Push Proxy Gateway Service

Approved Version 2.3 – 22 Nov 2011

Open Mobile Alliance

OMA-TS- PPGService –V2_3-20111122-A

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Use of this document is subject to all of the terms and conditions of the Use Agreement located at http://www.openmobilealliance.org/UseAgreement.html.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the "OMA IPR Declarations" list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE "OMA IPR DECLARATIONS" LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Figures

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

1. Scope

This specification defines the functional requirements for Push Proxy Gateways (PPG), as part of the OMA Push enabler.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2. References

2.1 Normative References

[Mobitex] "Mobitex Interface Specification (MIS) ". Rev R4A. Document number LZY 232 105. Ericsson. URL://www.ericsson.com/

[PushArch] "Push Architecture". Open Mobile Alliance™. OMA-AD-Push-V2_3. URL:http://www.openmobilealliance.org/

[PushMsg] "Push Message Specification". Open Mobile Alliance™. OMA-TS-Push_Message-V2_3. URL:http://www.openmobilealliance.org/

[PushOTA] "Push OTA Protocol Specification". Open Mobile Alliance™. OMA-TS-PushOTA-V2_3. URL:http://www.openmobilealliance.org/

[PushPAP] "Push Access Protocol Specification". Open Mobile Alliance™. OMA-TS-PAP-V2_3. URL:http://www.openmobilealliance.org/

[RFC791] "Internet Protocol". J. Postel. September 1981. URL: http://www.ietf.org/rfc/rfc791.txt

[RFC822] "Standard for the Format of ARPA Internet Text Messages". David H. Crocker. August 1982. URL: http://www.ietf.org/rfc/rfc822.txt

[RFC1951] "DEFLATE Compressed Data Format Specification version 1.3". P. Deutsch. May 1996. URL: http://www.ietf.org/rfc/rfc1951.txt

[RFC1952] "GZIP file format specification version 4.3" . P. Deutsch. May 1996. URL: http://www.ietf.org/rfc/rfc1952.txt

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

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

[RFC3513] "IP Version 6 Addressing Architecture". R. Hinden, et al. July 1998. URL: http://www.ietf.org/rfc/rfc3513.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[RFC5627] "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)". J. Rosenberg, October 2009, URL:http://www.ietf.org/rfc/rfc5627.txt

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

[SIPPush] "Push using SIP". Open Mobile Alliance™. OMA-TS-SIP_Push-V1_0. URL: http://www.openmobilealliance.org/

[TS23.003] "Technical Specification Group Core Network and Terminals; Numbering, addressing and identification". 3rd Generation Partnership Project (3GPP). URL:http://www.3gpp.org

[TS23.041] "Technical realization of Cell Broadcast Service (CBS)". 3rd Generation Partnership Project (3GPP). URL:http://www.3gpp.org

[TS26.346] "Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS)". 3rd Generation Partnership Project (3GPP). URL: http://www.3gpp.org

[WBXML] "Binary XML Content Format Specification". WAP Forum™. WAP-192-WBXML URL:http://www.openmobilealliance.org/

[WDP] "Wireless Datagram Protocol". WAP Forum™. WAP-259-WDP. URL:http://www.openmobilealliance.org/

[WSP] "Wireless Session Protocol". WAP Forum™. WAP-230-WSP705 URL:http://www.openmobilealliance.org/

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Open Mobile Alliance™, OMA-ORG-Dictionary, URL:http://www.openmobilealliance.org/

[OMNA] "OMA Naming Authority". Open Mobile Alliance™. URL:http://www.openmobilealliance.org/tech/OMNA.aspx

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

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.

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

Within this document, `courier font` is used to identify literal names of elements, attributes, parameters, and values in referenced specifications. For example, the following table indicates that the `message-state` attribute contains the `"pending"` value.

PAP Attribute	Value
`message-state`	`"pending"`

3.2 Definitions

Application An implementation of a related set of functions that perform useful work, often enabling one or more services [OMADICT]

Application-Level Addressing the ability to address push content between a particular user agent on a client and push initiator on a server

Bearer Network a network used to carry the messages of a transport-layer protocol between physical devices. Multiple bearer networks may be used over the life of a single push session.

Client A device, user agent, or other entity that acts as the receiver of a service [OMADICT]

Contact Point address information that describes how to reach a push proxy gateway, including transport protocol address and port of the push proxy gateway.

Content Digitized work that is processed, stored, or transmitted. It includes such things as text, presentation, audio, images, video, executable files, etc. Content may have properties such as media type, mime type, etc [OMADICT]

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

Content Format actual representation of content.

Device Equipment which is normally used by users for communications and related activities. The definition can be extended to cover remote monitoring applications where there is no user present, but the communications to and from the remote monitor use the same communications channels as when used by users [OMADICT]

End-user An individual who uses services and content [OMADICT]

Multicast Message a push message containing a single address which implicitly specifies more than one OTA client address.

Point-to-Multipoint Push Push content delivery to a group of users through the OTA-PTM Push-OTA protocol variant.

Push Access Protocol a protocol used for conveying content that should be pushed to a client, and push related control information, between a Push Initiator and a Push Proxy/Gateway.

Push Framework the entire push system. The push framework encompasses the protocols, service interfaces, and software entities that provide the means to push data to user agents in the client.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Push Initiator	An entity or service that initiates Push content delivery to Push clients [OMADICT]
Push OTA Protocol	a protocol used for conveying content between a Push Proxy/Gateway and a certain user agent on a client.
Push Proxy Gateway	A gateway acting as a Push proxy for Push Initiators, providing over-the-air Push message delivery services to Push clients [OMADICT]
Push Session	A WSP session that is capable of conducting push operations.
Registration Context	a state where the PPG is aware of at least the last capabilities and preferences conveyed from the terminal.
Server	An entity that provides resources to clients in response to requests [OMADICT]
User	An entity which uses services. Example: a person using a device as a portable telephone [OMADICT]
User agent	Any software or device that acts on behalf of a user, interacting with other entities and processing resources [OMADICT]
WAP Push	Push content delivery to a specific user via the WAP1 (OTA-WSP) or WAP2 (OTA-HTTP) Push-OTA protocol variants.
XML	The Extensible Markup Language is a World Wide Web Consortium (W3C) standard for Internet markup language, of which WML is one such language [OMADICT]

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form
CBS	Cell Broadcast Service
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MAN	Mobitex Subscription Number
MBMS	Multimedia Broadcast Multicast Service
OMNA	Open Mobile Alliance Naming Authority
OTA	Over The Air
OTA-HTTP	(Push) OTA over HTTP
OTA-WSP	(Push) OTA over WSP
OTA-SIP	(Push) OTA over SIP
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
PTM	Point-to-Multipoint
QoS	Quality of Service
RFC	Request For Comments
SIP	Session Initiation Protocol
SIR	Session Initiation Request
URI	Uniform Resource Identifier
URI	Uniform Resource Locator
WAP	Wireless Application Protocol
WDP	Wireless Datagram Protocol

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

WSP	Wireless Session Protocol
WBXML	WAP Binary XML
XML	Extensible Mark-up Language

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

4. Introduction

The OMA Push enabler defines the functional requirements for architectural entities and interfaces enabling deployment of Push services. The OMA Push enabler architectural model and the enabler entities (PPG and Push Client), are introduced in [PushArch].

Figure 1 illustrates the OMA Push architectural model:

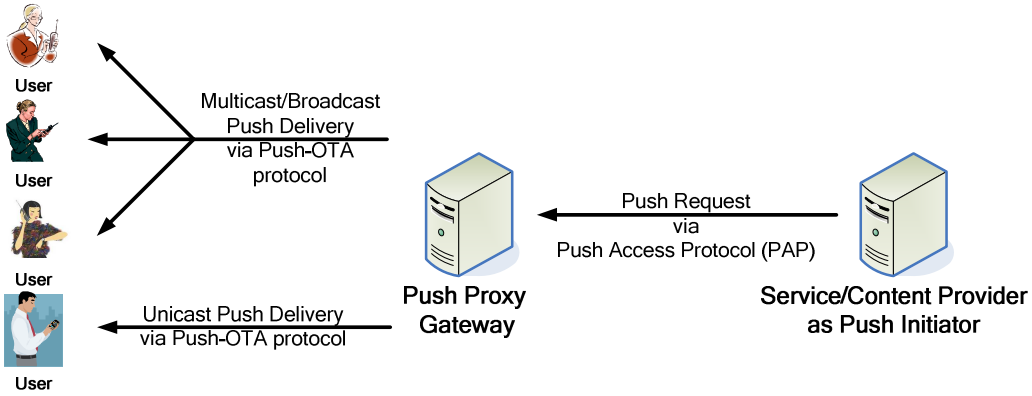

Figure 1: The Push Architectural Model

This specification focuses on the functions of Push Proxy Gateways (PPG). These functions are collectively referred to as PPG Service. They include:

- Push Access Protocol (PAP) interface operations
    - Push Submission (Push Message requests)
    - Result Notification
    - Status Query
    - Delivery Cancellation
    - Client Addressing
- Message transformation
- Push-OTA bearer and protocol selection
- OTA Session selection and creation
- Multi-terminal considerations

Figure 2 below provides a detailed end-to-end view of the service environment supported by the OMA Push enabler.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

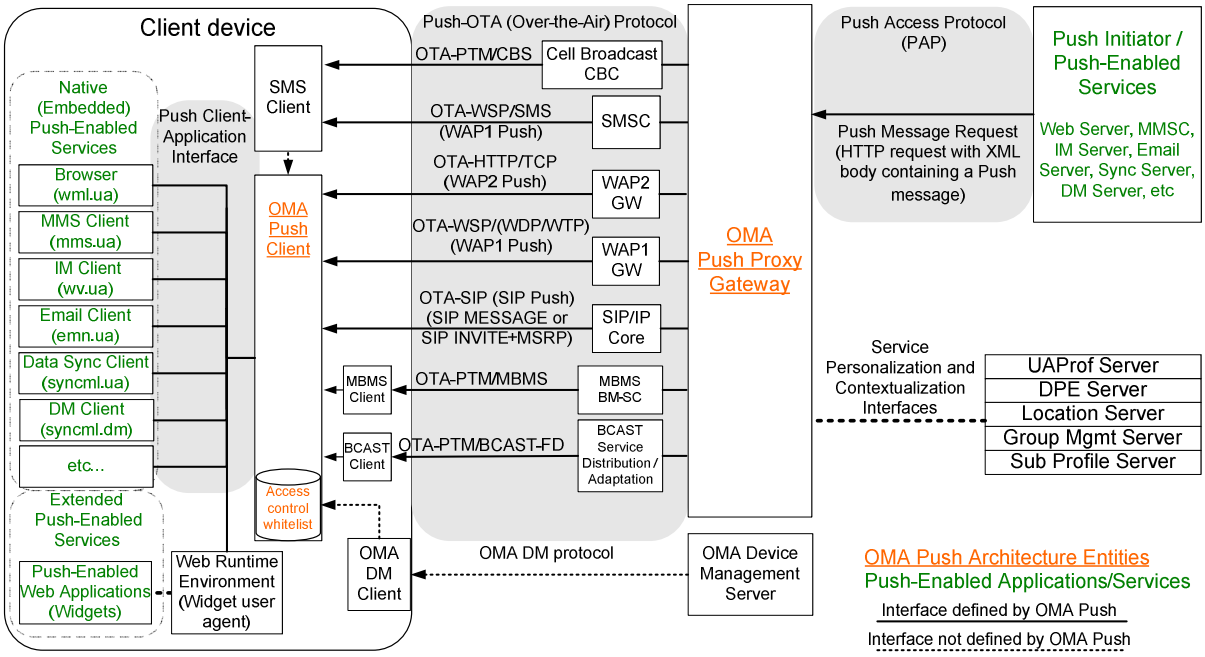

Figure 2: End-to-End Push Framework

4.1 Version 1.0

The PI is typically an application that runs on an ordinary web server. It communicates with the PPG using the *Push Access Protocol* (PAP). The PPG uses the *Push Over-The-Air* (OTA) *Protocol* to deliver the push content to the client. PAP is based on standard Internet protocols; XML is used to express the delivery instructions, and the push content can be any MIME media type. These standards help make WAP Push flexible and extensible.

The PPG is responsible for delivering the push content to the client. In doing so it potentially may need to translate the client address provided by the PI into a format understood by the mobile network, transform the push content to adapt it to the client's capabilities, store the content if the client is currently unavailable, etc. The PPG does more than deliver messages. For example, it may notify the PI about the final outcome of a push submission and optionally handle cancellation, replace, or client capability requests from the PI.

The OTA protocol relies upon Wireless Session Protocol (WSP).

The following diagram illustrates the push framework:

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

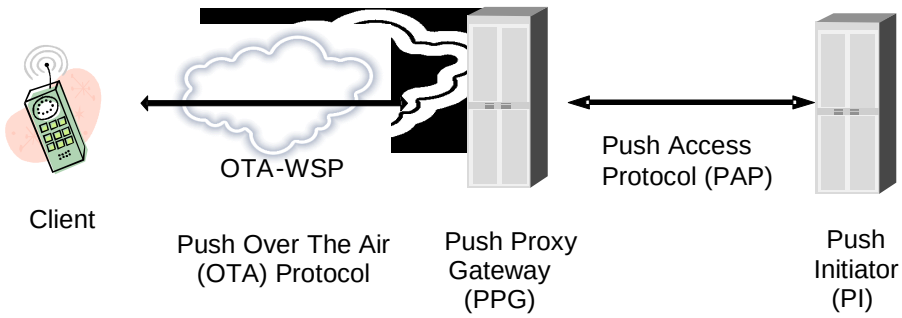

Figure 3: The Version 1.0 Push Framework

4.2 Version 2.0

WSP is suitable for use with low-bandwidth bearers that do not support TCP/IP, e.g. SMS, but can be used on top of TCP/IP as well. A new OTA (OTA-HTTP) variant is designed to run on top of HTTP 1.1 and is intended to be used in conjunction with bearers that support TCP/IP, e.g. GPRS, WIMAX or WLAN.

Figure 4 illustrates the enhanced push framework:

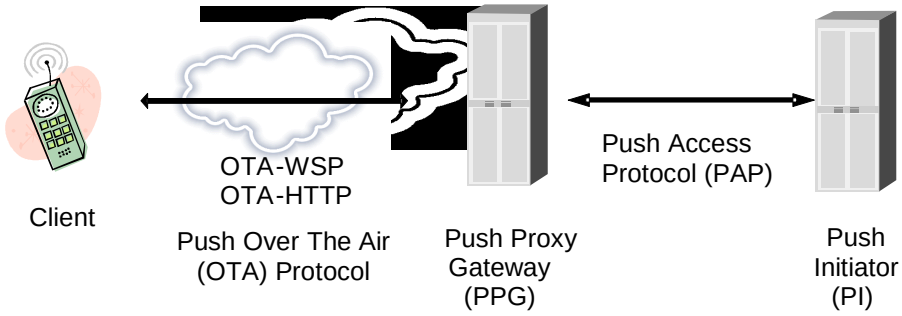

Figure 4: The Version 2.0 Push Framework

4.3 Version 2.1

The quality-of-service Element is extended to allow one-shot delivery and confirmed push with response. In the one-shot delivery service the PPG must attempt to deliver the message only once, and ensure that a one-shot delivery attempt can be made on the underlying bearer (e.g. not use the store-and-forward and retry capabilities provided by an SMSC). In the confirmed push with response service, the PI informs the PPG if it is ready to receive content from the client in response of the push request. The content is transferred to the PI in the corresponding resultnotification message.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

4.4 Version 2.2

In order to protect against denial of service attacks and push from unauthorized sources a Push Whitelist mechanism is defined. The Push Whitelist consists of a list of trusted PPGs and a list of trusted SMSCs. In addition to authenticating the source of the push message from a network perspective (either a PPG or intermediate SME / SMSC) a secondary mechanism may be used to verify the originating source of the content (either PI or PPG). The content is trusted due to it being 'signed' using a shared secret. This shared secret may be user defined or it might be some specific information that is related to the bearer or network.

A third Over The Air protocol is added to allow push messages be send over SIP in a convergent or IP only network.

The figure below illustrates the enhanced push framework:

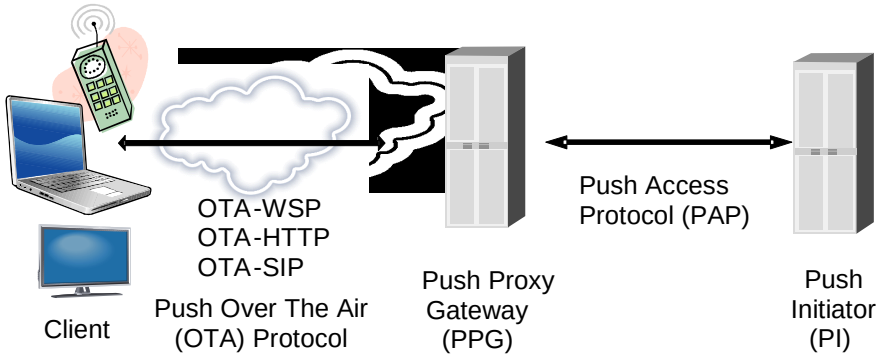

Figure 5: The Version 2.2 Push Framework

4.5 Version 2.3

This enabler release is an extension of the Push 2.2 Enabler release, and is referred to as Push 2.3 Point-to-Multipoint Push (PTM-Push). PTM-Push adds multipoint distribution methods to complement the existing point-to-point methods, enabling Push content delivery to a large number of clients simultaneously via network bearers supporting multicast and broadcast operation, e.g. MBMS, Cell Broadcast Service (CBS), and OMA BCAST.

New PPG service functions in version 2.3 include:

- General considerations of Push service in point-to-multipoint environments
- Push target addressing for Push messages to be delivered via PTM-Push

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5. PPG Operations

This section defines the operations performed by a PPG. These operations include push submission processing, result notification, delivery cancellation, and Push Access Protocol (PAP) status query.

PPG operations are defined as handling each push submission (and subsequent operations related to its push message) independently from other push submissions. However, there may be limited interaction between push submissions. For example, a PPG implementation MAY support multiple delivery priorities. This could cause one message to affect the time at which another (e.g., lower priority) message is processed, and consequently the ultimate success or failure of its delivery. Note that a PPG is not required to deliver push messages in any specific order.

5.1 Push Submission Processing

A Push Initiator (PI) triggers push message processing by sending the PPG a push message. Push submission processing includes four operations. The following three operations must be performed in order:

- push submission acceptance or rejection,
- over-the-air message delivery, if the message is accepted and can be delivered in accordance with PPG policies and PI requirements; and
- message delivery result notification, if the message is accepted and the push initiator has requested message delivery notification.

The fourth operation may, as determined by the PPG implementation, be performed at any time after push message acceptance:

- PAP push message response.

These four functions are described in this section.

5.1.1 Push Submission Acceptance or Rejection

Each PAP push submission received by the PPG is either accepted or rejected.

The PPG SHOULD accept a PAP push submission if it might ultimately be delivered to the OTA client. The PPG MUST reject any push submission containing a PAP `push-message` element that is not valid with respect to its document type definition (DTD). Additional criteria used to determine whether to accept or reject a `push-message` are implementation dependent.

An accepted, undelivered PAP push submission for which message handling (described in the next section) for over-the-air delivery have not been completed MUST have the following message status reportable:

PAP Attribute	Value
`message-state`	`"pending"`

5.1.1.1 Replacement of a Previously Submitted Push Message

This OPTIONAL function allows replacement of a previously submitted, still pending push message.

If the PPG supports replacement, and the message is in a state from which message replacement may be assured, the PPG MUST replace the message as requested by a PAP `push-message` message.

A PPG that does not support the replace operation MUST reject the push submission if the PI requests replacement.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.1.1.2 Request for Content from the Client

The PPG must support OTA-HTTP, OTA-SIP or PTM-Push to adequately serve a `push-message` indicating that the PI accepts content from the client in response to a confirmed push (by setting the `delivery-method` attribute to "`confirmed-with-response`" in the `quality-of-service` element). A PPG that does not support OTA-HTTP, OTA-SIP or PTM-Push MUST reject the push submission if the PI uses this feature.

5.1.2 Over-the-Air Message Delivery

Over-the-Air message delivery consists of two functions:

- Message handling
- Over-the-air message transmission.

These functions are described in this section.

5.1.2.1 Message Handling

The PPG may transform the push message entity contained within the push submission in preparation for over-the-air transmission. Typical reasons for transformation include compilations/optimisations for over-the-air efficiency, and translation of entities to a content type acceptable to the client. This section describes the transformations.

5.1.2.1.1 Entity and Header Transformation

A PPG MUST NOT transform the body of any entity, which falls under the scope of a No-Transform cache control directive as defined in [RFC2616]; otherwise, a PPG MAY translate entities in an implementation-dependent manner. The headers of all transformed entities MUST be revised as needed to correctly represent the transformed entity. All transformations MUST be in conformance with the requirements of [PushPAP].

5.1.2.1.1.1. WSP specific transformations

A PPG MUST support binary header encoding as specified in [WSP]. It MUST also encode content entities into their compact binary format [WBXML] (if such is specified) for transmission over OTA-WSP [PushOTA], unless it is positively known that the addressed terminal supports the non-encoded format. For example, Service Indication must usually be encoded into WBXML [WBXML] when delivered in connectionless mode.

5.1.2.1.1.2. HTTP specific transformations

A PPG SHOULD support content encoding for OTA transmission over OTA-HTTP [PushOTA] in order to minimize the volume of data sent over the air. When supported, the PPG MUST support deflate coding as specified in [RFC1951].

5.1.2.1.1.3. SIP specific transformations

A PPG SHOULD support content encoding for OTA transmission over OTA-SIP [PushOTA] in order to minimize the volume of data sent over the air. When supported, the PPG MUST support deflate coding as specified in [RFC1951]. Content entities MAY be encoded into their compact binary format [WBXML] when it is positively known that the addressed terminal supports the encoded format.

5.1.2.1.1.4. PTM specific transformations

A PPG SHOULD support content encoding for OTA transmission over OTA-PTM [PushOTA] in order to minimize the volume of data sent over the air. When supported, the PPG MUST support using the generic GZip algorithm [RFC1952]. UEs shall support GZip content decoding of FLUTE files [RFC1952].

5.1.2.1.2 `X-Wap-Application-Id` (Application-ID) header processing

A PPG MUST process a `X-Wap-Application-Id` (Application-ID) header as follows:

> If the header contains a `absoluteURI` format Application-ID for which an `app-assigned-code` has been registered with [OMNA], the PPG MUST remove any `app-assigned-code` format Application-ID (if present)

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

from the header and then substitute the registered `app-assigned-code` format Application-ID for the `absoluteURI` format Application-ID.

If the header contains a `absoluteURI` format Application-ID for which no `app-assigned-code` has been registered with [OMNA], the PPG MUST use this value unless a `app-assigned-code` format Application-ID is present. In this case (if the `app-assigned-code` format Application-ID is present), the `absoluteURI` format Application-ID must be removed.

A header containing only a `app-assigned-code` format Application-ID requires no substitutions or deletions.

If the resulting header identifies a default application known to the client, the PPG MAY delete this header.

If no `X-Wap-Application-Id` header is present in the push message, the PPG MUST, unless the client's default Application-ID is the WML user agent, add this header. If added, the Application-ID MUST be that of the WML user agent.

If OTA-SIP is used then a PPG MUST add the X-WAP-Application-Id header value in the URN form into the g.oma.pusheventapp feature tag as Push Resource Identifier. If it is a well known value registered with OMNA then only the Namespace Specific String SHOULD be used. Otherwise the URN Namespace Identifier MUST be included.

Otherwise a PPG MAY remove any header, which specifies a default value known to the client. This default may be specified in the over-the-air protocol, provisioned, or established using an implementation-dependent mechanism. For example, an `X-Wap-Application-Id` header might be removed if a client has only one push application, optimising over-the-air communications. `X-Wap-Application-Id` headers containing a registered value MUST NOT be sent over the air without being encoded in numeric format.

5.1.2.1.3 Message State

For each push submission for which errors are encountered in the steps above, or for which it is apparent that successful message delivery is not possible, message delivery MUST NOT be attempted. Note that this may cause a PAP `resultnotification-message` to be sent. Messages that fail the entity and header transformation process MUST have the following status reportable:

PAP Attribute	**Value**
`message-state`	`"undeliverable"`
`Code`	`"transformation-failure"`

If message handling is successfully completed, an undelivered message MUST have the following status reportable:

PAP Attribute	**Value**
`message-state`	`"pending"`

5.1.2.2 Over-the-Air Transmission

The purpose of this function is to deliver messages to the OTA client. Key elements of this function are selection of Push OTA protocol, selection of confirmed or unconfirmed push, and message delivery. A PPG implementation may include tests for message expiration and cancellation, message retransmission and delivery timeout, bearer management and WSP session (if OTA-WSP is used), registration context (if OTA-HTTP is used) management or registration status (if OTA-SIP is used).

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.1.2.2.1 Selection of Push OTA Protocol

A mobile terminal may support OTA-WSP, OTA-HTTP, OTA-SIP and OTA-PTM. OTA-WSP, OTA-SIP or OTA-PTM may be selected for the connectionless push. The PPG selects the OTA protocol variant for connection-oriented push (from among those supported by the terminal and/or as required by the Push Initiator) in an implementation dependent manner.

The PPG MAY be able to select the OTA protocol variants through getting the terminals information or network information, e.g.

- for example based upon SIP Registration between the terminal and PPG
- if the bearer is available,
- if the terminal supports the protocol variants based on terminal's capability,
- or the PI's indication, etc.

The PPG MAY also be able to hand over the decision to the terminal by sending a *Session Initiation Request* (SIR) that contains lists of contact points for OTA-WSP, OTA-HTTP and OTA-SIP. This approach and the SIR are defined in [PushOTA].

OTA-HTTP, OTA-SIP or OTA-PTM MUST be selected if the PI indicates that it accepts content from the client in response to a confirmed push (also see section 5.1.1.2). If the PPG fails to select OTA-HTTP, OTA-SIP or OTA-PTM, the PAP `resultnotification-message` MUST indicate failure of selecting the specified delivery method.

5.1.2.2.2 Bearer Network Selection

If the QoS section of the PAP `push-message` element requires a specific bearer and/or network to be used, the PPG MUST use the specified bearer and/or network, or fail to deliver the message with the following messages status reportable:

PAP Attribute	Value
`message-state`	`"undeliverable"`
`Desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of failure

The PPG MAY support OTA-PTM as defined in [PushOTA], for the various bearer network bindings, such as Push/MBMS, Push/BCAST and Push/CBS. For the Push message indicated to be delivered using OTA-PTM, the PPG will deliver the message via at least one bearer. If a specific OTA-PTM bearer has been explicitly indicated by the PI in the PAP message, e.g. indicated by a named user group, a broadcast/multicast address, targeted location, or network identifier, the PPG SHALL check whether the bearer is available, If yes, the PPG SHALL select the bearer as indicated, If the bearer indicated is not available or no specific bearer is indicated, the PPG SHALL select an available bearer from those associated with a Push Channel,,e.g. based on group or network information.. The PPG SHALL adapt the Push message to the selected bearers.

5.1.2.2.3 Session or Registration Context Selection/Creation

The PPG may use an existing WSP session (if OTA-WSP is used) or registration context (if OTA-HTTP is used), or take implementation-dependent action(s) to create a suitable WSP session or registration context (e.g. send an OTA Session Initiation Request). For OTA-SIP, the PPG SHOULD use the procedures described in [PushOTA] to determine when delivery actions can be taken. For OTA-PTM, the PPG SHALL use the procedures described in [PushOTA] to determine when delivery actions can be taken. If the PPG elects to attempt no further delivery action(s) due to the lack of and/or failure to create a suitable WSP session or registration context, the following messages status MUST be reportable:

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

PAP Attribute	Value
message-state	"undeliverable"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

5.1.2.2.4 Delivery Time Constraints

If the PPG supports delivery time constraints, the PPG MUST NOT deliver the push message prior to the PAP deliver-after-timestamp time and MUST, if unable to deliver by the PAP deliver-before-timestamp time, fail with the following message status reportable:

PAP Attribute	Value
message-state	"expired"
Desc	An appropriate, implementation-dependent value
event-time	Time or estimated time of failure

5.1.2.2.5 Delivery

Assuming no errors, if OTA-WSP is used for OTA delivery, the PPG MUST deliver either a confirmed (Po-ConfirmedPush) or unconfirmed (Po-Push or Po-Unit-Push) [PushOTA] push primitive; if OTA-HTTP is used for OTA delivery, the PPG MUST deliver messages by using the HTTP POST method. If OTA-SIP is used, the PPG MUST deliver messages using SIP INVITE/MSRP. If OTA-PTM is used the PPG MUST deliver messages using one of the point-to-multipoint methods defined in [PushOTA]. If OTA-HTTP, OTA-SIP or OTA-PTM is used and the PI indicates that it accepts content from the client in response to a confirmed push (also see section 5.1.1.2), the X-Wap-Push-Info header [PushOTA] MUST contain the "response" attribute token when the message is pushed to the client.

The use of confirmed or unconfirmed push depends on the PAP delivery-method attribute and implementation-dependent PPG policies.

5.1.2.2.5.1. Unconfirmed Push

A PPG MUST deliver "unconfirmed" messages using OTA-WSP (Po-Push.req or Po-Unit-Push.req primitive), OTA-HTTP, OTA-SIP or OTA-PTM. If OTA-HTTP, OTA-SIP or OTA-PTM is used, the PPG MUST report the same PAP result-notification message as if the message were pushed in an unconfirmed manner using OTA-WSP.

If the PPG sends a Po-Push.req or Po-Unit-Push.req primitive, or the PPG sends messages by using OTA-HTTP, OTA-SIP or OTA-PTM instead of these primitives, the following message status MUST be reportable:

PAP Attribute	Value
message-state	"delivered"
Delivery-method	"unconfirmed"
event-time	Time or estimated time of delivery

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.1.2.2.5.2. Confirmed Push

A PPG MUST deliver `"confirmed"` messages using OTA-WSP (`Po-ConfirmedPush.req` primitive), OTA-HTTP, OTA-SIP or OTA-PTM. If OTA-SIP is used then the INVITE/MSRP method MUST be used and a delivery confirmation MUST be requested by adding a Success-Report header set to yes. If OTA-PTM is used, then for the MBMS option reception reporting SHALL be requested by the PI as described in [[TS26.346]]. The remaining process depends on the type of push as follows:

If the PPG sends a `Po-ConfirmedPush.req` primitive or uses OTA-HTTP or uses the INVITE/MSRP method with delivery confirmation or uses reception reporting, the outcome depends as follows on whether or not the push message is acknowledged:

Success: If the PPG receives a `Po-ConfirmedPush.cnf` primitive indicating successful delivery to the OTA client, or a HTTP response including a `X-Wap-Push-Status` header indicating successful delivery or receives a MSRP Success-Report indicating successful delivery, possibly after a PPG's implementation-dependent retries, or receives a reception report request from the MBMS client, the following message status MUST be reportable:

PAP Attribute	Value
`message-state`	`"delivered"`
`Delivery-method`	`"confirmed"`
`event-time`	Time or estimated time of delivery

Failure due to abort: If the PPG receives a `Po-PushAbort.ind` primitive indicating an aborted push attempt (OTA-WSP), a `X-Wap-Push-Status` header indicating that the push message was rejected (OTA-HTTP) or MSRP Failure-Report indicating delivery failure (OTA-SIP) or a report type indicating failure from an MBMS client, the following message status MUST be reportable:

PAP Attribute	Value
`message-state`	`"aborted"`
`Code`	PAP-specified representation of the abort parameter specified in [PushOTA]
`Desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of aborted delivery attempt

Failure due to timeout: If OTA-WSP is used, a timeout occurs when the PPG does not receive an OTA `Po-ConfirmedPush.cnf` primitive within an implementation-dependent period of time. If OTA-HTTP is used, a timeout occurs when the PPG does not receive a response to a HTTP POST request within an implementation-dependent period of time. If OTA-SIP is used a timeout occurs when the PPG does not receive a MSRP Success-Report indicating delivery success within an implementation-dependent period of time. If OTA-PTM is used, no timeout occurs since the MBMS client re-establishes connection for reception reporting in case all timers time out. If the PPG elects to attempt no further delivery action(s) when a timeout occurs, the following messages status MUST be reportable:

PAP Attribute	Value
`message-state`	`"timeout"`
`Desc`	An appropriate, implementation-dependent value

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

event-time	Time or estimated time of last delivery attempt

5.1.2.2.5.3. Oneshot delivery

A PPG MUST deliver "oneshot" messages as described in section 5.1.2.2.5.1. In addition the PPG MUST attempt to deliver the message only once, and ensure that a one-shot delivery attempt can be made on the underlying bearer. The following message status MUST be reportable for a message delivered using this method:

PAP Attribute	Value
message-state	"delivered"
Delivery-method	"oneshot"
event-time	Time or estimated time of delivery

5.2 Result Notification

The PPG MUST, if requested by the push initiator during push message submission, send a PAP resultnotification-message to the push initiator or its designee.

5.2.1 Time of Result Notification

A result notification, if requested, should be sent as soon as practical after the completion (successful or unsuccessful) of the Over-the-Air message delivery process.

5.2.2 Result Notification Contents

The PAP resultnotification-message indicates the reportable message status, which includes the message state and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

For result notifications related to messages to multiple terminals via an OTA-PTM bearer, a list of successful and/or unsuccessful recipient addresses MAY be included in the resultnotification-message. The addresses SHALL be formatted as described in section 6. If more than one address is included in the successful-recipients element or the unsuccessful-recipients element, the addresses SHALL be comma-separated.

Assuming the PI requested a result notification and indicated that it accepts content from the client in response to a confirmed push (see section 5.1.1.2), content returned from the client in the response to a push via OTA-HTTP, if any, MUST be sent along with the resultnotification-message. If the PI did not indicate that it accepts content from the client in response to a confirmed push, the content entity MUST not be present when the resultnotification-message is returned to the PI. See [PushOTA] for further details.

5.3 PAP Status Query

This OPTIONAL function provides message status on receipt of a PAP statusquery-message.

The status query reply indicates the reportable message status, which includes the message state, and other information as specified earlier in this document. The status should reflect the message just before, within the limits of practicality, sending the result notification.

5.4 Delivery Cancellation

This OPTIONAL function allows delivery cancellation of a pending push message.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-Spec-20110101-I]

If the PPG supports cancellation of a push message, and the message is in a state from which delivery cancellation may be assured, the PPG MUST cancel delivery of the message as requested by a PAP `cancel-message`, and the following message status MUST be reportable:

PAP Attribute	**Value**
`message-state`	`"cancelled"`
`Desc`	An appropriate, implementation-dependent value
`event-time`	Time or estimated time of cancellation

If the PPG cannot assure cancellation of the message delivery, it MUST reject the delivery cancellation.

Successful cancellation of a push message will trigger a delivery result notification, if requested during the push message submission.

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6. Client Addressing

Push Initiators are able to identify clients to the PPG using a special textual address format. The PPG MUST transform these addresses into a form that can be used to deliver over the wireless network. Conversely, the PPG MUST transform network-specific addresses into the textual address format for communication to a Push Initiator. If a Push Initiator has used a particular address value to identify a client in a request sent to the PPG, this address value MUST be used when referring to this client in the corresponding response and any subsequent result notification.

A client address is composed of a client specifier and a PPG specifier. Inclusion of the PPG specifier provides a mechanism to ensure that the address is unambiguous, permitting requests to be routed through proxies. The PPG specifier does not necessarily identify a physical PPG, and is not required to be the hostname of the PPG receiving the address from a PI.

There are multiple types of client specifiers. A PPG MUST support at least one of these client specifier types:

a) User-defined identifiers

b) Device addresses

c) PLMN location addresses

User-defined identifiers are arbitrary values that are mapped to wireless network addresses in an unspecified manner. The PPG has complete control over which bearer-level address will be used in delivering the push message to the client. The user-defined identifier MAY be expanded to several bearer-level addresses for one or more clients. In this case the PPG MUST interact with the Push Initiator in the same way as when the user-defined identifier maps to a single bearer-level address. The interpretation of user-defined identifiers is based on a mutual understanding between the Push Initiator and the PPG. This permits them to be assigned values that are useful for the application using push services. For instance, they could be e-mail addresses.

Device addresses use static values from well-known network address spaces. One example is telephone numbers in the public land mobile network (PLMN). The PPG MAY use any of the client's bearer-level addresses in delivering the push message to the client. How the PPG determines this is not specified, but may be based, for instance, on the characteristics of the bearers used by the client.

PLMN location addresses use static values for network location areas, including Location Area Identification (LAI) or Cell Global Identification (CGI) [TS23.003]. PLMN location addresses provide a means to target groups of clients according to their network service area, with fine-grained selection (by cell site) or broader area (location area). The PLMN location address MAY be expanded to several bearer-level addresses. The PPG has complete control over which bearer-level address(es) will be used in delivering the push message to clients. However, the PPG MUST interact with the Push Initiator in the same way as an address that maps to a single bearer-level address. The interpretation of PLMN location addresses is based on a mutual understanding between the Push Initiator and the PPG.

The bearer-level address may invoke a point-to-multipoint delivery in the wireless network, for example, using cell broadcast. In this case there still MUST be a single result notification, if one has been requested.

6.1 Client Address Format

The external representation of addresses processed by the PPG is defined using ABNF [RFC4234]. The format is compatible with Internet e-mail addresses [RFC822]. The PPG MUST be able to parse this address format, and it MUST be able to determine whether it supports the specified address type or not.

```
wappush-address = ["/"] wappush-client-address ["/"] "@" ppg-specifier

wappush-client-address = "WAPPUSH" "=" client-specifier
ppg-specifier = dom-fragment *( "." dom-fragment )
dom-fragment = ( ALPHA / DIGIT ) *( ALPHA / DIGIT / "-" )

client-specifier = ( user-defined-identifier / device-address ) / plmn-location-address)
user-defined-identifier = ( escaped-value ext-qualifiers "/TYPE=USER" )
device-address = ( global-phone-number ext-qualifiers "/TYPE=PLMN" )
```

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
               / ( ipv4 ext-qualifiers "/TYPE=IPv4" )
               / ( ipv6 ext-qualifiers "/TYPE=IPv6" )
               / ( man ext-qualifiers "/TYPE=MAN" )
             / ( URI ext-qualifiers "/TYPE=URI" )
               / ( escaped-value ext-qualifiers "/TYPE=" address-type )
address-type = 1*address-char
; A network bearer address type [WDP]
address-char = ( ALPHA / DIGIT / "_" )
plmn-location = (mcc "-" mnc "-" lac ext-qualifiers "/TYPE=PLMN-LAI" )
              / (mcc "-" mnc "-" lac "-" ci ext-qualifiers "/TYPE=PLMN-CGI" )

ext-qualifiers = *( "/" keyword "=" value )
; for future extensions, e.g. special well-known user-defined identifier types
keyword = 1*( DIGIT / ALPHA / "-" )
value = 1*( %x20-2E / %x30-3C / %x3E-7E )
escaped-value = 1*( safe-char )
; the actual value escaped to use only safe characters by replacing
; any unsafe-octet with its hex-escape
safe-char = ALPHA / DIGIT / "+" / "-" / "." / "%" / "_"
unsafe-octet = %x00-2A / %x2C / %x2F / %x3A-40 / %x5B-60 / %x7B-FF
hex-escape = "%" 2HEXDIG     ; value of octet as hexadecimal value

global-phone-number = "+" 1*( DIGIT / written-sep )
written-sep = ( "-" / "." )
ipv4 = 1*3DIGIT 3( "." 1*3DIGIT )          ; IPv4 address value [RFC791]
ipv6 = 4HEXDIG 7( ":" 4HEXDIG )            ; IPv6 address value [RFC3513]
man = 8DIGIT                               ; Mobitex MAN address format [Mobitex]
mcc = 3DIGIT                ; MCC (Mobile Country Code) value [TS23.003]
mnc = 2DIGIT / 3DIGIT       ; MNC (Mobile Network Code) value [TS23.003]
lai = 4HEXDIG               ; LAC (Location Area Code) value [TS23.003]
ci  = 4HEXDIG               ; CI (Cell Identity) value [TS23.003]
```

Each value of a `user-defined-identifier` is a sequence of arbitrary octets. They can be safely embedded in this address syntax only by escaping potentially offending values. The conversion to `escaped-value` is done by replacing each instance of `unsafe-octet` by a `hex-escape` which encodes the numeric value of the octet.
URI is as defined in [RFC3986] with the additional requirement that any character not in the safe-char set needs to be hex escaped, e.g. it needs to match escaped-value.

6.2 Client Address Examples

Addresses using user-defined identifiers:

```
WAPPUSH=john.doe%40wapforum.org/TYPE=USER@ppg.carrier.com
; user-defined identifier for john.doe@wapforum.org

wappush=47397547589/type=user@carrier.com
; user-defined identifier for 47397547589

WAPPUSH=47397547589/TYPE=USER@Carrier.com
; equivalent to previous one

WAPPUSH=+155519990730/TYPE=USER@ppg.carrier.com
; user-defined identifier that looks like a phone number
```

Addresses using device addresses:

```
WAPPUSH=+155519990730/TYPE=PLMN@ppg.carrier.com
; device address for a phone number of some wireless network

WAPPUSH=FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/TYPE=IPv6@carrier.com
; device address for an IP v6 address
```

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
WAPPUSH=195.153.199.30/TYPE=IPv4@ppg.carrier.com
; device address for an IP v4 address
```

WAPPUSH=12345678/TYPE=MAN@ppg.carrier.com
; device address for a MAN address

WAPPUSH=sip%3Aalice%40atlanta.com/TYPE=URI@ppg.carrier.com
; device address for an URI address

6.3 Multi-Terminal Considerations

The user can have multiple terminals and each terminal may have its own capabilities. Push offers users with several terminals an explicit way to ensure Push content delivered across them. It is possible to deliver push content to all or subset of the terminals associated with the same user. When push content is delivered to a subset of the terminals, PPG uses a terminal's device ID to uniquely identify each terminal. For OTA-SIP as defined in [SIP Push], GRUU [RFC5627] is used to identify each terminal. For other Push-OTA protocol variants, other device identities may be used by the PPG, e.g. MSISDN, IMEI, IP address, etc.

Before directing Push messages to specific terminals for a user, the PPG SHALL obtain each terminal's device ID, e.g. through:

- the terminal registration and/or capability negotiation processes as defined in [PushOTA], e.g. for
  - OTA-WSP: through establishment of a WAP1 session
  - OTA-HTTP: through terminal registration
  - OTA-SIP: IMS third party registration, reg-event package subscription, or SIP OPTIONS
- Other unspecified means, e.g.
  - Pre-provisioning with the relationship of users to devices
  - Querying other network elements for the relationship of users to devices

Note that for some services, delivery of Push message to each terminal of the user may result in incorrect/undesired handling, e.g. services in which a Push service notification is expected to result in a single response by a service client (e.g. multimedia messaging, digital rights management, location services, email services, etc). In these cases, Push messages may need to be sent to a single terminal only. To direct a Push message to a specific terminal, the PPG:

- For OTA-SIP, the PPG MAY send Push message by using the user's identifier to the SIP/IP core which may choose the explicit terminal directly.
- If the PPG is directed to send the message to a specific device by the user's device ID (e.g. by a PI, through the PAP [PushPAP]), the PPG SHALL send the Push message to the specific terminal by using the device ID.

The PPG MAY decide which terminal to direct a Push message to depending upon the user's preference or some unspecified criteria, which indicates the user's preference of choosing which terminal to receive the upcoming Push message for that specific application. Such criteria could include for example:

- User's ID
- Device ID
- Application ID
- user's preference information

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- device capability
- users registration information, e.g. users status information
- etc.

In multi-terminal situations, the PPG can instruct a specific terminal for a user to establish a connection via the Session Initiation Request procedure defined in [PushOTA], directing the Session Initiation Request to the specific terminal via one of the supporting Push-OTA protocol variants.

In some cases, the registration of a Push application by a Push Client could impact the usability of related services on other terminals of the same user, for which there exists a current registration. In these cases, the PPG needs to ensure that Push messages for the application are delivered to only one terminal. In these cases:

- The PPG MAY reject a Push application registration which will impact the usability of the same application on another registered terminal with the same user's ID. In this situation, the terminal SHOULD be notified via an error response indicating that the registration was not possible due to conflict with other terminals.
- The PPG MAY use other unspecified means to ensure that only one terminal is chosen for related Push requests, e.g.
    - add the registering terminal to a list of the user's terminals for later decisions on delivery of Push messages to the user, e.g. a round-robin list via which the registered terminals are selected in sequence for Push delivery
    - add the registering terminal to a priority-ordered list of terminals for later decisions on delivery of Push messages to the user, e.g.
        - Put the new terminal at the head of a priority list, as the most recently registered terminal (thus the one most likely to be in current use)
        - Put the new terminal at the end of a priority list, as a terminal to use when higher-priority terminals are unregistered or unresponsive

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-PPGService-V2_3	22 Nov 2011	Status changed to Approved by TP: OMA-TP-2011-0406-INP_Push_V2_3_ERP_for_Final_Approval

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for Push Proxy Gateway

B.1.1 Predicates

These items are only used as predicates and do not state any requirements on the implementation.

Item	Function	Reference	Status	Requirement
PPG-CO-S-001	Confirmed push is supported		O	(OTA-CO-S-002 OR OTA-CO-S-003) AND PPG-GEN-S-013

B.1.2 Operations

Item	Function	Reference	Status	Requirement
PPG-GEN-S-001	Push Submission Rejection	5.1.1	M	
PPG-GEN-S-002	Incomplete message handling reportable	5.1.1	M	
PPG-GEN-S-003	Entity transformation under the scope of a No-Transform cache control directive	5.1.2.1.1	M	
PPG-GEN-S-004	Revising headers of transformed entities	5.1.2.1.1	M	
PPG-GEN-S-005	`X-Wap-Application-Id` header processing	5.1.2.1.2	M	
PPG-GEN-S-006	Registered `X-Wap-Application-Id` value sent over-the-air in numeric encoded format	5.1.2.1.2	M	
PPG-GEN-S-007	Reportable message states	5.1.2.1.3	M	
PPG-GEN-S-008	Bearer Network Selection (QoS)	5.1.2.2.2	M	
PPG-GEN-S-009	Reporting of failed Session or Registration Context Selection/Creation	5.1.2.2.3	M	
PPG-GEN-S-010	Delivery Time Constraints	0	M	
PPG-GEN-S-011	Delivery	0	M	
PPG-GEN-S-012	Reportable status associated with unconfirmed push	5.1.2.2.5.1	M	
PPG-GEN-S-013	Reportable statuses associated with confirmed push	5.1.2.2.5.2	O	
PPG-GEN-S-014	Sending of `resultnotification-message`	5.2	M	

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

PPG-GEN-S-015	PAP Status Query	5.3	O	PAP-OPS-S-004
PPG-GEN-S-016	Delivery Cancellation	5.4	O	
PPG-GEN-S-017	Handling message cancellation request	5.4	M	
PPG-GEN-S-018	Support for WSP specific transformations	5.1.2.1.1.1	M	
PPG-GEN-S-019	Support for HTTP specific transformations	5.1.2.1.1.2	O	OTA-CO-S-003
PPG-GEN-S-020	Support for push message replacement	5.1.1.1	O	
PPG-GEN-S-021	Support for binary header encoding	5.1.2.1.1.1	M	
PPG-GEN-S-022	Support for content encoding using WBXML	5.1.2.1.1.1	M	
PPG-GEN-S-023	Support for content encoding using 'deflate'	5.1.2.1.1.2	O	
PPG-GEN-S-024	Handling of push a `push-message` with the `delivery-method` attribute set to "`confirmed-with-response`" in the `quality-of-service` element	5.1.1.2	M	
PPG-GEN-S-025	Selection of Push OTA Protocol	5.1.2.2.1	M	
PPG-GEN-S-026	Inclusion of content returned from the client in a `resultnotification-message`	5.2.2	M	
PPG-GEN-S-027	Reportable statuses associated with oneshot delivery	5.1.2.2.5.3	O	

B.1.3 Client Addressing

Item	Function	Reference	Status	Requirement
PPG-ADD-S-001	Client Addressing	6	M	PPG-ADD-S-002 OR PPG-ADD-S-003
PPG-ADD-S-002	Support for user-defined identifiers	6	O	
PPG-ADD-S-003	Support for device addresses	6	O	
PPG-ADD-S-004	Support for client address format	6.1	M	

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.