Enabler Validation Plan for Authorization Framework for Network APIs

Candidate Version 1.0 – 13 May 2014

Open Mobile Alliance

OMA-EVP-Autho4API-V1_0-20140513-C

Contents

Figures

Tables

1. Scope

This document details the Validation plan for the Authorization Framework for Network APIs V1.0 Enabler Release. The successful accomplishment of the validation activities will be required for the Enabler to be considered for Approved status.

The validation plan for the Authorization Framework for Network APIs V1.0 Enabler Release specifications is based on testing expectations in the Enabler Test Requirements (ETR). While the specific test activities to be performed are described in the Enabler Test Specification (ETS) the test environment is described in this plan. This test environment details infrastructure, operational and participation requirements identified for the needed testing activities.

1.1 Assumptions

None.

1.2 Exclusions

None.

2. References

2.1 Normative References

[Autho4API_ER] "Authorization Framework for Network APIs", Version 1.0, Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL:http://www.openmobilealliance.org/

[Autho4API_ETR] "Enabler Test Requirements for Authorization Framework for Network APIs", Version 1.0, Open Mobile Alliance™, OMA-ETR-Autho4API-V1_0, URL:http://www.openmobilealliance.org/

[Autho4API_ETS] "Enabler Test Specification for Authorization Framework for Network APIs", Version 1.0, Open Mobile Alliance™, OMA-ETS-Autho4API-V1_0, URL:http://www.openmobilealliance.org/ [Note: ETS to be drafted and completed later]

[Autho4API_RD] "Authorization Framework for Network APIs Requirements", Version 1.0, Open Mobile Alliance™, OMA-RD-Autho4API-V1_0, URL:http://www.openmobilealliance.org/

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.11, Open Mobile Alliance™, OMA-ORG-IOP_Process-V1_11, 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

[SUP-XSD_rest_autho4api_url_prefixes] "XML data type definitions for OMA Autho4API", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_rest_autho4api_url_prefixes_for_granted_resources-V1_0, URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, 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.

3.2 Definitions

For the purpose of this specification, all definitions from the OMA Dictionary apply [OMADICT].

Access Token	Credentials used to access protected resources, further defined in section 1.4 of [draft-ietf-oauth-v2].
Authorization Grant	Credential representing the Resource Owner's authorization (to access its protected resources) used by the Autho4API Client to obtain an Access Token.
Refresh Token	Credentials used to obtain new Access Tokens, further defined in section 1.5 of [draft-ietf-oauth-v2].
Resource Owner	An entity capable of granting access to a protected resource (e.g. end-user).
Scope	The total resource access privileges of a given authorization process, either requested by Autho4API Client or issued to Autho4API Client in the form of an Access Token. It is expressed as a list of Scope Values.
Scope Value	A well-defined set of authorized operations on Network API resources.

3.3 Abbreviations

AES	Advanced Encryption Standard
API	Application Program Interface
Autho4API	Authorization Framework for Network APIs
CBC	Cipher Block Chaining
GSMA	Global System for Mobile communications Association
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MSISDN	Mobile Subscriber ISDN Number
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
OS	Operating System
SMS	Short Message Service
TLS	Transport Layer Security
URI	Unified Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
WAC	Wholesale Applications Community
XML	eXtensible Markup Language

4. Enabler Validation Description

The validation of this enabler will be made by the successful execution of test cases defined on the [Autho4API_ETS] and that satisfy the agreed test requirements listed in [Autho4API_ETR].

A summary of test results will be the evidence that those test cases were executed. The lack of open Problem Reports will be the evidence the test cases were successful executed and the reasons for any failure resolved as not a specification issue.

More details can be found later in this document.

5. TestFest Activities

5.1 Enabler Test Guidelines

5.1.1 Minimal Test Configuration

The enabler Autho4API v1.0 intends to test the components of the following diagram:

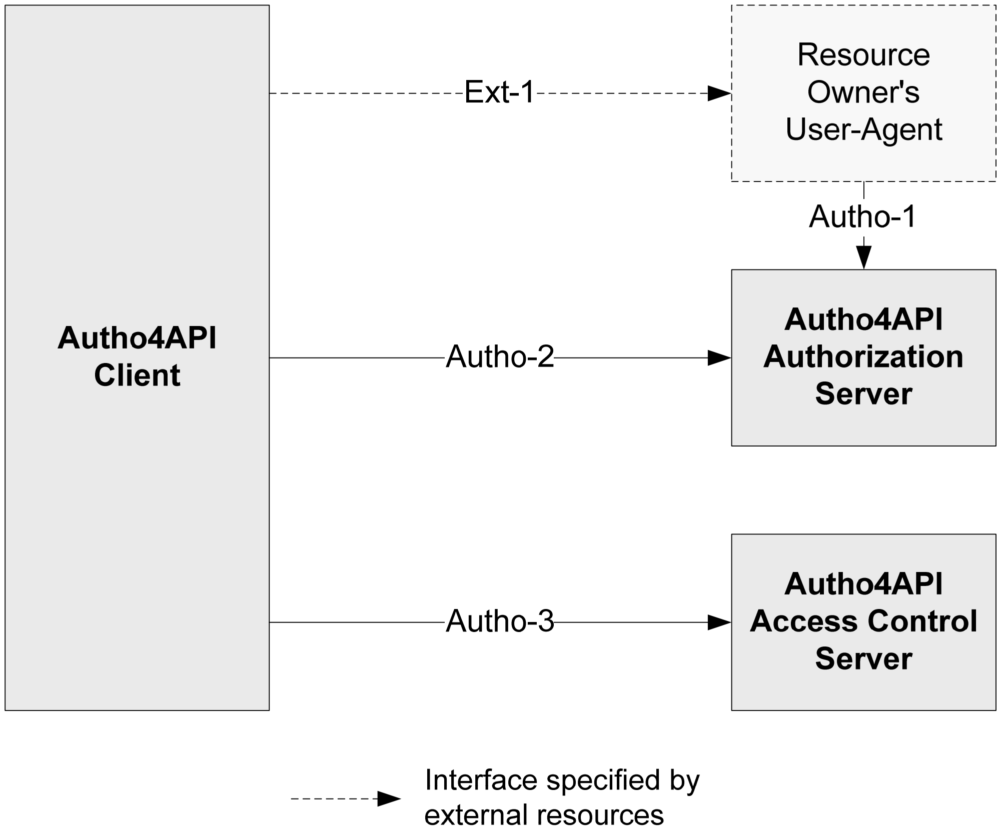
Autho4API Client
Ext-1
Resource Owner's User-Agent
Autho-1
Autho-2
Autho4API Authorization Server
Autho-3
Autho4API Access Control Server
Interface specified by external resources

Figure 1: Logical Architectural Diagram

The requirements for testing interoperability of the enabler Autho4API v1.0 are:

1. A public (or confidential) Autho4API Client registers and gets client_id (and client_secret), as specified in [Autho4API_ER].
2. Autho4API Client obtains Resource Owner’s authorization according to at least one of the following methods:
    - Using Authorization Code flow:
        - Autho4API Authorization Server identify Autho4API Client, verify scope value validity, verify redirect_uri validity (a client’s public HTTP URL), authenticate Resource Owner, obtain authorization from Resource Owner and deliver the Authorization Response via HTTP redirection to redirect_uri, as specified in [Autho4API_ER]. Autho4API Authorization Server enforce transport layer security on the interface Autho-1 in Figure 1
    - Using Implicit Grant flow:
        - Autho4API Authorization Server identify Autho4API Client, verify scope value validity, verify redirect_uri validity (a client’s public HTTP URL), authenticate Resource Owner, obtain authorization from Resource

Owner and deliver the Access Token Response via HTTP redirection to redirect_uri, as specified in [Autho4API_ER] . Autho4API Authorization Server enforce transport layer security on the interface Autho-1 in Figure 13

3. Autho4API Client obtains an Access Token according to different client types:
   - Obtain an Access Token for a public client (using the Authorization Code flow)
     - Autho4API Authorization Server identify Autho4API Client, verify authorization code validity, generate an Access Token as specified in [Autho4API_ER]. Autho4API Authorization Server deliver Access Token to Autho4API Client with enforcing transport layer security on the interface Autho-2 in Figure 1.
   - Obtain an Access Token for a confidential client (using the Authorization Code flow)
     - Autho4API Authorization Server authenticate Autho4API Client, verify authorization code validity, generate an Access Token as specified in [Autho4API_ER]. Autho4API Authorization Server deliver Access Token to Autho4API Client with enforcing transport layer security on the interface Autho-2 in Figure 1.
4. Autho4API Client accesses the protected resource
   - Autho4API Access Control Server check the validity of Access Token sent in the Authorization header of resource request and if valid return the protected resource, as specified in [Autho4API_ER]. Autho4API Access Control Server enforce transport layer security on the interface Autho-3 in Figure 1.

5.1.2 Minimal Participation Guidelines

Minimum

- 4 different Autho4API Client implementations, 2 clients supporting the authorization code flow and 2 supporting the implicit grant flow
- 2 different Autho4API Authorization Server implementations
- 2 different Autho4API Access Control Server implementations.

5.1.3 Optimal TestFest Achievement Test Case Priority Guidelines

This list represents the current highest priority test cases that the participants should attempt to perform at the event. In order to facilitate maximum test coverage of the functionality of the enabler over a number of TestFests, this list may be modified by the IOP WG between test events to reflect the latest priorities. Therefore the ETS Test Cases listed below represent a subset of all the Test Cases for the Enabler that it is thought can be executed in a single test session at an OMA TestFest. It is not intended to be the only tests executed at a TestFest, and teams are encouraged to execute more tests if they are able to do so in the time allowed.

The list includes:

5.1.3.1 Registration of confidential client

Description	Test Case Id	Priority
Client of type confidential registration	**Autho4API-1.0-int-001**	

5.1.3.2 Registration of public client

Description	Test Case Id	Priority
Client of type public registration	**Autho4API-1.0-int-002**	

5.1.3.3 Obtain Resources Owner’s authorization

Description	Test Case Id	Priority
Obtain Resources Owner’s authorization - Authorization Code flow	**Autho4API-1.0-int-003**	
Obtain Resources Owner’s authorization – Implicit Grant flow	**Autho4API-1.0-int-004**	

5.1.3.4 Obtain Access Token

Description	Test Case Id	Priority
Obtain Access Token for a public client - Authorization Code flow	**Autho4API-1.0-int-005**	
Obtain Access Token for a Confidential Client - Authorization Code Flow	**Autho4API-1.0-int-006**	

5.1.3.5 Access to Protected Resources

Description	Test Case Id	Priority
Access to protected resources	**Autho4API-1.0-int-007**	

5.1.3.6 Error Handling in Authorization

Description	Test Case Id	Priority
Error handling when requesting authorization with no response using the Authorization Code flow	**Autho4API-1.0-int-008**	
Error handling when requesting authorization with no response using the Implicit Grant flow	**Autho4API-1.0-int-009**	
Error handling when requesting authorization with wrong client ID in response using the Authorization Code flow	**Autho4API-1.0-int-010**	
Error handling when requesting authorization with wrong client ID using the Implicit Grant flow	**Autho4API-1.0-int-011**	

5.1.3.7 Error Handling in Access Token

Description	Test Case Id	Priority
Error handling when requesting an Access Token with an unauthorized client with a Public Client	**Autho4API-1.0-int-012**	
Error handling when requesting an Access Token with an unauthorized client using a Private Client	**Autho4API-1.0-int-013**	

5.1.3.8 Error Handling in Resources Access

Description	Test Case Id	Priority
Error handling when accessing protected content due to an invalid request due to missing parameter	**Autho4API-1.0-int-014**	
Error handling when accessing protected content due to an expired token used	**Autho4API-1.0-int-015**	
Error handling when accessing protected content due to a token that does not grant access to the scope of the request	**Autho4API-1.0-int-016**	

5.2 Enabler Test Requirements

5.2.1 Test Infrastructure Requirements

To execute the test cases referred on this EVP, it will be required a TCP-IP connection between the Auto4API client, the Autho4API Authorization Server and the Autho4API Access Control Server.

The Resource Owner's User Agent can be on the device where is the Autho4API client or shared with the Autho4API Authorisation Server.

It is required an REST API based enabler to exercise the Autho4API enabler. There are several possible enablers however to test Autho4API the more suitable is Payments REST API.

5.2.2 Enabler Execution Flow

5.2.2.1 Enabler Execution Flow including obtaining Authorization Code

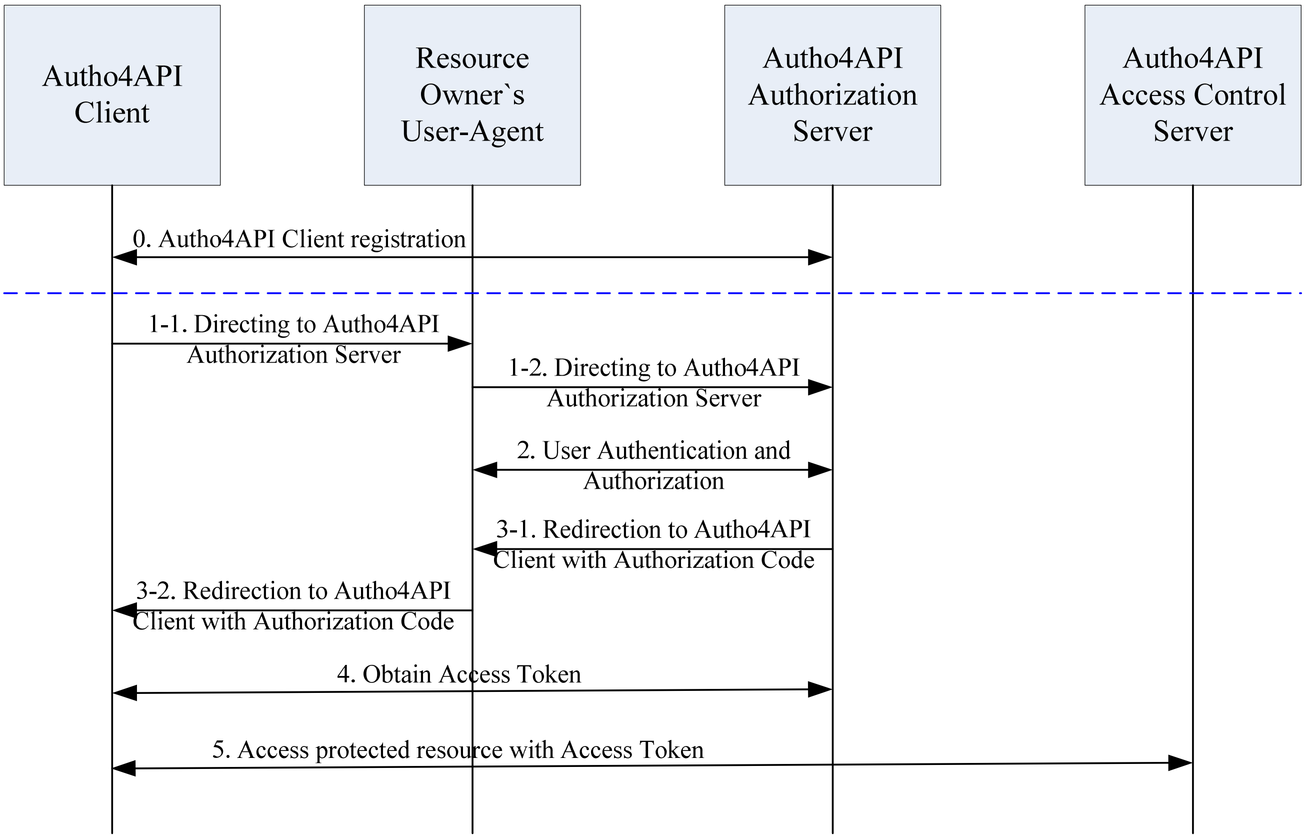
Autho4API Client
Resource Owner`s User-Agent
Autho4API Authorization Server
Autho4API Access Control Server
0. Autho4API Client registration
1-1. Directing to Autho4API Authorization Server
1-2. Directing to Autho4API Authorization Server
2. User Authentication and Authorization
3-1. Redirection to Autho4API Client with Authorization Code
3-2. Redirection to Autho4API Client with Authorization Code
4. Obtain Access Token
5. Access protected resource with Access Token

Figure 2: Enabler Execution Flow including obtaining Authorization Code

5.2.2.2 Enabler Execution Flow including obtaining Implicit Grant

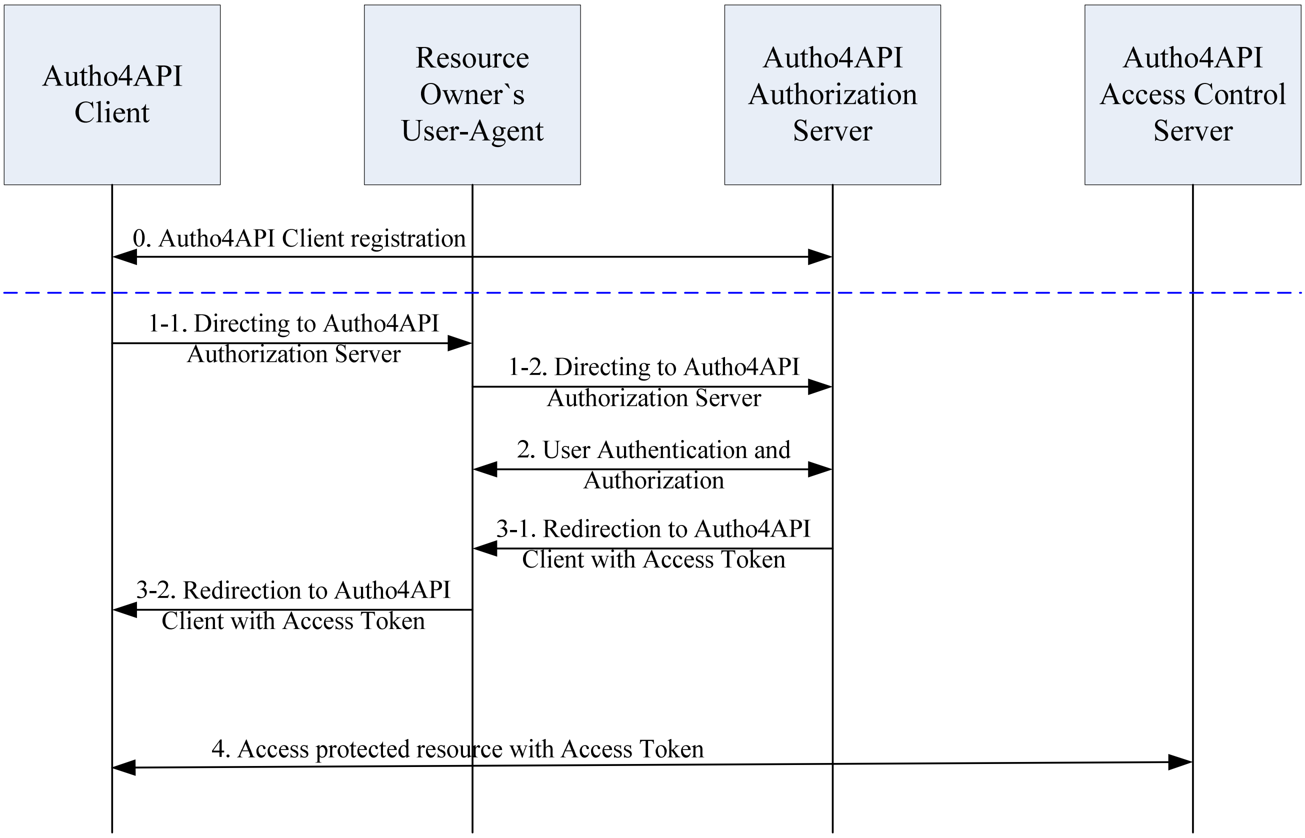
Autho4API Client
Resource Owner`s User-Agent
Autho4API Authorization Server
Autho4API Access Control Server
0. Autho4API Client registration
1-1. Directing to Autho4API Authorization Server
1-2. Directing to Autho4API Authorization Server
2. User Authentication and Authorization
3-1. Redirection to Autho4API Client with Access Token
3-2. Redirection to Autho4API Client with Access Token
4. Access protected resource with Access Token

Figure 3: Enabler Execution Flow including obtaining Implicit Grant

5.2.3 Test Content Requirements

At the moment it is foreseen that all the test content will have to be generated at the test event. If later is seen that some material can be reused it will be described in this EVP and created a TFP.

5.2.4 Test Limitations

5.2.4.1 Physical

There are no know limitations and even the test can be done remotely.

5.2.4.2 Resources

No limitation identified so far.

5.2.5 Test Restrictions

It will be difficult to evaluate the enabler performance without some type of generator of many authentications credentials requests and X-REST API enabler requests.

5.2.6 Test Tools

For the moment no test tool required. However a traffic generator of several authentication credentials requests and X-REST API requests will help understand the performance of the enabler.

5.2.6.1 Existing Tools to be Used

None.

5.2.6.2 Test Tool Requirements

Not applicable at the moment.

5.2.7 Resources Required

It will be required at least one resource to operate each of the client and another to the server.

5.3 Tests to be Performed

The following sections describe the tests related to the formal TestFest validation activities.

5.3.1 Entry Criteria for TestFest

The following tests need to be performed and passed by implementations by members wishing to participate in the TestFest. This ensures minimal requisite capability of the implementations. The tests are defined in [Autho4API-ETS] and any special comments are noted.

Test Case Id	Special Conditions
Autho4API-1.0-int-002	None
Autho4API-1.0-int-007	None

Table 1: Listing of Tests for Entry Criteria for TestFest

5.3.2 Testing to be Performed at TestFest

All the test cases need to be performed to fully cover the range of capabilities of the enabler and defined protocols. These tests are to be covered in the TestFest. The tests are defined in [Autho4API-ETS] and there are no special comments to be noted.

5.4 Enabler Test Reporting

5.4.1 Problem Reporting Requirements

The problem reports should be entered on the OMA PR tool at http://www.oma-tech.org/pr/Frontpage.aspx. The problems have to be first analysed by the IOP Champion and IOP BRO, then assigned to IOP, if it is a problem with the test specifications or to ARC working group if with the specifications.

5.4.2 Enabler Test Requirements

It will be used the normal process using the ETR produced by ARC and socialised with IOP BRO.

6. Alternative Validation Activities

The validation of the enabler can be done via test results achieved through normal test fests, via virtual test fests or bilateral test sessions.

Test results from events organised by other forums can be considered also relevant for the validation after analyses of ARC and IOP BRO.

7. Approval Criteria

The Autho4API 1.0 enabler can be put in the Approved state when:

- The Enabler has been tested successfully at 3 Test Fests or
- 4 clients, 2 supporting the authorisation code and 2 supporting 2 supporting implicit grant and 2 servers have successfully run bilateral tests sessions and
- No open PRs exist.

7.1 Enabler Validation Test Cases

The following table list the set of tests that are used for enabler validation.

Test Case Id	ETR Requirement Id	ETR Status	Notes
Autho4API-1.0-int-001	**Autho4API-001**		
Autho4API-1.0-int-002	**Autho4API-001**		
Autho4API-1.0-int-003	**Autho4API-002**	**M**	
Autho4API-1.0-int-004	**Autho4API-003**	**M**	
Autho4API-1.0-int-005	**Autho4API-004**	**M**	
Autho4API-1.0-int-006	**Autho4API-005**	**M**	
Autho4API-1.0-int-007	**Autho4API-006**	**M**	
Autho4API-1.0-int-008	**Autho4API-061**	**M**	
Autho4API-1.0-int-009	**Autho4API-061**	**M**	
Autho4API-1.0-int-010	**Autho4API-061**	**M**	
Autho4API-1.0-int-011	**Autho4API-061**	**M**	
Autho4API-1.0-int-012	**Autho4API-062**	**M**	
Autho4API-1.0-int-013	**Autho4API-062**	**M**	
Autho4API-1.0-int-014	**Autho4API-063**	**M**	
Autho4API-1.0-int-015	**Autho4API-063**	**M**	
Autho4API-1.0-int-016	**Autho4API-063**	**M**	

Table 2: Enabler Validation Test Cases

7.2 Non-Covered ETR Requirements

None of the optional test requirements are covered. However all of the mandatory ETR requirements are tested.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-EVP-Autho4API-V1_0	06 Jul 2012	All	Initial version, based on 2012 EVP Template
	18 Oct 2012		Incorporated agreed CRs: OMA-IOP-BRO-2012-0046R02-CR_Autho4API_v1.0_EVP_TestFest_Activities OMA-IOP-BRO-2012-0056R01-CR_Autho4API_EVP_remaing_sections Restored numbering of sections
	26 Mar 2014	5.1.3.1-5.1.3.8, 5.3.1, 5.3.2, 7.1, 7.2	Incorporated CR: OMA-IOP-2014-0039-CR_Update_Autho_EVP Editorial changes
Candidate Version OMA-EVP-Autho4API-V1_0	13 May 2014	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2014-0086-INP_Autho4API_V1_0_EVP_for_Candidate_approval