not match any existing voicemails, the EVVM Client MAY hold the notification for a limited amount of time (in an attempt to cope with delays, expecting the voicemail to show up within a reasonable timeframe), connect to the EVVM Server and perform reconciliation, or discard the notification.

## 7.6.6 IMAP4 session management

### 7.6.6.1 Capability agreement

The EVVM Client SHALL store its client preferences document prior to accessing the EVVM Server containing its applicable capabilities, as described in section 7.3.2.2 "*Client preferences*".

The EVVM Client SHALL obtain a copy of the global service preferences document and a copy of the subscription-specific service preferences document that applies to its user - provided that these documents exist - prior to accessing the EVVM Server. When both global and subscription-specific service preferences documents exists, the EVVM Client SHALL calculate the final set of service preferences as described in section 7.3.2.1 "*Service preferences*". The EVVM Client SHALL evaluate each EVVM Server capability versus its own capabilities. The smallest common denominator (of the client capabilities versus the server capabilities) constitutes the capability agreement.

After successful authentication to the EVVM Server, the EVVM Client SHALL identify itself to the EVVM Server, as specified in section 7.6.6.1.1 "*Client identification*". The capability agreement SHALL remain valid throughout the session - even if the service preferences are updated in the meantime - until the EVVM Client disconnects from the EVVM Server.

#### 7.6.6.1.1 Client identification

The EVVM Client SHALL identify itself to the EVVM Server using the IMAP4 ID extension defined in [RFC2971]. This specification defines the following field name for this purpose:

```
client-id                       = EVVM Client-ID, as defined in section 7.3.1.2 "Client Identifier".
client-id-confirmation          = A simple confirmation from the EVVM Server.
```

To ensure that the result of the procedure is unambiguous and reliable, the EVVM Client SHALL verify the response. In case the server does not send a 'client-id-confirmation' field with the value 'done' in response, or, in case the result is 'BAD', the EVVM Client SHALL either disconnect or continue the session as if it was a client based on the [OMTP VVM] specifications (changing its behavior by disabling all functionality that is not available in [OMTP VVM]).

Example:

```
C: a005 ID ("evvm-client-id" "340457b0-b969-11e1-afa6-0800200c9a66")
S: * ID ("evvm-server-confirmation" "done")
S: a005 OK ID completed
```

### 7.6.6.2 Uninterrupted service in an intermittent-connectivity situation

The EVVM Client MAY employ the following techniques to maintain an uninterrupted EVVM-based service in an intermittent-connectivity situation.

When the EVVM Client detects an interruption in its connection with the EVVM Server, the EVVM Client MAY reserve the current state (defined in [RFC3501]) of the IMAP4 session and other temporary information. After that, the EVVM Client MAY attempt periodically to reconnect within a certain period of time. After the reconnection is established, the EVVM Client MAY continue with the previous session based on the stored IMAP4 state and temporary information.

The EVVM Client MAY support the quick-mailbox-resynchronization procedures defined in [RFC5162].

### 7.6.6.3 IMAP Session Termination

When the EVVM Client receives an untagged BYE response from the EVVM Server without initially requesting to log out, as specified in the IMAP protocol [RFC3501], the EVVM Client SHALL end the ongoing EVVM session.

## 7.6.7 Content adaptation

### 7.6.7.1 Text to speech conversion

The text to speech conversion is requested by the EVVM Client according to the EVVM user preferences (see chapter 7.3.2.3.12 "*Text to speech conversion*").

The text to speech content adaptation is used for generating audio parts (synthesized speech) for text based greetings or messages.

A greeting or message containing only text can be created and edited on the client, using existing composition procedures. When such a greeting or message is stored on the EVVM server the EVVM server triggers the text to speech conversion to get the corresponding synthesized audio. The resulting synthesized audio part is attached to the text based greeting or message and replaces that greeting or message.

Only if the EVVM user settings for text to speech conversion are switched on (see chapter 7.3.2.3.12 "*Text to speech conversion*") the EVVM Client is allowed to store the text based greeting in the greeting folder.

Only if the EVVM user settings for text to speech conversion are switched on (see chapter 7.3.2.3.12 "*Text to speech conversion*") the EVVM Client is allowed to store messages which contain only textual part into outbox / draft folder.

### 7.6.7.2 Voice to text conversion

The voice to text conversion is requested by the EVVM Client according to the user preferences (see chapter 7.3.2.3.13 "*Voice to text conversion*").

The user preferences, described in chapter 7.3.2.3.13 "*Voice to text conversion*", relate to start / stop procedures.

The voice to text content adaptation is used for generating text from voicemails.

If the user requests an on demand transcription then it will be realized via XTRANSCRIBE_ command with parameter UID as described in [GSMA VVM]. EVVM Client MAY support XTRANSCRIBE_ command as defined in [GSMA VVM].

If the XTRANSCRIBE_ command is supported the EVVM Client SHALL support XTRANSCRIPTION_SERVICE_ command with parameter STATE and its values START/STOP as defined in [GSMA VVM].

To identify the original voice message of transcription EVVM Client SHALL support X-Original-Msg-UID and X-Transcription message headers as defined in [GSMA VVM].

## 7.6.8 Spam Reporting

When one or more voicemails need to be reported as spam(s), the EVVM Client SHALL deposit a spam-reporting message into the EVVM user's Spambox via an IMAP4 APPEND command specified in [RFC3501]. The spam-reporting message constitutes the literal argument of the APPEND command, and is an Internet message whose format conforms to [RFC5322], with its body section consisting of one or more spam-reporting requests defined as below, in the ABNF format described in [RFC5234].

```
      spam-report       = “Action” “:” action-type “;” “UID” “=” voicemail-uid “;” “Type” “=” spam-type
      action-type       = “New” / “Withdraw” / “Update”
                                ; “New” means to report a new voicemail as a spam.
                                ; “Withdraw” means the voicemail is no longer a spam.
                                ; “Update” means to update the abuse type of the spam voicemail.
      voicemail-uid     = UID-of-the-spam-voicemail
      spam-type         = “phishing” / “malware”
                                ; The abuse type of the spam voicemail.
```

After reporting voicemails as spams, the EVVM Client SHALL be able to change the spam types or withdraw the spam reports of the reported voicemails by generating the corresponding spam-reporting messages based on the above defined syntax and depositing the generated messages into the EVVM user's Spambox via APPEND commands.

See section D.4 "*Examples of spam-reporting messages*" for the examples of submitting spam-reporting messages.

### 7.6.9 Securing voicemail contents using application-level encryption

Security of EVVM voicemail contents can be achieved by the originating and terminating EVVM Clients only, without the intervention (encryption, decryption, and security suite exchange) of the EVVM Server and the Forwarding Gateway. If a voicemail is to be forwarded via MMS or e-mail and security mechanisms are applied to the voicemail contents, the Forwarding Gateway SHOULD maintain the security of the voicemail contents in the generated MMS or e-mail message.

Considering that the level of support at the endpoints cannot be determined and the end-to-end encryption has no impact on the interoperability, the EVVM Enabler leaves the choice of the encryption mechanism out of scope; OpenPGP, as specified in [RFC3156] and [RFC4880], S/MIME as specified in [RFC5751], or any other suitable mechanism can be used. Implementers should take into account various characteristics of the intended service deployment.

## 7.7 Procedures at the EVVM Server

The EVVM Server provides the following procedures:

- Authentication, as described in section 7.7.1 “Authentication”,
- Voicemail handling, as described in section 7.7.2 “Voicemail handling”,
- Greeting and Voice Signature management, as described in section 7.7.3 “Greeting and Voice Signature management”,
- Delivery/read report handling, as described in section 7.7.4 “Delivery/read report handling”,
- Notification handling, as described in section 7.7.5 “Notification handling”,
- IMAP4 session management, as described in section 7.7.6 “IMAP4 session management”,
- Content adaptation, as described in section 7.7.7 “Content adaptation”,
- Spam reporting, as described in section 7.7.8 “Spam Reporting”,
- Managing the external storage, as described in section 7.7.9 “*Managing the external storage*”.

### 7.7.1 Authentication

The EVVM Server SHALL support the authentication procedures described in VVM 1.3 [GSMA VVM] with the clarifications included in section 6.4.1 “*Authentication*”.

The EVVM Server can establish a secure session with the EVVM Client. Upon receiving a STARTTLS request, the EVVM Server SHALL process the request and return a response according to [RFC3501]. The EVVM Server SHALL at least support TLS as defined in [RFC5246] and TLS-PSK as defined in [RFC4279] for the TLS protocol negotiations.

The EVVM Server SHALL allow the user to authenticate using any one of his/her associated identifiers. After the EVVM Server completes the authentication process, it SHALL check the identifier-association information (serviceprefs-subscription/prefs/associations) to determine if there are other user identifiers associated with this authenticated identifier. Having completed the authentication successfully the EVVM Server SHALL consider all other identifiers authenticated.

### 7.7.2 Voicemail handling

#### 7.7.2.1 Voicemail submission and delivery

The EVVM Server provides two methods for receiving voicemails as follows: The EVVM Server SHALL support the SMTP procedures specified in [RFC5321]; The EVVM Server MAY support receiving voicemails via IMAP. If references are present in the voicemail, the EVVM Server SHALL handle the references as speicified in section 7.7.2.2 “*Handling references in voicemails*”.

The EVVM Server MAY support forwarding voicemails to the Forwarding Gateway. Upon receiving a voicemail via SMTP or receiving a voicemail in an EVVM user’s Outbox via IMAP, the EVVM Server SHALL check the voicemail’s forwarding indication, i.e., the “NSID” parameter. If an “NSID” parameter is present in the To header field in the voicemail’s MIME header part, the EVVM Server SHALL forward the voicemail towards the Forwarding Gateway.

Otherwise, the EVVM Sever SHALL submit or deliver the voicemail according to each recipient’s identifier(s) obtained either from the SMTP RCPT command or, in the case of submission via IMAP, from the To header field in the voicemail, as detailed below:

- If a recipient is served by this EVVM Server,
    - The EVVM Server SHALL check the recipient user preferences and the service provider’s policies and determine if the voicemail is acceptable to the recipient.
        - i. If the voicemail sender is in the recipient’s black list, or the sender is NOT in the recipient’s white list, the EVVM Server SHALL discard the voicemail.
        - ii. If the voicemail sender has been determined as a spammer by service provider, the EVVM Server SHALL discard the voicemail.
        - iii. If the recipient EVVM Client is not available and the recipient user preferences indicate that the recipient prefers to receive voicemails as SMS, MMS or e-mail messages,
            - a) If the recipient prefers to receive voicemails via SMS, the EVVM Server MAY convert the voice item in the voicemail into text and discard other attachments (if any); otherwise, the EVVM Server MAY choose to perform or not to perform voice-to-text conversion according to the recipient user preferences.
            - b) The EVVM Server MAY send the voicemail towards the Forwarding Gateway, as specified in section 7.7.2.3 “*Voicemail forwarding*”.
    - Otherwise the EVVM Server SHALL store the voicemail in the recipient’s voicemail box (according to the recipient’s user preferences).
- If a recipient is NOT served by this EVVM Server, the EVVM Server SHALL act as a Mail Transfer Agent (MTA) and dispatch the voicemail message towards the recipient’s Server Domain via SMTP according to the recipient’s identifier, as specified in [RFC5321].

If the voicemail sender is served by this EVVM Server and a Sentbox is provided to the sender, the EVVM Server SHALL deposit a copy of the voicemail into the sender’s Sentbox while submitting or delivering the voicemail to the recipient.

The EVVM Server SHALL handle delivery and/or read reports according to section 7.7.4 “*Delivery/read report handling*”.

#### 7.7.2.2 Handling references in voicemails

Upon receiving an IMAP GENURLAUTH command, the EVVM Server SHALL respond with a URLAUTH URL based on the URL received in the GENURLAUTH command, as specified in [RFC4467] and [RFC5550].

Upon receiving an SMTP BURL command while receiving a voicemail via SMTP, the EVVM Server SHALL fetch the stored voicemail or part of it corresponding to the URL received in the BURL command and assemble the complete message by inserting the fetched media in place of the URL, according to [RFC5550].

Upon receiving an IMAP APPEND command with the CATENATE Extension for depositing a voicemail into the EVVM user’s Outbox, the EVVM Server SHALL fetch the stored voicemail or part of it corresponding to the URL contained in the received voicemail and assemble the complete message by inserting the fetched media in place of the URL, according to [RFC5550].

#### 7.7.2.3 Voicemail forwarding

To forward a voicemail through the Forwarding Gateway, the EVVM Server SHALL append a Non-VM Service Identifier (NSID) parameter to the To header field in the voicemail’s MIME header part and SHALL act as a Mail Transfer Agent (MTA) to forward the voicemail to the Forwarding Gateway.

The following ABNF expression defines the Non-VM Service Identifier as a parameter of the To header field:

```
NSID                        = "nsid=" ("sms" / "mms" / "email")
```

Example:

```
To: Bob@Example.COM; nsid=mms
```

### 7.7.2.4 Storing draft voicemails in the Server Domain

If the EVVM Server supports the Draftbox and a Draftbox is provided to the EVVM user, then the Server SHALL support the IMAP APPEND command as defined in [RFC3501], and the EVVM Server SHALL deposit the received voicemail into the EVVM user's Draftbox.

### 7.7.2.5 Voicemail future release

The EVVM Server MAY allow an EVVM user (sender) to request to send a voicemail at a future time.

An EVVM Server MAY support SMTP submission service extension, Future Message Release, as defined in [RFC4865], the related parameter (hold-param) includes:

- Future-release-interval, and
- Future-release-date-time.

If an EVVM Server supports this extension above, when an EVVM Client sends a command EHLO as defined [RFC2821], the EVVM Server SHOULD respond an EHLO reply, including a FUTURERELEASE keyword as defined in [RFC4865].

If an EVVM Server does not support this extension above, the EHLO reply does not include the FUTURERELEASE keyword.

When an EVVM Server receives a voicemail containing valid hold-param from an EVVM Client,

- If the EVVM Server supports the extension above, then,
    - The EVVM Server releases the voicemail to the recipient(s) when meeting the deadline as defined by the hold-param, and
    - The EVVM Server sends a delivery report to the EVVM Client if requested.
- If the EVVM Server does not support the extension above, then
    - The EVVM Server rejects to accept the voicemail.

## 7.7.3 Greeting and Voice Signature management

The EVVM Server SHALL support all functionality described in section "2.7 Guidelines For Greetings And Voice Signature Management" in [GSMA VVM].

## 7.7.4 Delivery/read report handling

### 7.7.4.1 Delivery reports

The EVVM Server SHALL generate a DSN message after receiving a voicemail if the voicemail requests a delivery report (see [GSMA VVM]).

When the EVVM Server receives a DSN message from the Forwarding Gateway or a remote EVVM Server and the DSN message is destined for an EVVM user served by this EVVM Server itself, the EVVM Server SHALL deposit the DSN message into the original voicemail sender's voicemail box according to the identifier carried in the SMTP RCPT To command.

### 7.7.4.2 Read reports

The EVVM Server SHALL generate a MDN message after receiving a voicemail containing a MDN request (see [GSMA VVM]).

When the EVVM Server receives a MDN message from an EVVM Client, the Forwarding Gateway, or a remote EVVM Server and the MDN message is destined for an EVVM user served by this EVVM Server itself, the EVVM Server SHALL deposit the MDN message into the original voicemail sender's voicemail box according to the identifier carried in the SMTP RCPT To command.

## 7.7.5 Notification handling

The EVVM Server SHALL notify EVVM Clients that are connected to the EVVM Server using inband notifications supported by both endpoints.

The EVVM Server SHALL notify EVVM Clients that are disconnected from the EVVM Server using outband notifications supported by both endpoints.

The EVVM Server SHALL NOT send SYNC notifications to EVVM Clients that are connected to the EVVM Server.

### 7.7.5.1 Inband notifications

When the EVVM Server supports IMAP IDLE, the EVVM Server SHALL follow the notification procedures defined in [RFC2711].

When the EVVM Server supports IMAP NOTIFY, the EVVM Server SHALL follow the notification procedures defined in [RFC5465].

### 7.7.5.2 Outband notifications

As defined in section 7.6.6.1 "*Capability agreement*", EVVM Clients are REQUIRED to store their preferences prior to accessing the EVVM Server. The EVVM Server SHALL treat clients that do not store their preferences (clientprefs/prefs) as VVM clients, based on the [GSMA VVM] specification. For VVM clients, the EVVM Server SHALL follow the notification procedures defined in [GSMA VVM]. For EVVM Clients, the EVVM Server SHALL follow the notification procedures defined in this section.

Whenever an event occurs that may result in an outband notification, the EVVM Server SHALL:

- Check the service provider policies in the service preferences (serviceprefs-final/prefs/notifications@deac, serviceprefs-final/prefs/notifications@pref, serviceprefs-final/prefs/notifications@sync, serviceprefs-final/prefs/notifications@trans).
  - If the policy does not allow (the value is 'false') sending the type of notification *this* event would yield, the EVVM Server SHALL NOT notify any of the user's clients.
- Check the user's choice in the user preferences (userprefs/prefs/notifications@home, userprefs/prefs/notifications@roaming, userprefs/prefs/notifications@intlroaming).
  - If the user does not allow (where the corresponding values are 'false') notifying any of his/her EVVM Clients residing in devices having a certain roaming condition (in home network, roaming nationally, roaming internationally) , the EVVM Server SHALL NOT notify the user's EVVM Clients residing in devices that are under the corresponding roaming condition.
- Check the service provider policies in the service preferences (serviceprefs-final/prefs/notifications@home, serviceprefs-final/prefs/notifications@roaming, serviceprefs-final/prefs/notifications@intlroaming).
  - If the policy does not allow (where the corresponding values are 'false') notifying EVVM Clients residing in devices having a certain roaming condition (in home network, roaming nationally, roaming internationally), the EVVM Server SHALL NOT notify EVVM Clients residing in devices that are under the corresponding roaming condition.
- Check each client's level of support in their client preferences (clientprefs/prefs/notifications@deac, clientprefs/prefs/notifications@pref, clientprefs/prefs/notifications@sync, clientprefs/prefs/notifications@trans).
  - The EVVM Server SHALL NOT send notifications that the respective recipient EVVM Clients do not support.

- Generate the notification content corresponding to the type of the event, as defined in section 7.3.6.3 "*Deactivation notification*", section 7.3.6.4 "*Preferences notification*", section 7.3.6.2.2.2 "*Outband sync notification*" or section 7.3.6.5 "*Transcription notification*".
- Apply client-specific transformation(s) to the notification payload as specified in section 7.7.5.2.1 "*Encrypting notifications*".

#### 7.7.5.2.1 Encrypting notifications

The EVVM Server SHALL:

- Check the service provider's policy on encryption (serviceprefs-final/prefs/notifications@encrypt).
- Check the client's choice on encryption (clientprefs/prefs/notifications@encrypt).
- If the client's choice for encryption is exactly the opposite of the service provider's policy (i.e. encryption is set to 'always' on one side while it is set to 'never' on the other side, or vice versa), then the EVVM Server SHALL NOT send the notification to the corresponding EVVM Client. For clarification purposes, a truth table has been included in section D.2 "*Truth table for notification encryption*"
- Select an encryption method supported by both the EVVM Server (serviceprefs-final/prefs/notifications/encryption) and the EVVM Client (clientprefs/prefs/notifications/encryption).
    - If a suitable encryption mechanism was found, the EVVM Server SHALL encrypt the notification payload as specified according to the selected encryption algorithm, using the appropriate keys (clientprefs/prefs/notifications/encryption/publickey or clientprefs/prefs/notifications/encryption/sharedsecret) depending on the characteristics of the chosen encryption mechanism (asymmetric or symmetric).
    - If a suitable encryption mechanism was not found, the EVVM Server SHALL NOT send the notification to the corresponding EVVM Client.

## 7.7.6 IMAP4 session management

### 7.7.6.1 Capability agreement

The EVVM Server SHALL disclose its applicable capabilities in its service preferences either globally, on a per-subscriber basis, or both, as described in section 7.3.2.1 "*Service preferences*" and store the applicable service preferences documents.

The EVVM Server SHOULD obtain a copy of the corresponding client preferences document when the EVVM Client identifies itself as described in section 7.7.6.1.1 "*Client identification*". If the client does not identify itself, the EVVM Server SHALL treat the client as a client based on the [GSMA VVM] specifications. If the EVVM Client identifies itself but the EVVM Server cannot obtain the client preferences document for some reason, it SHALL either disconnect the EVVM Client, or, continue while treating the client as a client based on the [GSMA VVM] specifications. In case the client preferences document has been successfully obtained, the EVVM Server SHALL evaluate each EVVM Client capability versus its own capabilities. The smallest common denominator (of the client capabilities versus the server capabilities) constitutes the capability agreement. The capability agreement SHALL remain valid throughout the session - even if the service preferences are updated in the meantime - until the EVVM Client disconnects from the EVVM Server.

#### 7.7.6.1.1 Client identification

The EVVM Server SHALL support the IMAP4 ID extension defined in [RFC2971] with the additional clarifications in section 7.6.6.1.1 "*Client identification*".

### 7.7.6.2 Uninterrupted service in an intermittent-connectivity situation

The EVVM Server MAY employ the following techniques to maintain an uninterrupted EVVM-based service in an intermittent-connectivity situation.

When the EVVM Server loses its connection with an EVVM Client abnormally, the EVVM Server SHALL reserve, for a certain period of time, the current state (defined in [RFC3501]) of the IMAP4 session, as well as other temporary information, related to that specific EVVM Client. If that EVVM Client can reconnect within that certain period of time, the EVVM

Server SHALL continue to handle the requests from that EVVM Client based on the stored IMAP4 state and temporary information.

The EVVM Server MAY support the quick-mailbox-resynchronization procedures defined in [RFC5162].

#### 7.7.6.3 Network-initiated service deactivation

The EVVM Server SHALL be able to deactivate the EVVM-based service unilaterally, depending on service provider policy. Before deactivating the service, the EVVM Server SHALL send an untagged BYE response to the EVVM Client to inform the client that the server is about to end the ongoing EVVM session, as specified in the IMAP protocol [RFC3501]. After that, to deactivate the service, the EVVM Server SHALL also send an outband deactivation notification, as specified in section 7.3.6.3 "*Deactivation notification*", to the EVVM Client.

### 7.7.7 Content adaptation

#### 7.7.7.1 Text to speech conversion

The text to speech conversion is performed by the EVVM Server itself or by delegation to a Transcoder entity.

If the EVVM user setting for text to speech conversion is switched on (see chapter 7.3.2.3.12 "*Text to speech conversion*"), the text part of the text based greeting or text based message will be converted into synthesized audio and attached to the greeting or message and subsequently synchronized with the EVVM Client.

#### 7.7.7.2 Voice to text conversion

The voice to text conversion is performed, if supported, by the EVVM Server itself or by delegation to a Transcoder entity.

The user preferences, described in chapter 7.3.2.3.13 "*Voice to text conversion*", relate to start / stop procedures.

An on demand transcription will be realized via XTRANSCRIBE_ command with parameter UID as described in [GSMA VVM]. The EVVM Server SHALL support XTRANSCRIBE_ command as defined in [GSMA VVM].

For backward compatibility with VVM 1.3 clients the EVVM Server SHALL support XTRANSCRIPTION_SERVICE_ command with parameter STATE and its values START/STOP as defined in [GSMA VVM].

To identify the original voice message of transcription EVVM Server SHALL support X-Original-Msg-UID and X-Transcription message headers as defined in [GSMA VVM].

#### 7.7.7.3 Content adaptation based on client preferences

Upon receiving an EVVM Client's request for fetching a voicemail, the EVVM Server SHALL adapt the contents of the voicemail to be delivered to that EVVM Client based on that client's capabilities and client preferences, e.g., adapt the enclosed media items to the media formats supported by that EVVM Client, or filter the media items larger than a certain size, and deliver the voicemail with the adapted contents towards that EVVM Client.

### 7.7.8 Spam Reporting

The EVVM Server SHALL assign a Spambox, a specially defined mailbox/folder, to each EVVM user. The EVVM Server SHALL support receiving spam-reporting messages in that EVVM user's Spambox via the EVVM Client's IMAP APPEND commands, as specified in section 7.6.8 "*Spam Reporting*".

The EVVM Server can process and report spam voicemails to the service provider. How to process and report the spam voicemails is out of scope, depending on the implementation and deployment.

### 7.7.9 Managing the external storage

This specification uses the CPM-MSG interface defined in [OMA CPM TS MS] to manage the external storage.

When the user (using an EVVM Client, a VVM client or the TUI) is managing his/her external storage via the EVVM Server, the EVVM Server SHALL authenticate and manage the external storage on behalf of the user. For any other purposes, the EVVM Server SHALL authenticate and manage the external storage on behalf of an administrative entity.

Addition information is available about the use of external storage in section D.3 "*Deployment Options*".

## 7.8 Procedures at the Forwarding Gateway

The EVVM Enabler allows EVVM users to forward voicemails to users of SMS, MMS or e-mail and request delivery/read reports. Forwarding can be initiated by the EVVM Client, by sending a request to the EVVM Server using the EVVM-1 interface. The recipient can also request, by setting his/her relevant user preferences, to receive voicemails as SMS or MMS messages, or receive them as regular e-mails in another e-mail box.

The Forwarding Gateway forwards voicemails using the EVVM-3 interface and deposits delivery/read reports via the EVVM-4 interface. Forwarding voicemails to the following non-VM communication services are in the scope of this specification:

- SMS,
- MMS,
- E-mail.

Upon receiving a voicemail including an "NSID" parameter with its value set to "sms", "mms" or "e-mail", the Forwarding Gateway SHALL forward the voicemail towards SMS, MMS or e-mail systems, respectively.

While forwarding a voice mail, if the header "EVVM-EmotionIndication" exists in the voice mail, the Forwarding Gateway SHALL discard this header, if supporting emotion indication.

While forwarding a voicemail to SMS/MMS/E-mail, the Forwarding Gateway SHALL convert the received voicemail into the appropriate message format based on the conversion rules defined in Sections 7.8.1 "*Forwarding voicemails via SMS*", 7.8.2 "*Forwarding voicemails via MMS*" and 7.8.3 "*Forwarding voicemails via E-mail*", for SMS, MMS and e-mail, respectively.

While forwarding a voicemail to MMS or E-mail, if the voicemail includes any encrypted data or digitally signed data, the Forwarding Gateway SHOULD keep those items and their formats in the generated MMS or e-mail message.

The Forwarding Gateway MAY reject to forward a voicemail to SMS, if the voicemail includes any encrypted content or digital signature.

### 7.8.1 Forwarding voicemails via SMS

When forwarding a transcribed EVVM voicemail message to an SMS system or routing an SMS status report back to the EVVM environment, the EVVM Forwarding Gateway SHALL act as an External Short Message Entity (ESME) according to [SMPP].

When the EVVM Forwarding Gateway receives a voicemail transcription message that is to be sent to an SMS user, it SHALL handle the voicemail as described in Section 7.8.1.1 "*EVVM VPIM to SMS*".

NOTE: the encapsulation of an EVVM voicemail complies with the Voice Profile for Internet Mail (VPIM) specified in [RFC3801].

The EVVM user MAY request delivery reports for forwarded voicemails. When the EVVM Forwarding Gateway receives from the SMS system an SMS status report that is associated with a previously forwarded voicemail, it SHALL handle the SMS status report as described in Section 7.8.1.2 "*SMS status report to Voicemail Delivery Report*".

#### 7.8.1.1 EVVM VPIM to SMS

When the Forwarding Gateway receives a transcribed voicemail message, it SHALL check the size of the message. If the size of the message allows the message to be sent as one [SMPP] submit_sm request, the Forwarding Gateway SHALL generate a submit_sm request based on the received message in accordance with [SMPP], with the clarifications given in Table 7 below, and send the submit_sm request towards the SM-SC. If the size of the received message is too large for the message to be sent as one [SMPP] submit_sm request, the message SHALL be divided into concatenated submit_sm(s), with each of them conforming to [SMPP] and the clarifications given in Table 7 below.

If there is a NOTIFY parameter contained in the ESMTP RCPT command and the Forwarding Gateway supports the processing of delivery reports, the Forwarding Gateway SHALL store the envelope identifier specified by the ENVID parameter of the ESMTP MAIL command, according to [RFC3461], and, when the Forwarding Gateway receives the submit_sm_resp response from the SM-SC, it SHALL store the message_id contained in the submit_sm_resp response and associate the message_id with the stored envelope identifier.

| **Submit_SM parameters** | **SMPP parameters status** | **VPIM headers and SMTP commands (headers from [RFC3801] unless otherwise noted)** | **Comment** |
|---|---|---|---|
| dest_addr_ton,<br>dest_addr_npi,<br>destination_addr | Mandatory | (SMTP) RCPT TO:<br>or<br>To:, Cc: | Translated by the Forwarding Gateway to the corresponding routable target user's address.<br>To and CC are not used if there is only one recipient. If there is more than one recipient, a Submit-SM request is generated for every single recipient, with each request's "dest_addr_ton", "dest_addr_npi" and "destination_addr" parameters set in accordance with its corresponding recipient. |
| source_addr_ton,<br>source_addr_npi,<br>source_addr | Mandatory | (SMTP) MAIL FROM: | Translated by the Forwarding Gateway to the corresponding routable originating user's address. |
| validity_period | Mandatory | BY (i.e., parameter of extended MAIL FROM command of Deliver By SMTP Service Extension [RFC2852]) | Set based on the value of the BY parameter, if present; otherwise, set as per service provider policy. |
| service_type | Mandatory | | Set by the Forwarding Gateway to NULL. |
| esm_class | Mandatory | | Set by the Forwarding Gateway to "Store and Forward" as defined in [SMPP]. |
| protocol_id | Mandatory | | Set by the Forwarding Gateway to the appropriate value defined in [SMPP] based on the network type (e.g., GSM, CDMA). |
| schedule_delivery_time | Mandatory | Deferred-Delivery [RFC4021] | Set to the date-time given in the Deferred-Delivery header, if present; otherwise, set to NULL for immediate delivery. |
| registered_delivery | Mandatory | (SMTP) The NOTIFY parameter of the ESMTP RCPT command [RFC3461] | Set to xxxxxx01 based on [SMPP] when the [RFC3461] NOTIFY parameter is set to "SUCCESS" or "SUCCESS,FAILURE";<br>Set to xxxxxx10 based on [SMPP] when the [RFC3461] NOTIFY parameter is set to "FAILURE";<br>otherwise, set to NULL. |
| replace_if_present_flag | Mandatory | | Set to NULL. |
| data_coding | Mandatory | | Set according to service provider policy. |
| sm_default_msg_id | Mandatory | | Set by the Forwarding Gateway to NULL. |
| sm_length | Mandatory | | Set by the Forwarding Gateway to be the length of the payload of the resulting short message. |
| short_message | Mandatory | VPIM transcription message (body of SMTP DATA protocol unit) [RFC5321] | NOTE: The mapping is restricted to only the text. |
| language_indicator | Optional | | Set to NULL. |
| sar_msg_ref_num | Optional | | Set by the Forwarding Gateway to the newly generated reference number to the concatenated SMs based on [SMPP]. |
| sar_segment_seq_num | Optional | | Set by the Forwarding Gateway to the ordering number of each concatenated SM based on [SMPP]. |
| sar_total_segments | Optional | | Set by the Forwarding Gateway to the total number of concatenated SMs based on [SMPP]. |

**Table 7: EVVM VM message to SMS - Submit_SM request details**

### 7.8.1.2 SMS status report to Voicemail Delivery Report

When the Forwarding Gateway receives a deliver_sm request with a SMS status report from the SM-SC, it SHALL perform the following:

1. Checking if there is the same stored message_id received in submit_sm_resp as the receipted_message_id parameter received in deliver_sm with the following clarification:
   a. If there is no matched message_id, then responding with an error towards the SM-SC, setting command_status header field to ESME_RINVMSGID based on [SMPP];
   b. Otherwise, continuing with the rest of the steps;
2. Generating a DSN message as a voicemail delivery report according to [RFC3461] based on the received [SMPP] deliver_sm request with the clarifications given in Table 9 below.
3. Sending the delivery report generated in Step 2 towards the EVVM Server in an SMTP transaction described in Table 8 below.
4. Responding with a deliver_sm_resp towards the SM-SC following the rules and procedures in [SMPP].

| **SMTP Command (DSN)** | **deliver_sm**<br>**[status report]** | **Comment** |
|---|---|---|
| MAIL From | source_addr_ton,<br>source_addr_npi,<br>source_addr | Translated by the Forwarding Gateway to the corresponding routable user's address. |
| RCPT To: | dest_addr_ton,<br>dest_addr_npi,<br>destination_addr | Translated by the Forwarding Gateway to the corresponding routable target EVVM user's address. |
| DATA | | The generated DSN (Table 9). |

**Table 8: SMS Status Report to EVVM Delivery Report - SMTP transaction details**

<table>
<tr><th>DSN header<br>[headers from [RFC3461] unless otherwise noted]</th><th>DSN Header status</th><th>deliver_sm<br>[status report]</th><th>Comment</th></tr>
<tr><td>From</td><td>Mandatory</td><td>source_addr_ton,<br>source_addr_npi,<br>source_addr</td><td>Translated by the Forwarding Gateway to the corresponding routable user's address.</td></tr>
<tr><td>To</td><td>Optional</td><td>dest_addr_ton,<br>dest_addr_npi,<br>destination_addr</td><td>Translated by the Forwarding Gateway to the corresponding routable target EVVM user's address.</td></tr>
<tr><td>Date</td><td>Mandatory</td><td></td><td>Set according to service provider policy</td></tr>
<tr><td>Subject</td><td>Optional</td><td></td><td></td></tr>
<tr><td>MIME Version</td><td>Mandatory</td><td></td><td>Set to "1.0 (Voice 2.0)"</td></tr>
<tr><td>Content-Type</td><td>Mandatory</td><td></td><td>Set to " multipart/report; report-type=delivery-status" according to [RFC3464].<br>Format of the multipart/report content ([RFC3464]):
<table>
<tr><th>Field</th><th>Status</th><th>deliver_sm<br>[status report]</th><th>Comment</th></tr>
<tr><td>Content-Type</td><td>Mandatory</td><td></td><td>Set to "message/delivery-status"</td></tr>
<tr><td>Original-Envelope-Id</td><td>Mandatory</td><td>receipted_message_id</td><td>Set to the original voicemail's envelope identifier stored corresponding to the message_id indicated by the receipted_message_id in deliver_sm</td></tr>
<tr><td>Reporting-</td><td>Mandatory</td><td></td><td>The identifier of the SM-SC</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>DSN header<br>[headers from [RFC3461] unless otherwise noted]</th><th>DSN Header status</th><th>deliver_sm<br>[status report]</th><th>Comment</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>MTA</td><td></td><td></td><td></td></tr>
<tr><td>DSN-Gateway</td><td>Mandatory</td><td></td><td>The identifier of the Forwarding Gateway</td></tr>
<tr><td>Final-Recipient</td><td>Mandatory</td><td>source_addr_ton, source_addr_npi, source_addr</td><td>Translated by the Forwarding Gateway to the corresponding routable user’s address.</td></tr>
<tr><td>Action</td><td>Mandatory</td><td>message_state</td><td>Set to the value corresponding to message_state (e.g., DELIVERED = delivered, UNDELIVERABLE = failed, etc.) based on service provider policy.</td></tr>
<tr><td>Status</td><td>Mandatory</td><td>message_state</td><td>Set to the status-code corresponding to message_state (e.g., DELIVERED = 2.0.0, UNDELIVERABLE = 5.0.0, etc.) based on service provider policy.</td></tr>
</table>
</td></tr>
</table>

**Table 9: SMS Status Report to EVVM Delivery Report**

Note: data_sm and data_sm_resp messages can be used as alternatives for submit_sm/deliver_sm and submit_sm_resp/ deliver_sm_resp, respectively, as described in [SMPP]. Similar rules and procedures can be derived if data_sm and data_sm_resp messages are used.

## 7.8.2 Forwarding voicemails via MMS

When forwarding a voicemail to an MMS system or routing an MMS delivery/read report back to the EVVM environment, the EVVM Forwarding Gateway SHALL act as an MMS relay server node in the MMS network (see [3GPP TS23.140]).

When the EVVM Forwarding Gateway receives an EVVM voicemail that is to be sent to an MMS user, it SHALL determine the address of the recipient’s MMS Relay/Server (e.g., via an ENUM/DNS query) and handle the voicemail as described in section 7.8.2.1 “*EVVM VPIM to MMS*”.

The EVVM user MAY request delivery reports for forwarded voicemails. When the EVVM Forwarding Gateway receives from the MMS network an MMS delivery notification (i.e., an MM4_delivery_report.REQ [3GPP TS23.140]) that is associated with a previously forwarded voicemail, it SHALL handle the MMS delivery notification as described in section 7.8.2.2 “*MMS Delivery Report to Voicemail Delivery Report*”.

The EVVM user MAY request read reports for forwarded voicemails. When the EVVM Forwarding Gateway receives from the MMS network an MMS read reply (i.e., an MM4_read_reply.REQ [3GPP TS23.140]) that is associated with a previously forwarded voicemail, it SHALL handle the MMS read reply as described in section 7.8.2.3 “*MMS Read Reply to Voicemail Read Report*”.

### 7.8.2.1 EVVM VPIM to MMS

When the Forwarding Gateway receives a voicemail, it SHALL perform the following:

1. Determining the address of the recipient’s MMS relay server in his/her home network (e.g., via an ENUM/DNS query).

2. Generating an MM4_forward.REQ request based on the received voicemail and in accordance with [3GPP TS23.140], with the clarifications given in Table 11 below.

3. Sending the MM4_forward.REQ message in an SMTP transaction with the details described in Table 10 below.

If there is a NOTIFY parameter contained in the ESMTP RCPT command, with its value set to "SUCCESS", "FAILURE" , "DELAY" or any combination of them, and the Forwarding Gateway supports the processing of delivery reports, then the Forwarding Gateway SHALL store the envelope identifier specified by the ENVID parameter of the ESMTP MAIL command, according to [RFC3461], and, after generating the Message ID for the MMS message, the Forwarding Gateway SHALL store this Message ID and associate this Message ID with the stored envelope identifier.

If there is a Disposition-Notification-To header in the received voicemail message and the Forwarding Gateway supports the processing of read reports, the Forwarding Gateway SHALL store the message-id contained in the received voicemail message, and, after generating the Message ID for the MMS message, the Forwarding Gateway SHALL store this Message ID and associate this Message ID with the corresponding voicemail's message-id.

| SMTP Command (MMS) | SMTP Command (VPIM) | Comment |
|---|---|---|
| MAIL From | MAIL From | |
| RCPT To: | RCPT To: | |
| DATA | | The generated MM4_forwards.REQ request (Table 11). |

**Table 10: VPIM to MMS - SMTP transaction details**

| MM4 Information element | MM4 Parameter status | VPIM headers and SMTP commands<br>[headers from [RFC3801] unless otherwise noted] | Comment |
|---|---|---|---|
| 3GPP MMS Version | Mandatory | | Set according to the MMS protocol version supported by the EVVM Forwarding Gateway. |
| Message Type | Mandatory | | Set to "MM4_forward.REQ" |
| Transaction ID | Mandatory | | Unique transaction identifier generated by the Forwarding Gateway. |
| Message ID | Mandatory | Message-id | Forwarding Gateway generates a unique Message ID for the MMS message, related to the Message-id of received VPIM. |
| Recipient(s) address | Mandatory | (SMTP) RCPT TO:<br>or<br>To, Cc | |
| Sender address | Mandatory | (SMTP) MAIL FROM: | |
| Content type | Mandatory | Content-Type | Set to multipart/mix if Content-Type is "multipart/voice-message", otherwise "multipart/related" |
| Date and time | Mandatory | Date | |
| Time of Expiry | Conditional | | Set according to service provider policy. |
| Delivery report | Conditional | (SMTP) The NOTIFY parameter of the ESMTP RCPT command [RFC3461] | Set to true when the [RFC3461] NOTIFY parameter is set to "SUCCESS", "FAILURE", "DELAY" or any combination of them. |
| Priority | Conditional | Importance | Set to corresponding value of Importance<br>non-urgent= low<br>normal= normal<br>urgent= high |
| Sender visibility | Conditional | | Set according to service provider policy |
| Read reply | Conditional | Disposition-Notification-To<br>[RFC3798] | Set to "Yes" if [RFC3798] "Disposition-Notification-To" exists in the received voicemail message. |
| Subject | Conditional | Subject | |
| Acknowledgement Request | Optional | | Set per service provider policy |
| Forward counter | Conditional | | Set per service provider policy. |
| Originator-System-Address | Optional | | Set by the Forwarding Gateway to its own address, formatted as an MMS R/S address. |
| Forward-Route | Optional | | Set by the Forwarding Gateway to its own address, formatted as an MMS R/S address. |
| Content | Conditional | Body of SMTP DATA | Set to corresponding body received in SMTP DATA. |

**Table 11: Voicemail to MMS - MM4_forward.REQ details**

### 7.8.2.2 MMS Delivery Report to Voicemail Delivery Report

When the Forwarding Gateway receives an MM4_delivery_report.REQ, it SHALL perform the following:

1. Generating a DSN message as a voicemail delivery report according to [RFC3461] based on the received MM4_delivery_report.REQ with the clarifications given in Table 13 below.
2. Sending the delivery report generated in Step 1 towards the EVVM Server in an SMTP transaction in Table 12 below.
3. Generating an MM4_delivery_report.RES response in accordance with [3GPP TS23.140], with the clarifications given in Table 15 below.
4. Sending the MM4_delivery_report.RES in an SMTP transaction, with the clarifications given in Table 14 below.

| SMTP Command (DSN) | SMTP Command (MMS) | Comment |
|---|---|---|
| MAIL From | MAIL From | |
| RCPT To: | RCPT To: | |
| DATA | | The generated DSN (Table 13). |

**Table 12: MMS Delivery Report to EVVM Delivery Report - SMTP transaction details**

<table>
<tr><th>DSN header<br>[headers from [RFC3461] unless otherwise noted]</th><th>DSN Header status</th><th>MM4 Information element</th><th>Comment</th></tr>
<tr><td>From</td><td>Mandatory</td><td>Recipient Address</td><td></td></tr>
<tr><td>To</td><td>Optional</td><td>Sender Address</td><td></td></tr>
<tr><td>Date</td><td>Mandatory</td><td>Date and Time</td><td>Set the Date per the MM4 Date and Time.</td></tr>
<tr><td>Subject</td><td>Optional</td><td></td><td></td></tr>
<tr><td>MIME Version</td><td>Mandatory</td><td></td><td>Set to “1.0 (Voice 2.0)”</td></tr>
<tr><td>Content-Type</td><td>Mandatory</td><td></td><td>Set to “ multipart/report; report-type=delivery-status” according to [RFC3464].<br>Format of the multipart/report content ([RFC3464]):
<table>
<tr><th>Field</th><th>Status</th><th>MM4 Information element</th><th>Comment</th></tr>
<tr><td>Content-Type</td><td>Mandatory</td><td></td><td>Set to “message/delivery-status”</td></tr>
<tr><td>Original-Envelope-Id</td><td>Mandatory</td><td>Message ID</td><td>Set to the original voicemail’s envelope identifier stored corresponding to the Message ID indicated in the received MM4_delivery_report.REQ</td></tr>
<tr><td>Reporting-MTA</td><td>Mandatory</td><td>Recipient address</td><td>The identifier of the original MMS message recipient</td></tr>
<tr><td>DSN-Gateway</td><td>Mandatory</td><td></td><td>The identifier of the Forwarding Gateway</td></tr>
<tr><td>Final-Recipient</td><td>Mandatory</td><td>Recipient address</td><td>Translated by the Forwarding Gateway to the corresponding routable user’s address.</td></tr>
<tr><td>Action</td><td>Mandatory</td><td>MM Status</td><td>Set to the value corresponding to MM Status (e.g., retrieved = delivered, rejected = failed, etc.) based on service provider policy.</td></tr>
<tr><td>Status</td><td>Mandatory</td><td>MM Status</td><td>Set to the status-code corresponding to MM Status (e.g., retrieved = 2.0.0,</td></tr>
</table>
</td></tr>
</table>

| DSN header<br>[headers from [RFC3461] unless otherwise noted] | DSN Header status | MM4 Information element | Comment |
|---|---|---|---|
| | | | rejected = 5.0.0, etc.) based on service provider policy. |

Table 13: MMS delivery report to EVVM Delivery Report

| SMTP/MM4_delivery_report.RES | Comment |
|---|---|
| MAIL From | Set by the Forwarding Gateway to its own SMTP address. |
| RCPT To: | Set by the Forwarding Gateway to the “Mail From” command parameter that was carried in the MM4_delivery_report.REQ. |
| DATA | The generated MM4_delivery_report.RES response (Table 15). |

Table 14: MM4_delivery_report.RES - SMTP transaction details

| MM4 Information element | MM4 Parameter status | Comment |
|---|---|---|
| 3GPP MMS Version | Mandatory | Set according to the MMS protocol version supported by the Forwarding Gateway. |
| Message Type | Mandatory | Set to “MM4_delivery_report.RES”. |
| Transaction ID | Mandatory | Set by the Forwarding Gateway to the “Transaction ID” received in the MM4_delivery_report.REQ. |
| Message ID | Mandatory | Set by the Forwarding Gateway to the “Message ID” received in the MM4_delivery_report.REQ. |
| Request Status | Mandatory | Set by the Forwarding Gateway to the appropriate value, as defined for X-Mms-Request-Statius-Code. |
| Originator-Recipient-Address | Optional | Set by the Forwarding Gateway, adding its own address (formatted as an MMS R/S address), after the one received in the MM4_delivery_report.REQ. |
| Return-Route | Optional | Set per the content of the Forward-Route received in the MM4_delivery_report.REQ. |

Table 15: MM4_delivery_report.RES details

### 7.8.2.3 MMS Read Reply to Voicemail Read Report

When the Forwarding Gateway receives an MM4_read_reply_report.REQ, it SHALL perform the following:

1. Generating MDN message as a voicemail read report according to [RFC3798] based on the received MM4_read_reply_report.REQ with the clarifications given in Table 17 below.
2. Sending the read report generated in step 1 towards the EVVM Server in an SMTP transaction in Table 16 below.
3. Generating an MM4_read_reply_report.RES response in accordance with [3GPP TS23.140], with the clarifications given in Table 19 below.
4. Sending the MM4_read_reply_report.RES in an SMTP transaction, with the clarifications given in Table 18 below.

| SMTP Command (MDN) | SMTP Command (MMS) | Comment |
|---|---|---|
| MAIL From | MAIL From | |
| RCPT To: | RCPT To: | |
| DATA | | The generated MDN (Table 17). |

Table 16: MMS Read Reply to EVVM Read Report - SMTP transaction details

<table>
<tr><th>MDN header<br>[headers from [RFC3801] and[RFC3798] unless otherwise noted]</th><th>VPIM Header status</th><th>MM4 Information element</th><th>Comment</th></tr>
<tr><td>From</td><td>Mandatory</td><td>Recipient address</td><td></td></tr>
<tr><td>To</td><td>Optional</td><td>Sender address</td><td></td></tr>
<tr><td>Date</td><td>Mandatory</td><td>Date and time</td><td>Set the Date per the MM4 Date and Time.</td></tr>
<tr><td>Subject</td><td>Optional</td><td></td><td></td></tr>
<tr><td>MIME Version</td><td>Mandatory</td><td></td><td>Set to “1.0 (Voice 2.0)”</td></tr>
<tr><td>Content-Type</td><td>Mandatory</td><td></td><td>Set to “Multipart/report; report-type=disposition-notification” according to [RFC3798].<br>Format of the multipart/report content (RFC[3798]):
<table>
<tr><th>Field</th><th>Status</th><th>MM4 Information element</th><th>Comment</th></tr>
<tr><td>content-type</td><td>Mandatory</td><td></td><td>Set to “message/disposition-notification”</td></tr>
<tr><td>Reporting-UA</td><td>Optional</td><td>Recipient address</td><td>The identifier of the original MMS message recipient</td></tr>
<tr><td>Final-Recipient</td><td>Mandatory</td><td>Recipient address</td><td>Translated by the Forwarding Gateway to the corresponding routable user’s address.</td></tr>
<tr><td>Original-Message-ID</td><td>Mandatory</td><td>Message ID</td><td>Set to the original voicemail’s message-id stored corresponding to the Message ID indicated in the received MM4_read_reply_report.REQ</td></tr>
<tr><td>MDN-Gateway</td><td>Mandatory</td><td></td><td>The identifier of the Forwarding Gateway</td></tr>
<tr><td>Disposition</td><td>Mandatory</td><td>Read Status</td><td>Set to the disposition-type corresponding to Read Status (e.g., set to “displayed” if Read Status is “Read”) based on service provider policy.</td></tr>
</table></td></tr>
</table>

**Table 17: MMS Read Reply to EVVM Read Report**

| SMTP/MM4_read-reply_report.RES | Comment |
|---|---|
| MAIL From | The Forwarding Gateway sets to its own SMTP address. |
| RCPT To: | Set to the “Mail From” command parameter that was carried in the MM4_read_reply_report.REQ. |
| DATA | The generated MM4_read_reply_report.RES response (Table 19). |

**Table 18: MM4_read_reply_report.RES - SMTP transaction details**

| MM4 Information element | MM4 Parameter status | Comment |
|---|---|---|
| 3GPP MMS Version | Mandatory | Set according to the MMS protocol version supported by the Forwarding Gateway. |
| Message Type | Mandatory | Set to “MM4_read_reply_report.RES”. |
| Transaction ID | Mandatory | Set to the “Transaction ID” received in the MM4_read_reply_report.REQ. |
| Message ID | Mandatory | Set to the “Message ID” received in the MM4_read_reply_report.REQ. |
| Request Status | Mandatory | Set to the appropriate value, as defined for X-Mms-Request-Status-Code. |
| Originator-Recipient-Address | Optional | The Forwarding Gateway adds its own address (formatted as an MMS R/S address), after the one received in the MM4_read_reply_report.REQ. |
| Return-Route | Optional | Set per the content of the Forward-Route received in the MM4_read_reply_report.REQ. |

**Table 19: MM4_read_reply_report.RES details**

### 7.8.3 Forwarding voicemails via E-mail

When forwarding an EVVM voicemail to an e-mail system or routing an e-mail Delivery Status Notification (DSN) or Message Disposition Notification (MDN) back to the EVVM environment, the EVVM Forwarding Gateway SHALL act as a Message Transfer Agent (MTA) according to [RFC5598]. The EVVM Forwarding Gateway SHALL relay the forwarded voicemail from the EVVM Server towards the recipient's Mail Delivery Agent (MDA) and relay the DSN and MDN back to the EVVM Server as the EVVM delivery and read reports, respectively, as specified in [RFC5321], [RFC3461], [RFC3798] and [RFC5598].

# 8. Release Information

## 8.1 Supporting File Document Listing

| Doc Ref | Permanent Document Reference | Description |
|---|---|---|
| **Supporting Files** | | |
| [EVVM_MO] | OMA-SUP-MO_EVVM-V1_0-20150915-A | Device Description File for EVVM MO. It is aligned with the [OMA EVVM MO] spec.<br>Working file in Managed Object directory:<br>file: evvm-mo-v1_0.ddf<br>path:http://www.openmobilealliance.org/tech/omna/evvm_mo |
| [XSD_CP] | OMA-SUP-XSD_EVVM_CP-V1_0-20150915-A | XML schema definitions for EVVM Client Preferences<br>Working file in schema directory:<br>file: evvm-cp-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_GSP] | OMA-SUP-XSD_EVVM_GSP-V1_0-20150915-A | XML schema definitions for EVVM Global Service Preferences<br>Working file in schema directory:<br>file: evvm-gsp-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_SSP] | OMA-SUP-XSD_EVVM_SSP-V1_0-20150915-A | XML schema definitions for EVVM Subscription-specific Service Preferences<br>Working file in schema directory:<br>file: evvm-ssp-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_UP] | OMA-SUP-XSD_EVVM_UP-V1_0-20150915-A | XML schema definitions for EVVM User Preferences<br>Working file in schema directory:<br>file: evvm-up-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_DEAC] | OMA-SUP-XSD_EVVM_DEAC-V1_0-20150915-A | XML schema definitions for EVVM Deactivation Notification<br>Working file in schema directory:<br>file: evvm-deac-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_PREF] | OMA-SUP-XSD_EVVM_PREF-V1_0-20150915-A | XML schema definitions for EVVM Preferences Notification<br>Working file in schema directory:<br>file: evvm-pref-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_SYNC] | OMA-SUP-XSD_EVVM_SYNC-V1_0-20150915-A | XML schema definitions for EVVM Synchronization Notification<br>Working file in schema directory:<br>file: evvm-sync-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |
| [XSD_TRAN] | OMA-SUP-XSD_EVVM_TRAN-V1_0-20150915-A | XML schema definitions for EVVM Transcription Notification<br>Working file in schema directory:<br>file: evvm-tran-v1_0.xsd<br>path:http://www.openmobilealliance.org/tech/profiles |

**Table 20: Listing of Supporting Documents in EVVM Release**

## 8.2 OMNA Considerations

### 8.2.1 OMNA PUSH Application ID

The following Push Application ID has been registered under the well known values:

| Number | URN | Description |
|---|---|---|
| 0x16 | x-oma-application:evvm.ua | PUSH Application ID is used while dispatching and receiving various OMA EVVM event notifications from the EVVM Server to the OMA EVVM Client, using the OMA Push Enabler. Requested by OMA EVVM WG and documented in the OMA EVVM 1.0 Enabler release. |

**Table 21: OMNA PUSH Application ID**

## 8.3 IANA Considerations

According to [RFC3501], IMAP4 capabilities are registered by publishing a standards track or IESG approved experimental RFC. The registry is currently located at: http://www.iana.org/assignments/imap4-capabilities

# Appendix A. Change History (Informative)

## A.1 Approved Version History

| Reference | Date | Description |
|---|---|---|
| OMA-ER-EVVM-V1_0-20150915-A | 15 Sep 2015 | Status changed to Approved by TP<br>TP Ref # OMA-TP-2015-0145-INP_EVVM_V1_0_ERP_for_final_Approval |

# Appendix B. Static Conformance Requirements (Normative)

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

## B.1 SCR for EVVM Client

| Item | Function | Reference | Requirement |
|---|---|---|---|
| EVVM-C-001-M | Composing a voicemail | 7.6.2.1 | |
| EVVM-C-002-O | Composing a voicemail including references to remote contents | 7.6.2.1, 7.6.2.3.1, 7.6.2.3.3 | |
| EVVM-C-003-O | Composing a voicemail including a background media item | 7.3.4.2 | |
| EVVM-C-004-O | Composing a voicemail including a remote background media item | 7.3.4.2, 7.6.2.1 | |
| EVVM-C-005-O | Including emotion indications in voicemails | 7.6.2.1 | |
| EVVM-C-006-O | Encrypting the voicemail or part of it while composing the voicemail | 7.6.2.1 | |
| EVVM-C-007-O | Including digital signature in the voicemail | 7.6.2.1 | |
| EVVM-C-008-M | Requesting a delivery report and a read report for a voicemail | 7.6.2.2, 7.6.2.3.1, 7.6.2.3.3 | |
| EVVM-C-009-M | Sending a voicemail via SMTP | 7.6.2.3.1 | |
| EVVM-C-009a-M | Sending a voicemail via IMAP | 7.6.2.3.3 | |
| EVVM-C-010-O | Sending a voicemail to a recipient possessing multiple associated identifiers | 7.6.2.3 | |
| EVVM-C-011-O | sending to a user a voicemail containing a background media item | 7.6.2.3 | |
| EVVM-C-012-M | Forwarding a voicemail without downloading it | 7.6.2.1, 7.6.2.3.1, 7.6.2.3.3 | |
| EVVM-C-013-M | Retrieving voicemails. | 7.6.2.5 | |
| EVVM-C-014-M | Retrieving voicemails with any one of the user’s identifiers after logging into the user’s voicemail box, if multiple identifiers are associated with the user’s voicemail box. | 7.6.2.5 | |
| EVVM-C-015-M | Associating delivery/read reports | 7.6.4 | |

| Item | Function | Reference | Requirement |
|---|---|---|---|
| | with the corresponding voicemails when displaying delivery/read reports. | | |
| EVVM-C-016-O | Presenting the correct emotion to the user according to the emotion indication in the voicemail being presented | 7.6.2.4 | |
| EVVM-C-017-O | playing simultaneously the voice and the background audio | 7.6.2.4 | |
| EVVM-C-018-M | Reporting a spam message without having to upload the whole voicemail | 7.6.8 | |
| EVVM-C-019-M | Reporting that a voicemail previously reported as spam is no longer spam | 7.6.8 | |
| EVVM-C-020-M | Being able to end IMAP session while receiving the network-initiated deactivation announcement. | 7.6.6.3 | |
| EVVM-C-021-M | Exchanging capability with the EVVM Server | 7.6.6.1 | |
| EVVM-C-022-O | Maintaining the service continuity in challenged environments | 7.6.6.2 | |
| EVVM-C-023-O | Voicemail Future Release | 7.6.2.3.2 | |

**Table 22: SCR for EVVM Client**

## B.2 SCR for EVVM Server

| Item | Function | Reference | Requirement |
|---|---|---|---|
| EVVM-S-001-M | Receiving voicemail via SMTP | 7.7.2.1 | |
| EVVM-S-002-O | Receiving voicemail via IMAP | 7.7.2.1 | |
| EVVM-S-003-M | Replacing the received URL with the corresponding media stored in the voicemail storage | 7.7.2.2 | |
| EVVM-S-004-M | Exchanging capability with the EVVM Client | 7.7.6.1 | |
| EVVM-S-005-M | Receiving spam report | 7.7.8 | |
| EVVM-S-006-O | Receiving and delivering voicemail requested for | 7.7.2.5 | |

| Item | Function | Reference | Requirement |
|---|---|---|---|
| | future release. | | |
| EVVM-S-007-M | Deactivating the service from the network side | 7.7.6.3 | |
| EVVM-S-008-O | maintaining an uninterrupted EVVM-based service in an intermittent-connectivity situation | 7.7.6.2 | |
| EVVM-S-009-O | Text to speech conversion | 7.7.7.1 | |
| EVVM-S-010-O | Voice to text conversion | 7.7.7.2 | |
| EVVM-S-011-M | Encrypting notifications | 7.7.5 | |
| EVVM-S-012-O | Managing the external storage | 7.7.9 | |

**Table 23: SCR for EVVM Server**

## B.3 SCR for Forwarding Gateway

| Item | Function | Reference | Requirement |
|---|---|---|---|
| EVVM-F-001-O | Forwarding voicemails via SMS | 7.8.1.1 | |
| EVVM-F-002-O | Forwarding SMS status reports as voicemail delivery reports | 7.8.1.2 | |
| EVVM-F-003-O | Forwarding voicemails via MMS | 7.8.2.1 | |
| EVVM-F-004-O | Forwarding MMS delivery reports as voicemail delivery reports | 7.8.2.2 | |
| EVVM-F-005-O | Forwarding MMS read replies as voicemail read reports | 7.8.2.3 | |
| EVVM-F-006-O | Forwarding voicemails via E-mail service | 7.8.3 | |
| EVVM-F-007-O | Preserving the integrity and confidentiality in the generated MMS or e-mail message when interworking with MMS or e-mail. | 7.8 | |

**Table 24: SCR for Forwarding Gateway**

# Appendix C. Flows (informative)

## C.1 Notifications

The flows in this section depict examples of exchanges described in section 7.3.6 “*Notifications*”.

### C.1.1 Client-initiated subscriber status notifications, VVM 1.3

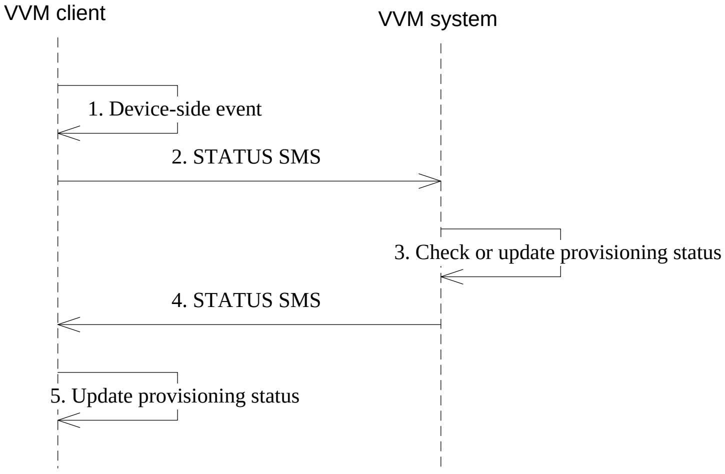
VVM client
VVM system
1. Device-side event
2. STATUS SMS
3. Check or update provisioning status
4. STATUS SMS
5. Update provisioning status

**Figure 5: Client-initiated subscriber status notifications, VVM 1.3**

1. A device-side event takes place, triggering the flow. For example, a new SIM card is inserted.

2. Client-originated SMS is sent to the server, querying, activating or deactivating the service,

3. Based on the contents of the client-originated SMS the VVM system checks or updates the provisioning status of the user.

4. Server-originated SMS is sent to the client, including information depending on the preceding request.

5. The VVM client is configured with the new information.

### C.1.2 Server-initiated subscriber status notifications, VVM 1.3

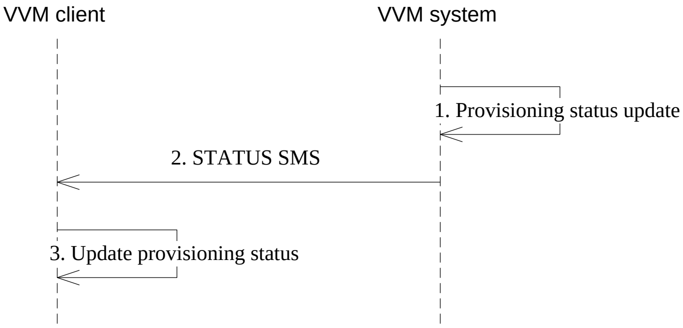

**Figure 6: Subscriber status notifications, VVM 1.3**

1. The provisioning status is updated on the server side without a request from the client, triggering the flow. For example, the password for the IMAP server was updated.

2. Server-originated SMS is sent to the client, including the updated information.

3. The VVM client is configured with the new information.

## C.1.3 Sync notification, VVM 1.3

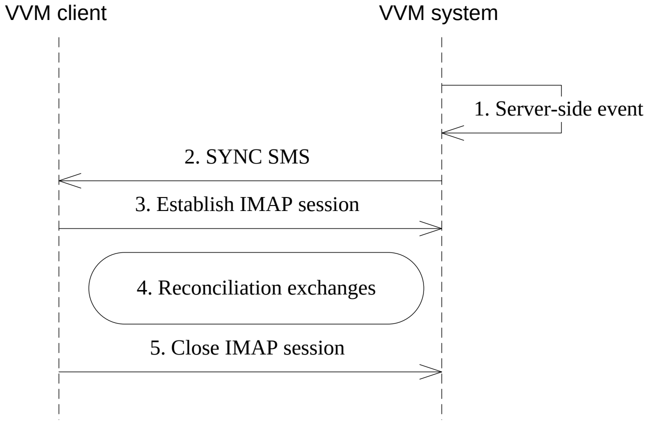

**Figure 7: Sync notification, VVM 1.3**

1. A service-side event takes place, triggering the flow. For example, a new voicemail has been deposited.
2. A server-originated SYNC SMS is sent to the client, containing information about the change in the voicemail box.
3. The client establishes an IMAP4 session.
4. Reconciliation takes place, updating the client with server-side changes and vice versa.
5. The client closes the IMAP4 session.

## C.1.4 Sync notification, EVVM 1.0

### C.1.4.1 Inband sync notifications

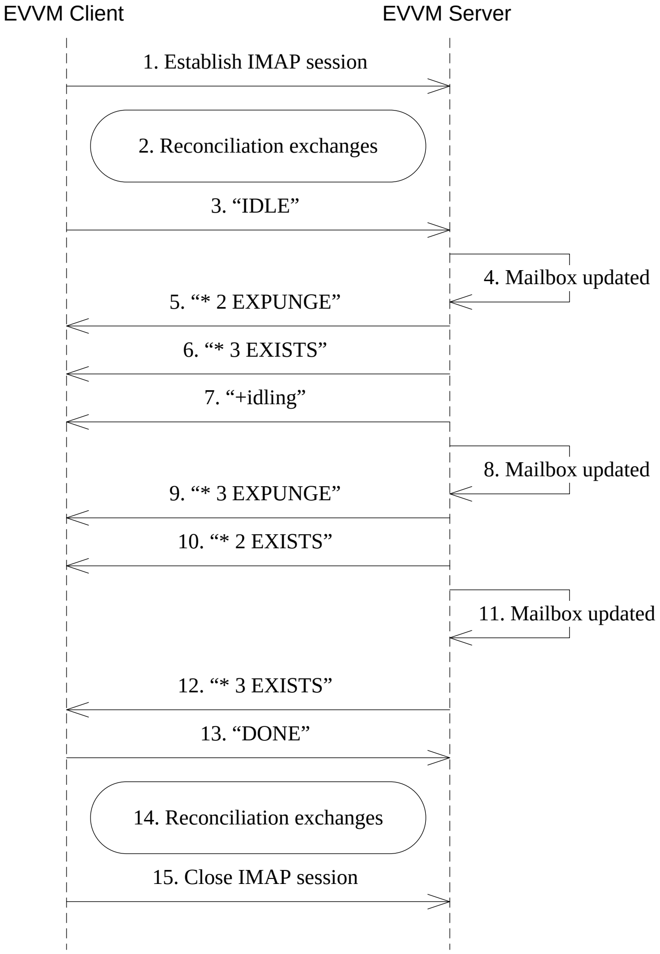

**Figure 8: Inband sync notification using IDLE**

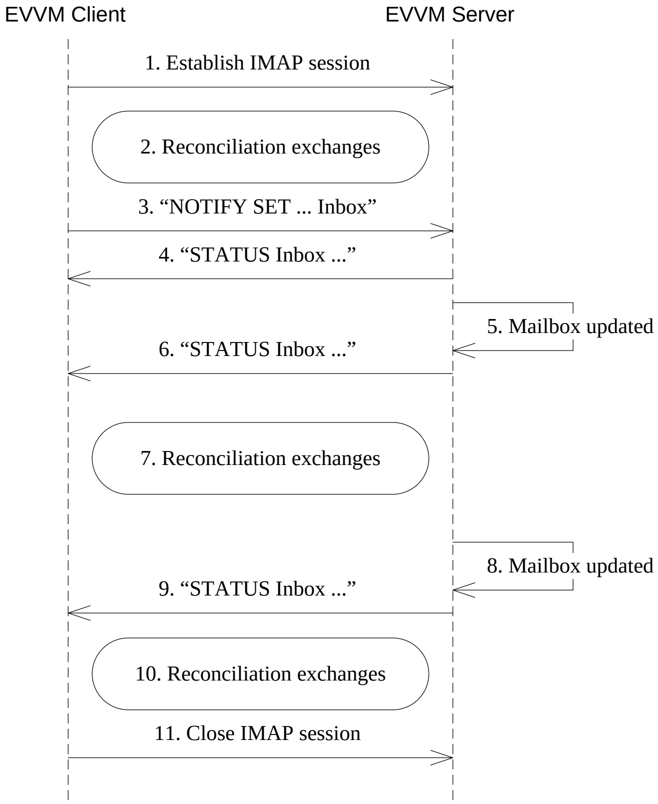

**Figure 9: Inband sync notification using NOTIFY**

### C.1.4.2 Outband sync notification

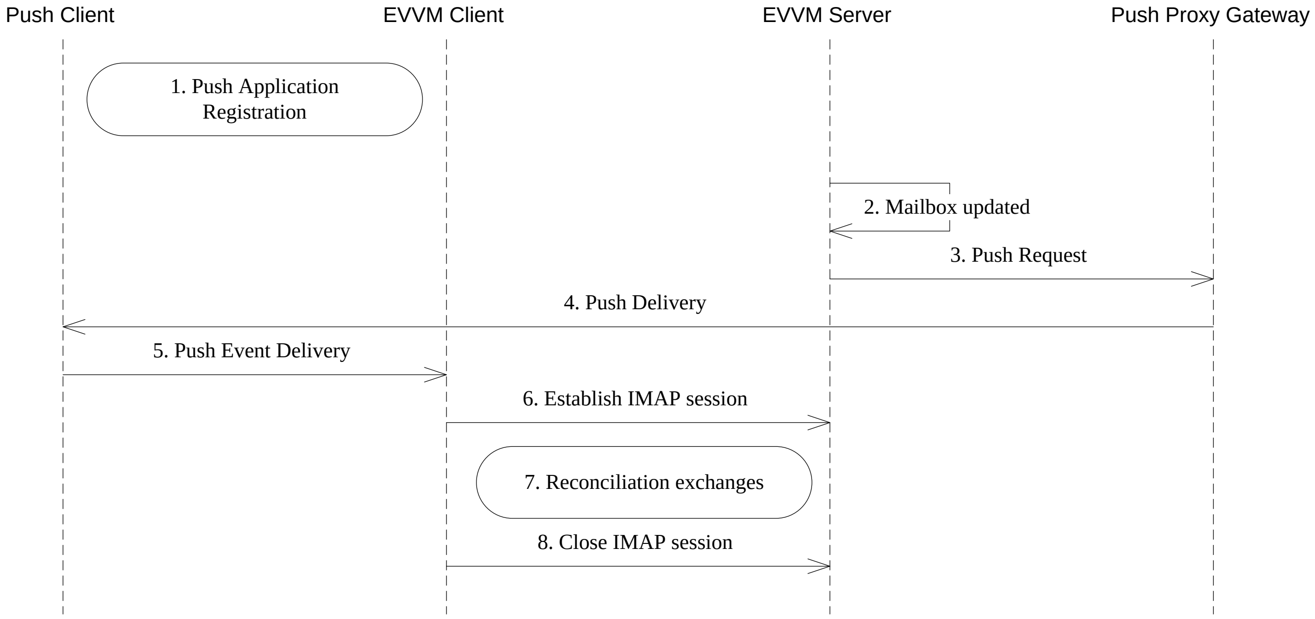

**Figure 10: Outband sync notification, EVVM 1.0**

1. The EVVM Client registers itself to the Push Client as a pre-requisite to receiving push notifications.

2. A service-side event takes place, triggering the main flow. For example, a new voicemail has been deposited.
3. The EVVM Server requests push delivery from the Push Proxy Gateway and hands over the notification payload (containing information about the change).
4. The Push Proxy Gateway delivers the notification to the Push Client (note: while the figure indicates unicast push delivery, this may actually be multicast push delivery).
5. The Push Client hands over the notification payload to the EVVM Client.
6. The EVVM Client establishes an IMAP4 session.
7. Reconciliation takes place, updating the client with server-side changes and vice versa.
8. The EVVM Client closes the IMAP4 session.

## C.1.5 XDM-EVVM notification framework

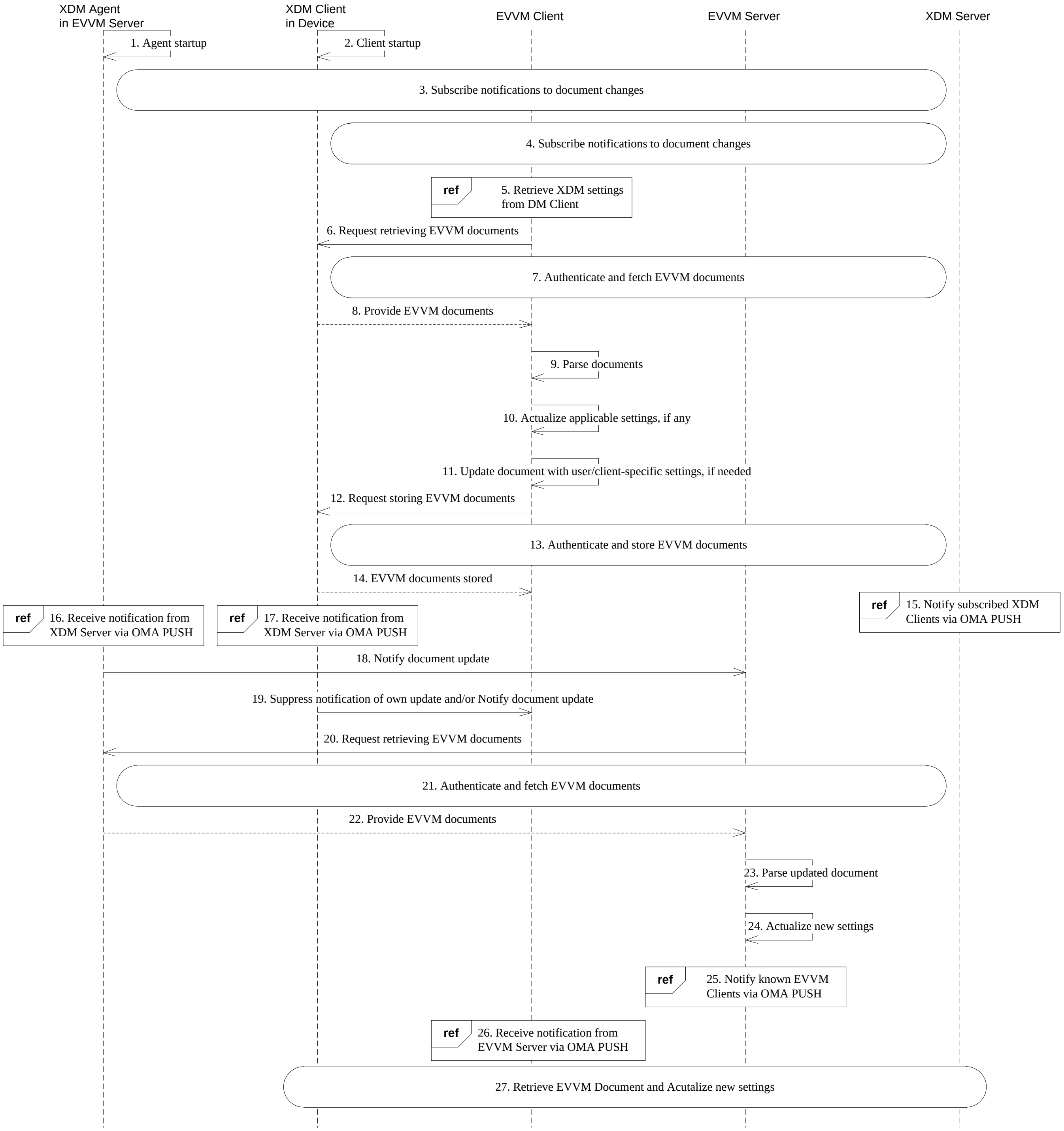
XDM Agent
in EVVM Server
XDM Client
in Device
EVVM Client
EVVM Server
XDM Server
1. Agent startup
2. Client startup
3. Subscribe notifications to document changes
4. Subscribe notifications to document changes
ref
5. Retrieve XDM settings from DM Client
6. Request retrieving EVVM documents
7. Authenticate and fetch EVVM documents
8. Provide EVVM documents
9. Parse documents
10. Actualize applicable settings, if any
11. Update document with user/client-specific settings, if needed
12. Request storing EVVM documents
13. Authenticate and store EVVM documents
14. EVVM documents stored
ref
15. Notify subscribed XDM Clients via OMA PUSH
ref
16. Receive notification from XDM Server via OMA PUSH
ref
17. Receive notification from XDM Server via OMA PUSH
18. Notify document update
19. Suppress notification of own update and/or Notify document update
20. Request retrieving EVVM documents
21. Authenticate and fetch EVVM documents
22. Provide EVVM documents
23. Parse updated document
24. Actualize new settings
ref
25. Notify known EVVM Clients via OMA PUSH
ref
26. Receive notification from EVVM Server via OMA PUSH
27. Retrieve EVVM Document and Acutalize new settings

**Figure 11: XDM-EVVM notification framework**

1. The XDM Agent in the EVVM Server starts up when the EVVM Server is started.

2. The XDM Client in the Device starts up when the Device is started.

3. The XDM Client in the EVVM Server subscribes to changes (in the EVVM XML documents) with the XDM Server.

4. The XDM Client in the Device subscribes to changes (in the EVVM XML documents) with the XDM Server.
5. When the EVVM Client is started, it receives the XDM settings from the DM Client in the Device.
6. The EVVM Client request retrieving the EVVM XML documents from the XDM Client.
7. The XDM Client retrieves the requested documents from the XDM Server (unless already cached).
8. The XDM Client returns the requested documents to the EVVM Client.
9. The EVVM Client parses the XML documents.
10. The EVVM Client actualizes applicable settings, if any.
11. The EVVM Client updates the XML documents with user or client-specific settings, if any.
12. The EVVM Client request updating the EVVM XML document(s) from the XDM Client.
13. The XDM Client updates the requested document(s) on the XDM Server.
14. The XDM Client informs the EVVM Client that the XML document(s) have been stored.
15. The XDM Server sends out notifications to all subscribed XDM Clients via the OMA PUSH Enabler.
16. The XDM Agent in the EVVM Server is notified about the update.
17. The XDM Client in the Device is notified about the update.
18. The XDM Agent in the EVVM Server notifies the EVVM Server.
19. The XDM Client in the Device suppresses the notification resulting from its own document update and/or notifies the EVVM Client.
20. The EVVM Server requests its XDM Agent to retrieve the updated XML document(s).
21. The XDM Agent retrieves the requested documents from the XDM Server.
22. The XDM Agent returns the requested documents to the EVVM Server.
23. The EVVM Server parses the XML documents.
24. The EVVM Server actualizes the new settings.
25. The EVVM Server sends PREF notifications to all known EVVM Clients via the OMA PUSH Enabler, informing them that the new settings can be applied.
26. The EVVM Client receives the PREF notification from EVVM Server via PUSH Client in the Device.
27. The EVVM Client obtains the documents that is retrieved by the XDM Client and takes the new settings into use.

## C.1.6 EVVM notification examples

### C.1.6.1 DEAC, temporary

Example of temporary deactivation event notification.

The entity identified as "admin@example.com" using key "MagicNumber" disables the client for 5 minutes.

```
<deac v="1.0" c="aBcDeF11223344" s="1">
   <d et="t" i="admin@example.com" k="MagicNumber" t="300"/>
</deac>
```

### C.1.6.2 DEAC, permanent

Example of permanent deactivation event notification.

The entity identified as "admin@example.com" using key "MagicNumber" disables the client permanently, including securely wiping EVVM data.

```
<deac v="1.0" c="aBcDeF11223344" s="1">
   <d et="p" i="admin@example.com" k="MagicNumber" w="true"/>
</deac>
```

### C.1.6.3 PREF

Example of preferences event notification.

The server:

– applied new global service preferences.

– applied new subscription-specific service preferences.

– applied new user preferences.

– applied new client preferences.

```
<pref v="1.0" c="aBcDeF11223344" s="1">
   <p i="urn:oma:xml:evvm:serviceprefs-global"/>
   <p i="urn:oma:xml:evvm:serviceprefs-subscription"/>
   <p i="urn:oma:xml:evvm:userprefs"/>
   <p i="urn:oma:xml:evvm:clientprefs"/>
</pref>
```

### C.1.6.4 SYNC, minimal

Example of a minimalistic synchronization event notification; a 'ping' if you like, without additional information.

There is no information about what happened. The client can connect and reconcile.

```
<sync v="1.0" c="aBcDeF11223344" s="2"/>
```

### C.1.6.5 SYNC, complex

Example of a complex synchronization event notification.

The notification includes details of the events that took place, allowing the client to avoid connecting to the server.

These details are:

A new 10seconds long 'happy' voice message has arrived from +1234567890 on the 13th of Oct, just before 5pm.

The message has been read and marked as deleted.

The message has been deleted (expunged); it is no longer available on the server).

There are 6 unread and 1 read messages in Inbox.

There are no unread and 35 read messages in Sentbox.

```
<sync v="1.0" c="aBcDeF11223344" s="1">
   <m i="1234@evvm.example.com" et="n" sn="+1234567890" d="20111013T164526Z" ml="10" e="happy"/>
   <m i="1234@evvm.example.com" et="f" f="\Seen \Deleted"/>
   <m i="1234@evvm.example.com" et="d"/>
   <f i="INBOX" r="1" u="6"/>
   <f i="SENT" r="0" u="35"/>
</sync>
```

### C.1.6.6 TRAN

Example of message transcription event notification.

The notification includes transcription of the message, allowing the user to avoid listening to the actual voice message.

Details:

The transcription is of the voice message that has arrived from +1234567890 on the 13th of Oct, just before 5pm.

The complete flag is 'false' indicating that the voice message contains at least one part that could not be transcoded - and the user might lose information by using the transcription only.

```
<tran v="1.0" c="aBcDeF11223344" s="1">
   <m i="1234@evvm.example.com" sn="+1234567890" d="20111013T164526Z" c="false">hi tom this is jack call
me tonight after six @ 1234567890</m>
</tran>
```

# Appendix D. Additional Information (informative)

## D.1 Figures

### D.1.1 Detailed Enabler Interactions

The following figure contains additional information to Figure 3, showing the main interfaces of each Enabler from the EVVM Enabler's perspective, as well as the interfaces between Enablers.

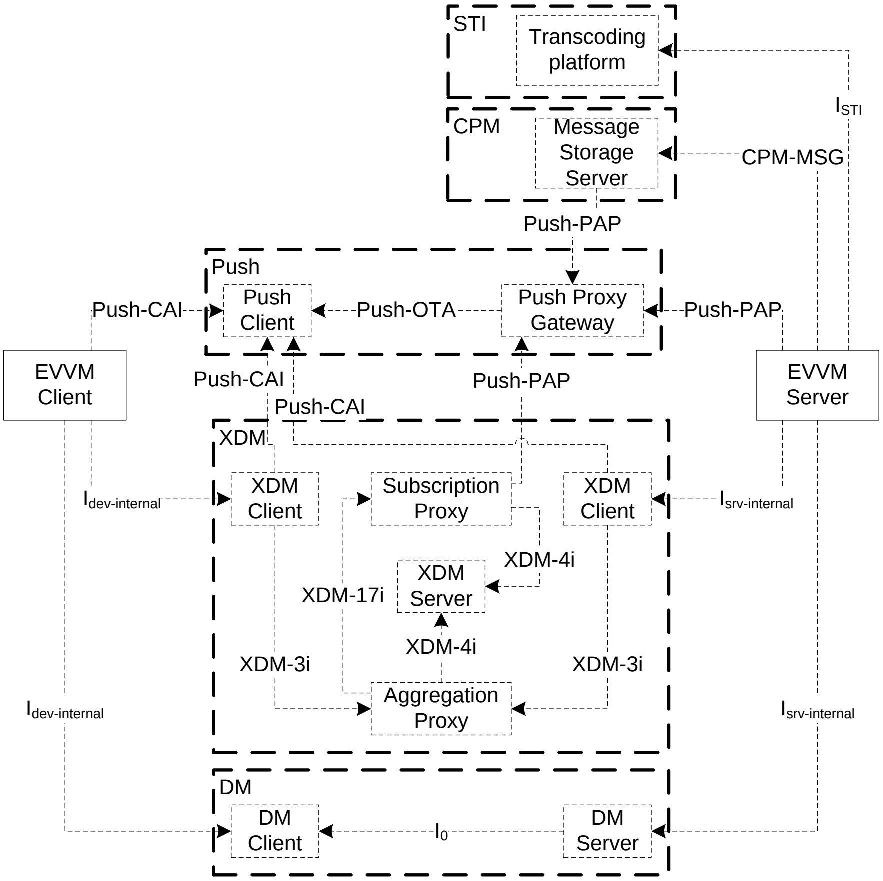

**Figure 12 - Detailed interactions between the EVVM Enabler functional components and the supporting Enablers.**

See Figure 3 for the detailed explanations of $I_{srv\text{-}internal}$ and $I_{dev\text{-}internal}$.

## D.2 Truth table for notification encryption

The following table provides clarification for the procedure described in section 7.7.5.2.1 “*Encrypting notifications*”.

Service provider preference: serviceprefs-final/prefs/notifications@encrypt

Client preference: clientprefs/prefs/notifications@encrypt

| Service provider preference | Client preference | Result |
|---|---|---|
| always | always | Encrypted notification will be sent |
| always | ifneeded | Encrypted notification will be sent |
| always | never | Notification will not be sent |
| ifneeded | always | Encrypted notification will be sent |
| ifneeded | ifneeded | Notification will be sent with or without encryption.<br>The EVVM Server decides whether it will be encrypted (possibly based on the transport used). |
| ifneeded | never | Notification will be sent without encryption |
| never | always | Notification will not be sent |
| never | ifneeded | Notification will be sent without encryption |
| never | never | Notification will be sent without encryption |

**Table 25: Truth table for notification encryption**

## D.3 Deployment Options

The EVVM Enabler and other IMAP-based-message-storing Enablers, such as the CPM Enabler, can share a common storage server. This deployment option allows the service provider to co-locate the voicemail messages with other stored objects of the subscribers in a centralized location, the External Message Storage Server. The External Message Storage Server provides the same functionalities as the CPM Message Storage Server (MSS). The External Message Storage Server is a stand-alone network-based server and, for EVVM, provides all functionalities required for the storage of the EVVM user’s voicemails, delivery/read reports, greetings and voice signatures and their attachments. The interface between the EVVM Server and External Message Storage Server shall be CPM-MSG, as defined in [OMA CPM TS MS].

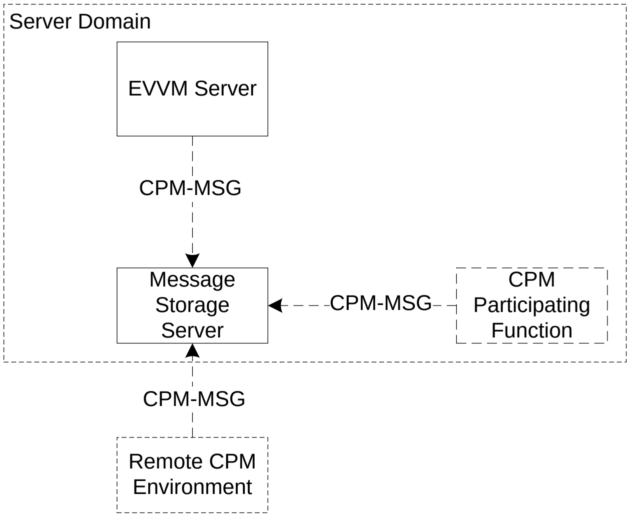

**Figure 13 – Deployment option: Using External Message Storage Server as voicemail storage**

The deployment scenario to include the CPM Message Storage Client in the EVVM capable device is shown in Figure 14. With this deployment option, the EVVM user will be able to directly access the stored objects in his/her network-based CPM Message Storage Server that is used as the EVVM stand-alone network-based server as described in Section 6.3.1.4 “*Message Storage Server*”.

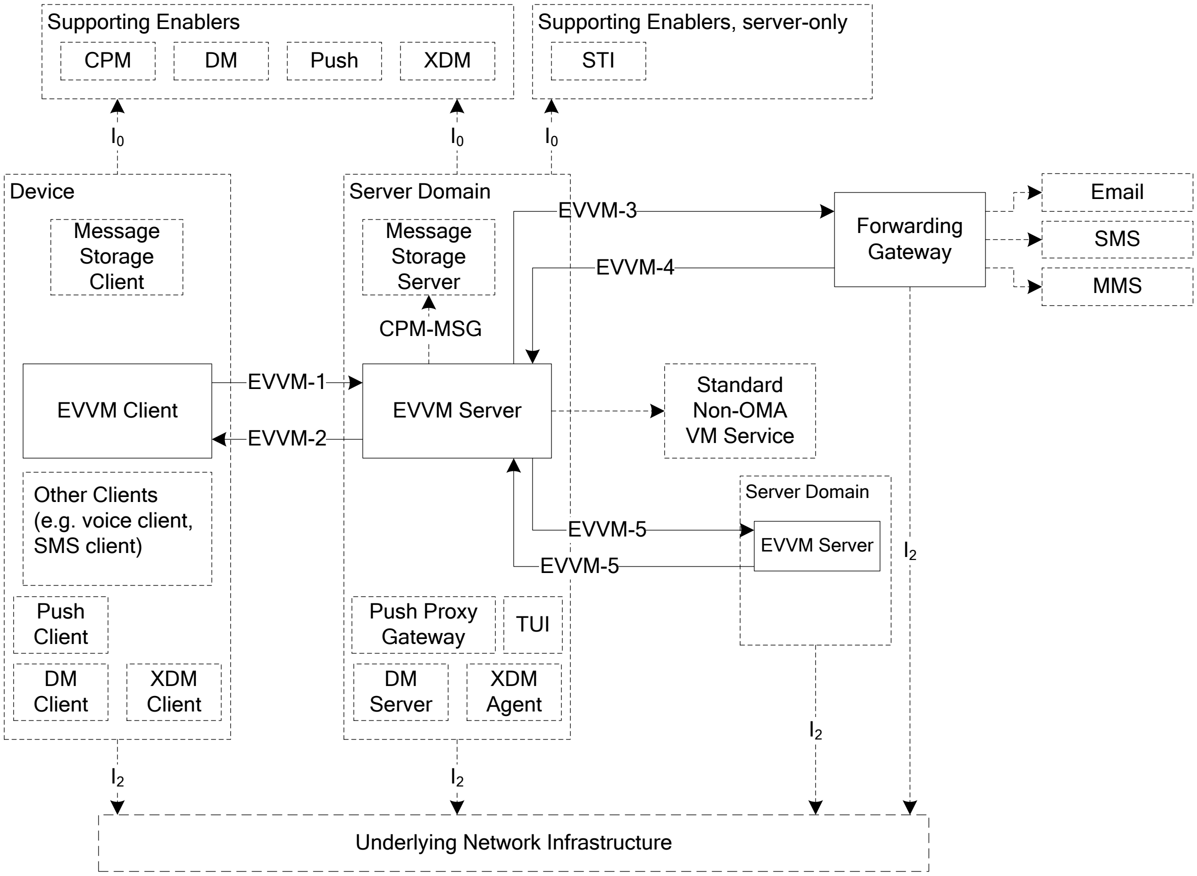

**Figure 14 – Depiction of the optional deployment in the OMA EVVM logical architecture**

In this external-storage situation, some features, such as authentication, voicemail management and voicemail storing procedures are needed to be specified with more details.

## D.3.1 Authentication

When the user (using an EVVM Client, a VVM client or the TUI) is managing his/her external storage via the EVVM Server, the EVVM Server shall authenticate and manage the external storage on behalf of the user. For any other purposes, the EVVM Server shall authenticate and manage the external storage on behalf of an administrative entity.

When the EVVM Server receives an authentication request from the EVVM Client, it should transfer the authentication request to the External Message Storage Server, on behalf of the EVVM user. After the EVVM Server receive the response from the External Message Storage Server, it needs to transfer the response to the EVVM Client.

## D.3.2 Voicemail management

The EVVM Server needs to act as a middleman to transfer the voicemail management requests and corresponding responses between the EVVM Client and the External Message Storage Server.

In response to the EVVM User/Client request, e.g., for fetching, deleting or storing voicemails via IMAP, as specified in section 7.6.2.5 "*Voicemail management*", the EVVM Server performs the corresponding operations to manage the stored objects in the user's voicemail folders. The objects include voicemails, delivery/read reports, greetings and voice signatures. These operations to manage the stored objects are described in [OMA CPM TS SD] and specified in [OMA CPM TS MS].

When the EVVM Server needs to resolve the references (to one or more stored voicemails) contained in a voicemail, the EVVM Server communicates with the MSS and fetches the referenced voicemails from the EVVM user's voicemail folders in the MSS, according to [OMA CPM TS SD] and [OMA CPM TS MS], for the subsequent delivery to the recipient.

### D.3.3 Voicemail Storing

When the EVVM Server needs to store message objects (including voicemails, delivery/read reports, greetings and voice signatures) in the EVVM user's voicemail folders, the EVVM Server performs the operations described in [OMA CPM TS SD] and specified in [OMA CPM TS MS], with the following clarifications:

- Upon receiving a request for storing a voicemail in the EVVM user's Draftbox, the EVVM Server communicates with the Message Storage Server (MSS) and stores the voicemail in the EVVM user's Draftbox in the MSS.
- Upon receiving a voicemail or delivery/read report that is to be delivered to an EVVM user served by this EVVM Server, the EVVM Server communicates with the MSS and stores the received voicemail or delivery/read report in the EVVM user's Inbox in the MSS, subject to the user preferences.
- Upon receiving the EVVM user's request for submitting a voicemail to another user, the EVVM Server performs the requested voicemail submission and, at the same time, communicates with the MSS and stores a copy of the voicemail in the EVVM user's Sentbox in the MSS, subject to the user preferences.
- Upon receiving a request for storing a greeting or voice signature in the EVVM user's Greetings folder, the EVVM Server communicates with the MSS and stores the greeting or voice signature in the EVVM user's Greetings folder in the MSS.

## D.4 Examples of spam-reporting messages

A spam-reporting message is defined in section 7.6.8 "*Spam Reporting*".

Here is an example of using the APPEND command defined in [RFC3501] to report voicemails as spams, where the voicemail with UID 12340 is reported as a phishing spam and the voicemail with UID 12350 is reported as a malware spam:

```
C: A003 APPEND Spambox
S: + Ready for literal data
C: Date: Mon, 12 March 2012 12:00:00 -0800 (PST)
C: From: Fred Foobar <foobar@Blurdybloop.COM>
C:
C: Action: New; UID=12340; Type=phishing
C: Action: New; UID=12350; Type=malware
C:
S: A003 OK APPEND completed
```

Here is an example of using the APPEND command defined in [RFC3501] to withdraw previous spam reports or change spam types, where the previous spam report regarding the voicemail with UID 12340 is withdrawn and the abuse type of the spam voicemail with UID 12350 is changed into "phishing":

```
C: A003 APPEND Spambox
S: + Ready for literal data
C: Date: Mon, 12 March 2012 12:00:00 -0800 (PST)
C: From: Fred Foobar <foobar@Blurdybloop.COM>
C:
C: Action: Withdraw; UID=12340; Type=phishing
C: Action: Update; UID=12350; Type=phishing
C:
S: A003 OK APPEND completed
```

NOTE: in the above examples, "C:" and "S:" indicate lines sent by the client and server respectively.