Enabler Test Requirements for Push 2.3

Candidate Version 2.3 – 16 Mar 2010

Open Mobile Alliance

OMA-ETR-Push-V2_3-20100316-C

Contents

Figures

Tables

1. Scope

The Enabler Test Requirements (ETR) document for the Enabler under consideration is created and maintained by the Technical Working Group (TWG) responsible for the technical specifications for the corresponding Enabler.

The ETR document is intended to cover at least those requirements collected in the Requirements Document (RD) and the Architecture Document (AD) in addition to any other items the TWG has identified as important enough to warrant attention from interoperability perspective and identify any technical functionalities that should be covered by testing.

2. References

2.1 Normative References

[CacheOp] "WAP Cache Operation", WAP Forum™. WAP-175-CacheOp. URL: http://www.openmobilealliance.org/

[ERELD] "Enabler Relese Document for Push 2.3", Open Mobile Alliance™, OMA-ERELD-Push-V2_3, URL:http://www.openmobilealliance.org/

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.3, Open Mobile Alliance™, OMA-ORG-IOP_Process-V1_3, URL:http://www.openmobilealliance.org/

[PPGService] "Push Proxy Gateway Service Specification". Open Mobile Alliance™. OMA-TS-PPGService-V2_3. URL:http://www.openmobilealliance.org/

[Push2.3] "Enabler Release Definition for Push Version 2.3", Open Mobile Alliance™. OMA-ERELD-Push-V2_3. URL: http://www.openmobilealliance.org/

[PushCAI] "Push Client - Application Interface Specification". Open Mobile Alliance™. OMA-TS-PushCAI-V1_0. URL:http://www.openmobilealliance.org/

[PushCBS] "Push Over the Air Technical Specification – CBS Adaptation" . Open Mobile Alliance™. OMA-TS-Push_CBS_Adaptation-V1_0. URL:http://www.openmobilealliance.org/

[PushOTA] "Push OTA Protocol Specification". Open Mobile Alliance™. OMA-WAP-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/

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

[PushMO] "Push Management Object". Open Mobile Alliance™. OMA-TS-Push_MO-V1_1. URL: http://www.openmobilealliance.org/

[ServiceInd] "Service Indication", WAP Forum™. WAP-167-ServiceInd. URL: http://www.openmobilealliance.org/

[ServiceLoad] "Service Loading", WAP Forum™. WAP-168-ServiceLoad. URL: http://www.openmobilealliance.org/

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

2.2 Informative References

[Browsing] "Enabler Release Definition for Browsing", Open Mobile Alliance™. OMA-ERELD-Browsing-V2_3-20050614-C URL:http://www.openmobilealliance.org/

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

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

[PROVARCH] "Provisioning Architecture Overview 1.1". Open Mobile Alliance_.OMA-WAP-ProvArch-v1_1. URL: http://www.openmobilealliance.org/

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

[Push2.2] Enabler Release Definition for Push Version 2.2", Open Mobile Alliance™. OMA-ERELD-Push-V2_2. URL: http://www.openmobilealliance.org/

[Push2.1] "Enabler Release Definition for Push Version 2.1", Open Mobile Alliance™. OMA-ERELD-Push-V1_1.

URL:http://www.openmobilealliance.org/

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.

3.2 Definitions

Application	An implementation of a related set of functions that perform useful work, often enabling one or more services [OMADICT].
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].
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 Channel	A Push content resource identified by a URI.
Push Client –Application Interface	A device-internal interface provided by Push Clients, via which Push applications can register for Push services with application-specified options, and receive notifications of Push events.
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.
Push Initiator	An entity or service that intiates 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 proxy gateway that provides push proxy services
Push Session	a WSP or HTTP context that is capable of conducting push operations.
Registration	refers to a procedure where the PPG becomes aware of the terminal’s current capabilities and preferences.
Registration Context	a state where the PPG is aware of at least the last capabilities and preferences conveyed from the terminal
Server	a device (or service) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client. A server may initiate a connection to a client as part of

	a service (push).
TestFest	Multi-lateral interoperability testing event
User	a user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content
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
AD	Architecture Document
BCAST	OMA Broadcast Services
CBS	Cell Broadcast Service
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MBMS	Multimedia Broadcast/Multicast Service
OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
OTA	Over The Air
PAP	Push Access Protocol
PI	Push Initiator
PPG	Push Proxy Gateway
RD	Requirements Document
SIP	Session Initiation Protocol
WAP	Wireless Application Protocol

4. Introduction

The purpose of this Enabler Test Requirements document is to help guide the testing effort for the Push 2.3 Enabler, documenting those areas where testing is most important, to ensure interoperability of implementations.

This document provides the cumulative set of test requirements for implementations of the OMA Push enabler, as of the Push 2.3 enabler release. For implementations that have already been validated against earlier versions of the OMA Push enabler, the new Push 2.3 enabler test requirements are identified in this document, and can be the focus for enabler validation and implementation conformance test activities.

The Enabler under consideration comprises the following specifications:

- OMA-TS-PPGService-V2_3: Push Proxy Gateway Service Specification
- OMA-TS-PushCAI-V1_0: Push Client – Application Interface Specification
- OMA-TS-PAP-V2_3: Push Application Protocol (PAP) Specification
- OMA-TS-PushOTA-V2_3: Push Over the Air (OTA) Specification
- OMA-TS-Push_Message-V2_3: Push Message Specification Specification
- OMA-TS-Push_CBS_Adaptation-V1_0: Push Over the Air Technical Specification – CBS Adaptation
- OMA-TS-Push-MO-V1_0: Push Management Object Specification
- WAP-167-ServiceInd: Push Service Indication Specification
- WAP-168-ServiceLoad: Push Service Load Specification
- WAP-175-CacheOp: Push Cache Operation Specification

Push allows Push Initiators and Application Servers to intiate service-related transactions and content delivery to user devices. OMA Push has evolved over several releases, beginning as WAP Push 1.x releases which supported the WAP1 bearers, and releases from WAP Push 2.0 up to OMA Push 2.2 which extended support to WAP2 (HTTP) and SIP bearers. The service environment enabled by OMA Push is illustrated in Figure 1, which shows the two main options for Push service deployment:

- A service/content provider, acting as Push Intiator, requests the OMA Push-based delivery of content to a user through a Push Proxy Gateway (PPG)
- An Application Server directly uses OMA Push to deliver content to a user.

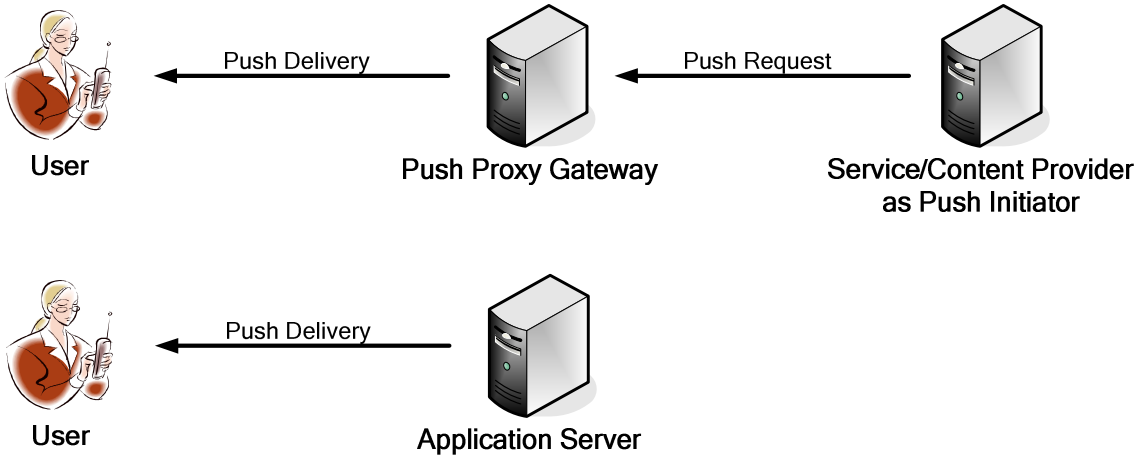

Figure 1: OMA Push service environment

In contrast, the service environment enabled by OMA Push with the enhancement of PTM-Push is illustrated in Figure 2:

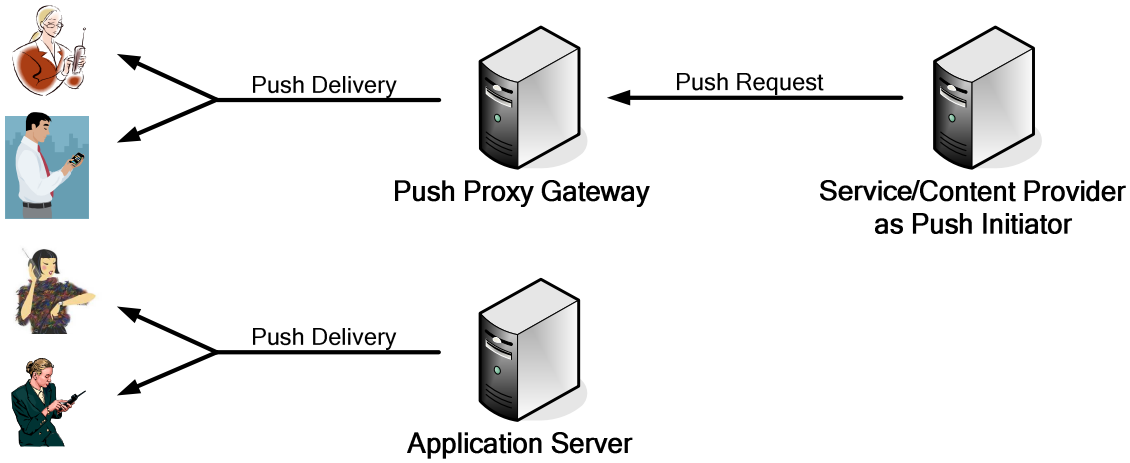

Figure 2: OMA Push service environment with PTM-Push support

OMA Push defines interfaces allowing a *Push Initiator* (PI) to transmit *push content* and *delivery instructions* to a *Push Proxy Gateway* (PPG), and delivery of the push content to the Push Client according to the delivery instructions. The Push Client subsequently delivers the push content to an OMA enabler user-agent or application in the device (hereafter referred to as the "client application"). The PPG and Push Client are the two architectural entities specified by the OMA Push enabler.

The PI may be an application that runs on an ordinary web server, an OMA enabler entity (e.g. multimedia messaging server, device management server,…), or any other application designed to deliver content to users via OMA Push. 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 Push Client. Note the name Push-OTA is based upon the historical focus of OMA Push on mobile data services, but the protocol is also usable over wired connections. Push-OTA is also directly usable by OMA enabler entities or application servers acting directly as a Push Server.

Client applications may be OMA enabler user agents (e.g. browsers, multimedia messaging clients, instant messaging clients, etc) or other device-resident applications that are supported by the Push Client.

Generally, the testing activity should aim at validating the normal working behaviour of the client/server interactions, as well as testing the error conditions whenever it is possible to set up the appropriate scenarios. The following sections provide a more detailed description of the testing requirements for DPE V1.0.

This document also intends to provide some guidance on the prioritization of the specifications and features to be tested within Push v2.3.

4.1 Push Version 2.2

A full description of Push V2.2 [Push2.2] can be found in the ERELD and specifications.

Push 2.2 represents an evolution of Push 2.1 to provide

(a) mechanisms for filtering and authorizing the source of push content.

(b) Guidelines and minimum conformance for the use of segmentation and reassembly when using SMS for push delivery.

(c) Addition of OTA-SIP as a new Push-OTA protocol variant

The changes between the previous version of OMA Push, Push 2.1 [Push2.1] are as follows:

- Push Security
    - white list filtering on the client
    - Managed object to store white list or new push management object [PushMO]
    - Security parameters on push content types to authorize the source of the push message
- Push using SMS
    - Minimum conformance specification for the device when re-assembling push content (if it is segmented over a number of SMS messages)
    - Minimum conformance on the server for segmenting push content over a number of SMS messages
    - Guidelines for push initiators (via PAP Quality of Service) for the most efficient use of the SMS bearer in the context of push.
- Addition of OTA-SIP as a new Push-OTA protocol variant

The scope of the push security changes should provide mechanisms for filtering and verifying the source of push content.

4.2 Push Version 2.3

A full description of Push V2.3 [Push2.3] can be found in the ERELD and specifications.

This enabler release is an extension of the Push 2.2 Enabler release [Push2.2][Push2.2] 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.

The changes between the previous version of OMA Push, Push 2.2 [Push2.2] are as follows:

- Addition of OTA-PTM as a protocol variant, with bindings to the bearers MBMS, OMA BCAST, and CBS

- Ability to delivery Push messages to groups of users over point-to-point and/or point-to-multipoint bearers
- An explicit Push service registration process for coordination of configuration parameters between PPG and Push Client

For simplicity, these new bindings are referred to as Push/MBMS, Push/BCAST, and Push/CBS.

5. Test Requirements

These are the guidelines to interpret these requirements.

- The components being tested are assumed to have reached to a certain level of maturity through implementor's internal conformance methodologies. The internal conformance methodologies are out of scope of these ETRs.
- The test suites derived from these requirements are targeted to test the interoperability between the component implementations.
- The requirements specified below do not force any new behaviors that are not explicitly stated in the detailed specifications.

These test specifications do not include content types not included in this enabler release, such as those defined by Device Management (DM). The testing of these content types is the responsibility of other enabler releases, such as the enabler release for Download Management [ERELDDM].

These requirements are divided into specific test requirements for:

- Push Proxy Gateway
- Push Client

5.1 Enabler Test Requirements

The test requirements collected in this section are related to the Enabler Push V2.3. While this document provides the cumulative set of test requirements for implementations of the OMA Push enabler as of the Push 2.3 enabler release, it is expected that for implementations that have already been validated against earlier versions of the OMA Push enabler, the new Push 2.3 enabler test requirements identified in this document can be the focus for enabler validation and implementation conformance test activities.

In this section, it should be defined what specific functionalities of this Enabler shall or should be tested to ensure adequate operational of the implementations, including any security requirements and constraints on usage if specified (e.g. user can forward a media object but can not visualize it). That means that devices (clients/serves) shall do what they have to do and they shall not do what they are not allowed to do. Both types of test requirements (positive and negative testing) should be included here if so required.

Besides this information, OMA Architecture specifies a "Framework Architecture", consisting of a set of common functions that need to be invoked in most use cases involving the different Service Enablers. The functionality requirements defined in the OMA Framework Architecture, i.e. authentication, authorization, charging, billing, common directory, etc. should also be listed in this table.Use cases are the main input to identify test requirements.

The following test requirements should cover both Conformance test requirements (i.e. functionality to be tested to verify wheter it is implemented either in the client side or in the server side) and Interoperability test requirements (i.e. client/server interactions one with another)

The following sections (Mandatory and Optional test requirements) could also be separated for client and server test requirements.

The tables for the mandatory and optional test requirements include the following columns:

FEATURE KEY:	A set of characters uniquely identifying the enabler test requirement to be tested. It is suggested that the Feature Key is no longer than 4 to 5 characters. The purpose of the Feature Key is that when used, it distinctly refers to only one feature to be tested.
FEATURE DESCRIPTION:	A description of a technical specification feature to be tested.

FEATURE TEST REQUIREMENTS: A description of what shall be tested for the feature,

Following are the Feature Keys as used in the test requirements tables. The "new/updated/extended" defines the relationship of test requirements to the enabler releases.

- New: the related enabler requirements are new in the release. Testing must validate support for the new feature.
- Updated: the related enabler requirements have been updated in the the release, including changes to common requirements and context-specific requirements extensions (e.g. for support of new Push-OTA protocol variants). Testing must validate support for the updated feature. Testing should validate support for the dependent Push-OTA protocol variants, depending upon the scope of the update (common or context-specific).
- Extended: support for the related enabler requirements has been extended to new contexts, but there are no new requirements. Testing should optionally validate support for the feature in the new contexts.

Feature Key	Description	New/updated/extended	
		Push 2.2	Push 2.3
APPID	Application Identification	Updated	Updated
CPRES	Content Presentation (handing of standard Push content types)		
CTXL	Content Translation (between textual and tokenized forms)		
CTYPE	Support for standard content types	Updated	Extended
OTACL	Push-OTA Connectionless Push Service	Updated	Note 1)
PAP	Push Access Protocol	Extended	Updated
PTM	Point-to-Multipoint Push (new in Push 2.3)	n/a	New
PUSHMSG	Push Message Format	Updated	Extended
REG	Push Service Registration	Updated	Updated
SEC	Security related features	Updated	Updated
SIR	Session Initiation Request	Extended	Updated
SMS	Short Message Service related functions	New	

Table 1 Feature Keys

Table Notes:

1. Connectionless Push for OTA-PTM is addressed as part of the PTM feature tests

5.1.1 Mandatory Test Requirements

Mandatory test requirements should cover those features and use cases that require validation in order to approve the enabler. These include areas with complex interactions between the different functional components of the enabler architecture or where the complexity of the specification(s) is such that there is some uncertainty that they have been correctly specified.

These features and use cases SHOULD cover mandatory and MAY recommend prioritisation of optional implementation features. If testing of some of the mandatory features is not required, then the ETR SHALL contains an explanation for their exclusion.

NOTE: This table needs to be filled out at a level where ambiguity is not present but details are not overwhelming.

Ambiguity means that the details do not have several meanings nor have more than one possible implementation path following.

5.1.2 Push Proxy Gateway

	Feature Key	Feature Description	Feature Test Requirements
Normal Flow	OTACL-001	Connectionless Push (OTA-WSP, OTA-SIP)	The test campaign must verify that Connectionless Push (OTA-WSP or OTA-SIP) is supported as defined in **[**PushOTA**]**
	PAP-001	Push Access Protocol	The test must confirm that all the mandatory elements of the push access protocol **[**PushPAP**]** are supported by the Push Proxy Gateway (PPG)
	PUSHMSG-001	Push Message Format	Support of the Push Message Format. **[**PushMsg**]** must be tested; this also includes registered application identifiers [OMNA].
	CTXL-001	Content Translation (Service Indication)	The testing must verify whether content translation is supported by the Push Proxy, for Service Indication [ServiceInd] content type, as defined in the Push Proxy Service Specification **[**PPGService**]**
	APPID-001	Application Addressing	The test campaign must verify whether application addressing data is conveyed in the push meta-data and that the application numeric identifiers, registered by OMNA, are supported
	CTYPE-001	Content Types (OTA-WSP, OTA-SIP)	The test campaign needs to verify support for application/vnd.wap.multipart.mixed, application/vnd.wap.multipart.related and application/vnd.wap.multipart.alternative (OTA-WSP) or multipart/mixed, multipart/related and multipart/alternative (OTA-SIP) for PUSH MIME type over the air **[**PushOTA**]**
	SMS-001	SMS Segmentation	The testing must verify that a Push Proxy can, at a minimum, segment content into 4 SMS messages
	PTM-001	Point-to-Multipoint Push	The test campaign must verify that Point-to-Multipoint Push (OTA-PTM) is supported as defined in **[**PushOTA**]**
Error Flow	PAP-002	Push Access Protocol	In the case where the push submission (PAP) is malformed, i.e. not as specified in the **[**PushPAP**]** specification it must be rejected by a push proxy **[**PPGService**]**

Table 2: Applicability Table for Enabler Specific Mandatory Test Requirements of the Push Proxy Gateway

5.1.3 Push Client

	Feature Key	Feature Description	Feature Test Requirements
Normal Flow	CPRES-001	Content Presentation (Service Indication)	The test campaign must verify that Service Indication Content [ServiceInd] is supported.
	APPID-002	Application Addressing	The test campaign must verify whether application addressing data is conveyed in the push meta-data and that the application numeric identifiers, registered by OMNA, are supported
	CTYPE-002	Content Types (OTA-WSP, OTA-SIP)	The test campaign needs to verify support for application/vnd.wap.multipart.mixed, application/vnd.wap.multipart.related and application/vnd.wap.multipart.alternative (OTA-WSP) or multipart/mixed, multipart/related and multipart/alternative (OTA-SIP) for PUSH MIME type over the air **[**PushOTA**]**
	SEC-001	Push Security Whitelist	The test campaign will verify the function of push whitelists as defined in **[**PushOTA**]**, for whitelists provisioned using device management object and with whitelists defined using VENDORCONFIG extensions defined in [PROVARCH]
	SMS-002	SMS Reassembly	The testing must verify that, at a minimum, a client must be able to re-assemble at least 4 SMS messages into a correctly formed push message
	PTM-002	Point-to-Multipoint Push	The test campaign must verify that Point-to-Multipoint Push (OTA-PTM) is supported as defined in **[**PushOTA**]**
Error Flow	CPRES-002	Content Presentation (Service Indication)	If the same Service Indication [ServiceInd] is sent more than once to the same device – using the same Service Indication Identifier (si-id) the subsequent service indications must NOT be presented to the user.
	SEC-002	Push Security Whitelist	In the event that whitelists are supported but not provisioned the push content must be processed and should be presented to the user as per the semantics of the content type being pushed.

Table 3: Applicability Table for Enabler Specific Mandatory Test Requirements of the Push Client

5.1.4 Optional Test Requirements

Optional test requirements should cover those features and use cases that are not mandated to be tested, but it is still felt that their inclusion will enhance the quality of the enabler validation.

Additionally, important conformance test requirements MAY be listed.

These features and use cases SHOULD cover optional and MAY cover mandatory implementation features. In case a mandatory feature is listed here, the Feature Test Requirements column should provide an explanation why testing of this feature is not mandated.

NOTE: This table needs to be filled out at a level where ambiguity is not present but details are not overwhelming.

Ambiguity means that the details do not have several meanings nor have more than one possible implementation path following.

5.1.5 Push Proxy Gateway

	Feature Key	Feature Description	Feature Test Requirements
Normal Flow	OTACO-001	Connection Orientated Push (OTA-WSP, OTA-HTTP, or OTA-SIP)	If connection Orientated push is supported the test campaign MUST verify whether the mandatory OTA-WSP, OTA-HTTP, or OTA-SIP is supported as defined in **[**PushOTA**]**
	SIR-001	Session Initiation	Depending on the type of connectivity supported (OTA-WSP, OTA-HTTP, or OTA-SIP) the testing is required to verify that session initiation application content type (SIA) is supported as defined in **[**PushOTA**]**;
	PAP-003	Push Access Protocol	The test campaign SHALL verify if the optional parts of the Push Access Protocol are supported as defined in the PAP specification **[**PushPAP**]**.
	CTYPE-003	Content Presentation (Service Load and Cache Operation)	The testing SHALL verify whether the content types Service Load [ServiceLoad] and Cache operation [CacheOp] are correctly actioned by the device as per specification
	CTYPE-004	Content Types (OTA-HTTP)	The test campaign needs to verify support for multipart/mixed, multipart/related and multipart/alternative for PUSH MIME type over HTTP
	CTXL-002	Content Translation (Service Load and Cache Operation)	The testing must verify whether content translation is supported by the Push Proxy, for Service Indication [ServiceInd] content type, as defined in the Push Proxy Service Specification **[**PPGService**]**
	PAP-004	PAP Capabilities Request	The testing shall verify the operation of the PAP CCQ (Client Capabilities Query).
	PAP-005	PAP Result Notification	The testing shall verify the generation of 'ResultNotification' **[**PushPAP**]** by a push proxy conforming to the **[**PPGService**]** specification.
	REG-001	Push Service Registration (WAP1, WAP2, SIP)	The testing shall verify the operation of capability negotiation (OTA-WSP) and registration (OTA-HTTP or OTA-SIP) as specified in the **[**PushOTA**]** specification.
	OTACL-002	One shot	The testing shall verify the operation of 'one-shot' push as specified in the **[**PPGService**]** specification and as requested via the PAP Quality of Service Element **[**PushPAP**]**.
	OTACO-002	Confirmed With Response	The testing shall verify the operation of 'confirmed with response' push as specified in the **[**PPGService**]** specification and as requested via the PAP Quality of Service Element **[**PushPAP**]**.
	SEC-003	Security Considerations	The test campaign will verify the security functionality defined in OTA-SEC-S-001, OTA_HTTP-S-013-O, and OTA-SIP-S-008-O **[**PushOTA**]**
	PTM-003	Point-to-Multipoint Push	If OTA-PTM binding to MBMS is supported, the test campaign must verify that Push/MBMS is supported as defined in **[**PushOTA**]**
	PTM-004	Point-to-Multipoint Push	If OTA-PTM binding to BCAST is supported, the test campaign must verify that Push/BCAST is supported as defined in **[**PushOTA**]**
	PTM-005	Point-to-Multipoint Push	If OTA-PTM binding to CBS is supported, the test campaign must verify that Push/CBS is supported as defined in **[**PushOTA**]**

	Feature Key	Feature Description	Feature Test Requirements
	PTM-006	Push Channel	If Push Channel definition is supported, the test campaign must verify that Push messages are delivered per the Push Channel, as defined in **[**PushOTA**]**
	REG-002	Push Service Registration (OTA-PTM)	The testing shall verify the operation of registration (OTA-PTM) as specified in the **[**PushOTA**]** specification.
Error Flow	PAP-006	Push Access Protocol	The testing shall verify that malformed PAP messages, received from the Push Initiator, are rejected by a Push Proxy Gateway conforming to **[**PPGService**]**. Examples of malformed PAP submissions are: Invalid control entity : recipient addressing Invalid control entity: use of Quality of Service Invalid content entities

Table 4: Applicability Table for Enabler Specific Optional Test Requirements of the Push Proxy Gateway

5.1.5.1 Interoperability

Table 5 Proxy ETR for PUSH Priorities for IOP Test

Summary Requirement	Priority
Verify Push content delivery over each supported bearer • Connectionless Push (Mandatory) • Connection Orientated Push (Optional)	High
Verify push Security changes (content type) parameters is supported	High
Verify Push Proxy support **[**PPGService**]**. for Push Initiators via the Push Access Protocol **[**PushPAP**]**	Medium

5.1.6 Push Client

	Feature Key	Feature Description	Feature Test Requirements
Normal Flow	OTACO-003	Connection Orientated Push (OTA-WSP and/or OTA-HTTP)	If connection orientated push is supported the test campaign MUST verify whether the mandatory OTA-WSP or OTA-HTTP is supported as defined in **[**PushOTA**]**
	SIR-002	Session Initiation	Depending on the type of connectivity supported (OTA-WSP or OTA-HTTP) the testing is required to verify that session initiation application content type (SIA) is supported as defined in **[**PushOTA**]**;
	CPRES-003	Content Presentation (Service Load and Cache Operation)	The testing SHALL verify whether the content types Service Load [ServiceLoad] and Cache operation [CacheOp]are correctly actioned by the device as per specification
	CTYPE-005	Content Types (OTA-HTTP)	The test campaign needs to verify support for multipart/mixed, multipart/related and multipart/alternative for PUSH MIME type over HTTP
	OTACL-003	Confirmed With Response	The testing shall verify the operation of 'confirmed with response' push as specified in the **[**PPGService**]** specification and as requested via the PAP Quality of Service Element **[**PushPAP**]** and as responded to by a compatible device **[**PushOTA**]**
	SEC-004	Security	The testing shall verify the operation of content type parameter extensions to authenticate the source of the push content as defined in **[**PushOTA**]**
	PTM-007	Point-to-Multipoint Push	If OTA-PTM binding to MBMS is supported, the test campaign must verify that Push/MBMS is supported as defined in **[**PushOTA**]**
	PTM-008	Point-to-Multipoint Push	If OTA-PTM binding to BCAST is supported, the test campaign must verify that Push/BCAST is supported as defined in **[**PushOTA**]**
	PTM-009	Point-to-Multipoint Push	If OTA-PTM binding to CBS is supported, the test campaign must verify that Push/CBS is supported as defined in **[**PushOTA**]**
	REG-003	Push Service Registration (OTA-PTM)	The testing shall verify the operation of registration (OTA-PTM) as specified in the **[**PushOTA**]** specification.
Error Flow			

Table 6: Applicability Table for Enabler Specific Optional Test Requirements of the Push Client

5.1.6.1 Interoperability

Table 7 Push Client ETR for PUSH Priorities for IOP Test

Summary Requirement	Priority
Verify Push content is presented by the device when received for the following content types: • Service Indication [ServiceInd] • Service Load [ServiceLoad]	High

Summary Requirement	Priority
• Cache Operation [CacheOp]	
Verify push white listing is supported (either via Device Management or WAP Provisioning)	High
Verify that support for session initiation multiple Session Initiation requests in a short space of time	High

5.2 Backwards Compatibility

This enabler release is backward compatible with earlier releases, and the test suite addresses validation requirements to ensure backward compatibility.

5.3 Enabler Dependencies

None.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft Version 2.3 History

Document Identifier	Date	Sections	Description
Draft Version OMA-ETR-Push-V2.3	30 Sept 2009	n/a	Initial version of this document for Push 2.3, based upon OMA-ETR-Push-V2_2-20090609
	19 Oct 2009	All	Updated for agreed CR: OMA-CD-PUSH-2009-0103-CR_PTM_Push_ETR_edits
	21 Oct 2009	5.1, 5.1.x	Updates from ETR review OMA-IOP-BRO-2009-0111R02-INP_PTM_Push_ETR_for_review
	27 Jan 2010	All	Updated for agreed CR: OMA-CD-PUSH-2010-0011-CR_2.3_CONRR_ETR.doc Updated template
CandidateVersion OMA-ETR-Push-V2.3	16 Mar 2010	All	Status changed to Candidate by TP: OMA-TP-2010-0106-INP_PUSH_V2_3_ERP_for_Candidate_Approval