Enabler Test Specification for the

Web Runtime API (WRAPI) Push API

Candidate Version 1.0 – 08 Oct 2013

Open Mobile Alliance

OMA-ETS-WRAPI-V1_0-20131008-C

Contents

Figures

1. Scope

This document describes in detail the Interoperability test cases for the Web Runtime API V1.0 Enabler Release (WRAPI 1.0).

2. References

2.1 Normative References

[IOPPROC] “OMA Interoperability Policy and Process”, Version 1.9, Open Mobile Alliance™, OMA-ORG-IOP_Process-V1_9, 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

[WRAPI-ETR] "Enabler Test Requirements for the Web Runtime API (WRAPI)". Open Mobile Alliance™. OMA-ETR-WRAPI-V1_0 URL: http://www.openmobilealliance.org/

[WRAPI-EVP] "Enabler Validation Plan for the Web Runtime API (WRAPI) Push API". Open Mobile Alliance™. OMA-EVP-WRAPI-V1_0 URL: http://www.openmobilealliance.org/

[WRAPI-Push] "Web Runtime API (WRAPI) – Push". Open Mobile Alliance™. OMA-TS-WRAPI_Push-V1_0 URL: http://www.openmobilealliance.org/

[WRAPI-Patterns] " Web Runtime API (WRAPI) – Design Patterns". Open Mobile Alliance™. OMA-TS-WRAPI_Design_Patterns-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/

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”, are normative, unless they are explicitly indicated to be informative.

The following numbering scheme is used:

xxx-y.z-con-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’con’	Indicating this test is a conformance test case
number	Leap number for the test case

Or

xxx-y.z-int-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’int’	Indicating this test is a interoperability test case
number	Leap number for the test case

3.2 Definitions

API Patterns	Design guidelines and requirements for definition of APIs
Browser Context	Web applications executing under a Web browser as Web runtime environment.
ECMAScript	Use definition from [OMADICT].
JavaScript	Use definition from [OMADICT].
User Agent	Use definition from [OMADICT].
Web	The World Wide Web, a content and application framework based upon hypertext and related technologies, e.g. XML, JavaScript/ECMAScript, CSS, etc.
Web Application	An application designed using Web technologies.
Web IDL	An IDL language for Web application APIs
Web Runtime	Client software that supports the execution of Web Applications
Widget Context	Web applications installed and executing under a W3C Widget [W3C-Widgets] engine as Web runtime environment.
Uniform Resource Identifier	Use definition from [OMADICT].

3.3 Abbreviations

API	Application Programming Interface
HTTP	HyperText Transfer Protocol
IDL	Interface Definition Language

JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
SCR	Static Conformance Requirements
SMS	Short Message Service
TS	Technical Specification
UA	User Agent
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAC	Wholesale Applications Community
W3C	World Wide Web Consortium
WRAPI	The OMA Web Runtime API enabler
WRT	Web runtime environment
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The purpose of this document is to provide test cases for the WRAPI 1.0 Enabler.

Test configuration requirements are defined in [WRAPI_EVP].

5. Test Cases

5.1 Push Client as Local Push API Server

5.1.1 EventSource Creation

Test Case ID	WRAPI-1.0-PushClient-1
Test Object	Push Client
Test Case Description and Purpose	Verify creation of EventSource connections for various event source types.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-004-M
ETR Reference	WRAPI_Push-M-001, WRAPI_Push-M-006
Tool	User Agent and Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection as described below, for each event source (OMA Push, SMS, and SIP) supported by the Push Client as Push API server. 2. The WebApp requests a new EventSource object creation with URL parameter as required for event delivery via a local Push API server, e.g. in the forms: http://localhost:4035 http://localhost:4035/?push-accept-source=* http://localhost:4035/?push-accept-source=urn:oma:xml:push http://localhost:4035/?push-accept-source=sms:+14255551212 http://localhost:4035/?push-accept-source=sip:bob@testserver.net For each of these requests, the following steps occur in the test: 3. The User Agent returns a new EventSource object. 4. The WebApp assigns a function to the onopen attribute of the returned EventSource object, as handler for “open” events, and assigns a function to the onerror attribute of the returned EventSource object, as handler for “error” events. 5. If the source is supported, the Push Client returns a successful response to the User Agent as described in [WRAPI-Push], including an “Access-Control-Allow-Origin” header with the value of the “Origin:” header received in the GET request. If the source is not supported, the Push Client returns a 403 FORBIDDEN response.

	6. Consequently, if the source is supported, the WebApp receives an "open" event to the onopen handler. If the source is not supported, the WebApp receives an "error" event to the onerror handler.
Pass-Criteria	The WebApp receives either an "open" event or an "error" event for each request, according to whether the source is supported or not.

5.1.2 EventSource Creation Error: Invalid Event Source

Test Case ID	WRAPI-1.0-PushClient-2
Test Object	Push Client
Test Case Description and Purpose	Verify failure of EventSource setup for invalid push-accept-source values.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-004-M
ETR Reference	WRAPI_Push-M-006
Tool	User Agent and Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. Load a Test WebApp designed to attempt EventSource establishment for invalid event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation with an invalid push-accept-source parameter, e.g.: http://localhost:4035/?push-accept-source=xyz 3. The User Agent returns a new EventSource object. 4. The WebApp assigns a function to the onerror attribute of the returned EventSource object, as handler for "error" events. 5. The Push Client returns a 403 FORBIDDEN response. 6. The WebApp receives an "error" event to the onerror handler.
Pass-Criteria	The WebApp receives an "error" event to the onerror handler when an unsupported Push API event source is requested.

5.1.3 Event Delivery for OMA Push

Test Case ID	WRAPI-1.0-PushClient-3

Test Object	Push Client
Test Case Description and Purpose	Verify delivery of OMA Push events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, PPG, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-1 Push Initiator is configured for access to Push service via a PPG, e.g. using PAP or PushREST APIs.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation for OMA Push event delivery as in WRAPI-1.0-PushClient-1. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns a function to the onmessage attribute of the EventSource object, as handler for "message" events. 4. Upon the notification, the Push Initiator sends a Push Message request to the PPG, targeted to the device in which the WebApp is running. 5. The PPG accepts the Push Message request and delivers a Push-OTA message to the Push Client via the applicable bearer. 6. The Push Client receives the Push-OTA message, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection. 7. The WebApp receives a "message" event to the onmessage handler.
Pass-Criteria	The WebApp receives the OMA Push message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.1.4 Event Delivery for SMS

Test Case ID	WRAPI-1.0-PushClient-4
Test Object	Push Client
Test Case Description and Purpose	Verify delivery of SMS events.

Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, SMSC, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-1 Push Initiator is configured for sending SMS messages from a specific SMS source address, e.g. using an SMS API.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SMS event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation for SMS event delivery as in WRAPI-1.0-PushClient-2, setting the push-accept-source parameter of the EventSource URL to the SMS source address to be used by the Push Initiator. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for "SMS" events. 4. Upon the notification, the Push Initiator sends an SMS to the device in which the WebApp is running. 5. The SMS message is delivered by the SMSC to the target device. 6. The Push Client receives the SMS message, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection. 7. The WebApp receives a "SMS" event to the assigned event listener.
Pass-Criteria	The WebApp receives the SMS message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.1.5 Event Delivery for SIP

Test Case ID	WRAPI-1.0-PushClient-5
Test Object	Push Client
Test Case Description and Purpose	Verify delivery of SIP events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-C-002-M

	WRAPI-PUSH-C-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-1 Push Initiator and Push Client are registered with the SIP Proxy.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SIP event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation for SIP event delivery as in WRAPI-1.0-PushClient-2, setting the push-accept-source parameter of the EventSource URL to the SIP source address to be used by the Push Initiator. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for "SIP" events. 4. Upon the notification, the Push Initiator sends a SIP MESSAGE to the device in which the WebApp is running. 5. The SIP MESSAGE message is delivered by the SIP Proxy to the target device. 6. The Push Client receives the SIP MESSAGE, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection. 7. The WebApp receives a "SIP" event to the onmessage handler.
Pass-Criteria	The WebApp receives the SIP MESSAGE data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.1.6 Push API Connection Closure by WebApp

Test Case ID	WRAPI-1.0-PushClient-6
Test Object	Push Client
Test Case Description and Purpose	Verify Push API connection closure by WebApp.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-001-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, PPG | SMSC | SIP Proxy, User Agent, Test WebApp

Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-3 | WRAPI-1.0-PushClient-4 | WRAPI-1.0-PushClient-5
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event (“message” | “SMS” | “SIP”), the WebApp calls the close() method on the EventSource object, and sends a notification to the Push Initiator that the EventSource has been closed. The WebApp does not remove the onmessage handler as it will be used to verify that no further events are delivered by the Push Client via that EventSource connection. 4. The User Agent closes the local connection to the Push Client as Push API server. 5. Upon the notification, the Push Initiator sends another message to the device in which the WebApp is running. 6. The Push Client, if it is still listening for messages on the related bearer, does not deliver the incoming message to the EventSource connection (which should already be closed anyway). 7. No further message events are received by the WebApp within a reasonable timeout, e.g. 10 seconds.
Pass-Criteria	After closing the EventSource connection, the WebApp receives no further message events via the EventSource object.

5.1.7 Multiple Push API Connections

Test Case ID	WRAPI-1.0-PushClient-7
Test Object	Push Client
Test Case Description and Purpose	Verify ability to create multiple Push API connections.
Specification Reference	[WRAPI-Push] Section 7
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-003-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, PPG | SMSC | SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-3 | WRAPI-1.0-PushClient-4 | WRAPI-1.0-PushClient-5

Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event ("message" | "SMS" | "SIP"), the WebApp opens another EventSource connection of the same or different type, and receives another event via that connection as in the prerequisite test.
Pass-Criteria	The WebApp is able to open and simultaneously use two EventSource connections of the same or different type.

5.1.8 Event Filtering for OMA Push

Test Case ID	WRAPI-1.0-PushClient-8
Test Object	Push Client
Test Case Description and Purpose	Verify ability to filter events for OMA Push.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, PPG, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-3
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push as in WRAPI-1.0-PushClient-3. 2. The WebApp initiates two EventSource connections for OMA Push event delivery as in WRAPI-1.0-PushClient-3. One connection includes a filter parameter on the Push Application ID, and the other includes a filter parameter on the content type of Push events. http://localhost:4035/?push-accept-source=urn:oma:xml:push&push-accept-application-id=myapp.com http://localhost:4035/?push-accept-source=urn:oma:xml:push&push-accept-content-type=text/vnd.wap.si 3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in WRAPI-1.0-PushClient-3. 4. The Push Initiator requests delivery of two Push Messages via the PPG. One message includes the Push Application ID indicated by the WebApp for the first EventSource connection, and the other includes the content type indicated by the

	WebApp for the second EventSource connection. 5. The Push messages are delivered as in WRAPI-1.0-PushClient-3, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.
Pass-Criteria	The first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.

5.1.9 Event Filtering for SMS and SIP

Test Case ID	WRAPI-1.0-PushClient-9
Test Object	Push Client
Test Case Description and Purpose	Verify ability to filter events for SMS and SIP.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-C-002-M WRAPI-PUSH-C-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, SMSC | SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PushClient-4 | WRAPI-1.0-PushClient-5 Push Initiator is configured with ability to send messages using two different source addresses, e.g. via two different SMS API keys or using two different SIP registrations.
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (SMS | SIP), with the additional functions described below. 2. The WebApp initiates two EventSource connections for event delivery as in the prerequisite test. The first connection includes an event source filter parameter with one of the source addresses used by the Push Initiator, and the second connection includes an event source filter with the value ‘*’. 3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in the prerequisite test. 4. The Push Initiator sends a message using the source address specified by the WebApp in the first connection, and a second message from another source address. 5. The Push messages are delivered as in the prerequisite test, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first

	message is received at both EventSource objects, and the second message by only the second EventSource object.
Pass-Criteria	The first message is received at both EventSource objects, and the second message by only the second EventSource object.

5.1.10 Restricting Access to Local Push Service

Test Case ID	WRAPI-1.0-PushClient-10
Test Object	Push Client
Test Case Description and Purpose	Verify Push Client serves only locally-originated Push API requests.
Specification Reference	[WRAPI-Push] Section 8.1
SCR Reference	WRAPI-PUSH-C-007-M
ETR Reference	WRAPI_Push-M-005
Tool	User Agent, Test WebApp, Push API Request Injection Tool
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. From a Push API Request Injection Tool on a local network to which the Push API supporting device under test is connected, send a correctly formed Push API connection to the device under test. For example, if the device under test has IP address 192.168.1.1: `GET http://192.168.1.1:4035` 2. If possible, ensure the request is delivered to the Push Client listening on that port, i.e. disable any firewall function in the device under test. 3. If received by the Push Client, the request is silently ignored.
Pass-Criteria	Non-locally-originated Push API requests are silently ignored by the Push Client.

5.1.11 User Consent for Push API Service Access

Test Case ID	WRAPI-1.0-PushClient-11
Test Object	Push Client
Test Case Description and Purpose	Verify that in cases which pre-arranged trust relationships do not apply (i.e. are not supported, or not applicable for the specific WebApp), the Push Client ensures user consent for access to the Push API service by the WebApp.
Specification Reference	[WRAPI-Push] Section 8.3
SCR Reference	WRAPI-PUSH-C-009-M

ETR Reference	WRAPI_Push-M-005
Tool	User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. From a WebApp for which pre-arranged trust relationships do not apply (i.e. are not supported, or not applicable for the specific WebApp), initiate a Push API request. 2. The Push Client informs the tester of the request including the WebApp's origin, and asks for consent to allow the request. Note that no specific UI is mandated, but per the overall intent of the requirement, a user informed consent process must occur. 3. The tester accepts the request, and the connection is successfully established e.g. as in WRAPI-1.0-PushClient-1. 4. The WebApp initiates another Push API request, using a different source address filter. 5. The Push Client informs the tester of the new request including the WebApp's origin, and asks for consent to allow the request. 6. The tester accepts the request, and the second connection is successfully established e.g. as in WRAPI-1.0-PushClient-1. 7. Through a provided user interface, the tester revokes granted API permission, either for one or both of the existing requests (UI or capabilities may vary). 5. The Push Client terminates the EventSource connection for the revoked permissions, and the WebApp receives an "error" event at the EventSource object related to the revoked Push API connections.
Pass-Criteria	Users are allowed to express informed consent for establishment of Push service for a WebApp, and are able to revoke earlier granted permissions.

5.2 User Agent as Virtual Push API Server

5.2.1 EventSource Creation

Test Case ID	WRAPI-1.0-UA-1
Test Object	User Agent
Test Case Description and Purpose	Verify creation of EventSource connections for various event source types.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-UA-001-M WRAPI-PUSH-UA-002-M

ETR Reference	WRAPI_Push-M-001, WRAPI_Push-M-006
Tool	Test WebApp
Test Code/Files	To be defined
Preconditions	None

Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection as described below, for each event source (OMA Push, SMS, and SIP) supported by the User Agent as Push API server.
	2. The WebApp requests a new EventSource object creation with URL parameter with origin field (host and port) set to a value recognized by the User Agent as an origin at which it supports operation as a virtual Push API server, e.g. in the forms:
	`http://localhost:4035`
	`http://localhost:4035/?push-accept-source=*`
	`http://localhost:4035/?push-accept-source=urn:oma:xml:push`
	`http://localhost:4035/?push-accept-source=sms:+14255551212`
	`http://localhost:4035/?push-accept-source=sip:bob@testserver.net`
	For each of these requests, the following steps occur in the test:
	3. If the source is supported, the User Agent returns a new EventSource object. Otherwise, the User Agent throws a SYNTAX_ERR exception, and the steps below do not apply for that specific request.
	4. The WebApp assigns a function to the onopen attribute of the returned EventSource object, as handler for “open” events.
	5. The WebApp receives an “open” event to the onopen handler.

Pass-Criteria	The WebApp receives either an “open” event or a SYNTAX_ERR exception for each request, according to whether the source is supported or not.

5.2.2 EventSource Creation Error: Invalid Event Source

Test Case ID	WRAPI-1.0-UA-2
Test Object	User Agent
Test Case Description and Purpose	Verify failure of EventSource setup for invalid push-accept-source values.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-004-M
ETR Reference	WRAPI_Push-M-006

Tool	Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. Load a Test WebApp designed to attempt EventSource establishment for invalid event delivery at an origin served by the User Agent as a virtual Push API server. 2. The WebApp requests a new EventSource object creation with an invalid push-accept-source parameter, e.g.: `http://localhost:4035/?push-accept-source=xyz` 3. The User Agent throws a SYNTAX_ERR exception.
Pass-Criteria	The WebApp receives a SYNTAX_ERR exception when an unsupported Push API event source is requested.

5.2.3 Event Delivery for OMA Push

Test Case ID	WRAPI-1.0-UA-3
Test Object	User Agent
Test Case Description and Purpose	Verify delivery of OMA Push events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, PPG, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-1 Push Initiator is configured for access to Push service via a PPG, e.g. using PAP or PushREST APIs.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push event delivery at an origin served by the User Agent as a virtual Push API server. 2. The WebApp requests a new EventSource object creation for OMA Push event delivery as in WRAPI-1.0-UA-1. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns a function to the onmessage attribute of the EventSource object, as handler for "message" events.

	4. Upon the notification, the Push Initiator sends a Push Message request to the PPG, targeted to the device in which the WebApp is running. 5. The PPG accepts the Push Message request and delivers a Push-OTA message to the User Agent via the applicable bearer. 6. The User Agent receives the Push-OTA message, maps the content to the text/event-stream MIME type, and delivers a “message” event to the EventSource object. 7. The WebApp receives a “message” event to the onmessage handler.
Pass-Criteria	The WebApp receives the OMA Push message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.2.4 Event Delivery for SMS

Test Case ID	WRAPI-1.0-UA-4
Test Object	User Agent
Test Case Description and Purpose	Verify delivery of SMS events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, SMSC, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-1 Push Initiator is configured for sending SMS messages from a specific SMS source address, e.g. using an SMS API.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SMS event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation for SMS event delivery via a local Push API server as in WRAPI-1.0-UA-2, setting the push-accept-source parameter of the EventSource URL to the SMS source address to be used by the Push Initiator. 3. Upon the “open” event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for “SMS” events. 4. Upon the notification, the Push Initiator sends an SMS to the device in which the WebApp is running.

	5. The SMS message is delivered by the SMSC to the target device. 6. The User Agent receives the SMS message, maps the content to the text/event-stream MIME type, and delivers an "SMS" event to the EventSource object. 7. The WebApp receives an "SMS" event to the assigned event listener.
Pass-Criteria	The WebApp receives the SMS message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.2.5 Event Delivery for SIP

Test Case ID	WRAPI-1.0-UA-5
Test Object	User Agent
Test Case Description and Purpose	Verify delivery of SIP events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, SIP Proxy, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-1 Push Initiator and User Agent are registered with the SIP Proxy.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SIP event delivery via a local Push API server. 2. The WebApp requests a new EventSource object creation for SIP event delivery via a local Push API server as in WRAPI-1.0-UA-2, setting the push-accept-source parameter of the EventSource URL to the SIP source address to be used by the Push Initiator. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for "SIP" events. 4. Upon the notification, the Push Initiator sends a SIP MESSAGE to the device in which the WebApp is running. 5. The SIP MESSAGE message is delivered by the SIP Proxy to the target device. 6. The User Agent receives the SIP MESSAGE, maps the content to the text/event-stream MIME type, and delivers a "SIP" event to the EventSource object. 7. The WebApp receives a "SIP" event to the assigned event listener.

Pass-Criteria	The WebApp receives the SIP MESSAGE data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.2.6 Push API Connection Closure by WebApp

Test Case ID	WRAPI-1.0-UA-6
Test Object	User Agent
Test Case Description and Purpose	Verify Push API connection closure by WebApp.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-001-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, PPG | SMSC | SIP Proxy, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-3 | WRAPI-1.0-UA-4 | WRAPI-1.0-UA-5
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event (“message” | “SMS” | “SIP”), the WebApp calls the close() method on the EventSource object, and sends a notification to the Push Initiator that the EventSource has been closed. The WebApp does not remove the onmessage handler as it will be used to verify that no further events are delivered by the User Agent via that EventSource connection. 4. The User Agent closes the EventSource connection. 5. Upon the notification, the Push Initiator sends another message to the device in which the WebApp is running. 6. The User Agent, if it is still listening for messages on the related bearer, does not deliver the incoming message to the EventSource connection (which should already be closed anyway). 7. No further message events are received by the WebApp within a reasonable timeout, e.g. 10 seconds.
Pass-Criteria	After closing the EventSource connection, the WebApp receives no further message events via the EventSource object.

5.2.7 Multiple Push API Connections

Test Case ID	WRAPI-1.0-UA-7
Test Object	User Agent
Test Case Description and Purpose	Verify ability to create multiple Push API connections.
Specification Reference	[WRAPI-Push] Section 7
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-003-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, PPG | SMSC | SIP Proxy, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-3 | WRAPI-1.0-UA-4 | WRAPI-1.0-UA-5
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event (“message” | “SMS” | “SIP”), the WebApp opens another EventSource connection of the same or different type, and receives another event via that connection as in the prerequisite test.
Pass-Criteria	The WebApp is able to open and simultaneously use two EventSource connections of the same or different type.

5.2.8 Event Filtering for OMA Push

Test Case ID	WRAPI-1.0-UA-8
Test Object	User Agent
Test Case Description and Purpose	Verify ability to filter events for OMA Push.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, PPG, Test WebApp
Test Code/Files	To be defined

Preconditions	WRAPI-1.0-UA-3
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push as in WRAPI-1.0-UA-3. 2. The WebApp initiates two EventSource connections for OMA Push event delivery as in WRAPI-1.0-UA-3. One connection includes a filter parameter on the Push Application ID, and the other includes a filter parameter on the content type of Push events. http://localhost:4035/?push-accept-source=urn:oma:xml:push&push-accept-application-id=myapp.com http://localhost:4035/?push-accept-source=urn:oma:xml:push&push-accept-content-type=text/vnd.wap.si 3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in WRAPI-1.0-UA-3. 4. The Push Initiator requests delivery of two Push Messages via the PPG. One message includes the Push Application ID indicated by the WebApp for the first EventSource connection, and the other includes the content type indicated by the WebApp for the second EventSource connection. 5. The Push messages are delivered as in WRAPI-1.0-UA-3, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.
Pass-Criteria	The first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.

5.2.9 Event Filtering for SMS and SIP

Test Case ID	WRAPI-1.0-UA-9
Test Object	User Agent
Test Case Description and Purpose	Verify ability to filter events for SMS and SIP.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-UA-002-M WRAPI-PUSH-UA-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, SMSC | SIP Proxy, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-UA-4 | WRAPI-1.0-UA-5

	Push Initiator is configured with ability to send messages using two different source addresses, e.g. via two different SMS API keys or using two different SIP registrations.
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (SMS | SIP), with the additional functions described below. 2. The WebApp initiates two EventSource connections for event delivery as in the prerequisite test. The first connection includes an event source filter parameter with one of the source addresses used by the Push Initiator, and the second connection includes an event source filter with the value ‘*’. 3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in the prerequisite test. 4. The Push Initiator sends a message using the source address specified by the WebApp in the first connection, and a second message from another source address. 5. The Push messages are delivered as in the prerequisite test, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first message is received at both EventSource objects, and the second message by only the second EventSource object.
Pass-Criteria	The first message is received at both EventSource objects, and the second message by only the second EventSource object.

5.2.10 User Consent for Push API Service Access

Test Case ID	WRAPI-1.0-UA-10
Test Object	User Agent
Test Case Description and Purpose	Verify that in cases which pre-arranged trust relationships do not apply (i.e. are not supported, or not applicable for the specific WebApp), the Push Client ensures user consent for access to the Push API service by the WebApp.
Specification Reference	[WRAPI-Push] Section 8.3
SCR Reference	WRAPI-PUSH-UA-008-M
ETR Reference	WRAPI_Push-M-005
Tool	Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. From a WebApp for which pre-arranged trust relationships do not apply (i.e. are not supported, or not applicable for the specific WebApp), initiate a Push API request. 2. The User Agent informs the tester of the request including the WebApp’s origin, and asks for consent to allow the request. Note that no specific UI is mandated, but

	per the overall intent of the requirement, a user informed consent process must occur. 3. The tester accepts the request, and the connection is successfully established e.g. as in WRAPI-1.0-UA-1. 4. The WebApp initiates another Push API request, using a different source address filter. 5. The User Agent informs the tester of the new request including the WebApp's origin, and asks for consent to allow the request. 6. The tester accepts the request, and the second connection is successfully established e.g. as in WRAPI-1.0-UA-1. 7. Through a provided user interface, the tester revokes granted API permission, either for one or both of the existing requests (UI or capabilities may vary). 5. The User Agent terminates the EventSource connection for the revoked permissions, and the WebApp receives an "error" event at the EventSource object related to the revoked Push API connections.
Pass-Criteria	Users are allowed to express informed consent for establishment of Push service for a WebApp, and are able to revoke earlier granted permissions.

5.3 PPG as Push API Server

5.3.1 EventSource Creation

Test Case ID	WRAPI-1.0-PPG-1
Test Object	PPG
Test Case Description and Purpose	Verify creation of EventSource connections for various event source types.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-S-001-M WRAPI-PUSH-S-002-M
ETR Reference	WRAPI_Push-M-001, WRAPI_Push-M-006
Tool	User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	None
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection as described below, for each event source (OMA Push, SMS, and SIP) supported by the PPG as Push API server. 2. The WebApp requests a new EventSource object creation with URL parameter

	as required for event delivery via the PPG, e.g. in the forms: http://example.com http://example.com/?push-accept-source=* http://example.com/?push-accept-source=urn:oma:xml:push http://example.com/?push-accept-source=sms:+14255551212 http://example.com/?push-accept-source=sip:bob@testserver.net For each of these requests, the following steps occur in the test: 3. The User Agent returns a new EventSource object. 4. The WebApp assigns a function to the onopen attribute of the returned EventSource object, as handler for “open” events, and assigns a function to the onerror attribute of the returned EventSource object, as handler for “error” events. 5. If the source is supported, the PPG returns a successful response to the User Agent as described in [WRAPI-Push], including an “Access-Control-Allow-Origin” header with the value of the “Origin:” header received in the GET request. If the source is not supported, the Push Client returns a 403 FORBIDDEN response. 6. Consequently, if the source is supported, the WebApp receives an “open” event to the onopen handler. If the source is not supported, the WebApp receives an “error” event to the onerror handler.
Pass-Criteria	The WebApp receives either an “open” event or a SYNTAX_ERR exception for each request, according to whether the source is supported or not.

5.3.2 EventSource Creation Error: Invalid Event Source

Test Case ID	WRAPI-1.0-PPG-2
Test Object	PPG
Test Case Description and Purpose	Verify failure of EventSource setup for invalid push-accept-source values.
Specification Reference	[WRAPI-Push] Section 7.1.1
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-004-M
ETR Reference	WRAPI_Push-M-006
Tool	User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	None

Test Procedure	1. Load a Test WebApp designed to attempt EventSource establishment for invalid event delivery via a PPG as Push API server. 2. The WebApp requests a new EventSource object creation with an invalid push-accept-source parameter, e.g.: http://example.com/?push-accept-source=xyz 3. The User Agent returns a new EventSource object. 4. The WebApp assigns a function to the onerror attribute of the returned EventSource object, as handler for "error" events. 5. The PPG returns a 403 FORBIDDEN response. 6. The WebApp receives an "error" event to the onerror handler.
Pass-Criteria	The WebApp receives a SYNTAX_ERR exception when an unsupported Push API event source is requested.

5.3.3 Event Delivery for OMA Push

Test Case ID	WRAPI-1.0-PPG-3
Test Object	PPG
Test Case Description and Purpose	Verify delivery of OMA Push events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-1 Push Initiator is configured for access to WRAPI Push API service via the PPG, e.g. using PAP or PushREST APIs.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push event delivery via a PPG as Push API server. 2. The WebApp requests a new EventSource object creation for OMA Push event delivery as in WRAPI-1.0-PPG-1. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns a function to the onmessage attribute of the EventSource object, as handler for "message" events. 4. Upon the notification, the Push Initiator sends a Push Message request to the

	PPG, targeted to the device in which the WebApp is running. 5. The PPG accepts the Push Message request, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection. 6. The WebApp receives a “message” event to the onmessage handler.
Pass-Criteria	The WebApp receives the OMA Push message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.3.4 Event Delivery for SMS

Test Case ID	WRAPI-1.0-PPG-4
Test Object	PPG
Test Case Description and Purpose	Verify delivery of SMS events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-1 Push Initiator is configured for access to WRAPI Push API service via the PPG, e.g. using an SMS API.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SMS event delivery via a PPG as Push API server. 2. The WebApp requests a new EventSource object creation for SMS event delivery as in WRAPI-1.0-PushClient-2, setting the push-accept-source parameter of the EventSource URL to the SMS source address to be used by the Push Initiator. 3. Upon the “open” event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for “SMS” events. 4. Upon the notification, the Push Initiator sends an SMS to the device in which the WebApp is running. 5. The PPG accepts the SMS message request, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection.

	6. The WebApp receives a "SMS" event to the assigned event listener.
Pass-Criteria	The WebApp receives the SMS message data, mapped to the text/event-stream MIME type as described in [WRAPI-Push].

5.3.5 Event Delivery for SIP

Test Case ID	WRAPI-1.0-PPG-5
Test Object	PPG
Test Case Description and Purpose	Verify delivery of SIP events.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-006-M
ETR Reference	WRAPI_Push-M-002
Tool	Push Initiator, SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-1 Push Initiator and PPG are registered with the SIP Proxy. PPG is configured to serve the SIP address of the target device, and forward SIP MESSAGE events to the target device over EventSource connections.
Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for SIP event delivery via a PPG as Push API server. 2. The WebApp requests a new EventSource object creation for SIP event delivery as in WRAPI-1.0-PPG-2, setting the push-accept-source parameter of the EventSource URL to the SIP source address to be used by the Push Initiator. 3. Upon the "open" event, the Webapp sends a notification to the Push Initiator indicating that the EventSource is opened, and assigns an event listener to the EventSource object, as handler for "SIP" events. 4. Upon the notification, the Push Initiator sends a SIP MESSAGE to the device in which the WebApp is running. 5. The SIP MESSAGE message is delivered by the SIP Proxy to the PPG**.** 6. The PPG accepts the SIP MESSAGE, maps the content to the text/event-stream MIME type, and delivers an event to the User Agent via the EventSource connection. 7. The WebApp receives a "SIP" event to the assigned event listener.
Pass-Criteria	The WebApp receives the SIP MESSAGE data, mapped to the text/event-stream

	MIME type as described in [WRAPI-Push].

5.3.6 Push API Connection Closure by WebApp

Test Case ID	WRAPI-1.0-PPG-6
Test Object	PPG
Test Case Description and Purpose	Verify Push API connection closure by WebApp.
Specification Reference	[WRAPI-Push] Section 7.3
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-001-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-3 | WRAPI-1.0-PPG-4 | WRAPI-1.0-PPG-5
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event ("message" | "SMS" | "SIP"), the WebApp calls the close() method on the EventSource object, and sends a notification to the Push Initiator that the EventSource has been closed. The WebApp does not remove the onmessage handler as it will be used to verify that no further events are delivered by the User Agent via that EventSource connection. 4. The User Agent closes the EventSource connection. 5. Upon the notification, the Push Initiator sends another message to the device in which the WebApp is running. 6. The PPG does not deliver the new request via the EventSource connection (which should already be closed anyway). 7. No further message events are received by the WebApp within a reasonable timeout, e.g. 10 seconds.
Pass-Criteria	After closing the EventSource connection, the WebApp receives no further message events via the EventSource object.

5.3.7 Multiple Push API Connections

Test Case ID	WRAPI-1.0-PPG-7
Test Object	PPG
Test Case Description and Purpose	Verify ability to create multiple Push API connections.
Specification Reference	[WRAPI-Push] Section 7
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-003-M
ETR Reference	WRAPI_Push-M-001
Tool	Push Initiator, SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-3 | WRAPI-1.0-PPG-4 | WRAPI-1.0-PPG-5
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (OMA Push, SMS, SIP), with the additional functions described below. 2. The WebApp opens an EventSource connection and receives an event as described in the prerequisite test. 3. Upon the applicable event (“message” | “SMS” | “SIP”), the WebApp opens another EventSource connection of the same or different type, and receives another event via that connection as in the prerequisite test.
Pass-Criteria	The WebApp is able to open and simultaneously use two EventSource connections of the same or different type.

5.3.8 Event Filtering for OMA Push

Test Case ID	WRAPI-1.0-PPG-8
Test Object	PPG
Test Case Description and Purpose	Verify ability to filter events for OMA Push.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, User Agent, Test WebApp
Test Code/Files	To be defined

Preconditions	WRAPI-1.0-PPG-3

Test Procedure	1. Load a Test WebApp designed to establish an EventSource connection for OMA Push as in WRAPI-1.0-PPG-3.
	2. The WebApp initiates two EventSource connections for OMA Push event delivery as in WRAPI-1.0-PPG-3. One connection includes a filter parameter on the Push Application ID, and the other includes a filter parameter on the content type of Push events.
	`http://exampl.com/?push-accept-source=urn:oma:xml:push&push-accept-application-id=myapp.com`
	`http://example.com/?push-accept-source=urn:oma:xml:push&push-accept-content-type=text/vnd.wap.si`
	3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in WRAPI-1.0-PPG-3.
	4. The Push Initiator requests delivery of two Push Messages via the PPG. One message includes the Push Application ID indicated by the WebApp for the first EventSource connection, and the other includes the content type indicated by the WebApp for the second EventSource connection.
	5. The Push messages are delivered as in WRAPI-1.0-PPG-3, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.

Pass-Criteria	The first message is received at the first EventSource object, and the second message at the second EventSource object. The EventSource objects receive only one message each.

5.3.9 Event Filtering for SMS and SIP

Test Case ID	WRAPI-1.0-PPG-9
Test Object	User Agent
Test Case Description and Purpose	Verify ability to filter events for SMS and SIP.
Specification Reference	[WRAPI-Push] Section 7.2
SCR Reference	WRAPI-PUSH-S-002-M WRAPI-PUSH-S-005-M
ETR Reference	WRAPI_Push-M-003
Tool	Push Initiator, SIP Proxy, User Agent, Test WebApp
Test Code/Files	To be defined
Preconditions	WRAPI-1.0-PPG-4 | WRAPI-1.0-PPG-5

	Push Initiator is configured with ability to send messages using two different source addresses, e.g. via two different SMS API keys or using two different SIP registrations.
Test Procedure	1. Load Test WebApp which is designed per the specific scenario under test (SMS | SIP), with the additional functions described below. 2. The WebApp initiates two EventSource connections for event delivery as in the prerequisite test. The first connection includes an event source filter parameter with one of the source addresses used by the Push Initiator, and the second connection includes an event source filter with the value '*'. 3. The EventSource connections are successfully established, and the WebApp informs the Push Initiator as in the prerequisite test. 4. The Push Initiator sends a message using the source address specified by the WebApp in the first connection, and a second message from another source address. 5. The Push messages are delivered as in the prerequisite test, and received by the WebApp as message events to the appropriate EventSource objects, i.e. the first message is received at both EventSource objects, and the second message by only the second EventSource object.
Pass-Criteria	The first message is received at both EventSource objects, and the second message by only the second EventSource object.

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/Candidate Version 1.0 History

Document Identifier	Date	Section	Description
Draft Versions OMA-ETS-WRAPI-V1_0	18 Sep 2012	All	Baseline agreed in OMA-IOP-BRO-2012-0055-INP_WRAPI_ETS_Baseline
Candidate Version OMA-ETS-WRAPI-V1_0	08 Oct 2013	N/A	Status changed to Candidate by TP TP Ref#OMA-TP-2013-0315-INP_WRAPI_V1_0_ETS_for_Candidate_approval