D8.8

Integration Testing Report (1)

Document Identification	
Date	29.08.2018
Status	Final
Version	1.0

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

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

To conceive truly “global trust lists”, a key objective of LIGHT*est* is global operation and global acceptance. By using the existing and well established Domain Name System as the basis of the LIGHT*est* trust infrastructure, global operation is possible from the start without requiring any additional roll out of enabling infrastructure. By covering most needs through the well-accepted DNS and involving non-European stakeholders from the very start, LIGHT*est* promises to overcome the “not invented here” syndrome and reach real global acceptance and uptake easily. By using the mature, already well-tested DNS with its existing software components, combined with massive integration testing, LIGHT*est* can quickly reach Technology Readiness Level.

Integration testing is the effort to verify that the integrated system works in a harmonized way including all its components. During the development process, the modules that work fine individually do not tend to work together correctly for the first time. Integration testing verifies whether the whole system components perform collaboratively. In this task integration testing will be performed.

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 D8.8 – Integration Testing Report (1) is the first testing report on integration testing of LIGHT*est* at the system level.

2. Document Information

2.1 Contributors

Name	Partner
Berkay TOPÇU	TUBITAK
Burçin BOZKURT GÜNAY	TUBITAK
Edona FASLLIJA	TUBITAK
Elif ÜSTÜNDAĞ SOYKAN	TUBITAK
Emine BİRCAN	TUBITAK

2.2 History

Version	Date	Author	Changes
0.0	06/08/2018	Emine BİRCAN, TUBITAK	First Draft
0.1	10/08/2018	TUBITAK	Initial Version
1.0	29/08/2018	TUBITAK	Final Version

3. Table of Contents

3.1 Table of Figures

3.2 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 Provider 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. Integration Test Overview

4.1 System Components Overview

The LIGHTest reference architecture has four main components to be tested: Trust Schema Publication Authority (TSPA), Trust Translation Authority (TTA), Delegation Publisher (DP) and Automatic Trust Verifier (ATV) as seen in Figure 1.

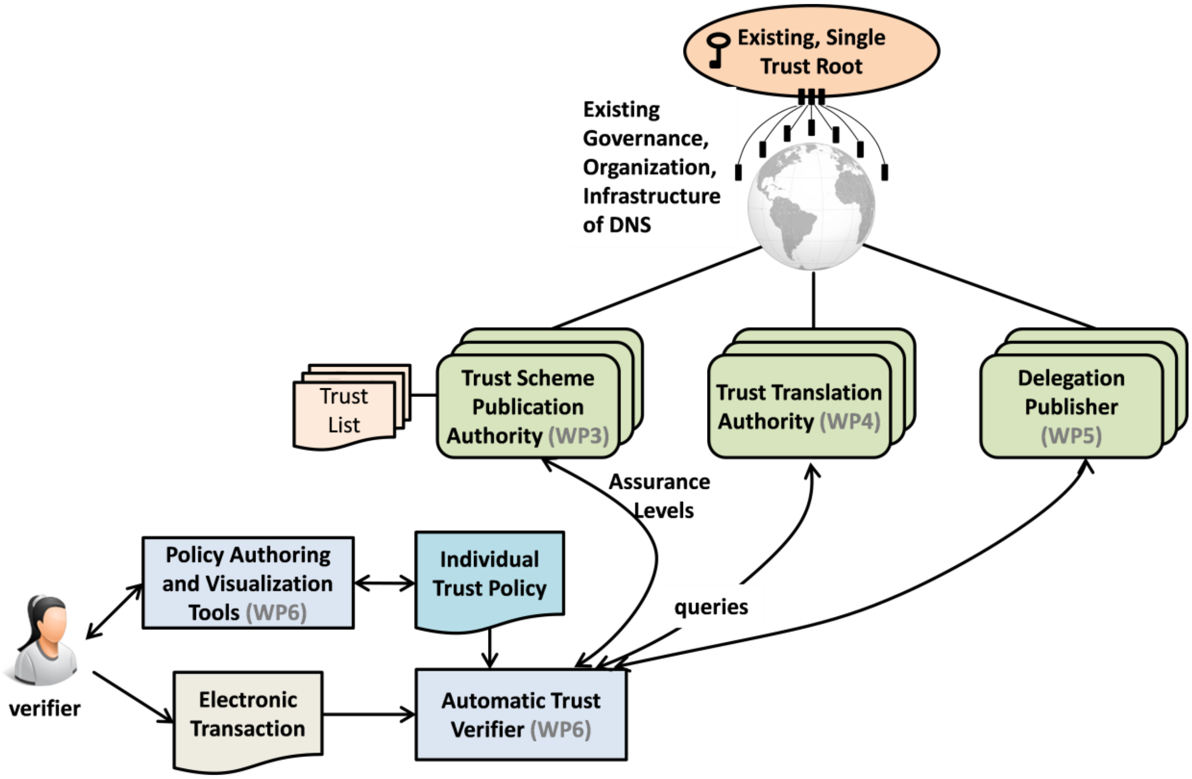
Existing, Single Trust Root
Existing Governance, Organization, Infrastructure of DNS
Trust List
Trust Scheme Publication Authority (WP3)
Trust Translation Authority (WP4)
Delegation Publisher (WP5)
Assurance Levels
queries
Policy Authoring and Visualization Tools (WP6)
Individual Trust Policy
verifier
Electronic Transaction
Automatic Trust Verifier (WP6)

Figure 1 LIGHT^{est} Reference Architecture

ATV is the component that verifies whether the electronic transaction satisfies the verifier's trust policy. TSPA manages the publishing of the trust schemes. TTA provides the necessary translation data to map the levels of assurance of the foreign trust scheme to its equivalent in the domestic trust scheme in a cross-jurisdiction setting. DP permits verifiers to query delegations and mandates in case data records that compose an electronic transaction are not directly signed by the legal entity responsible for it, but by a natural person that acts as an authorized representative for the former based on a delegation.

4.2 Testing Methodology

In this section we provide an overview of the Minder Test Assertion Model and a short guideline on how to write test assertions.

4.2.1 Minder Test Assertion Model

Minder Test Assertion Model is based on the OASIS Test Assertion Model (http://docs.oasis-open.org/tag/guidelines/v1.0/cn02/guidelines-v1.0-cn02_files/image004.jpg).

A **conformance clause (CC)** is a high-level statement that references to normative sources: *Example: "An implementation conforms to the Component Requirements if it meets the following conditions: (Either other clauses or normative statements listed here).* For a component level integration testing a conformance clause can be *"An interface between Component_A and Component_B is integrated seamlessly if it satisfies the conditions provided in the normative statements NS_1 to NS_n."*

A **normative statement (NS)** is a clear sentence that contains a subject, a may or must clause, and provides concise instructions (the word "SOME" is not used). *Example normative statement:*

A TSPA implementation MUST sign the URI Resource Records that are published in its DNS Name Server.

A **target** is the implementation of the component that is referred. It is either the artifact or the tool to be tested. The target of a TA is the subject of the test. When the test assertion succeeds, we claim that the target has passed the test.

Example targets: TSPA, TTA, DP

A **prerequisite (optional)** is a logical statement that must be satisfied before the assertion. If the prerequisite evaluates to false, then the context is not valid for the current test assertion and the test is not done.

Example prerequisite for the normative statement given before:

The received DNS Response is contains the _trust. prefix.

A **predicate** is the logical statement that includes the assertion of one or more normative statements. It may be similar to an if statement (in terms of a programming language). It evaluates whether an action/statement/etc. true or false.

Example predicate:

- *The DNSSEC Validation of the Resource Record retrieved is successful.*

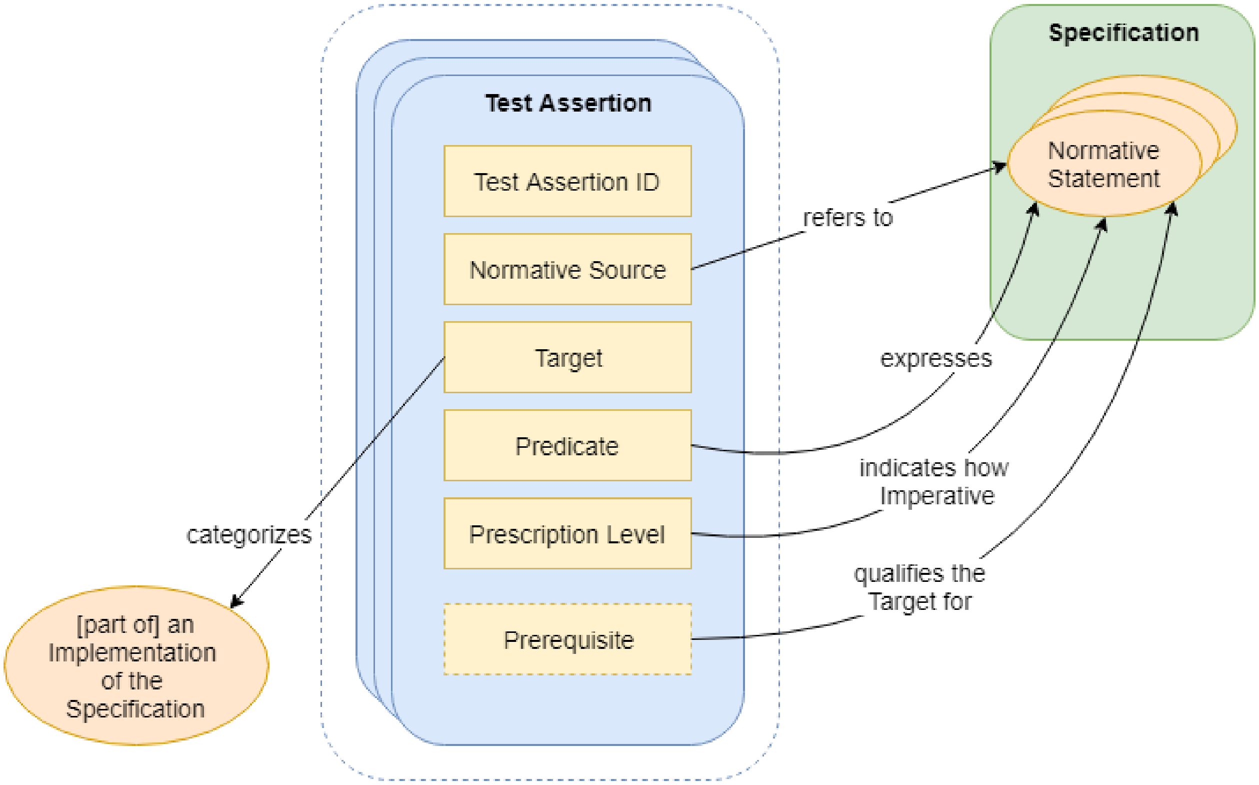

Figure 2 General Anatomy of a Test Assertion

4.2.2 Test Assertions Guideline

TA’s are generated in the following order:

(1) **Conformance clauses** are generated for a specification (e.g. a LIGHT*est* component design specification).

(2) **Normative statements** are extracted from the conformance clauses.

(3) **Targets** are identified.

(4) **Prerequisites** are generated for the statements

(5) **Predicates** are generated for the normative statements.

(6) The predicates-prerequisites-normative statements are formalized using the OASIS format. (Please note that the OASIS standard does not directly claim the step (2) but that is our interpretation for ease of flow, and is valid in OASIS context).

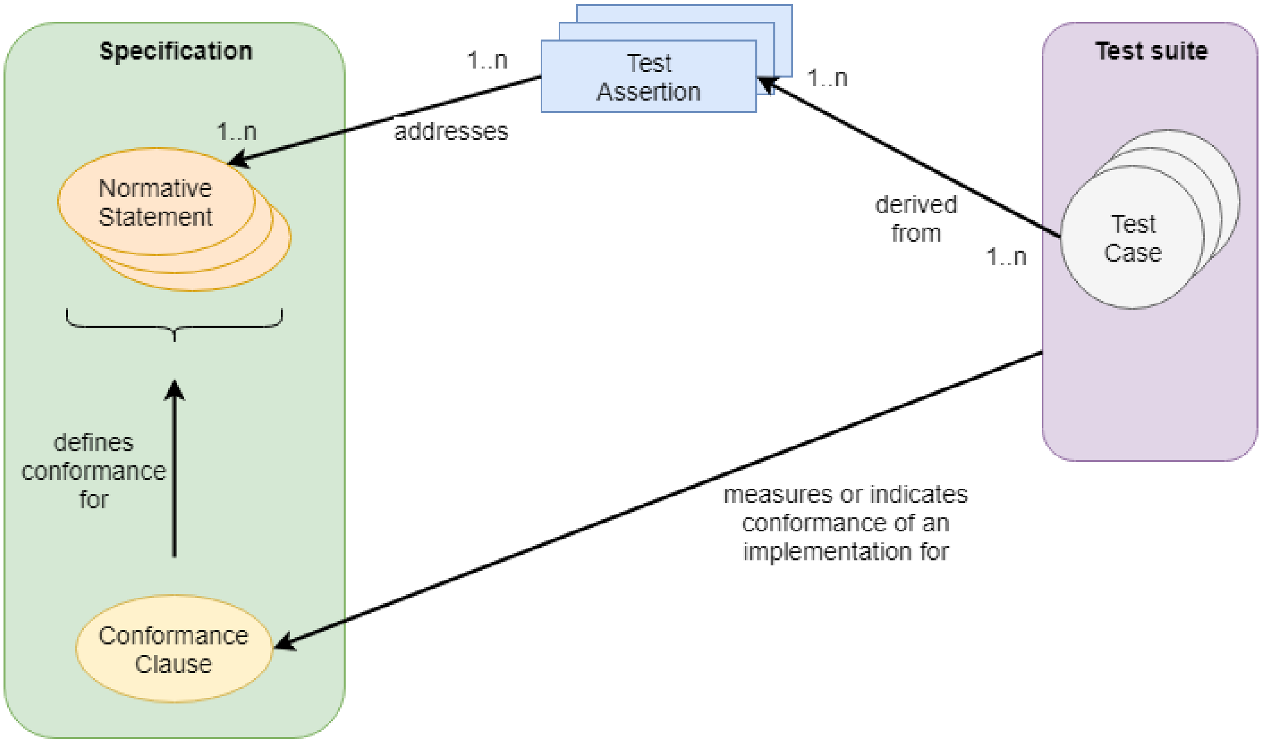

Figure 3 The relation between Test Assertions and Specifications[1].

4.3 Testing Architecture Overview

A typical Integration Testing focuses on checking data communication amongst the software components. There are different levels of Integration Testing which depends on the context of the project. According to International Software Testing Qualifications Board (ISTQB), integration testing involves:

- Component integration testing that establishes the interaction between integrated modules in one system.
- System integration testing featuring the controls on the integration of several interacting systems.

The LIGHT*est* reference architecture (RA) has been provided in Figure 1. As the LIGHT*est* system is comprised of ATV, TSPA,TTA and DP components, the integration testing methodology will be, naturally, the component integration testing.

The integration testing plan includes the inspection of the interfaces between the components to ensure that they work together seamlessly after a full deployment. All the communication between the deployed components will be intercepted, inspected and modified if necessary, then forwarded to its actual destination by Minder. The scripting support on Minder will help a full automation of the tests, and the interception of the messages flowing between the components will be accomplished by the so called *Minder Adapters*.

A *Minder Adapter* provides two aspects (views) in terms of interfaces. The first side interacts with the interfaces of system component being tested (i.e. SUT). This side ensures that the SUT does not know any detail about testing and 'believes' that it is sending its message to the actual target. In that sense, a Minder Adapter acts as a 'target component'. The second aspect of the adapter interacts with Minder and converts the messages flowing within the system to the Minder signals so that Minder can understand and interpret them. For the Minder Testbed applied architecture overview please see Figure 4.

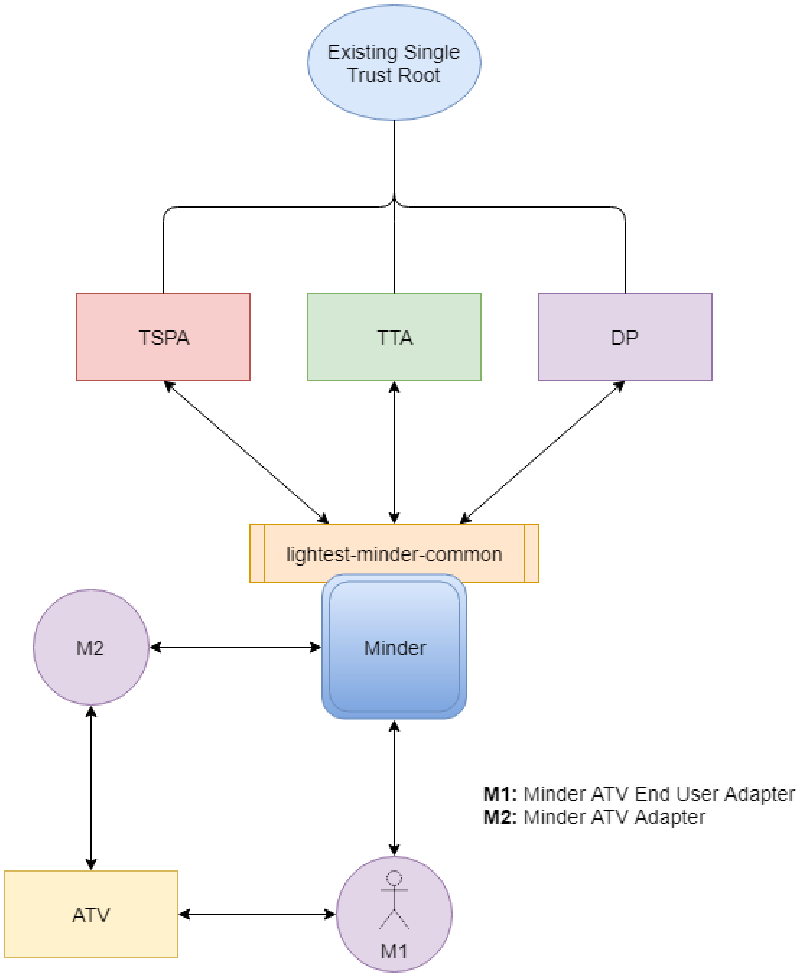

Figure 4 Minder Testbed Applied Architecture

The Minder adapters are depicted as circular components in the architecture diagram. There is a noticeable interception of the links between the actual LIGHT^{est} components. The messages flowing between Minder and the ATV component of LIGHT^{est} are intercepted by the M1 and M2 adapters. Minder acts as a man-in-the-middle between the components. The communication between Minder and the TSPA, and TTA, can be 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.

A sample Minder adapter is depicted in Figure 5. The ATVMinderAdapter, receives the DNS query that was issued by the ATV and coverts it into datagramRequestReceived signal call to Minder.

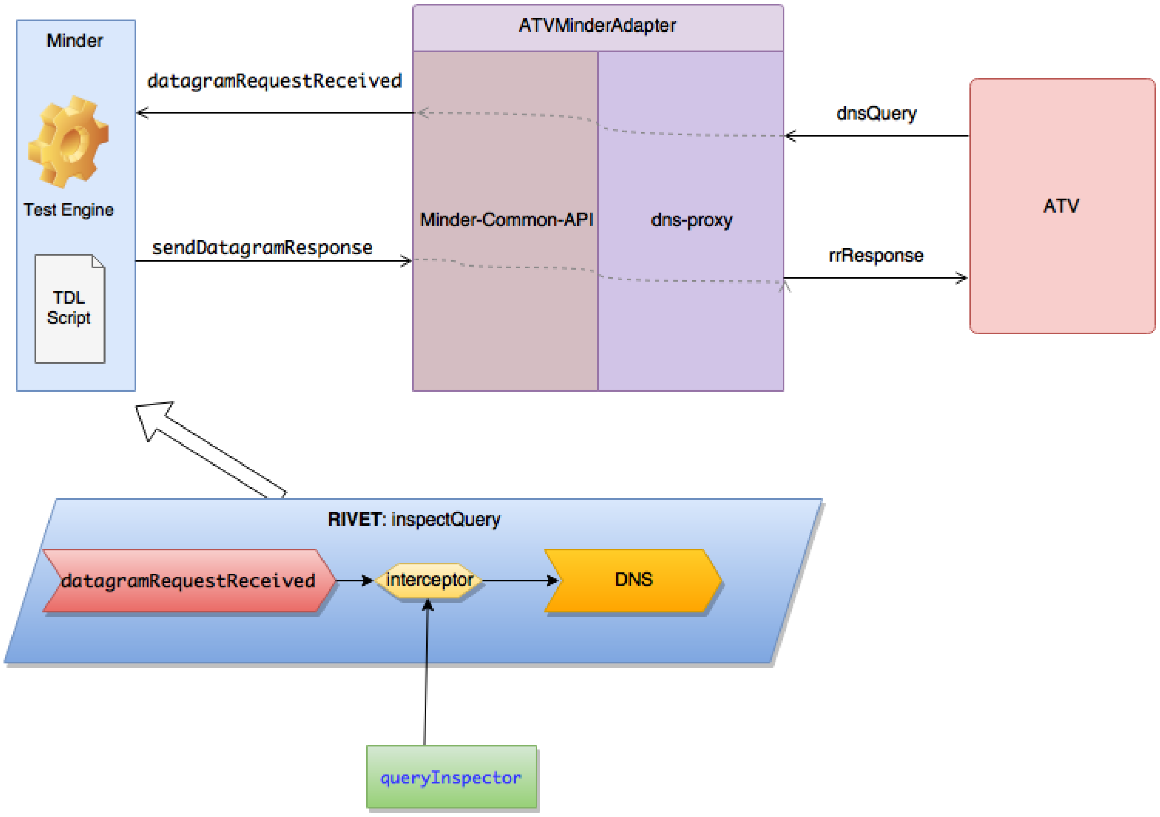

Figure 5 Sample Figure for Minder-TSPA Communication

At the arrival of the signal, TDL scripts for the corresponding Test Cases consisting of multiple rivets are executed. Each rivet can be seen as a single step of a test case that is run on the Minder Test Engine. The rivet enables the inspection of the query before being submitted to the LIGHT^{est} DNS components. This is achieved through the *queryInspector* function, which takes a query having the DNS-like parameters, and returns it after the necessary inspections. The inspection might also involve the modification of the query if it is required in the test assertion corresponding to the test case. Minder uses Minder-common API and a DNS client API to convert the actual DNS queries and responses to signals/slots and vice versa.

5. Integration Testing Scenarios:

Integration Testing scenarios aim to investigate whether two or more components successfully communicate based on interfaces defined between them. Interfaces are extracted from the general scenarios detailed in D2.14, D3.2, D3.4, D4.2, D4.4, D5.2 and D5.4. The basic scenarios that are used to generate integration testing scenarios are as follows:

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

5.1 ATV-TSPA Integration Testing Scenarios

Prerequisites:

1. *Trust policy* and *electronic transaction* exist as test assets.

Main Flow

1. Minder triggers the Querying of Trust Scheme Membership Test Scenario.
   1.1. Minder calls the startTest slot of the M1 Adapter that provides the electronic transaction and the trust policy test assets to ATV and triggers the initiation of the integration test scenario.
2. The ATV issues a IssuerName_DNS_query for the Issuer Name extracted from the electronic transaction.
   2.1. The IssuerName_DNS Query issued from ATV is intercepted by M2.
   2.2. M2 converts the IssuerName_DNS_query into IssuerName_DNS_query Minder Signal.
   2.3. Minder inspects the DNS query and issues the DNS query to TSPA via its DNS client API.
   2.4. The TSPA returns a RR response containing the Scheme Name associated with the Issuer Name.
   2.5. Minder inspects the SchemeName RR response and calls the RR response slot of M2, which in turn forwards the response to the ATV.
3. The ATV issues a SchemeName_DNS_query for the Issuer Name extracted from the RR response of step 2.5.
   3.1. M2 intercepts the SchemeName_DNS_query issued from the ATV.
   3.2. M2 converts the SchemeName_DNS_query into SchemeName_DNS_query Minder Signal.
   3.3. Minder inspects the DNS query and issues the SchemeName_DNS_query to TSPA.
   3.4. The TSPA returns a RR response containing the Trust Scheme Provider Location associated with the Scheme Name.

3.5. Minder inspects the RR response and calls the RR response slot of M2, which in turn forwards it to the ATV.

4. The ATV issues a SchemeProviderLocation_IssuerName_query for the Issuer Name and Scheme Provider location obtained from the previous DNS queries.
   4.1. M2 intercepts the SchemeProviderLocation_IssuerName_query,
   4.2. M2 converts the SchemeProviderLocation_IssuerName_query into SchemeProviderLocation_IssuerName_query Minder Signal.
   4.3. Minder inspects the query and issues the SchemeProviderLocation_IssuerName_query to TSPA
   4.4. The TSPA returns a signed Association_Statement response to Minder.
   4.5. Minder inspects the AssociationStatement response and calls the Association response slot of M2, which in turn forwards the response to the ATV.
5. The ATV issues a SchemeNameConstraints_DNS_query to TSPA.
   5.1. M2 intercepts the SchemeNameConstraints_DNS_query issued by the ATV.
   5.2. M2 converts the SchemeNameConstraints_DNS_query to the SchemeNameConstraints_DNS_query Minder signal.
   5.3. Minder inspects the SchemeNameConstraints_DNS_query and issues it to TSPA via its DNS Client API.
   5.4. TSPA returns a SMIME/RR response containing the Constraints to Minder.
   5.5. Minder inspects the RR response and calls the ConstraintsRR response Slot of M2., which in turn forwards it to the ATV.

5.2 ATV-TTA Integration Testing Scenarios

Prerequisites

1. Trust policy and electronic transaction exist as test assets.
2. After *Querying of Trust Scheme Membership Scenario in TSPA is completed,* it ends with a *false* Boolean value meaning that ATV completes the trust policy analysis for the trust scheme and does not find matching between the trust scheme of the transaction and the policy. Then ATV decides to query for the trust translations
3. For Tuple&Ordinal trust schemes, Trust levels for the trust scheme has already defined in TSPA.

Main Flow

1. Minder calls the startTest slot of the M1 Adapter.
   1.1. M1 provides the electronic transaction and the trust policy test assets to ATV and triggers the initiation of the integration scenario.
2. The ATV issues a SchemeName_DNS_query to look for a matching trust translation scheme for the translation provider
   2.1. The SchemeName_DNS_query issued from ATV is intercepted by M2,
   2.2. M2 converts SchemeName_DNS_query into SchemeName _DNS_query Minder Signal.

2.3. Minder inspects the DNS query and issues the SchemeName_DNS_query to the DNS component of the TTA.
2.4. The TTA returns a RR response containing the pointer to the Trust Translation List associated with the Scheme Name.
2.5. Minder inspects the RR response and calls the RR Response Slot of M2, which in turn forwards the RR Response to ATV.

3. The ATV issues a SchemeTranslationProvider_SchemeName_TrustLevel_query in order to look for the TTA provider with the Scheme Name and Level in the form of DomainName
3.1. The SchemeTranslationProvider_SchemeName_TrustLevel_query issued from ATV is intercepted by M2
3.2. M2 converts SchemeTranslationProvider_SchemeName_TrustLevel_query to Minder Signal
3.3. Minder inspects the DNS query and issues the SchemeTranslationProvider_SchemeName_TrustLevel_query to the DNS component of TTA
3.4. The TTA returns a RR response containing the equivalent signed trust scheme level list document
3.5. Minder inspects the RR response and calls the RR Response Slot of M2, which in turn forwards the RR Response to ATV

4. The ATV issues a SchemeNameConstraints_DNS_query in order to retrieve the certificate constraints
4.1. The SchemeNameConstraints_DNS_query issued from ATV is intercepted by M2
4.2. M2 converts SchemeNameConstraints_DNS_query to Minder signal
4.3. Minder inspects the DNS query and issues the SchemeNameConstraints_DNS_query to the DNS component of TTA
4.4. The TTA returns a RR response containing the certificate constraints
4.5. Minder inspects the TT response and calls the RR Response Slot of M2, which in turn forwards the RR response to ATV

5.3 ATV-DP Integration Testing Scenarios

Prerequisites

1. Trust policy and electronic transaction exist as test assets.
2. Delegation file exists in the electronic transaction.
3. Mandator should delegate DP to revoke the respective delegation.
4. Mandator should issue a certificate to DP for revocation purposes.

Main Flow

1. Minder triggers the Querying of Trust Delegation Discovery Integration Test Scenario.
1.1. Minder calls the startTest slot of the M1 Adapter.
1.2. M1 provides the test assets (electronic transaction in Asic format) to ATV and triggers the initiation of the test scenario. (Note that electronic transaction includes delegation)

2. The ATV issues a Revocation_List_Query with the Delegation_Hash that is the hash of delegation extracted from the electronic transaction.
   2.1. The Revocation_List_Query issued from ATV is intercepted by M2.
   2.2. M2 converts the Revocation_List_Query into Revocation_List_Query Minder Signal.
   2.3. Minder inspects the Revocation_List_Query and sends it to DP for the query.
   2.4. The DP returns a Revocation_Status response to Minder if the delegation hash corresponding to the queried hash exists.
   2.5. Minder inspects the Revocation_Status response and calls the Revocation_Status response slot of M2, which in turn forwards the response to the ATV.

6. Integration Test Assertions

This section lists the first round of test assertions to verify that components TSPA, TTA, and DP are very well integrated and work in a harmonized way. These test assertions were extracted from the respective Conceptual Framework, Requirements and Uses and Design related deliverables (D2.3, D2.14, D3.3, D3.4, D4.4, D5.2). Following the methodology described in the previous section, we first list the normative sources together with their references, and then derive the test assertions to test for integration interfaces.

For traceability purposes, the identifiers of the items (conformance clause, normative sources and test assertions) were done using the following convention: **CC/NS/TA_LightestComponentName(TSPA/TTA/DP)_Number.**

6.1 TSPA

6.1.1 ATV-TSPA Integration Testing

Integration Testing activity for TSPA aims to verify that scenarios from ATV to TSPA fulfill the software interface requirements that can be derived from the sequence flows in the Reference Architecture (D2.14) and scenarios in the D3.4. The conformance clause, the normative statements and the test assertions are provided to satisfy the interfaces related with TSPA so that integrated system works in a harmonized way including all its components.

6.1.1 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.

6.1.2 ATV-TSPA Integration Normative Sources

NS ID	NS_TSPA_1
Reference	FR-05.00, D2.3 Section 5.1
Description	The Trust Scheme Publication Authority MUST be able to operate an off-the-shelf DNS Name Server with the DNSSEC extension

NS ID	NS_TSPA_2
Reference	D3.3 Section 5.3

Description	Subject Alternative Name field of the Certificate used to sign the electronic transaction points to the domain name (domain name used as Identifier) of the entity.

NS ID	NS_TSPA_3
Reference	D3.3 Section 5.3
Description	Issuer Alternative Name field of the Certificate used to sign the electronic transaction points to the domain name (domain name used as Identifier) of the issuer of the entity that issued the certificate.

NS ID	NS_TSPA_4
Reference	D3.3 Section 5.4
Description	The Authenticity of Trust Declarations of TSPA is ensured as follows: A URI Record is authentic if DNSSEC validation succeeds. A PTR Record is authentic if DNSSEC validation succeeds A Trust List is authentic: - if the certificate used to sign the list is valid under the constraints of the SMIMEA records published under the same domain name. - if the signature is valid

NS ID	NS_TSPA_5
Reference	D3.3 Section 6.1
Description	A TSPA is composed of a DNS Name Server with DNSSEC extension that contains: Resource Records (PTR) for certificate Issuer (Issuer Name) – pointing to the URI RR of the Trust Scheme

	Resource Records (URI) for Trust Scheme (Scheme Name) – pointing to the Trust Scheme Provider Resource Records (SMIMEA) for Trust Scheme (Scheme Name)

NS ID	NS_TSPA_6
Reference	D3.3 Section 6.1
Description	A TSPA is composed of a HTTP Server (Trust Scheme Provider) that contains: - Signed Trust Lists - Tuple-based (ordinal and Boolean included) representations of Trust Schemes, provided as pointer from Trust List.

NS ID	NS_TSPA_7
Reference	D3.4 Section 5.3
Description	URI Resource Record contains exactly one URI as its record data.

NS ID	NS_TSPA_8
Reference	D3.4 Section 5.3
Description	PTR Resource Record contains more than one domain as its record data.

NS ID	NS_TSPA_9
Reference	D3.4 Section 9.1.2
Description	SMIMEA Resource Record contains four possible constraint fields, such as: - CA constraints - Service Certificate Constraints

	- Trust Anchor Assertion - Domain-Issued Certificate

NS ID	NS_TSPA_10
Reference	D3.4 – Section 6.1
Description	Trust Schemes published in the DNS Name Server of TSPA can be either Boolean or Ordinal.

NS ID	NS_TSPA_11
Reference	D3.4 Section 6.1
Description	IssuerName Query must be of the following structure: _scheme._trust.IssuerDomainName IN PTR

NS ID	NS_TSPA_12
Reference	D3.4 Section 6.2
Description	According to the format of the Trust Scheme, the Response to the IssuerName Query is a PTR Resource Record that contains : - the domain name of the SchemeName if Boolean - levelName.domainName of the SchemeName if Ordinal

NS ID	NS_TSPA_13
Reference	D3.4 Section 6.2
Description	SchemeName Query utilizes the Domain Name of the SchemeName obtained from NS_TSPA_10 and must be of the following structure: _scheme._trust.SchemeNameDomainName IN URI

NS ID	NS_TSPA_14
Reference	D3.4 Section 6.2
Description	The Response to the SchemeName Query is a URI Resource Record that points to the Trust Scheme Provider under which the trust list of the Scheme is published.

NS ID	NS_TSPA_15
Reference	D3.4 Section 6.2
Description	The response of a Trust List query (IssuerName_SchemeName_association query) MUST be a signed Association Statement/signed trust list (boolean)

NS ID	NS_TSPA_16
Reference	D3.4 Section 9.1.2
Description	A Certificate Constraint Query must be of the following structure: _scheme._trust. SchemeNameDomainName IN SMIMEA

6.1.3 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 contains published trust scheme membership declarations in the form of PTR Records. 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 and contains published scheme locations declarations in the form of URI Records. 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 and contains published trust list declarations in the form of signed trust lists. 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 and contains published trust list declarations in the form of signed trust lists. 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 and contains published trust scheme membership declarations in the form of PTR Records. 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 and contains published trust scheme membership declarations in the form of PTR Records. 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 and contains published trust scheme membership declarations in the form of PTR Records. 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 and contains published trust scheme membership declarations in the form of PTR Records. 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 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 and contains published trust scheme membership declarations in the form of PTR Records. 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.

6.2 TTA

6.2.1 ATV-TTA Integration Testing

Integration Testing activity for TTA aims to verify that scenarios from ATV to TTA fulfill the software interface requirements that can be derived from the sequence flows in the Reference Architecture (D2.14) and scenarios in the D4.4. The conformance clause, the normative statements and the test assertions are provided to satisfy the interfaces related with TTA so that integrated system works in a harmonized way including all its components.

TTA is composed of two functionalities: Trust Translation List Provider and Trust Translation Publisher (DNS with DNSSec Extension).

Integration testing scope does not include the interfaces among the TTA sub-components Trust Translation List Provider and Trust Translation Publisher. TTA functionalities are taken as a whole and interfaces from/to TTA are taken into consideration.

6.2.2 TTA Integration Testing Conformance Clauses

We have only one integration conformance clause:

CC_TTA_1: An implementation of TTA is conforming to TTA if it satisfies the conditions provided in the normative statements (NS_TTA_1-10) under the next section 6.2.3.

6.2.3 ATV-TTA Integration Normative Sources

Normative Sources are elicited from the scenarios that includes the flows among the components defined in D2.14 and D4.4.

NS ID	NS_TTA _1
Reference	FR-06.00- TTA: Integratable with DNSSEC, FR-06.06- TTA: Discoverability, FR-06.02- TTA: Utilities to Load selected Trust Translation Data
Description	ATV's request for trust translation lists for the trust scheme is handled by DNS Name Server with DNSSEC extension for translation publish and discovery in TTA

NS ID	NS_TTA _2
Reference	FR-06.01- TTA: Trust Data Flexibility, D4.4 Section 7.1
Description	The TTA provides multiple Trust Translation Lists under different subdomains for each recognized trust level with either XML, TPL, or both.

Other Normative Sources extracted from the WP4 Documents are given below:

NS ID	NS_ TTA_3

Reference	D4.4 Section 7.3, NS_TTA_1
Description	TTA will publish a pointer to the Trust Translation List for the group in the form of a series URI resource records

NS ID	NS_ TTA_4
Reference	D4.4 Section 6.2, NS_TTA_1, NS_TTA_9, NS_TTA_15
Description	For the authenticity of Trust Translation List declarations, the certificate used to sign the list is verified under the constraints of the SMIMEA records

NS ID	NS_TTA_5
Reference	D4.4 Section 6.2, NS_TTA_1, NS_TTA_3,NS_TTA_2
Description	A TTA publishes the trust translation list as the following: a. Resource Records (URI) for Trust Scheme (Scheme Name) – pointing to the Translation Lists with either XML or TPL format b. Resource Records (SMIMEA) for Trust Translation Lists for the Trust Scheme (Scheme Name)

NS ID	NS_TTA_6
Reference	D4.4 Section 6.1
Description	A TTA provides a public Restful API that provides Signed Trust Translation Lists

NS ID	NS_TTA_7
Reference	NS_TTA_2, NS_TTA_6
Description	In case of XML file type, TTA provides the list of the trust levels equivalents to the one requested with level name and trust scheme name

NS ID	NS_TTA_8
Reference	NS_TTA_2, NS_TTA_6

Description	In case of TPL file type, TTA should return the list of the trust levels equivalents to the one requested with level name, trust scheme name and TPL description

NS ID	NS_TTA_9
Reference	D4.4 Section 6.2, NS_TTA_1
Description	TTA provides certificate constraints to use for the verification of the translation list signature.

6.2.4 ATV - TTA Integration Test Assertions

TTA Test Assertions are associated with CC_TTA_1 conformance clause in order to define the integration testing specifications and later on test cases will be derived from test assertions that address the normative statements of the specification.

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 _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 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_3
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and contains published trust translation list declarations.
Prescription Level	Mandatory
Predicate	TTA publishes a pointer to the trust translation list for the group in the form of a series URI resource records

TA ID	TA_TTA_5
Normative Source	NS_TTA_2, NS_TTA_5
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and contains published trust translation list declarations.

Prescription Level	Preferred
Predicate	TTA returns more than one file for each recognized trust level with Boolean, Ordinal or Tuple trust scheme types in XML or TPL format.

TA ID	TA_TTA_6
Normative Source	NS_TTA_2, NS_TTA_6, NS_TTA_7
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and 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	In case of XML, TTA returns the list of the trust levels equivalents to the one requested with level name and trust scheme name.

TA ID	TA_TTA_7
Normative Source	NS_TTA_2, NS_TTA_6, NS_TTA_8
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and contains published trust translation list declarations. Trust translation lists are already defined in the TPL 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	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_8
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 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_9
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 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_10
Normative Source	NS_TTA_10
Target	ATV – TTA Interface
Prerequisite	The TTA DNS Name Server is up and running and 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.

6.3 ATV-DP Integration Testing

In order to test if the Delegation Provider (DP) implementation is integrated with the other LIGHTest components seamlessly, conformance clause, testing scenario, the normative statements and the test assertions are given below.

In DP integration testing scenarios, delegation publisher interface between DP and Mandator is omitted as the main focus of the LIGHTest integration testing is to test the harmonization between the Lightest components. Therefore, the Mandator interface is not taken into account.

6.3.1 DP Integration Testing Conformance Clause

IC_DP_1: An interface between DP and ATV is integrated seamlessly if it satisfies the conditions provided in the normative statements NS_DP_1 to NS_DP_7.

6.3.2 ATV – DP Integration Normative Statements

NS ID	NS_DP_1
Reference	D5.2 Section 9
Description	DP provides a RESTFUL API to publish and download delegations.

NS ID	NS_DP_2
Reference	D5.2 Section 9.3
Description	DP should possess a revocation list to check if the delegation is revoked. If so, DP should return a notification saying that the delegation is revoked.

NS ID	NS_DP_3
Reference	D5.2 Section 9.4
Description	DP MUST respond only one revocation query at the time.

NS ID	NS_DP_4
Reference	D5.2 Section 9.4.2
Description	DP MUST sign the revocation response with the certificate that is issued by Mandator for the revocation purpose.

NS ID	NS_DP_5
Reference	D5.2 Section 9.4.2
Description	Revocation query that is sent to DP MUST include hash of the delegation (delegation_id) to be queried for revocation status.

NS ID	NS_DP_6
Reference	D5.2 Section 9.4.2
Description	Revocation response MUST include the delegation that is given to DP, the certificates that is used to sign, and all certificates to build the trust chain. DP should possess a revocation list to check if the delegation is revoked. If so, DP should return a notification saying that the delegation is revoked.

NS ID	NS_DP_7
Reference	D5.2 Section 9.4.2
Description	The ATV is required to check that the DP indeed has a valid entry containing (amongst others) a hash of the delegation. This check is to ensure that the delegation has not been revoked by the mandator.

6.3.3 ATV – DP Integration Test Assertions

TA ID	TA_DP_1
Normative Source	NS_DP_1
Target	DP-ATV interface
Prerequisite	Delegation file is needed.
Prescription Level	Mandatory
Predicate	DP MUST provide a Restful interface to ATV to validate delegations

TA ID	TA_DP_2
Normative Source	NS_DP_2
Target	DP-ATV interface
Prerequisite	Revoked Delegation Delegation file is needed.
Prescription Level	Mandatory
Predicate	DP MUST return a notification saying that the delegation is revoked if the delegation is found and revoked.

TA ID	TA_DP_3
Normative Source	NS_DP_2
Target	DP-ATV interface
Prerequisite	Valid Delegation file is needed.
Prescription Level	Mandatory
Predicate	DP MUST return a notification saying that the delegation is valid.

TA ID	TA_DP_4
Normative Source	NS_DP_3
Target	DP-ATV interface
Prerequisite	Revoked Delegation file is needed.
Prescription Level	Mandatory
Predicate	DP MUST return error if ATV sends more than one revocation query at the time.

TA ID	TA_DP_5
Normative Source	NS_DP_4
Target	DP
Prerequisite	Revoked Delegation file is needed. Mandator should delegate DP to revoke the respective delegation. To revoke a delegation, the Mandator must identify the delegation which delegation to revoke first.
Prescription Level	Mandatory
Predicate	DP MUST respond to ATV's revocation query with a signed revocation response

TA ID	TA_DP_6
Normative Source	NS_DP_5, NS_DP_7
Target	DP
Prerequisite	Revoked Delegation file is needed. Mandator should delegate DP to revoke the respective delegation. To revoke a delegation, the Mandator must identify the delegation which delegation to revoke first.
Prescription Level	Mandatory

Predicate	ATV MUST send the Revocation query with the hash of the delegation (delegation_id).

TA ID	TA_DP_7
Normative Source	NS_DP_6
Target	DP
Prerequisite	Revoked Delegation file is needed. Mandator should delegate DP to revoke the respective delegation. To revoke a delegation, the Mandator must identify the delegation which delegation to revoke first.
Prescription Level	Mandatory
Predicate	DP's response for ATV's revocation query MUST include the delegation that is given to DP, the certificates that is used to sign and all certificates to build the trust chain if the delegation is found and valid.

TA ID	TA_DP_8
Normative Source	NS_DP_6
Target	DP
Prerequisite	Revoked Delegation file is needed. Mandator should delegate DP to revoke the respective delegation. To revoke a delegation, the Mandator must identify the delegation which delegation to revoke first.
Prescription Level	Mandatory
Predicate	DP MUST respond to ATV's revocation query with an error/notification message if delegation is not found.

7. References

The LIGHTest Project, D2.3 – Requirements and Use Cases, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D2.3.pdf

The LIGHTest Project, D2.14 - Reference Architecture, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D2.14.pdf

The LIGHTest Project, D3.3 – DNS-based Publication of Trust Schemes, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D3.3.pdf

The LIGHTest Project, D3.4 – Discovery of Trust Scheme Publication Authorities, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D3.4.pdf

The LIGHTest Project, D4.3 – DNS-based Publication of Trust Translation Schemes, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D4.3.pdf

The LIGHTest Project, D4.4 – Discovery of Trust Translation Authorities, Project Deliverable, 2017; https://www.lightest.eu//static/deliverables/D4.4.pdf

The LIGHTest Project, D5.2 – Conceptual Framework for Delegations (2), Project Deliverable, 2018.

8. 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), IBM Danmark (DK) 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.