D8.9

Integration Testing Report (2)

Document Identification	
Date	25.02.2019
Status	Final
Version	1.0

Related WP	WP 3, WP4, WP 5, WP6	**Related Deliverable(s)**	D3.2, D4.2, D5.2, D6.2, D8.8
Lead Authors	TUBITAK	**Dissemination Level**	PU
Lead Participants	TUBITAK	**Contributors**	TUBITAK
Reviewers	ATOS, UPRC		

This document is issued within the frame and for the purpose of the LIGHT^{est} project. LIGHT^{est} has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No 700321.

1. Executive Summary

This document is the deliverable D8.9 “Integration Testing Report (2)” of the project “Lightweight Infrastructure for Global Heterogeneous Trust management in support of an open Ecosystem of Stakeholders and Trust schemes” (LIGHT*est*, project nr. 700321) with the objective to create a global cross domain trust infrastructure that renders it transparent and easy for verifiers to evaluate electronic transactions. This deliverable is constructed upon the deliverable D8.8 “Integration Testing Report (1)” and it covers the second iteration on integration testing efforts.

Throughout the course of the project, integration testing will be carried out in three iterations and at each iteration a periodic report on integration testing will be published. This document D8.9 – Integration Testing Report (2) is the second iteration testing report on integration testing of LIGHT*est* components at the system level. The main contents of this deliverable include test cases that are derived from normative statements and test assertions given in D8.8 according to the applied Minder Test Assertion Model.

Results of the testing will be provided in the third iteration of the report.

2. Document Information

2.1 Contributors

Name	Partner
Berkay TOPÇU	TUBITAK
Burçin BOZKURT GÜNAY	TUBITAK
Neslihan KIZILBEY	TUBITAK
Asiye BOZKURT	TUBITAK
Sedat ÇİFTÇİ	TUBITAK

2.2 History

Version	Date	Author	Changes
0.1	07/01/2019	TUBITAK	First draft
0.2	12/02/2019	TUBITAK	Update and inclusion of Test cases
1.0	25/02/2019	TUBITAK	Final version

3. Table of Contents

3.1 Table of Figures

3.2 Table of Tables

3.3 Table of Acronyms

API	Application Program Interface
ATV	Automatic Trust Verifier
CC	Conformance Clause
DNS	Domain Name System
DNSSEC	Domain Name System SECurity extensions
DP	Delegation Publisher
eIDAS	Electronic Identification, Authentication and trust (Services)
eT	Electronic transaction
FR	Functional Requirement
HTTP(S)	Hypertext Transfer Protocol (Secure)
ISTQB	International Software Testing Qualifications Board
MTDL	Minder Test Definition Language
M1	Minder END User ATV Adapter
M2	Minder ATV Adapter
NS	Normative Statement
OASIS	Advancing Open standards for information society
PDF	Portable Document Format
RA	Reference Architecture
PTR	Pointer
REST	Representational State Transfer (service)
RR	Resource Record
S/MIME	Secure/Multipurpose Internet Mail Extensions
SUT	System Under Test
TA	Test Assertion
TA id	Test Assertion Identification Number
TCP/IP	Transmission Control Protocol / Internet Protocol
TP	Trust Policy
TPL	Trust Policy Language
TSLTS	Trust Service Status List Technical Specification
TSL	Trust Service Status List
TSP	Trust Service Provider

TSPA	Trust Scheme Publication Authority
TTA	Trust Translation Authority
URI	Uniform Resource Identifier
XML	Extensible Markup Language

4. Scope of the deliverable

4.1 Overview

The overall focus of the LIGHT*est* project is to develop a lightweight trust infrastructure providing parties of electronic transactions with automatic validation of trust based on their individual trust policies. By using an existing infrastructure of the global Domain Name System (DNS) for publication, querying, and cross-jurisdiction translation of information relevant to make such decisions, including levels of assurance, LIGHT*est* aims to enable the use of truly "global trust lists". With this approach LIGHT*est* basically provides an infrastructure to realize the most important principles and driving factors of eIDAS on a global level.

Integration testing is a level of software testing where individual LIGHT*est* components are combined and tested as a group. The purpose of this level of testing is to expose faults/defects in the interfaces and in the interactions between integrated components or systems. Task 8.4 is dedicated to integration testing. In this task, we will test outputs of other WPs in order to see whether they exchange and use information properly, interpret the exchanged information meaningfully, and multiple entities work together in a harmonious way.

This deliverable is structured as follows. Section 1 presents the executive summary. Section 2 basically includes document information and Section 3 gives the table of contents. Section 4 presents an overview of WP8 and scope of this deliverable. Section 5 summarizes the testing architecture with Minder Test Manager inclusion and revisits the testing methodology. Section 6 presents integration testing scenarios and Section 7 presents the integration test cases derived from test assertions for TSPA, TTA and DP.

4.2 Scope

Within the course of the LIGHT*est* project, integration testing of the software components developed in WP3, 4, 5, and 6 will be carried out by using Minder Testbed and the results will be reported periodically. D8.9 Integration testing report (2) is the second report of this series and focuses on defining test cases to perform integration testing of the LIGHT*est* components. Although the testing duties within the LIGHT*est* project include conformance and interoperability testing and code quality review within the WP8 activities, Task 8.4 is dedicated to the automatic integration testing of the LIGHT*est* components.

5. Integration Testing Overview

5.1 System Components Overview

The LIGHT^{est} reference architecture has three main components to be tested: Trust Scheme Publication Authority (TSPA), Trust Translation Authority (TTA), and Delegation Publisher (DP) seen in Figure 1.

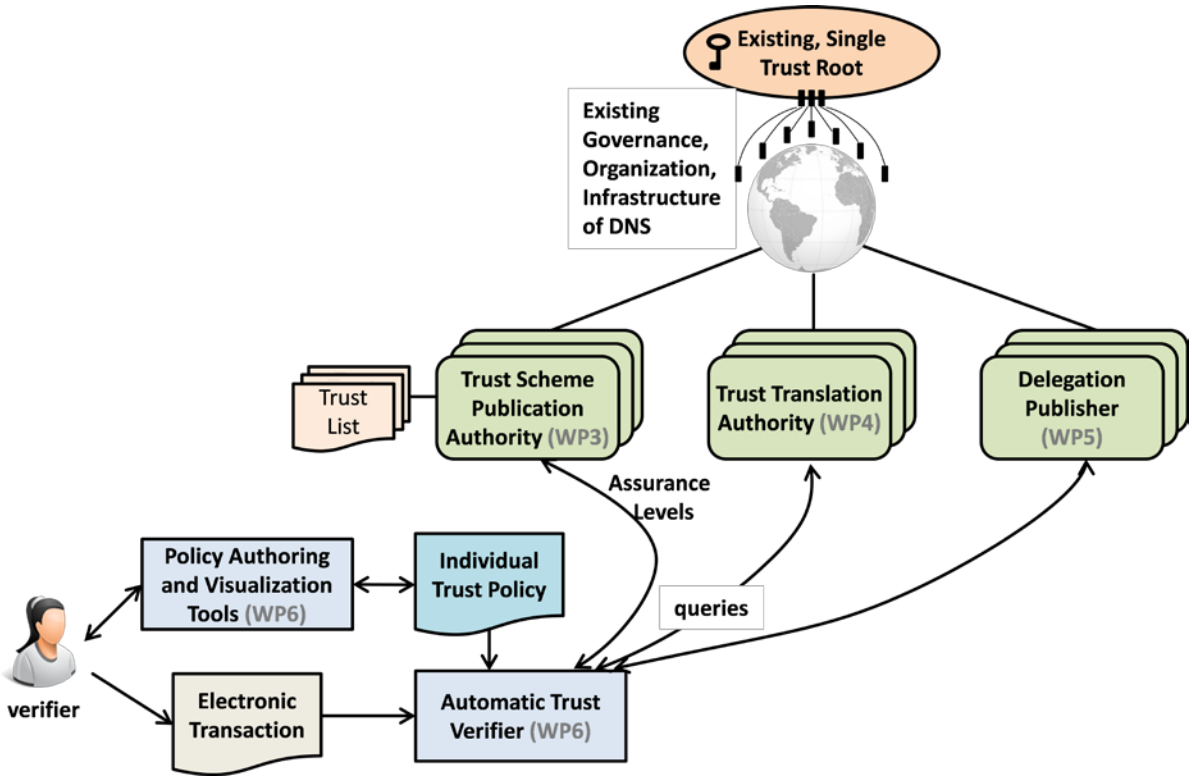

Figure 1 LIGHT^{est} Reference Architecture

5.2 Testing Methodology

The main testing methodology is already described in D8.8 Testing Methodology section. Figure 2 depicts the general anatomy of a OASIS Test Model Case, where Test cases are derived from Test assertions in form of Test Suites.

Test Suite/Case management feature is supported by Minder Test Manager.

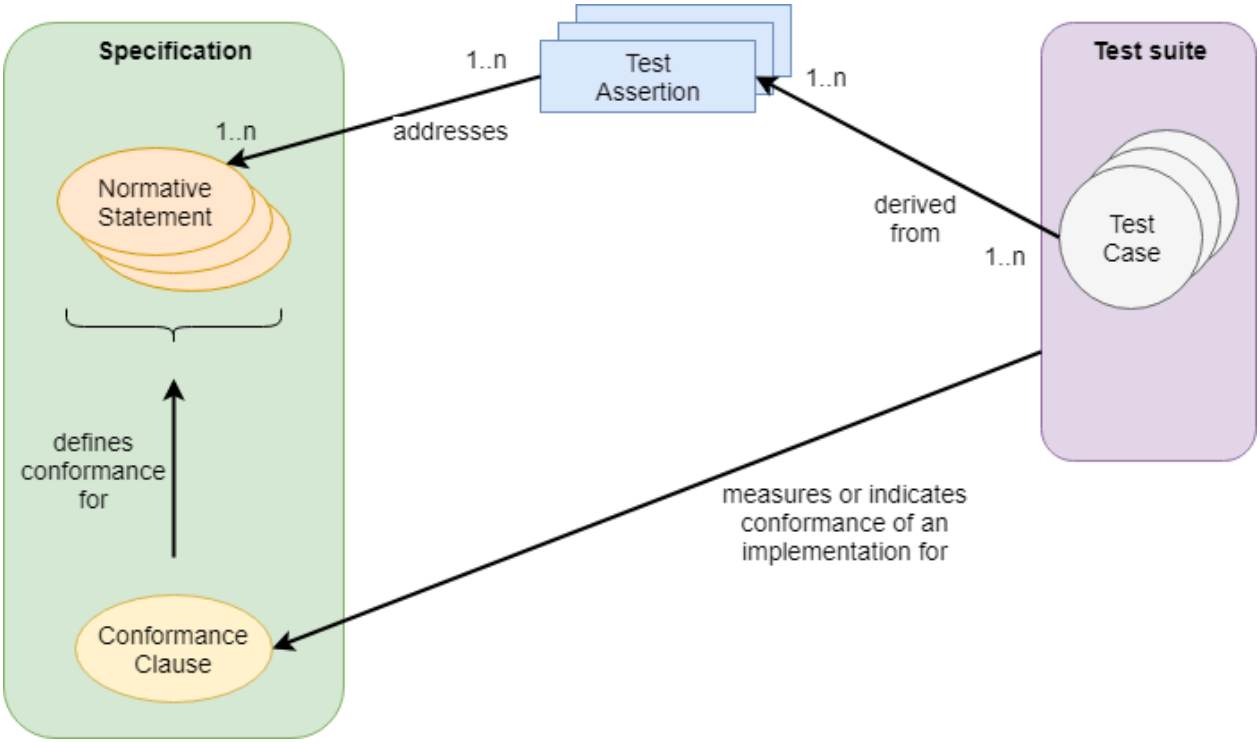

Figure 2 The relation between Test Assertions and Specifications.

5.2.1 Test Case Generation Guideline

Following the same methodology in D8.8 Section 5, test cases are derived from test assertions. The details about the test case derivation methodology is given in *Appendix A. Deriving a Test Case from a Test Assertion* given in (http://docs.oasis-open.org/tag/guidelines/v1.0/guidelines-v1.0.pdf).

As a summary, conditions to derive a test case from a test assertion are given as follows:

- When a Target instance is not qualified for a Test Assertion, a Test Case derived from this Test Assertion does not indicate whether the Target instance fulfills or not the Normative Statement addressed by the Test Assertion.
- When a Target instance is qualified for a Test Assertion and satisfies the Test Assertion Predicate, a Test Case derived from this Test Assertion either indicates that the Target instance fulfills the Normative Statement addressed by the test assertion, or does not indicate anything.
- When a Target instance is qualified for a Test Assertion and does not satisfy the Test Assertion Predicate, a Test Case derived from this Test Assertion either indicates that the Target instance does not fulfill the Normative Statement addressed by the test assertion, or does not indicate anything.

5.3 Testing Architecture Overview

As already defined in D8.8 Testing Architecture Overview, integration testing involves component integration testing that establishes the interaction between integrated modules in one system.

The second iteration integration testing plan includes the identification of test cases derived from test assertions that aims to verify the integration of TSPA, TTA and DP components with *Minder Adapters*.

For the Minder Testbed applied architecture overview please see Figure 3.

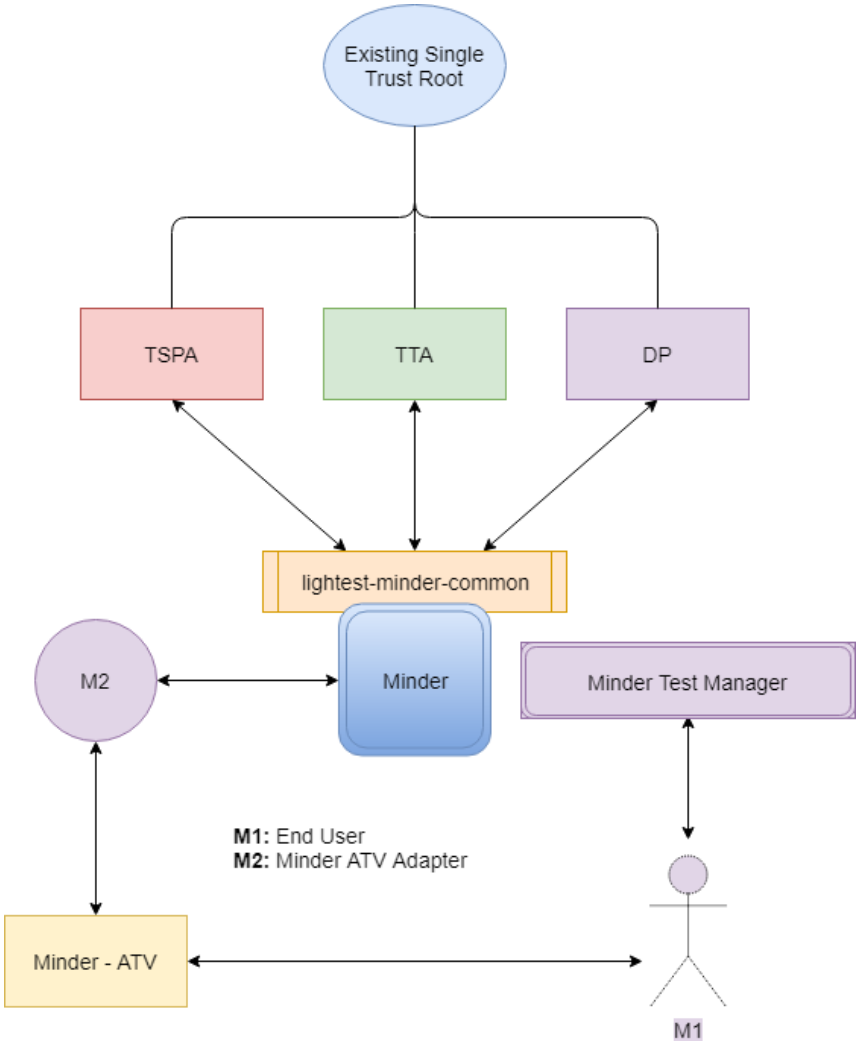

Figure 3 Minder Testbed Applied Architecture

The communication between Minder and the TSPA, and TTA, is handled directly via Minder's own DNS Client component, likewise, the communication between Minder and the DP can be

handled via Minder's HTTP component eliminating the need for adapters for all these components.

Automatic Trust Verifier (Minder-ATV) component has been re-implemented/developed and included in the testing architecture for testing purposes to manage the scenarios that include the usage of REST services provided by TSPA, TTA and DP components.

During the test case identification, features that support the management of test cases/suits and test executions are also developed within Minder Testbed architecture. The new feature is called as "Minder Test Manager" and the source codes are included in the Minder Testbed sources. The "Minder Test Manager" also uses the same DNS Client and HTTP Client components to gain access to the APIs provided by TSPA, TTA and DP components.

6. Integration Testing Scenarios:

The basic integration testing scenarios that are extracted from the general scenarios detailed in D2.14, D3.2, D3.4, D4.2, D4.4, D5.2 and D5.4. are already given in D8.8 Integration Testing Scenarios section. The scenarios are given as follows.

1. Querying of Trust Scheme Membership
2. Querying of Trust Translation List
3. Discovering of Trust Delegation

7. Integration Test Assertions and Test Cases

This section lists the second release of test assertions for TSPA, TTA, and DP, and first release of test cases derived from test assertions TSPA, TTA, and DP for respectively. First release of test assertions are already given in D8.3. This section includes the updated version of assertions. Also, this section includes the initial version of test cases that will be used for test executions.

Following the methodology described in the previous section, we first analyze the normative sources together with their references, and the test assertions to elicit test cases for conformance to these specifications.

For traceability purposes, the identifiers of the test case definitions are done using the following convention: **TC_LightestComponentName(TSPA,TTA,DP)_TestCaseID**

7.1 ATV – TSPA Integration Testing

7.1.1 ATV – TSPA Integration Testing Conformance Clauses

CC_TSPA_1: ATV queries an electronic transaction from TSPA, where the trust lists are managed, whether the transaction is trustworthy.

7.1.2 ATV – TSPA Integration Normative Sources

For brevity, we removed ATV – TSPA Integration Normative Sources from this deliverable. Since there has not been any modifications to the normative statements, readers can refer to D8.8 Integration Testing Report (1) for further details.

7.1.3 ATV – TSPA Integration Test Assertions

TA ID	TA_ TSPA_1
Normative Source	NS_ TSPA_1
Target	ATV – TSPA Interface
Prerequisite	Trust policy and electronic transaction exist as test assets.
Prescription Level	Mandatory
Predicate	IP address of the TSPA DNS server exists and can be listed on the configurations and is already set on the TCP/IP Properties (DNS Server Address settings)

TA ID	TA_TSPA_2
Normative Source	NS_TSPA_4, NS_TSPA_5
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running and TSPA contains trust scheme membership declarations. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the IssuerName query is a PTR Record and its DNSSEC validation is successful.

TA ID	TA_ TSPA_3
Normative Source	NS_TSPA_4, NS_TSPA_5, NS_TSPA_14
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. TSPA contains trust scheme membership declarations. The ATV has issued a SchemeNameLocation query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the SchemeNameLocation query is a URI Record and its DNSSEC validation is successful

TA ID	TA_TSPA_4
Normative Source	NS_TSPA_4, NS_TSPA_5, NS_TSPA_15
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. TSPA contains trust scheme membership declarations with signed trust list URIs. The ATV

	has issued an IssuerName_SchemeNameAssociation query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the IssuerName_SchemeNameAssociation query is a signed association and its signature validation is successful

TA ID	TA_TSPA_5
Normative Source	NS_TSPA_4, NS_TSPA_5
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued a SchemeNameTuples query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the SchemeNameTuples query is a set of tuples retrieved from the pointer of the respective trust list entry.

TA ID	TA_ TSPA_6
Normative Source	NS_TSPA_11
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form scheme._trust.IssuerDomainName IN PTR

TA ID	TA_TSPA_7
Normative Source	NS_TSPA_13, NS_TSPA_7
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued a SchemeNameLocation query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form _scheme._trust.SchemeNameDomainName IN URI

TA ID	TA_TSPA_8
Normative Source	NS_TSPA_16
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued a CertificateConstraints query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form _scheme._trust. SchemeNameDomainName IN SMIMEA

TA ID	TA_TSPA_9
Normative Source	NS_TSPA_12, NS_TSPA_10, NS_TSPA_8
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued an IssuerName query to the TSPA.
Prescription	Mandatory

Level	
Predicate	The RR response to the IssuerName query is a PTR Record containing the domain name of the SchemeName if the queried trust scheme is Boolean.

TA ID	TA_TSPA_10
Normative Source	NS_TSPA_12, NS_TSPA_10, NS_TSPA_8
Target	ATV – TSPA Interface
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the IssuerName query is a PTR Record containing levelName.domainName of the SchemeName if the queried trust scheme is Ordinal.

7.1.4 ATV - TSPA Integration Test Cases

This section includes the list of test cases and the descriptions of the test cases.

7.1.4.1 ATV – TSPA Integration Test Case List

Table 1 List of ATV – TSPA integration test cases

ID	Purpose
TC_TSPA_1	Check if TSPA-DNS server exists and is configured to be used by TSPA.
TC_TSPA_2	Verify that RR response to the issuername is a PTR record and DNSSec validation is successfull
TC_TSPA_3	Verify that RR response to the issuername is a URI record and DNSSec validation is successfull
TC_TSPA_4	Verify that trust list pointed on the URI RR record, signature validation is successful

TC_TSPA_5	Verify that an invalid trust list pointed on the URI RR record, signature validation fails
TC_TSPA_6	Verify that TSPA could provide trust scheme declarations successfully
TC_TSPA_7	Verify that TSPA could provide trust list URLs for the trust scheme successfully
TC_TSPA_8	Verify that TSPA-DNS provides certificate constraints to validate trust lists successfully
TC_TSPA_9	Verify that trust list validation fails in case of invalid certificate
TC_TSPA_10	Verify that TSPA can persist Boolean trust scheme declarations successfully
TC_TSPA_11	Verify that TSPA can persist ordinal trust scheme declarations successfully
TC_TSPA_12	Verify that received DNS works in synchronization with TSPA
TC_TSPA_13	Verify that received TSPA works in synchronization with DNS entries

7.1.4.2 ATV – TSPA Integration Test Case Details

ID	**TC_TSPA_1**	
Assertion(s)	TA_TSPA_1	
Test Purpose	Check if TSPA-DNS server exists and is configured to be used by TSPA.	
Pre-Test Conditions	TSPA should already be deployed DNS deployment should be available	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig lightest.nlnetlabs.nl*	; <<>> DiG 9.10.6 <<>> lightest.nlnetlabs.nl ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8688 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;lightest.nlnetlabs.nl. IN A

		;; ANSWER SECTION: lightest.nlnetlabs.nl. 3600 IN A 185.49.141.61
2	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/scheme with eidas.kamusm.gov.tr-example scheme name*	The service should return HTTP 200 OK
3	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR _scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_2**	
Assertion(s)	TA_TSPA_2	
Test Purpose	Verify that RR response to the issuername is a PTR record and DNSSec validation is successfull	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,

		ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_3**	
Assertion(s)	TA_TSPA_3	
Test Purpose	Verify that RR response to the issuername is a URI record and DNSSec validation is successfull	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TSL-XML.xml trust list* parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*

ID	TC_TSPA_4
Assertion(s)	TA_TSPA_4
Test Purpose	Verify that trust list pointed on the URI RR record, signature validation is successful
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA A valid trust list is defined in TSPA (TC_TSPA_4 should be executed)

Step	Test Activity	Expected Result
1	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with http://www. mindertestbed.org:8081/trust/TSL-XML.xml parameter	The trust list should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761

		;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TSPA_5**	
Assertion(s)	TA_TSPA_4	
Test Purpose	Verify that an invalid trust list pointed on the URI RR record, signature validation fails	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/Invalid-TSL-XML.xml trust list* parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/InvalidTSL-*

		XML.xml
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with http://www.mindertestbed.org:8081/trust/TSL-XML.xml parameter	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _scheme._trust.eidas.kamusm.gov.tr-example SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust.*eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV checkCertificatefromSMIMEA service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successfull

ID	**TC_TSPA_6**		
Assertion(s)	TA_TSPA_5, TA_TSPA_6		
Test Purpose	Verify that TSPA could provide trust scheme declarations successfully		
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA		
Step	**Test Activity**		**Expected Result**

1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_7**	
Assertion(s)	TA_TSPA_7	
Test Purpose	Verify that TSPA could provide trust list URLs for the trust scheme successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK with response data including eidas.kamusm.gov.tr-example trust scheme
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761

		;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP GET request: https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list *with http://www. mindertestbed.org:8081/trust/TSL-XML.xml parameter*	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*

ID	**TC_TSPA_8**	
Assertion(s)	TA_TSPA_8	
Test Purpose	Verify that TSPA-DNS provides certificate constraints to validate trust lists successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA	
Step	**Test Activity**	**Expected Result**

1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK with response data including eidas.kamusm.gov.tr-example scheme
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP PUT request: https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with *http://www. mindertestbed.org:8081/trust/TSL-XML.xml parameter*	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
5	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761

		;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in http://www.mindertestbed.org:8081/trust/TSL-XML.xml	Trust List validation result should be successful
7	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should be successful

ID	**TC_TSPA_9**	
Assertion(s)	TA_TSPA_8	
Test Purpose	Verify that trust list validation fails in case of invalid certificate	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with eidas.kamusm.gov.tr-example should be published with turkey.lightest.nlnetlabs.nl domain name in TSPA A valid trust list URI record is already defined on TSPA-DNS An SMIMEA record including an invalid certificate to be used in trust list validation	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

		;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in *http://www. mindertestbed.org:8081/trust/TSL-XML.xml*	Trust List validation result should be successful
4	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should NOT be successful

ID	**TC_TSPA_10**	
Assertion(s)	TA_TSPA_9	
Test Purpose	Verify that TSPA can persist Boolean trust scheme declarations successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**

1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ /turkey.lightest.nlnetlabs.nl/schemes with boolean.eidas.kamusm.gov.tr-example parameter*	The service should return HTTP 200 OK. (Total 2 trust scheme with *.eidas.kamusm.gov.tr-example and Boolean. .eidas.kamusm.gov.tr-example should be available)*
2	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example*
3	Minder sends to TSPA the following GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ scheme/turkey.lightest.nlnetlabs.nl/schemes*	The service should return HTTP 200 OK where it contains the 2 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_11**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that TSPA can persist ordinal trust scheme declarations successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ /turkey.lightest.nlnetlabs.nl/schemes with ordinallevelname.eidas.kamusm.gov.tr*	The service should return HTTP 200 OK. (Total 3 trust scheme with *.eidas.kamusm.gov.tr-example and Boolean. .eidas.kamusm.gov.tr-example and ordinallevelname.eidas.kamusm.gov.tr-example should be available)*

	-example parameter	
2	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. ordinallevelname.eidas.kamusm.gov.tr-example*
3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl/schemes*	The service should return HTTP 200 OK where it contains the 3 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example* *Ordinallevelname. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_12**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that received DNS works in synchronization with TSPA	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

		; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. ordinallevelname.eidas.kamusm.gov.tr-example*
2	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK where it contains the 3 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example* *Ordinallevelname. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_13**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that received TSPA works in synchronization with DNS entries	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *Turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. Ordinallevelname.eidas.kamusm.gov.tr-example*

2	On the terminal, login to DNS with ssh with ssh -i id_rsa tubitak@lightest.nlnetlabs.nl Goto /usr/home/zonemgr/etc folder Edit Open the zone file with the following command: vim lightest.nlnetlabs.nl Delete the corresponding record with turkey.lightest.nlnetlabs.nl and *Ordinallevelname.eidas.kamusm.gov.tr-example trust scheme* Close the ssh session	DNS should be updated.
3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK where it contains the 2 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example*

7.2 ATV – TTA Integration Testing

7.2.1 ATV – TTA Integration Testing Conformance Clauses

Conformance clauses are given in D8.8 Integration Testing Report (1).

7.2.2 ATV – TTA Integration Normative Sources

For brevity, we removed ATV – TTA Integration Normative Sources from this deliverable. Since there are no modifications to the normative statements, readers can refer to D8.8 Integration Testing Report (1) for further details.

7.2.3 ATV – TTA Integration Test Assertions

The updated assertions are given below.

TA ID	TA_TTA_1
Normative Source	NS_TTA_1
Target	ATV – TTA Interface
Prerequisite	The name and details (characteristics) of the trust scheme are defined in the TSPA and received from TSPA
Prescription Level	Mandatory
Predicate	ATV issues a DNS call for the trust scheme, with DNS record set as “_translate” for the aspect and “_trust” for the application with the following format and TTA

returns the URI resource record for the trust scheme:

;; QUESTION SECTION: Client/ATV to the TTA
;_translate._trust.etimestamp.eidas.eu. IN URI

In case of Tuple&Ordinal Schemes, the assurance level (obtained from TSPA for the trust scheme) is included with the following format

;; QUESTION SECTION: Client/ATV to the TTA
;_translate._trust.**qualified**.eseal.eidas.eu. IN URI

TA ID	TA_TTA_2
Normative Source	NS_TTA_1, NS_TTA_2, TA_TTA_1
Target	ATV – TTA Interface
Prerequisite	The name and details (characteristics) of the trust scheme are defined in the TSPA and received from TSPA
Prescription Level	Mandatory
Predicate	For Boolean trust scheme, TTA returns the resource record with the following format: ;; QUESTION SECTION: Client/ATV to the TTA ;_translate._trust.etimestamp.eidas.eu. IN URI ;; ANSWER SECTION: from the TTA _translate._trust.etimestamp.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedTimestampEidas1.tpl l _translate._trust.etimestamp.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedTimestampEidas1.xml

TA ID	TA_TTA_3
Normative Source	NS_TTA_1, NS_TTA_2, TA_TTA_1

Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations. The names of the assurance levels just published by the TSPA have to be already retrieved from the TSPA by ATV
Prescription Level	Mandatory
Predicate	For Ordinal&Tuple Trust Scheme, TTA returns the resource record with the following format:: ;; QUESTION SECTION: Client/ATV to the TTA ;_translate._trust.qualified.eseal.eidas.eu. IN URI ;; ANSWER SECTION: from the TTA _translate._trust.qualified.eseal.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedSealEidas1.tpl _translate._trust.qualified.eseal.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedSealEidasN.tpl _translate._trust.qualified.eseal.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedSealEidas1.xml _translate._trust.qualified.eseal.eidas.eu. IN URI https://lightest.eu/ttl_qualifiedSealEidasN.xml

TA ID	TA_TTA_4
Normative Source	NS_TTA_2,NS_TTA_3, NS_TTA_5
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations.
Prescription Level	Mandatory

Predicate	TTA provides more than one translation schemes for the trust scheme. TTA provides/publishes a pointer to the trust translation list for each recognized trust level with Boolean, Ordinal or Tuple trust scheme types in the form of a series URI resource records.

TA ID	TA_TTA_5
Normative Source	NS_TTA_2, NS_TTA_6, NS_TTA_7, NS_TTA_8
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations. Trust translation lists are already defined in the XML format for the trust schemes The names of the assurance levels just published by the TSPA have to be already retrieved from the TSPA by ATV, in order to build the right domain name for asking for the translation.
Prescription Level	Preferred
Predicate	Trust Translation Provider provides a file for each recognized trust level with XML and TPL . In case of XML, TTA returns the list of the trust levels equivalents to the one requested with level name and trust scheme name. In case of TPL, TTA returns the list of the trust levels equivalents to the one requested with level name, trust scheme name and TPL description

TA ID	TA_TTA_6
Normative Source	N NS_TTA_4, NS_TTA_5, NS_TTA_9
Target	TTA
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations.
Prescription Level	Mandatory

Predicate	For Boolean trust scheme, TTA-DNS checks whether the certificate used for signing the translation files is valid according to the content of DNS-SMIMEA resource record. ;; QUESTION SECTION: Verifying authenticity ;_translate._trust.etimestamp.eidas.eu. IN SMIMEA ;; ANSWER SECTION: _translate._trust.etimestamp.eidas.eu. IN SMIMEA <SMIMEA record data>

TA ID	TA_TTA_7
Normative Source	NS_TTA_4, NS_TTA_5, NS_TTA_9
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations.
Prescription Level	Mandatory
Predicate	For Ordinal&Tuple trust scheme, TTA-DNS checks whether the certificate used for signing the translation files is valid according to the content of DNS-SMIMEA resource record including the trust scheme and level of assurance

```
;; QUESTION SECTION: Verifying authenticity
;_translate._trust.qualified.eseal.eidas.eu.   IN  SMIMEA

;; ANSWER SECTION:
_translate._trust.qualified.eseal.eidas.eu.   IN  SMIMEA  <SMIMEA
record data>
```

TA ID	TA_TTA_8
Normative Source	NS_TTA_10
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and TTA contains published trust translation list declarations.

	The TTA should return the signed trust translation lists
Prescription Level	Mandatory
Predicate	TTA-DNS should provide certificate constraints to use for the verification of the translation list signature.

7.2.4 ATV – TTA Integration Test Cases

This section includes the list of test cases and the descriptions of the test cases.

TTA Test cases assume that scheme information is obtained from TSPA and electronic transaction is parsed on Minder-ATV and is sent to TSPA to conform the trust scheme membership.

7.2.4.1 ATV – TTA Integration Test Cases List

Table 2 List of ATV – TTA integration test cases

ID	Purpose
TC_TTA_1	Verify that TTA published more than one translation schemes for a boolean trust scheme
TC_TTA_2	Verify that TTA published more than one translation schemes for an ordinal trust scheme
TC_TTA_3	Verify that TTA published more than one translation schemes for a tuple trust scheme
TC_TTA_4	Verify that RR response to the issuername is a URI record for a boolean scheme and DNSSec validation is successful
TC_TTA_5	Verify that RR response to the issuername is a URI record for ordinal schemes and DNSSec validation is successful
TC_TTA_6	Verify that TTA published more than one translation schemes for a tuple trust scheme
TC_TTA_7	Check that the verification result of translation list signature is successful for a boolean trust scheme
TC_TTA_8	Check that the verification result of translation list signature is successful for

	an ordinal trust scheme
TC_TTA_9	Check that the verification result of translation list signature is successful for a tuple trust scheme
TC_TTA_10	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for ordinal trust scheme
TC_TTA_11	Verify that an invalid trust list, pointed on the URI RR record, signature validation fails for tuple trust scheme
TC_TTA_12	Verify that the certificate provided by DNS is not valid and translation list verification fails due to certificate validation

7.2.4.2 ATV – TTA Integration Test Case Details

ID	**TC_TTA_1**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for a boolean trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation* *with "test-agreement"* Boolean trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with "test-agreement"*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA "test-agreement"
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR,

		id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	**TC_TTA_2**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for an ordinal trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation with "test-agreement-ordinal"* Ordinal trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with "test-agreement-ordinal"*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA "test-agreement-ordinal"
3	On the terminal, type the following command: *dig _translate._trust.turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust.turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer:

		;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl …* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*

ID	**TC_TTA_3**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for a tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation with "test-agreement-tuple"* tuple trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA "test-agreement-tuple"

	lightest.eu:8080/ttaFM/mng/rsc/getTranslation *with "test-agreement-ordinal"*	
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*

ID	**TC_TTA_4**
Assertion(s)	TA_TTA_2
Test Purpose	Verify that RR response to the issuername is a URI record for a boolean scheme and DNSSec validation is successful
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for "test-agreement"

Step	Test Activity	Expected Result
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with with "test-agreement"* boolean trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK and JSON return value is the agreement details given in Test Scenario for TTA. The result should include the following translation lists: *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	TC_TTA_5
Assertion(s)	TA_TTA_1, TA_TTA_5
Test Purpose	Verify that RR response to the issuername is a URI record for ordinal schemes and DNSSec validation is successful
Pre-Test Conditions	TTA should already be deployed and running

	DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for “test-agreement-ordinal“

Step	Test Activity	Expected Result
1	Minder sends to TTA the following *HTTP GET request: http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with “test-agreement-ordinal”* Ordinal trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK. The result should include the following translation lists: *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with “test-agreement-ordinal”*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA “test-agreement-ordinal”
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl … _translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl _translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml _translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEid*

		asN.xml

ID	**TC_TTA_6**
Assertion(s)	TA_TTA_1
Test Purpose	Verify that TTA published more than one translation schemes for a tuple trust scheme
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for “test-agreement-tuple“

Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “test-agreement-tuple”* tuple trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK. The following translation files should be included: *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “test-agreement-tuple”*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA “test-agreement-tuple”
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI

		http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*

ID	**TC_TTA_7**	
Assertion(s)	TA_TTA_8	
Test Purpose	Check that the verification result of translation list signature is successful for a boolean trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running A valid translation with test-agreement scheme name is defined on TTA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimesta*

		mpEidasN.tpl *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* for the following translation lists http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml	The trust lists should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust. turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TTA_8**
Assertion(s)	TA_TTA_8, TA_TTA_8
Test Purpose	Check that the verification result of translation list signature is successful for an ordinal trust scheme
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running

	A valid translation with test-agreement-ordinal scheme name is defined on TTA

Step	Test Activity	Expected Result
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl …* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* for the following translation lists: http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml	The trust lists should be downloaded and should be opened via an XML editor

3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust. turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TTA_9**	
Assertion(s)	TA_TTA_7, TA_TTA_8	
Test Purpose	Check that the verification result of translation list signature is successful for a tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running A valid translation with test-agreement-tuple scheme name is defined on TTA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI

		;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* for the following translation lists : http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl	The trust lists should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust. turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TTA_10**
Assertion(s)	TA_TTA_06
Test Purpose	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for ordinal trust scheme
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running There exists an translation agreement “invalid-agreement-ordinal“ that

	includes an invalid trust list for a ordinal trust scheme

Step	Test Activity	Expected Result
1	Minder sends to TTA the following *HTTP GET request: http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with "invalid-agreement-ordinal"* parameter	The service should return HTTP 200 OK. The following translations should be listed: http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml*
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with the following files: http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> *DiG 9.10.6 _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA* *;; global options: +cmd* ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

		;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate._trust turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successful,

ID	**TC_TTA_11**	
Assertion(s)	TA_TTA_07	
Test Purpose	Verify that an invalid trust list, pointed on the URI RR record, signature validation fails for tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running There exists an translation agreement “invalid-agreement-tuple“ that includes an invalid trust list for a tuple trust scheme	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following *HTTP GET request: http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with “invalid-agreement-tuple” parameter*	The service should return HTTP 200 OK. The following translations should be listed: *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl*

		http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with the following files: http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> *DiG 9.10.6 _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA* *;; global options: +cmd* ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate._trust turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successful,

ID	**TC_TTA_12**
Assertion(s)	TA_TTA_07
Test Purpose	Verify that the certificate provided by DNS is not valid and translation list verification fails due to certificate validation
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running

	A valid trust list URI record is already defined on TSPA-DNS An SMIMEA record including an invalid certificate to be used in translation list validation exists on the DNS

Step	Test Activity	Expected Result
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*

3	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml	Trust List validation result should be successfull
4	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should NOT be successfull

7.3 ATV – DP Integration Testing

7.3.1 ATV – DP Integration Testing Conformance Clause

Conformance clause for ATV-DP is given in D8.8 Integration Testing Report (1)

7.3.2 ATV – DP Integration Normative Statements

For brevity, we removed ATV – DP Integration Normative Sources from this deliverable. Since there has not been any modifications to the normative statements, readers can refer to D8.8 Integration Testing Report (1) for further details.

7.3.3 ATV – DP Integration Test Assertions

For brevity, we removed ATV – DP Integration Test Assertions from this deliverable. Since there has not been any modifications to the assertions, readers can refer to D8.8 Integration Testing Report (1) for further details.

7.3.4 ATV – DP Integration Test Cases

This section includes the list of test cases and the descriptions of the test cases.

7.3.4.1 ATV – DP Integration Test Case List

Table 3 List of ATV – DP integration test cases

ID	Purpose
TC_DP_1	Verify that DP publishes the delegation successfully

TC_DP_2	Verify that DP does not publish an invalid delegation
TC_DP_3	Verify that DP does not publish a valid delegation with publicKeyHash not belonging to Proxy
TC_DP_4	Check that DP verifies the delegation successfully
TC_DP_5	Check the content of delegation from the response when delegation is verified.
TC_DP_6	Check the response of the revoked delegation
TC_DP_7	Check the response of the valid delegation queried on DP
TC_DP_8	Check the response that DP should return error if verifier sends more than one revocation query at the time
TC_DP_9	Check if a revoke command interface on DP is available
TC_DP_10	Check revocation response when verifier sends a revocation query. Check the signed revocation response with the certificate that is issued by Mandator for the revocation purpose. Check if delegation id id hash of delegation. Check if the response includes the delegation that is given to DP, the certificates that is used to sign and all certificates to build the trust chain.
TC_DP_11	Check if the delegation id is the hash of delegation when querying on DP.
TC_DP_12	Check that DP publishes delegation key successfully
TC_DP_13	Verify that DP respond with an error/notification message if delegation is not found

7.3.4.2 ATV – DP Integration Test Case Details

ID	**TC_DP_1**
Assertion(s)	TA_DP_1
Test Purpose	Verify that DP publishes the delegation successfully

Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed

Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including delegation type, Mandator's private key, Proxy public key, validity time, domain settings and address of DP.
2	Execute Minder-ATV *downloadservice* with the following parameter: delegation URL	https://mindertestbed.org/delegation/tubitak_delegation.xml should be downloded and opened in an XML editor

ID	**TC_DP_2**
Assertion(s)	TA_DP_1
Test Purpose	Verify that DP does not publish an invalid delegation
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed

Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (invaliddelegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 500.

ID	TC_DP_3	
Assertion(s)	TA_DP_1, TA_DP_3	
Test Purpose	Verify that DP does not publish a valid delegation with publicKeyHash not belonging to Proxy	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed It is assumed that DP does not publish delegation in this case.	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters* Note that publicKeyHash does not belong to Proxy	The service should return HTTP 500

ID	TC_DP_4
Assertion(s)	TA_DP_1, TA_DP_3
Test Purpose	Check that DP verifies the delegation successfully
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP **dpUrlAddress**: to be defined when DP deployment is completed

Step	Test Activity	Expected Result
1	Mandator creates a delegation for TUBITAK Tester.	Delegation.xml that conforms to ETSI 119 621 is created
2	Execute Minder-ATV *verifydelegationservice* with the following parameter: delegation.xml	The expected result is True
2	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including delegation type, Mandator's private key, Proxy public key, validity time, domain settings and address of DP.

ID	TC_DP_5
Assertion(s)	TA_DP_1, TA_DP_3
Test Purpose	Check the content of delegation from the response when delegation is verified.
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed

Step	Test Activity	Expected Result
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including validity time, Sequence Number, IssuedDate, Proxy, Mandator, validity, notAfter, notBefore, flags, server fields.
2	Execute Minder-ATV *downloadservice* with the following parameter: delegation URL	https://mindertestbed.org/delegation/tubitak_delegation.xml should be downloaded and opened in an XML editor

ID	TC_DP_6	
Assertion(s)	TA_DP_2, TA_DP_1, TA_DP_6	
Test Purpose	Check the response of the revoked delegation	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed A Revoked delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends http request below to DP searchServer to query status of delegation https://dpUrlAddress/search?delegation{id}?&token={token}	DP service sends a response that the delegation is revoked. To be defined when implemented.
2	Minder-ATV executes *verifyRevocationResponse service to validate the revocation result*	The verification result should be successful.

ID	TC_DP_7
Assertion(s)	TA_DP_1, TA_DP_3
Test Purpose	Check the response of the valid delegation queried on DP
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP

	Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends HTTP request below to DP searchServer to query status of delegation https://dpUrlAddress/search?delegation{id}?&token={token}	DP service sends a response that the delegation is valid. To be defined when implemented.

ID	**TC_DP_8**	
Assertion(s)	TA_DP_4	
Test Purpose	Check the response that DP should return error if verifier sends more than one revocation query at the time	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends two revocation query at the same time. *https://dpUrlAddress/search?delegation{idbfgbfgbgf}?&token={token}* *https://dpUrlAddress/search?delegation{iddddd}?&={token}*	Service returns error message. To be defined when implemented.

ID	TC_DP_9
Assertion(s)	TA_DP_1, TA_DP_5
Test Purpose	Check if a revoke command interface on DP is available
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester **dpUrlAddress**: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP

Step	Test Activity	Expected Result
1	Mandator sends HTTP request below to DP searchServer to query status of delegation https://dpUrlAddress/search?delegation{id}?&token={id}	DP service sends a response that the delegation is valid. To be defined when implemented.
2	Mandator sends delegation Id (hash of delegation), certificate signed by mandator and revocation delegation to DP	DP revokes the delegation and stores the revocation time with the revocation. To be defined when implemented.
3	Minder-ATV executes *verifyRevocationResponse service to validate the revocation response*	Verification should be successful

ID	TC_DP_10
Assertion(s)	TA_DP_1, TA_DP_6, TA_DP_7
Test Purpose	Check revocation response when verifier sends a revocation query. Check the signed revocation response with the certificate that is issued by Mandator for the revocation purpose.

	Check if delegation id id hash of delegation. Check if the response includes the delegation that is given to DP, the certificates that is used to sign and all certificates to build the trust chain.	
Pre-Test Conditions	1. Delegation Provider is accessible. 2. Revoked delegation file is needed.	
Step	**Test Activity**	**Expected Result**
1	Verifier sends http request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={id}*	DP searches revocation archive and prepares the response. To be defined when implemented.
2	DP sends response which is signed.	Verifier checks the response. To be defined when implemented.

ID	**TC_DP_11**	
Assertion(s)	TA_DP_6, TA_DP_7	
Test Purpose	Check if the delegation id is the hash of delegation when querying on DP.	
Pre-Test Conditions	Delegation Provider is accessible. Delegation does not exist on DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends HTTP request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={id}*	DP searches revocation archive and sends error message. To be defined when implemented.

ID	TC_DP_12	
Assertion(s)	TA_DP_2, TA_DP_1	
Test Purpose	Check that DP publishes delegation key successfully	
Pre-Test Conditions	Delegation Provider is accessible. Encrypted delegation key is needed.	
Step	**Test Activity**	**Expected Result**
1	Client sends the encyrpted delegation key to publication server. HTTP POST request to the following address: *https://dpUrlAddress/1/publish_key*	The service should return HTTP 201

ID	TC_DP_13	
Assertion(s)	TA_DP_1, TA_DP_8	
Test Purpose	Verify that DP respond with an error/notification message if delegation is not found	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP	
Step	**Test Activity**	**Expected Result**
1	Mandator sends HTTP request below to DP searchServer to query status of delegation https://dpUrlAddress/search?delegation{id}?&token={id} id is the id that does not exist on DP	DP service sends a response that the delegation is not found for the intended id. To be defined when implemented.

8. References

LIGHTest. (2017, 04 28). *D2.3 – Requirements and Use Cases.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20539492

LIGHTest. (2017, 02 28). *D2.14 – Reference Architecture.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20534069

LIGHTest. (2017, 04 28). *D3.3 – DNS-based Publication of Trust Schemes.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20539083

LIGHTest. (2017, 04 28). *D3.4 – Discovery of Trust Scheme Publication Authorities.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20531317

LIGHTest. (2017, 04 28). *D4.3 – DNS-based Publication of Trust Translation Schemes.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20531524

LIGHTest. (2017, 04 28). *D4.4 – Discovery of Trust Translation Authorities.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20538593

LIGHTest. (2017, 04 28). *D5.2 – Conceptual Framework for Delegations (2).* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20816230

LIGHTest. (2017, 04 28). *D8.8 – Integration Testing Report (1).* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=21147387

9. Project Description

LIGHTest project to build a global trust infrastructure that enables electronic transactions in a wide variety of applications

An ever increasing number of transactions are conducted virtually over the Internet. How can you be sure that the person making the transaction is who they say they are? The EU-funded project LIGHTest addresses this issue by creating a global trust infrastructure. It will provide a solution that allows one to distinguish legitimate identities from frauds. This is key in being able to bring an efficiency of electronic transactions to a wide application field ranging from simple verification of electronic signatures, over eProcurement, eJustice, eHealth, and law enforcement, up to the verification of trust in sensors and devices in the Internet of Things.

Traditionally, we often knew our business partners personally, which meant that impersonation and fraud were uncommon. Whether regarding the single European market place or on a Global scale, there is an increasing amount of electronic transactions that are becoming a part of peoples everyday lives, where decisions on establishing who is on the other end of the transaction is important. Clearly, it is necessary to have assistance from authorities to certify trustworthy electronic identities. This has already been done. For example, the EC and Member States have legally binding electronic signatures. But how can we query such authorities in a secure manner? With the current lack of a worldwide standard for publishing and querying trust information, this would be a prohibitively complex leading to verifiers having to deal with a high number of formats and protocols.

The EU-funded LIGHTest project attempts to solve this problem by building a global trust infrastructure where arbitrary authorities can publish their trust information. Setting up a global infrastructure is an ambitious objective; however, given the already existing infrastructure, organization, governance and security standards of the Internet Domain Name System, it is with confidence that this is possible. The EC and Member States can use this to publish lists of qualified trust services, as business registrars and authorities can in health, law enforcement and justice. In the private sector, this can be used to establish trust in inter-banking, international trade, shipping, business reputation and credit rating. Companies, administrations, and citizens can then use LIGHTest open source software to easily query this trust information to verify trust in simple signed documents or multi-faceted complex transactions.

The three-year LIGHTest project starts on September 1st and has an estimated cost of almost 9 Million Euros. It is partially funded by the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321. The LIGHTest consortium consists of 14 partners from 9 European countries and is coordinated by Fraunhofer-Gesellschaft. To reach out beyond Europe, LIGHTest attempts to build up a global community based on international standards and open source software.

The partners are ATOS (ES), Time Lex (BE), Technische Universität Graz (AT), EEMA (BE), G&D (DE), Danmarks tekniske Universitet (DK), TUBITAK (TR), Universität Stuttgart (DE), Open Identity Exchange (GB), NLNet Labs (NL), CORREOS (ES), University of Piraeus Research Center (GR) and Ubisecure (FI). The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.
The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.