



Abstract Test Suite for Interoperable APIs V1.0

Deliverable D3.6

29/08/2019



ID & Title :	D3.6 Abstract Test Suite for Interoperable APIs		
Version :	V1.0	Number of pages :	28
Short Description			
Deliverable D3.6 describes the Abstract Test Suite (ATS) that will allow to validate the proper implementation by third-parties of the APIs defined in D3.4.			
Revision history			
Version	Date	Modifications' nature	Author
0.1	23/05/2019	Creation of the structure	Olivier Genest (TRIALOG)
0.2	13/06/2019	First Draft	Pierre Mauvy (TRIALOG)
0.3	15/07/2019	Completion of sections outside the test suite	Pierre Mauvy (TRIALOG)
0.4	15/07/2019	Internal review	Olivier Genest (TRIALOG)
0.5	19/07/2019	Modification following the review	Pierre Mauvy (TRIALOG)
0.6	25/07/2019	Modification following a meeting with the implementation team	Pierre Mauvy (TRIALOG)
0.7	26/07/2019	First complete version for WP3 review	Olivier Genest (TRIALOG)
0.8	02/08/2019	Modification following the WP3 review	Pierre Mauvy (TRIALOG)
1.0	29/08/2019	Final correction	Pierre mauvy (TRIALOG)
Accessibility			
<input checked="" type="checkbox"/> Public	<input type="checkbox"/> Consortium + EC	<input type="checkbox"/> Restricted to a specific group + EC	<input type="checkbox"/> Confidential + EC
Owner/Main responsible			
Name(s)	Function	Company	Visa
Marco Cupelli		RWTH	
Author(s)/contributor(s): company name(s)			
Olivier Genest, Pierre Mauvy : TRIALOG Jonas Baude, Marco Cupelli, Amir Ahmadifar: RWTH Aachen University			
Reviewer(s): company name(s)			
Company			Name(s)
Enedis, Avacon, CEZ Distribuce, E.ON, Enexis, RWTH, TRIALOG			
Approver(s): company name(s)			
Company			Name(s)
Enedis, Avacon, CEZ Distribuce, E.ON, Enexis, RWTH			
Work Package ID	WP 03	Task ID	T3.1

Disclaimer: This report reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

EXECUTIVE SUMMARY

Major DSOs are working together with market players and other stakeholders within the Horizon 2020 LCE-02-2016 project InterFlex to develop an application programming interface (API) to guarantee interoperability between the DSO and aggregator systems. Standardized interfaces will be developed to integrate the platforms of different players.

The interoperability needs to be ensured between each platform to guarantee the easy access to the flexibility whatever the platform used is. This deliverable provides an abstract test suite, based on the specification defined in the D3.4 deliverable, offering a set of tests allowing every platform to check its conformance to the API specification.

This deliverable defines the approach, procedures and parameters needed to run the tests. These tests are divided between the Request-based and the Offer-based flexibility negotiation: the two negotiation schemes require specific tests since their processes are different.

Finally, the teste suite contains 16 tests:

- 2 tests common to both negotiation;
- 8 tests specific to the Request-based flexibility negotiation;
- 6 tests specific to the Offer-based flexibility negotiation.

TABLE OF CONTENT

- 1. INTRODUCTION6
 - 1.1. Scope of the document6
 - 1.2. Abbreviations and acronyms6
 - 1.3. Motivation6
 - 1.4. References.....7
- 2. APPROACH FOR APIS TESTING8
 - 2.1. Scope8
 - 2.2. Test architecture and coverage.....8
- 3. ABSTRACT TEST SUITE (ATS).....9
 - 3.1. Test procedure9
 - 3.1.1. Test setup9
 - 3.1.2. Functional requirements on the IUT9
 - 3.1.3. IUT configuration9
 - 3.2. Protocol Implementation Conformance Statements (PICS)9
 - 3.3. Protocol Implementation eXtra Information for Testing (PIXIT)9
 - 3.4. Variables and support functions 10
 - 3.5. Common interconnection tests 11
 - 3.5.1. Phase 0: Authentication..... 11
 - 3.6. Interconnection tests for “Request-based flexibility negotiation between DSO and aggregator”..... 13
 - 3.6.1. Phase 1: Flexibility Requests 13
 - 3.6.2. Phase 2: Flexibility Offers 14
 - 3.6.3. Phase 3: Flexibility Activation 19
 - 3.7. Interconnection tests for “Offer-based flexibility negotiation between DSO and aggregator”..... 22
 - 3.7.1. Phase 1: Flexibility Offers 22
 - 3.7.2. Phase 2: Flexibility Activation 26

LIST OF TABLES AND FIGURES

Table 1 - List of acronyms.....	6
Figure 1 -Test Setup	8

1. INTRODUCTION

1.1. Scope of the document

The goal of this document is to specify the abstract test suite used to validate the conformance of InterFlex API as defined in the Deliverable D3.4 [1].

This specification includes both Request-based and Offer-based flexibility negotiation processes.

1.2. Abbreviations and acronyms

The table below provides an overview of the notations, abbreviations and acronyms used in the document.

ATS	Abstract Test Suite
DSO	Distribution System Operator
ESCO	Energy Service Company
EC	European Commission
EC-GA	European Commission Grant Agreement
EU	European Union
GA	General Assembly
GWP	General Work Package
ID	Identifier
IUT	Implementation Under Test
KPI	Key Performance Indicator
PC	Project Coordinator
PCO	Points of Control and Observation
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation eXtra Information for Testing
SC	Steering Committee
TC	Technical Committee
TD	Technical Director
WP	Work Package
WPL	Work Package Leader

Table 1 - List of acronyms

1.3. Motivation

The deliverable D3.4 of the InterFlex project [1] details an API for flexibility negotiation and activation, interconnecting DSOs and aggregators or prosumers through a flexibility cloud platform.

As this API involves several actors, interoperability is for sure an essential requirement to benefit from it. Such interoperability can be achieved by ensuring a common understanding and verifying the proper implementation of this common understanding.

The Abstract Test Suite (ATS) defined in this document contributes to these two objectives:

- The definition of the expected behaviour in each test case requires to have a clear and explicit definition of this behaviour in the API specification. Thus, during the ATS development, any grey zone in the API specification is identified and clarified to allow a common understanding. In addition, as this ATS definition is performed at the same time as the reference implementation covered by D3.5, it offers the opportunity to confront the understandings of two different parties (namely TRIALOG, defining the ATS, and RWTH Aachen, developing the reference implementation), allowing to identify additional grey zones and challenge the clarifications between several parties in order to achieve one clear and common understanding.
- The main purpose of the ATS is to define test cases that can be used by implementers or third parties in order to verify that their implementation is compliant to the expected behaviour. This ATS can be the foundation stone of an industrial certification program, allowing labs to assess and certify the compliance of any implementation to the API specification.

1.4. References

- [1] Deliverable D3.4 - version 2.0 - Interoperable APIs Specification
- [2] ISO 9646 : Conformance test methodology

2. APPROACH FOR APIS TESTING

2.1. Scope

The API aims at abstracting characteristic properties of flexibilities in order to provide a unified interface to stakeholders such as DSOs or BRPs.

The abstract test suite defines test cases to test the API. These tests cover the communication with the DSOs (or BRPs) and the aggregators through the InterFlex API.

2.2. Test architecture and coverage

The Points of Control and Observation (PCOs) are the two separate APIs used to communicate with the DSOs and Aggregator. In each test case, appropriate messages are sent to the IUT and the responses are observed.

The test covers the interconnection tests as defined in ISO/IEC 9646-1 §6.1.2 for both APIs. The exchanges on these APIs depend on the chosen flexibility negotiation approaches, Request-based flexibility negotiation or Offer-based flexibility negotiation, and the tests cover both negotiations mechanism.

The tester will have two main parts: a virtual DSO to exchange with the Upper Layer API and a virtual Aggregator to exchange with the Lower Layer API. The tester needs to be able to simulate several virtual DSOs or Aggregators.

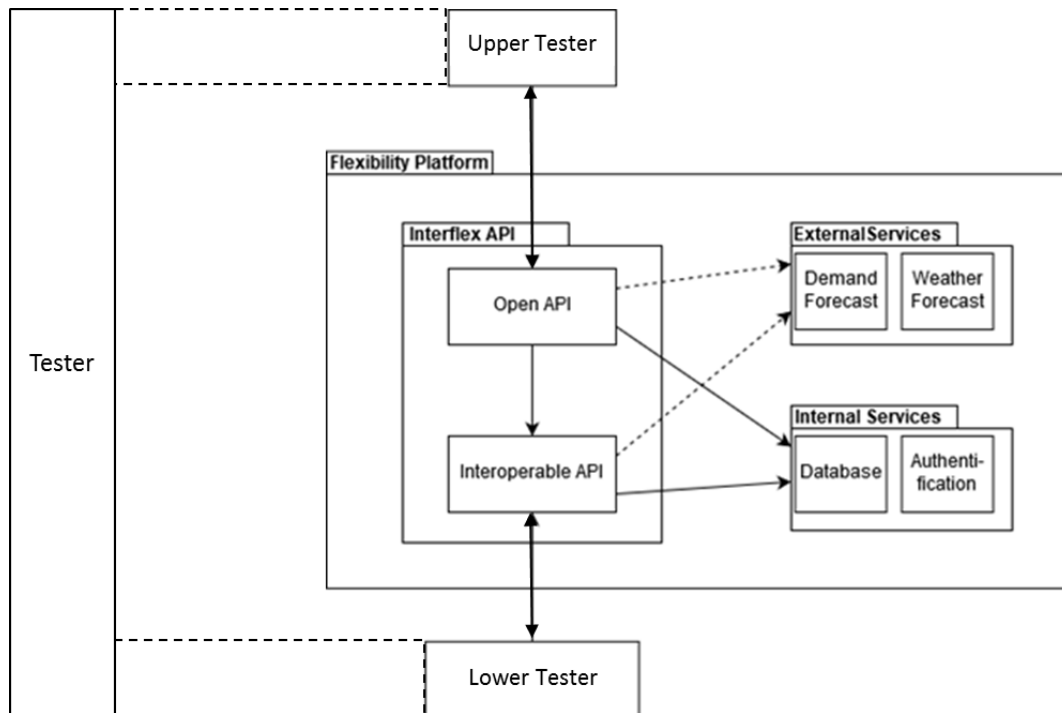


Figure 1 -Test Setup

3. ABSTRACT TEST SUITE (ATS)

3.1. Test procedure

3.1.1. Test setup

The IUT is considered as a black box with two PCOs:

- the Upper Layer API, which corresponds to the interface with the DSO
- the Lower Layer API, which corresponds to the interface with the aggregator or the prosumer.

The setup of the tester uses the two PCOs through an Upper Tester and a Lower Tester which are connected to the APIs.

3.1.2. Functional requirements on the IUT

The IUT should implement a Request-based or Offer-based flexibility negotiation. The IUT should be able to use the authentication internal services.

3.1.3. IUT configuration

The IUT must be set to the current date and hour of the testing. This may be done by being in the GMT+0 time zone.

The IUT must be able to accept authentication from virtual DSO and aggregator with the authorization identifiers and secrets defined in the PIXIT section.

3.2. Protocol Implementation Conformance Statements (PICS)

The following PICS are defined:

Name	Description	Value
pc_negociation_type	Type of negotiation supported by the IUT	0: Request-based negotiation 1: Offer-based negotiation
pc_api_version	Version of the InterFlex API	v1

3.3. Protocol Implementation eXtra Information for Testing (PIXIT)

The following PIXIT are defined:

Name	Description	Type
px_authorization_id_DSO	Authorization id for a DSO valid for the IUT	String
px_secret_DSO	Secret for the corresponding DSO id	String
px_authorization_id_aggregator_1	Authorization id for a first Aggregator valid for the IUT	String
px_secret_aggregator_1	Secret for the first Aggregator id	String
px_authorization_id_aggregator_2	Authorization id for a second Aggregator valid for the IUT	String
px_secret_aggregator_2	Secret for the second Aggregator id	String

3.4. Variables and support functions

The following variables are defined:

Name	Description	Type
vc_current_date	The current date	String
vc_DSO_authenticator	Template of the authenticator of the DSO as defined in [1]	Authenticator
vc_Aggregator_authenticator_1	Template of the authenticator of the first Aggregator as defined in [1]	Authenticator
vc_Aggregator_authenticator_2	Template of the authenticator of the second Aggregator as defined in [1]	Authenticator
vc_offer_id	The offer identifier to which corresponds the vc_flexibility and the vc_price	String
vc_flexibility	Template of the flexibility negotiated in the <code>f_flexibility_exchange()</code> function	Flexibility_t
vc_price	Template of the price negotiated in the <code>f_flexibility_exchange()</code> function	Price_t

The following constants are defined:

Name	Description	Value
tsc_entity_id_DSO	The entity_id of the virtual DSO	DSO
tsc_entity_id_Aggregator_1	The entity_id of the virtual Aggregator 1	Aggregator1
tsc_entity_id_Aggregator_2	The entity_id of the virtual Aggregator 2	Aggregator2

Name	Description	Parameters	Value
tsc_metadata_DSO	The metadata of the virtual DSO	Api_version	pc_api_version
		Entity_id	tsc_entity_id_DSO
tsc_metadata_Agg1	The metadata of the virtual Aggregator 1	Api_version	pc_api_version
		Entity_id	tsc_entity_id_Aggregator_1
tsc_metadata_Agg2	The metadata of the virtual Aggregator 2	Api_version	pc_api_version
		Entity_id	tsc_entity_id_Aggregator_2

The following functions are defined:

Name	Description
<code>f_flexibility_exchange()</code>	Function to exchange a <i>FlexibilityRequest</i> and a <i>FlexibilityOffer</i> by the Aggregator 1 and complete the vc_offer_id , vc_flexibility and vc_price variables

3.5. Common interconnection tests

The tests defined in this section are relevant for the Request-based flexibility negotiation and for the Offer-based flexibility negotiation. They should be applied irrespective of the value of `pc_negotiation_type`.

3.5.1. Phase 0: Authentication

3.5.1.1. Valid Behaviour: DSO authentication - TC_AUTH_VB_001_DSO

Description

This test validates that an authorized DSO can receive an authentication token from the IUT

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.2 Entity Authentication

Preconditions

- None

Initial State

- Virtual DSO not authenticated to the IUT

Test Body

1. The tester sends an *authenticationRequest* to the IUT through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *authorization_id* = *px_authorization_id_DSO_1*
 - *secret* = *px_secret_DSO_1*
2. The IUT must send an *authenticationResponse* through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *expires* > *vc_current_date*
 - *role* = DSO
3. Based on the *authenticationResponse* content:
 - Set *vc_DSO_authenticator.auth_token* = *token*
 - Set *vc_DSO_authenticator.refresh_token* = *refresh_token*

3.5.1.2. Valid Behaviour: Aggregator Authentication - TC_AUTH_VB_002_AGGREGATOR

Description

This test validates that an authorized Aggregator can receive an authentication token from the IUT

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.2 Entity Authentication

Preconditions

- None

Initial State

- Virtual Aggregator not authenticated to the IUT

Test Body

1. The tester sends an *authenticationRequest* to the IUT through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *authorization_id* = *px_authorization_id_aggregator_1*
 - *secret* = *px_secret_aggregator_1*
2. The IUT must send an *authenticationResponse* through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *expires* > *vc_current_date*
 - *role* = Aggregator
3. Based on the *authenticationResponse* content:
 - Set *vc_Aggregator_authenticator_1.auth_token* = *token*
 - Set *vc_Aggregator_authenticator_1.refresh_token* = *refresh_token*

3.6. Interconnection tests for “Request-based flexibility negotiation between DSO and aggregator”

The tests defined in this section are relevant only if `pc_negotiation_type` equals 1.

3.6.1. Phase 1: Flexibility Requests

In Request-based flexibility negotiation, the flexibility requests should be transferred to the Aggregator. The Phase 1 validates this mechanism.

3.6.1.1. Valid Behaviour: Flexibility request transmission -
TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST

Description

This test validates that the IUT sends back a flexibility request to an authenticated aggregator when it receives a flexibility request from a DSO

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (`pc_negotiation_type` = 1)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - `v_deadline` = `vc_current_date` + 30 seconds
 - `v_region` = a valid region id
 - `v_flexibility` = any correct flexibility
2. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - `metadata_t` = `tsc_metadata_DSO`
 - `offer_deadline` = `v_deadline`
 - `region_id` = `v_region`
 - `flexibility` = `v_flexibility`
3. The IUT must send a *FlexibilityRequest* to the IUT through the Lower Layer API before 30 seconds with:
 - `metadata_t` = `tsc_metadata_DSO`
 - `offer_deadline` = `v_deadline`
 - `region_id` = `v_region`

- $flexibility = v_flexibility$

3.6.1.2. Valid Behaviour: Flexibility request no aggregator register -
TC_REQUEST_P1_VB_002_FLEXIBILITY_REQUEST_NO_AGGREGATOR

Description

This test validates that the IUT sends a response with no flexibility when it receives a flexibility request from a DSO but there is no Aggregator authenticated

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation ($pc_negotiation_type = 1$)
- TC_AUTH_VB_001_DSO is PASS

Initial State

- Virtual DSO authenticated
- No virtual Aggregator authenticated

Test Body

1. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API
2. The IUT must not send *FlexibilityOffer* to the IUT through the Upper Layer API

3.6.2. Phase 2: Flexibility Offers

In Request-based flexibility negotiation, the flexibility offers are sent if a flexibility request is transferred to an Aggregator. The Phase 2 validates this mechanism.

3.6.2.1. Valid Behaviour: Flexibility offer -
TC_REQUEST_P2_VB_001_FLEXIBILITY_OFFER

Description

This test validates that the IUT sends a flexibility offer to the DSO when it receives a flexibility request from it and sends a valid flexibility offer in response from an Aggregator

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Response
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (`pc_negotiation_type = 1`)
- `TC_AUTH_VB_001_DSO` is PASS
- `TC_AUTH_VB_002_AGGREGATOR` is PASS
- `TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST` is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - `v_deadline = vc_current_date + 30 seconds`
 - `v_region = a valid region id`
 - `v_flexibility = any correct flexibility_t`
 - `v_price = any correct price_t`
2. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - `metadata_t = tsc_metadata_DSO`
 - `offer_deadline = v_deadline`
 - `region_id = v_region`
 - `flexibility = v_flexibility`
3. The IUT must send a *FlexibilityRequest* through the Lower Layer API to the Aggregator 1 before 30 seconds with:
 - `metadata_t = tsc_metadata_DSO`
 - `offer_deadline = v_deadline`
 - `region_id = v_region`
 - `flexibility = v_flexibility`
4. The tester sends a *FlexibilityOffer* through the Lower Layer API to the IUT with:
 - `metadata_t = tsc_metadata_Agg1`
 - `valid_until = v_deadline`
 - `region_id = v_region`
 - `flexibility_t = v_flexibility`
 - `price_t = v_price`
5. The IUT must send a *FlexibilityOffer* through the Upper Layer API to the DSO with:
 - `metadata_t = tsc_metadata_Agg1`
 - `valid_until = v_deadline`
 - `region_id = v_region`
 - `flexibility_t = v_flexibility`
 - `price_t = v_price`

3.6.2.2. Valid Behaviour: Multiple flexibility offers from multiple aggregators -
`TC_REQUEST_P2_VB_002_FLEXIBILITY_MULTIPLE_OFFERS_MULTIPLE_AGGREGATOR`

Description

This test validates that the IUT sends all the flexibility Offers when it receives a flexibility request from a DSO and several Flexibilities Offers from different Aggregators

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Response
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (`pc_negotiation_type = 1`)
- `TC_AUTH_VB_001_DSO` is PASS
- `TC_AUTH_VB_002_AGGREGATOR` is PASS
- `TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST` is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- Virtual Aggregator 2 authenticated

Test Body

1. The tester defines
 - `v_deadline` = `vc_current_date` + 30 seconds
 - `v_region` = a valid region id
 - `v_flexibility` = any correct `flexibility_t`
 - `v_price` = any correct `price_t`
 - `v_flexibility_request` = (`offer_deadline` = `v_deadline`, `region_id` = `v_region`, `flexibility` = `v_flexibility`)
2. The tester sends a `v_flexibility_request` to the IUT through the Upper Layer API
3. Before 30 seconds, in any order:
 - 3.1. The IUT must exchange with the Aggregator 1 through the Lower Layer API
 - 3.1.1. The IUT must send a `v_flexibility_request`
 - 3.1.2. The Aggregator 1 sends a `FlexibilityOffer`, `v_flex_offer_1` with:
 - `metadata_t` = `tsc_metadata_Agg1`
 - `offer_id` = `v_offer_id_1` = `tsc_entity_id_Aggregator_1` + "0001"
 - `valid_until` = `v_deadline`
 - `region_id` = `v_region`
 - `flexibility_t` = `v_flexibility`
 - `price_t` = `v_price`
 - 3.2. The IUT must exchange with the Aggregator 2 through the Lower Layer API
 - 3.2.1. The IUT must send a `v_flexibility_request`
 - 3.2.2. The Aggregator 2 sends a `FlexibilityOffer`, `v_flex_offer_2` with:
 - `metadata_t` = `tsc_metadata_Agg2`
 - `offer_id` = `v_offer_id_2` = `tsc_entity_id_Aggregator_2` + "0002"
 - `valid_until` = `v_deadline`
 - `region_id` = `v_region`
 - `flexibility_t` = `v_flexibility`
 - `price_t` = `v_price`

4. The IUT must send a `v_flex_offer_1` and `v_flex_offer_2` to the tester through the Upper Layer API with the corresponding `metadata_t`

3.6.2.3. Valid Behaviour: Flexibility multiple offer from unique aggregator -
TC_REQUEST_P2_VB_003_FLEXIBILITY_MULTIPLE_OFFERS

Description

This test validates that the IUT sends all the flexibility offers when it receives a flexibility request from a DSO and several Flexibilities offers from a unique Aggregator

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Response
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (`pc_negotiation_type = 1`)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - `v_deadline` = `vc_current_date` + 30 seconds
 - `v_region` = a valid region id
 - `v_flexibility` = any correct `flexibility_t`
 - `v_price` = any correct `price_t`
2. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - `metadata_t` = `tsc_metadata_DSO`
 - `offer_deadline` = `v_deadline`
 - `region_id` = `v_region`
 - `flexibility` = `v_flexibility`
3. The IUT must send a *FlexibilityRequest* through the Lower Layer before 30 seconds with:
 - `metadata_t` = `tsc_metadata_DSO`
 - `offer_deadline` = `v_deadline`
 - `region_id` = `v_region`
 - `flexibility` = `v_flexibility`
4. The tester sends a *FlexibilityOffer*, `v_flex_offer_1` through the Lower Layer API to the IUT with:

- *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *v_offer_id_1* = *tsc_entity_id_Aggregator_1* + “0001”
 - *valid_until* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility_t* = *v_flexibility*
 - *price_t* = *v_price*
5. The tester sends a *FlexibilityOffer*, *v_flex_offer_2* through the Lower Layer API to the IUT with:
- *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *v_offer_id_2* = *tsc_entity_id_Aggregator_1* + “0002”
 - *valid_until* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility_t* = *v_flexibility*
 - *price_t* = *v_price*
6. The IUT must send *v_flex_offer_1* and *v_flex_offer_2* to the tester through the Upper Layer API

3.6.2.4. Invalid Behaviour: Flexibility request no response from Aggregator -
TC_REQUEST_P2_IB_001_FLEXIBILITY_REQUEST_NO_REQUEST RESPONSE

Description

This test validates that the IUT sends a response with no flexibility when it receives no response from the Aggregator to a transmitted flexibility request

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (*pc_negociation_type* = 1)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - *v_deadline* = *vc_current_date* + 30 seconds
 - *v_region* = a valid region id
 - *v_flexibility* = any correct flexibility
2. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*

- *offer_deadline* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility* = *v_flexibility*
3. The tester starts a timer *tc_t1* of 30 seconds
 4. The IUT must send a *FlexibilityRequest* to the IUT through the Lower Layer API before 30 seconds with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_deadline* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility* = *v_flexibility*
 5. The tester does not respond until the end of *tc_t1*
 6. The IUT must not send a *FlexibilityOffer* to the IUT through the Upper Layer API

3.6.3. Phase 3: Flexibility Activation

The flexibility activation is the last part of a flexibility exchange after an exchange of flexibility requests and offers. The Phase 3 validates this mechanism.

3.6.3.1. Valid Behaviour: Flexibility activation -
TC_REQUEST_P3_VB_001_FLEXIBILITY_ACTIVATION

Description

This test validates that the IUT relays the messages when it receives a flexibility activation request and Acknowledgment

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.5 Flexibility Activation Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.6 Flexibility Activation Acknowledgment
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (*pc_negotiation_type* = 1)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST is PASS
- TC_REQUEST_P2_VB_001_FLEXIBILITY_OFFER is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- *f_flexibility_exchange()* done

Test Body

1. The tester sends a *FlexibilityActivation* to the IUT through the Upper Layer API with:

- *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
2. The IUT must send a *FlexibilityActivation* to the tester through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
 3. The tester sends a *FlexibilityActivationAck* to the IUT through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 4. The IUT must send a *FlexibilityActivationAck* to the tester through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*

3.6.3.2. Invalid Behaviour: Flexibility activation rejected -
 TC_REQUEST_P3_IB_001_FLEXIBILITY_ACTIVATION_REJECTED

Description

This test validates that the IUT relays the messages when it receives a flexibility activation request and Negative Acknowledgment

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.5 Flexibility Activation Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.7 Flexibility Activation Unacknowledgment
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (*pc_negotiation_type* = 1)
- *TC_AUTH_VB_001_DSO* is PASS
- *TC_AUTH_VB_002_AGGREGATOR* is PASS
- *TC_REQUEST_P1_VB_001_FLEXIBILITY_REQUEST* is PASS
- *TC_REQUEST_P2_VB_001_FLEXIBILITY_OFFER* is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- *f_flexibility_exchange()* done

Test Body

1. The tester sends a *FlexibilityActivation* to the IUT through the Upper Layer API with:
 - *metadata_t* = **tsc_metadata_DSO**
 - *offer_id* = **vc_offer_id**
 - *flexibility_t* = **vc_flexibility**
 - *price_t* = **vc_price**
2. The IUT must send a *FlexibilityActivation* to the tester through the Lower Layer API with:
 - *metadata_t* = **tsc_metadata_DSO**
 - *offer_id* = **vc_offer_id**
 - *flexibility_t* = **vc_flexibility**
 - *price_t* = **vc_price**
3. The tester sends a *FlexibilityActivationNack* to the IUT through the Lower Layer API with:
 - *metadata_t* = **tsc_metadata_Agg1**
 - *offer_id* = **vc_offer_id**
 - *flexibility_t* = **vc_flexibility**
4. The IUT must send a *FlexibilityActivationNack* to the tester through the Upper Layer API with:
 - *metadata_t* = **tsc_metadata_Agg1**
 - *offer_id* = **vc_offer_id**
 - *flexibility_t* = **vc_flexibility**

3.7. Interconnection tests for “Offer-based flexibility negotiation between DSO and aggregator”

The tests defined in this section are relevant only if `pc_negotiation_type` equals 0.

3.7.1. Phase 1: Flexibility Offers

In Offer-based flexibility negotiation, the flexibility offers are sent to a DSO if a flexibility request is received by the IUT. Since the flexibility offer mechanism cannot be tested without a request in the negotiation, the Phase 1 validates the offer and request mechanisms.

3.7.1.1. Valid Behaviour: Flexibility offer and request -
TC_OFFER_P1_VB_001_FLEXIBILITY_REQUEST

Description

This test validates that the IUT sends a flexibility offer when it receives a flexibility offer followed by a flexibility request for which the flexibility offer is relevant

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.3 Offer-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses an Offer-based flexibility negotiation (`pc_negotiation_type` = 0)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - `v_deadline` = `vc_current_date` + 30 seconds
 - `v_validity_limit` = `vc_current_date` + 1 minute
 - `v_region` = a valid region id
 - `v_flexibility` = any correct *flexibility_t*
 - `v_price` = any correct *price_t*
2. The tester sends a *FlexibilityOffer* through the Lower Layer API to the IUT with:
 - `metadata_t` = `tsc_metadata_Agg1`
 - `valid_until` = `v_validity_limit`
 - `region_id` = `v_region`

- *flexibility_t* = *v_flexibility*
 - *price_t* = *v_price*
3. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_deadline* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility* = *v_flexibility*
 4. The IUT must send a *FlexibilityOffer* before 30 seconds through the Upper Layer API to the DSO with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *valid_until* = *v_validity_limit*
 - *region_id* = *v_region*
 - *flexibility_t* = *v_flexibility*
 - *price_t* = *v_price*

3.7.1.2. Valid Behaviour: Flexibility request multiple offers -
TC_OFFER_P1_VB_002_FLEXIBILITY_REQUEST_MULTIPLE_OFFERS

Description

This test validates that the IUT sends several flexibilities offers when it receives several flexibilities offers followed by a flexibility request for which the flexibility offers are relevant

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.3 Offer-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses an Offer-based flexibility negotiation (*pc_negotiation_type* = 0)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_OFFER_P1_VB_001_FLEXIBILITY_REQUEST is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- Virtual Aggregator 2 authenticated

Test Body

1. The tester defines
 - *v_deadline* = *vc_current_date* + 30 seconds
 - *v_validity_limit* = *vc_current_date* + 1 minute
 - *v_region* = a valid region id

- **v_flexibility** = any correct *flexibility_t*
 - **v_price** = any correct *price_t*
2. The tester sends a *FlexibilityOffer*, **v_flex_offer_1** through the Lower Layer API to the IUT with:
 - *metadata_t* = **tsc_metadata_Agg1**
 - *offer_id* = **v_offer_id_1** = **tsc_entity_id_Aggregator_1** + “0001”
 - *valid_until* = **v_validity_limit**
 - *region_id* = **v_region**
 - *flexibility_t* = **v_flexibility**
 - *price_t* = **v_price**
 3. The tester sends a *FlexibilityOffer*, **v_flex_offer_2** through the Lower Layer API to the IUT with:
 - *metadata_t* = **tsc_metadata_Agg2**
 - *offer_id* = **v_offer_id_2** = **tsc_entity_id_Aggregator_2** + “0001”
 - *valid_until* = **v_validity_limit**
 - *region_id* = **v_region**
 - *flexibility_t* = **v_flexibility**
 - *price_t* = **v_price**
 4. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - *metadata_t* = **tsc_metadata_DSO**
 - *offer_deadline* = **v_deadline**
 - *region_id* = **v_region**
 - *flexibility* = **v_flexibility**
 5. The IUT must send **v_flex_offer_1** and **v_flex_offer_2** to the tester through the Upper Layer API

3.7.1.3. Valid Behaviour: Flexibility request no offer -
 TC_OFFER_P1_VB_003_FLEXIBILITY_REQUEST_NO_OFFER

Description

This test validates that the IUT sends no flexibility offers when it receives a flexibility request but have not received any flexibility offer

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.3 Offer-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses an Offer-based flexibility negotiation (*pc_negotiation_type* = 0)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - **v_deadline** = **vc_current_date** + 30 seconds
 - **v_region** = a valid region id
 - **v_flexibility** = any correct flexibility
2. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - **metadata_t** = **tsc_metadata_DSO**
 - **offer_deadline** = **v_deadline**
 - **region_id** = **v_region**
 - **flexibility** = **v_flexibility**
3. The IUT must not send a *FlexibilityOffer* in a timeout of 30 seconds to the IUT through the Upper Layer API

3.7.1.4. Invalid Behaviour: Already expires flexibility offer offer -
TC_OFFER_P1_IB_001_FLEXIBILITY_OFFER_EXPIRATION

Description

This test validates that the IUT sends a void list of flexibility offers when it receives a flexibility request but has only outdated flexibility offers

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.3 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.4 Flexibility Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.3 Offer-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses an Offer-based flexibility negotiation (**pc_negotiation_type** = 0)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_OFFER_P1_VB_001_FLEXIBILITY_REQUEST is PASS
- TC_OFFER_P1_VB_003_FLEXIBILITY_REQUEST_NO_OFFER is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated

Test Body

1. The tester defines
 - **v_validity_limit** = **vc_current_date** + 30 seconds
 - **v_region** = a valid region id
 - **v_flexibility** = any correct *flexibility_t*
 - **v_price** = any correct *price_t*
2. The tester sends a *FlexibilityOffer* through the Lower Layer API to the IUT with:

- *metadata_t* = *tsc_metadata_Agg1*
 - *valid_until* = *v_validity_limit*
 - *region_id* = *v_region*
 - *flexibility_t* = *v_flexibility*
 - *price_t* = *v_price*
3. The tester waits for 35 seconds
 4. The tester defines *v_deadline* = *vc_current_date* + 30 seconds
 5. The tester sends a *FlexibilityRequest* to the IUT through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_deadline* = *v_deadline*
 - *region_id* = *v_region*
 - *flexibility* = *v_flexibility*
 6. The IUT must not send a *FlexibilityOffer* in a timeout of 30 seconds to the IUT through the Upper Layer API

3.7.2. Phase 2: Flexibility Activation

The flexibility activation is the last part of a flexibility exchange after an exchange of flexibility requests and offers. The Phase 3 validates this mechanism.

3.7.2.1. Valid Behaviour: Flexibility activation -
TC_OFFER_P2_VB_001_FLEXIBILITY_ACTIVATION

Description

This test validates that the IUT relays the messages when it receives a flexibility activation request and Acknowledgment

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.5 Flexibility Activation Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.6 Flexibility Activation Acknowledgment
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (*pc_negociation_type* = 1)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_OFFER_P1_VB_001_FLEXIBILITY_REQUEST is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- *f_flexibility_exchange()* done

Test Body

1. The tester sends a *FlexibilityActivation* to the IUT through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
2. The IUT must send a *FlexibilityActivation* to the tester through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
3. The tester sends a *FlexibilityActivationAck* to the IUT through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
4. The IUT must send a *FlexibilityActivationAck* to the tester through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*

3.7.2.2. Invalid Behaviour: Flexibility activation rejected -
 TC_OFFER_P2_IB_001_FLEXIBILITY_ACTIVATION_REJECTED

Description

This test validates that the IUT relays the messages when it receives a flexibility activation request and Negative Acknowledgment

Specification References

- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.5 Flexibility Activation Request
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §4.4.7 Flexibility Activation Unacknowledgment
- Interoperable APIs Specification, Deliverable D3.4 - version 2.0 - §5.2 Request-based flexibility negotiation between DSO and aggregator

Preconditions

- IUT uses a Request-based flexibility negotiation (*pc_negotiation_type* = 1)
- TC_AUTH_VB_001_DSO is PASS
- TC_AUTH_VB_002_AGGREGATOR is PASS
- TC_OFFER_P1_VB_001_FLEXIBILITY_REQUEST is PASS

Initial State

- Virtual DSO authenticated
- Virtual Aggregator 1 authenticated
- *f_flexibility_exchange()* done

Test Body

1. The tester sends a *FlexibilityActivation* to the IUT through the Upper Layer API with:

- *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
2. The IUT must send a *FlexibilityActivation* to the tester through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_DSO*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 - *price_t* = *vc_price*
 3. The tester sends a *FlexibilityActivationNack* to the IUT through the Lower Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*
 4. The IUT must send a *FlexibilityActivationNack* to the tester through the Upper Layer API with:
 - *metadata_t* = *tsc_metadata_Agg1*
 - *offer_id* = *vc_offer_id*
 - *flexibility_t* = *vc_flexibility*