D4.3

DNS-based Publication of Trust Translation Schemes

Document Identification	
Date	28.02.2018
Status	Final
Version	Version 1.0

Related WP	WP 4	**Related Deliverable(s)**	D4.1, D3.1, D3.2, D2.14
Lead Authors	Miguel Mateo Montero, Javier Presa, Miryam Villegas	**Dissemination Level**	PU
Lead Participants	ATOS	**Contributors**	NLNET, G&D, FHG
Reviewers	DTU, GS		

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

The purpose of deliverable D4.3 is to present the mechanisms to publish the trust translation declaration lists using the existing data elements of the DNS system. It results from the adoption of a common definition of Publication to apply in the three work packages managing publication of Trust related data; WP3, WP4 and WP5.

Section 5 defines some concepts used in the document, which serve as a basis understanding in the WP.

Section 6 provides the common approach adopted in the project regarding the publication of Trust data. This section can be found entirely o partially in the corresponding deliverables of WP3 and WP5, with the aim of simplifying the task of understanding concepts within the deliverable without looking for references in other documents.

Section 7 defines the Trust Translation Authority concept, which is explained across examples, providing a specification of data model and its functional modules.

Finally, section 8 includes an example of use of the TTA functionality in the real world.

2. Document Information

2.1. Contributors

Name	Partner
Frank-Michael Kamm	G&D
Javier Presa	ATOS
Martin Hoffmann	NLNET
Miguel Mateo Montero	ATOS
Miryam Villegas	ATOS
Heiko Roßnagel	FHG

2.2. History

Version	Date	Author	Changes
0.1	27/10/2017	Miguel Mateo Montero Javier Presa	ToC added.
0.2	7/12/2017	Miryam Villegas Javier Presa	ToC modification Section 6.
0.3	16/01/2018	Miguel A. Mateo Javier Presa Miryam Villegas	Chapters 5, 7, 8.
	16/01/2018	Martin Hoffmann Javier Presa	Chapter 6
0.4	25/01/2018	Miryam Villegas	Removal of the methodology chapter. Updating some figures. Adding an example of trust translation in chapter 6. Some edition tasks.
	26/01/2018	Frank-Michael Kamm	Use case of translation in mobile ID.
	01/02/2018	Miguel Mateo Montero	Functional architecture.
0.5	09/02/2018	Miryam Villegas Miguel Mateo Montero	Re-arrange of sections and figures
1.0	27/02/2018	Miryam Villegas Miguel Mateo Montero	Final spellchecking and semantic review. Addressing comments from internal review.

3. Table of Contents

3.1. Table of Figures

3.2. Table of Tables

3.3. Table of Acronyms

AAID	Authenticator ID
AC	Authenticator Certificate
ATV	Automated Trust Verifier
AV	Authenticator Version
CA	Certification Authority
DNS	Domain Name System
DNSSEC	Domain Name System SECurity extensions
eIDAS	Electronic IDentification And Signature (Regulation (EU) No 910/2014 on electronic identification (eID) and trust services for electronic transactions in the internal market)
FIDO	Fast IDentity Online
ID	Identity, identification
LoA	Level of Assurance
SE	Secure Element
TEE	Trust Execution Environment
TPL	Trust Policy Language
TSPA	Trust Scheme Publication Authority
TTA	Trust scheme Translation Authority
TTL	Trust Translation List

4. Scope of the Deliverable

This deliverable aims to document the design for the publication functionality of the Trust Scheme Translation within the LIGHTest architecture, using the DNS framework with DNSSEC extension as described in D4.1. This publication makes data become available to be discovered by the Automatic Trust Validator (ATV) and assist to it in the process of validating a transaction.

At this point we want to remark that translating a given trust scheme means *translating a given trust level of such trust scheme*. Thus, the Trust Translation Authority is going to provide a list of trust levels of trust schemes, which are equivalent to the trust level of the trust scheme that serves like input. (See [1])

The publication of Trust Translation Data does not differ from the publication of other Trust Related information such as Trust Schemes and Trust Delegations, which drive to provide a consolidated approach to the Trust-related data. Therefore this approach has been developed in cooperation with WP3 and WP5 and can be used to specify the publication either of Trust Schemes, or Trust Translations or Trust Delegations. This approach is included in the three deliverables related to the publication within the tree work packages; D3.3 [2], section 6 of this document and D5.3 [3].

5. Terminology

This section provides the terminology used in this deliverables in order to use the same words and terminology and ease, in this way, the collaboration with partners and third entities.

5.1. Trust Translation Authority

It provides the translation between trust schemes, by enabling trust scheme providers to indicate which trust schemes are recognized as trustworthy by their trust scheme. It consists of two functions: Trust Translation List Provider and Trust Translation Publisher.

5.2. Trust Translation List Provider

This function is used to feed the TTA with translations or equivalences between Trust Scheme Levels and transform this data into Trust Translation Declarations. It is always operated by the Trusted Scheme Operator and provides a list of the Recognized Trust Schemes.

In this document translation or equivalence words are used indistinctly in the sense of unidirectional equivalence or translation. This is, the fact that Trust Scheme Level X is equivalent or is translated into Trust Scheme Level Y does not imply that Trust Scheme Level Y is equivalent or could be translated into Trust Scheme Level X.

5.3. Trust Translation Publisher

This function provides the required interfaces to make Trust Translation data publicly available, and in further steps, its discovering.

5.4. Verifier

Person or entity that wants to check if a transaction is valid or not

5.5. Automated Trust Verifier

Component used by the Verifier to verify documents. This component queries the Trust Translation Publisher to get Trust Scheme Levels equivalent to the object of verification.

6. Publishing Trust-related Information in the DNS

The trust framework created by LIGHTest verifies the trustworthiness of an electronic transaction by attempting to establish a chain of trust from a set of pre-configured, well-known trust sources to this transaction. The links in this chain are assurances that the trust into a source can be extended to another entity.

Within the LIGHTest project, three working packages look at different aspects of such assurances. WP3 examines the most basic form where a trust source known as a trust scheme declares that some issuer of trust services, such as certificates or time stamps, conforms to the conditions and rules set out by the scheme and thus extend trust placed into it onto this trust service. WP4 assesses the relationship between multiple trust schemes allowing a trust scheme or some other trusted entity to declare if and how trust into one scheme extends to trust into another scheme. WP5, finally, looks into how individual entities can empower other entities to act on their behalf – extending trust into themselves onto that other entity within certain well-defined limits.

6.1. Trust Declarations

However different they may appear, each of these aspects follows a similar pattern: Some entity makes a *trust declaration* stating that trust into a certain entity extends to another entity, possibly providing conditions and limits of such an extension of trust. To simplify further discussion, it will be helpful to label the three entities involved in the process. The entity issuing the declaration shall be the *originator,* the entity that is already trusted is the *source* and the entity trusted as a result of the declaration is the *target.*

Within the aspects discussed as part of the LIGHTest project, in many cases the originator of a declaration is identical to the source. This is certainly true for trust membership publication in WP3, where the trust scheme itself declares which trust services are member. In the aspects of the other two work packages, similarly originator and source are most often identical. However, there may be use cases where this is not the case. For instance, a third party may declare a trust translation independently of the trust schemes that are source or target of this declaration. Similarly, a third party may declare a trust delegation. For instance, business registries often provide information about individuals that are allowed to sign on behalf of a company. Such information can be modeled as third-party trust delegation.

This definition specifies a single declaration to extend trust from exactly one source to exactly one target. In practice, the document formats used often contain multiple declarations according to this definition. For instance, the trusted lists defined in ETSI TS 119 612 used for the declarations in WP3 contain a list of all the targets that a trust scheme as a source wishes to extend trust to. This published form of declarations shall be called *trust declaration documents.*

At least in principle each such document can contain one or many declarations with any number of sources and targets.

For a trust declaration to be considered when building the chain of trust during validation of an electronic transaction, the declaration itself needs to be trusted. If a declaration document is treated like any other electronic transaction, this trust can in turn be established through verification using the LIGHTest framework. That is, there needs to be a chain of trust from pre-configured trust sources to the originator of the declaration document for the particular aspect of the declaration in question.

Note that in most cases where the originator of the declaration is identical to the source of the declaration, this chain exists implicitly and no extra checks are needed. It may, however, be possible that conditions for trusting the source as such and trusting declarations made by it are different and thus need to be verified independently.

6.2. Publication of Trust Declarations

In order to use trust declarations for automatic, unaided verification, a validator needs to have access to all potentially applicable declarations. There are two fundamental strategies: a declaration could either be actively supplied as part of the input or configuration or it is left to the verifier itself to discover it.

The prime example for the former case is that a declaration is provided as part of the electronic transaction to be verified. This is particularly useful if the creator of the transaction is aware that the declaration is necessary for verification. For instance, if an entity signs a document in their function as a proxy, the transaction will only ever verify if the declaration of trust delegation – the mandate – is known to the verifier. The proxy may very well include the mandate in the transaction right away.

For declarations that are potentially applicable to a large amount of transactions, such a strategy isn't very practical. Instead, it is better to make the declarations available publicly and provide means for a verifier to discover how and where it can retrieve them.

While the most likely method for publishing currently is the Hypertext Transfer Protocol (HTTP), other protocols may become available in the future. An extensible standard exist to describe both the method used for accessing a resource and all necessary parameters for a successful retrieval in the form of a Uniform Resource Identifier (URI). It encodes all information into a single string which can be easily stored or transmitted.

6.3. Publication of Trust Translation Declarations on DNS

A declaration for trust translation states that if you trust a certain entity, you can safely also trust another entity. Within LIGHTest, we only discuss trust translation between levels of trust schemes. In this case, the trust scheme level that is trusted already either because of configuration or already applied trust translations is called the *trusted scheme level* and is the source of the declaration. The target of the translation is called the *equivalent scheme level.* A trust translation declaration can be made by any entity but it can only be considered for validation if it is made by an entity that is trusted for the purpose of trust translation. Typically, the declaration therefore originates with the *trusted scheme as said in* [4].

The trust declaration and trust declaration document are to be defined as part of the LIGHTest project.

The originator of a trust translation declaration publishes its location under the *_translate._trust* prefix using the URI record. The source and target of a declaration can, if they are not also the originator, point the originator by publishing a PTR record with that entities domain name identifier under the selfsame prefix (see [5]).

In the following example, the use of the three resource records (PTR, URI, SMIMEA) proposed to be used in LIGHTest is shown during the request for translating a given level of trust in a certain trust scheme. This is an example of discovery although it is used as a proof of concept in our publication design.

Let's take the example, in particular the trust translation for the Advanced level of an eSeal in eIDAS regulation (or eIDAS trust scheme), with the corresponding hypothesis related to the existing bilateral agreements.

The *trust scheme* eidas.eu may publish claims to being an equivalent scheme. Being purist, it is publishing the different *trust services* that can be searched for their *trust level* equivalences:

[Note that we are assuming any trust scheme contains trust services in the current reasoning.]

```
;; QUESTION SECTION:
;_translate._trust.eidas.eu.  IN  PTR

;; ANSWER SECTION:
_translate._trust.eidas.eu.   IN  PTR
                    eID.eidas.eu \
                    eSignature.eidas.eu \
                    eSeal.eidas.eu \
                    eTimestamp.eidas.eu \
                    eDelivery.eidas.eu \
                    websiteCertificate.eidas.eu \
```

In particular, we are interested in the electronic seal:

```
;; QUESTION SECTION:
;_translate._trust.eSeal.eidas.eu.  IN  PTR

;; ANSWER SECTION:
_translate._trust.eSeal.eidas.eu.   IN  PTR
                    advanced.eSeal.eidas.eu \
                    qualified.eSeal.eidas.eu
```

For the advanced level we want to know about its equivalent levels in other trust schemes, we ask now. The verifier needs to check this claim by locating the trust translation declaration:

```
;; QUESTION SECTION:
;_translate._trust.advanced.eSeal.eidas.eu.  IN  URI

;; ANSWER SECTION:
_translate._trust.advanced.eSeal.eidas.eu.   IN  URI
                    https://lightest.eu/translation_list_xxx.xml
```

The verifier downloads the document and finds the list of other trust levels equivalent to the advanced level of the eIDAS electronic seal. It needs to check the certificates used for signing the document, so:

```
;; QUESTION SECTION:
;_translate._trust.advanced.eSeal.eidas.eu.  IN  SMIMEA

;; ANSWER SECTION:
_translate._trust.advanced.eSeal.eidas.eu.   IN  SMIMEA  \
                                  <SMIMEA record data>
```

Assuming the certificates verify as well, the verifier now knows that the certificate indeed is a valid qualified certificate as per the eIDAS regulation.

7. Trust Translation Authority Model

The TTA, as described in D2.14 [4](section 7.2.8) is a function that enables the interoperability of trust schemes published by different entities and communities (and possibly across different trust domains) by defining the relationships between the levels of trust of the different trust schemes. To perform this function the TTA has to provide two different services; the first service is dedicated to publish the Trust Translation data and make it public available while the aim of the second service is to provide the required mechanism to let users (in this case the ATV) to discover such information. As can be deduced the second service (discovery) provides a means to find the information provided by the first service (publication of trust translation data) therefore, although the present document focuses on the publication service, it depicts some aspects of the discovery which will be extended in future deliverables of this Work Package 4.

The Trust Translation in LIGHTest can be provided between a trusted Boolean or Ordinal Trust Scheme or tuple-based Trust Scheme but, on top of these cases, translation between Tuple-Based Trust Scheme Publications of Trusted and Recognized Schemes is the generic translation of the TTA model.

When translating a given trust level in a trust scheme (the trusted one), the TTA is going to find a list of equivalent trust levels in other trust schemes (the recognized ones). Such a Trust Translation List contains a mapping between the Trusted Scheme and one or multiple Recognized Schemes, as any Trusted Scheme could be equivalent to not only one Recognized Scheme.

Based on previous conceptual view presented on D4.1 [1], and after analysing the possible options on the trust translations between different schemes, we extend the TTA conceptual model previously submitted adding more capability to represent all the possible scheme translations:

- If the TTA receives a request for translating a boolean trust level, the equivalent list returned will be conformed with those boolean trust levels equivalent to the trusted one, and only the boolean ones.
- If the TTA receives a request for translating an ordinal trust level, the equivalent list returned will be compound by those boolean trust levels and those ordinal trust levels equivalent to the trusted one, and only the boolean/ordinal ones, since any boolean trust level can be expressed like an ordinal trust level (see D2.14 [4]).
- If the TTA receives a request for translating a tuple-based trust level, the equivalent list returned will be compound by those boolean trust levels, those ordinal trust levels, and

those tuple-based equivalent to the trusted one. Note since any boolean/ordinal trust level is doable of being expressed like a tuple-based trust level (D2.14 [4]), the very complete translation list for a given trust level is obtained when such trusted/source trust level is expressed like tuple-based.

Therefore, because of trust levels of a boolean or ordinal trust schemes can be expressed like tuple-based trust levels, (see D2.14 [4], D4.1 [1]) TTA will internally will manage the trust schemes without making distinctions if it is boolean, ordinal or tuple-based. This means that every trust level to be published will be kept in a tuple-based way, although internal DB will keep register of their nature.

7.1. Trust Translation Authority conceptual model design

Although the TTA conceptual model has been depicted in D4.1 [1] already, the Figure 1 provides an enhanced overview on the conceptual view of representation the model in the TTA. This view distinguishes between data required for discovery of the Trust Translation List (DNS-server side) where all types of Trust Scheme are represented, and data required for representing a translation between a Trusted Scheme and a Recognized Scheme (Trust Translation List Provider side). Any data package on the DNS is identified by the associated Boolean (<SchemeName>) Trust Scheme Name, ordinal/tuple-based (<Levelname>.<SchemeName>) Scheme Name. In addition, it is included the fingerprint of the Trust Translation List Provider Certificate, which is used to sign the Trust Translation List.

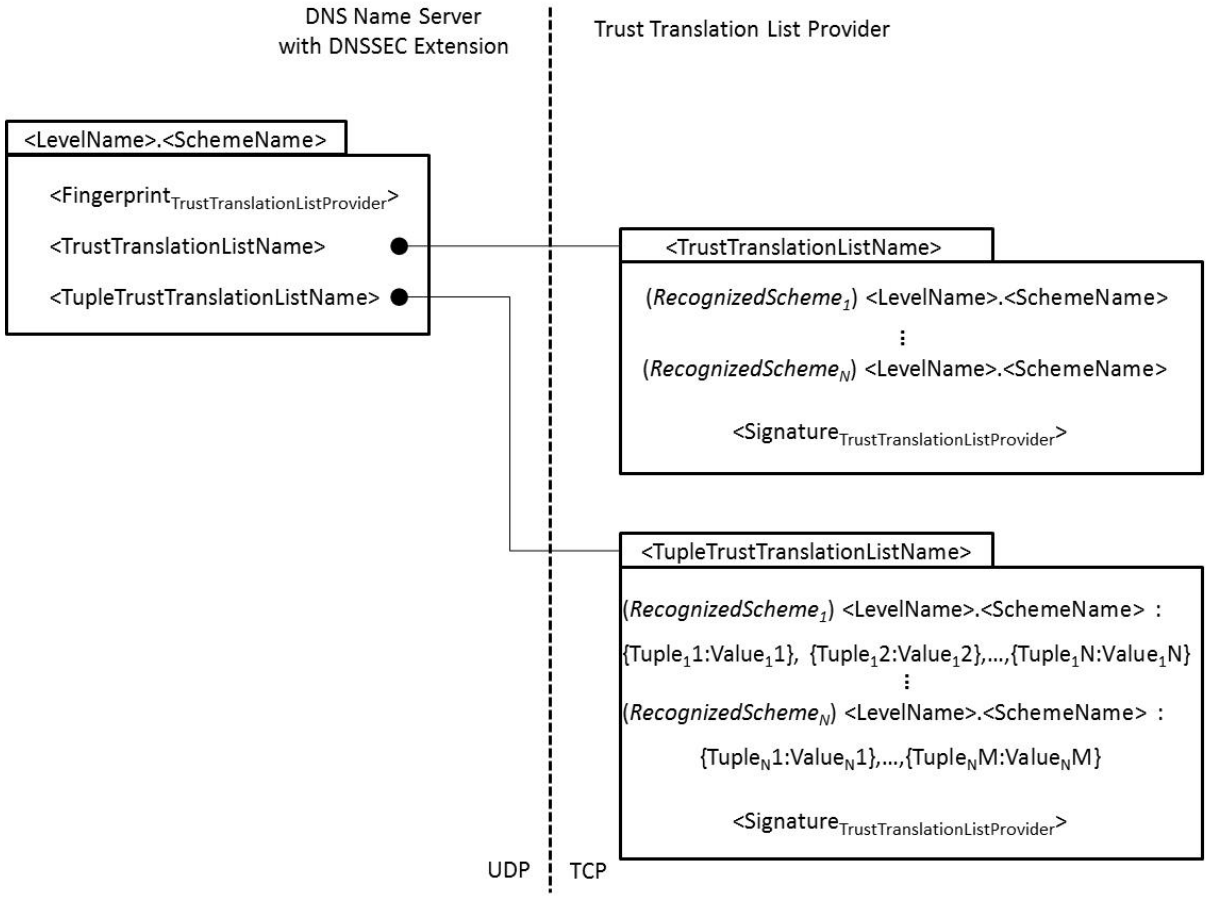

Figure 1: Conceptual model of Trust Translation Authority

As it could be derived from the figure, there are two types of Trust Translation List: Ordinal or TupleBased, and to note that a Trusted Scheme could be equivalent to one or multiple Recognized Trust Scheme.
In case of the Tuple Based Trust Scheme, the Recognized Trust Schemes do not have to contain the same number of tuples as their equivalent Trusted Scheme. It is important to highlight the Trust Translation Authority publishes equivalences between different Trust Levels, not between complete Trust Schemes.

7.2. Some use cases for the TTA

In this section, several use cases for trust level translations are described in order to check the previous TTA model is valid. We are going to suppose the corresponding bilateral agreements are signed and currently in use.

Boolean/ordinal use case:

Let's take the TTA providing translation lists for the electronic seal (eSeal) in an eIDAS scheme, so some lists of recognized trust schemes are published by the TTA for an **ordinal trust scheme**. In this case, the eSeal in eIDAS can be advanced or qualified (see D4.1 [1] and eIDAS regulation [6]), i.e. two values of level of trust that have to exist in the TSPA list. We will have two lists of equivalences/translations: one list for the *advanced trust level* and another list for the *qualified trust level* (remember there are agreements signed between the related trust bodies of each equivalence).

Imagine Spain's government has signed an agreement with the company X from a country that is not within eIDAS regulation. Such company X provides three levels of trust for its electronic seals: high, medium, low. As the result of the agreement, Spain's Government will accept the *high* eSeals from X like qualified eSeals, and the *medium* eSeals from X like advanced eSeals. In the same way, there could be a company Y, with qualified and not-qualified eSeals, signing an agreement with Spain's Government, and stating the qualified eSeals of Y are equivalent to the advance eSeals of Spanish eIDAS eSeals. Besides, in this example we are supposing that each agreement is valid in both directions[1]: the company X will accept the qualified eSeal of that government as its high eSeal, etc. Such hypothetical agreements are shown in the following tables:

Table 1: A hypothetical agreement on trust levels of electronic seals between Spain's Government and X Co.

Spain's Government (eIDAS)	X Company (X Co.)
eSeal level	**eSeal level**
Qualified	High
Advanced	Medium
	Low

[1] That double direction has to be specified in the bilateral agreement or in another one.

Table 2: A hypothetical agreement on trust levels of electronic seals between Spain's Government and Y Co.

Spain's Government (eIDAS)	Y Company (Y Co.)
eSeal level	**eSeal level**
Qualified	
Advanced	Qualified
	Not-qualified

Note, please, that since there is no agreement between the companies X and Y, there is no relation between their trust levels of eSeals (i.e. the relation of trust translation is *not* an *equivalence relation mathematically speaking*.)

The TTA will provide the list of equivalences for the advanced eSeal and for the qualified eSeal in the Spanish eIDAS scheme. The list of the advanced eIDAS eSeal will contain two elements: the medium X Co. eSeal, and the qualified Y Co. eSeal. The list of the qualified eSeal will contain just only the high X Co. eSeal like a recognized level of trust. Also, the TTA will publish the list of equivalences for the high eSeal and for the medium eSeal in the company X. Besides, there will be available a list of translations for the qualified Y Co. eSeal.

The above scenario of an ordinal trust scheme (Spanish eIDAS eSeal) and how its equivalences to an ordinal trust scheme (X Co. eSeal) and a boolean trust scheme (Y Co. eSeal) are published by the TTA is depicted below:

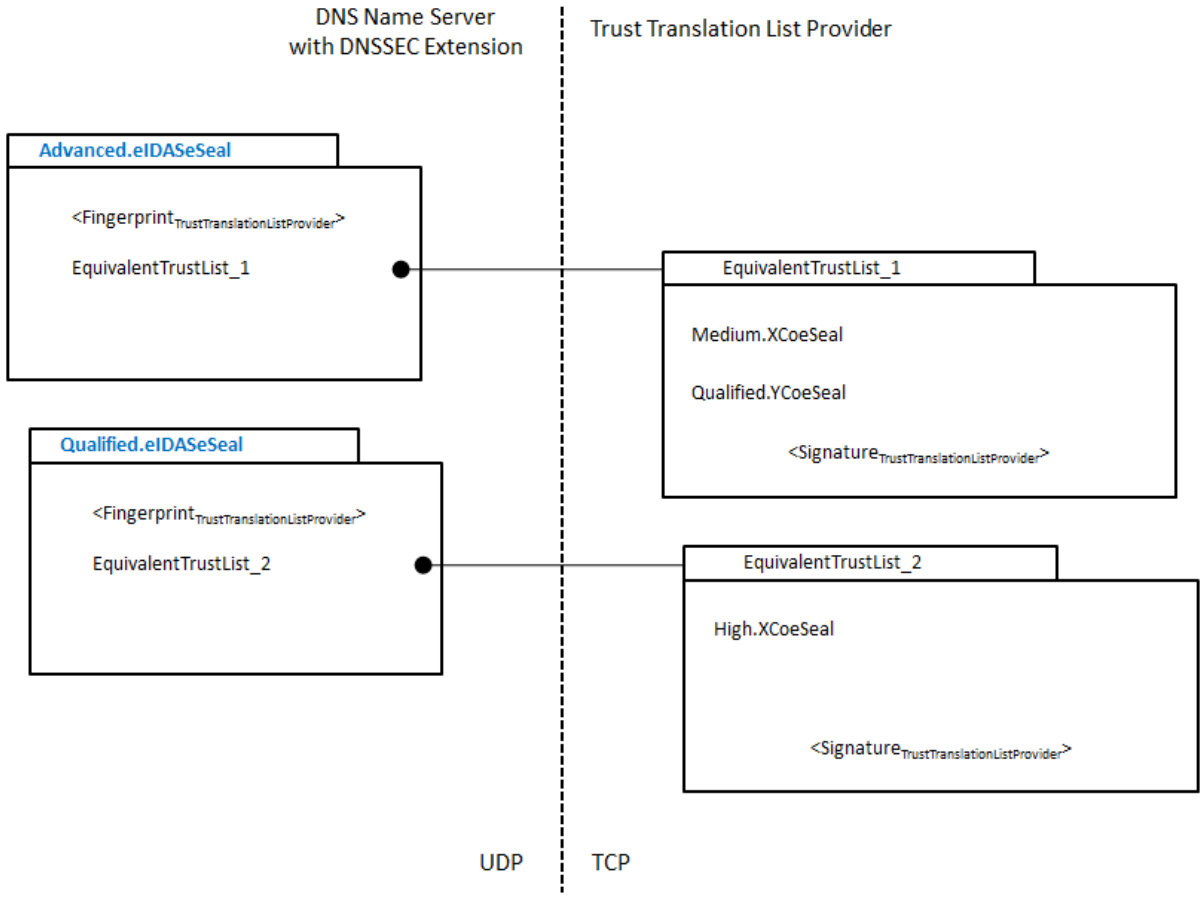

Figure 2: Samples of ordinal and boolean trust schemes being translated (one direction)

The way of reading the above figure is from left to right (as we explain in D3.1 and D4.1.): UDP protocol first; then, TCP protocol. The translations provided by the TTA are always settled by the bilateral agreement between one organization and another. The translation data of the TTA will never be deduced. However, if a given company using LIGHTest, wants to define a trust policy using the trust relation like a real equivalence relation among other companies or organisations, it is perfectly possible!

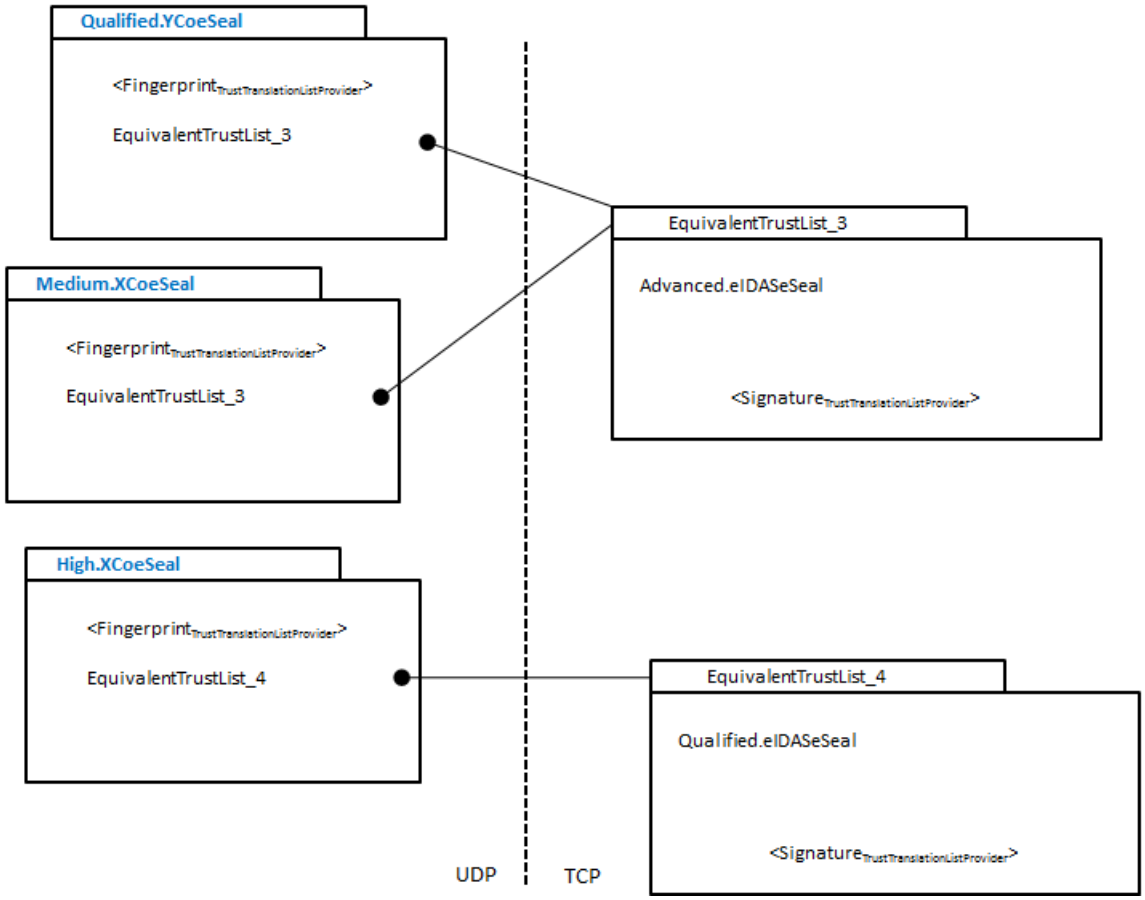

Figure 3: Samples of ordinal and boolean trust schemes being translated (the opposite direction)

Note in the above figures that here is only shown the list of the equivalences between trust levels in different trust schemes. In the TTA data model (see next section) we will see there are *conditions* under which those translation lists are valid or not. The conditions are related to the properties of the bilateral agreement itself (duration, status, etc.) The corresponding data model is shown in the next section.

We can express the former translations using the TPL (see [7]):

```
translate_seal(X_CO,EIDAS) :-
    attrib(EIDAS,schemename,eidas),
    attrib(X_CO, schemename,x_co),
    translate_adv(EIDAS,X_CO).

translate_seal(Y_CO,EIDAS) :-
    attrib(EIDAS,schemename,eidas),
    attrib(Y_CO, schemename,y_co),
    translate_adv(EIDAS,Y_CO).

translate_seal(X_CO,EIDAS) :-
    attrib(EIDAS,schemename,eidas),
    attrib(X_CO, schemename,x_co),
    translate_qual(EIDAS,X_CO).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Translations of Attribute                                  %%%
%%% eSeal: advanced and qualifed                               %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

translate_adv(EIDAS,X_CO) :-
    attrib(EIDAS,eSeal_level, advanced),
    attrib(X_CO, eSeal_level, medium).

translate_adv(EIDAS,Y_CO) :-
    attrib(EIDAS,eSeal_level, advanced),
    attrib(Y_CO, eSeal_level, qualified).

translate_qual(EIDAS,X_CO) :-
    attrib(EIDAS,eSeal_level, qualified),
    attrib(X_CO, eSeal_level, high).
```

Figure 4: TPL expressing the hypothetical translations from eIDAS to companies X and Y electronic seals

Tuple-based use case:

The second use case to be shown is related with a **tuple-based trust scheme**. For that, let's take FIDO authenticators and the equivalences between different versions of them, according to a *hypothetical* statement delivered by the FIDO Alliance for its Metadata Service. In such statement, some FIDO UAF authenticators version 1.5 are equivalent to some older FIDO U2F authenticators whenever the attestation process takes place. Also, the FIDO alliance would accept the attestation of the eID cards of the eIDAS infrastructure.

FIDO authenticators are described by a tuple-based trust scheme by means of these attributes:

- Authenticator Version
- Authenticator ID (AAID)
- Attestation Certificate
- User verification method such as fingerprint biometrics
- Whether keys are protected by Trusted Execution Environment (TEE) or Secure Element (SE)
- Whether biometrics are protected by TEE

For our example, let's consider that modern FIDO UAF authenticator with the following features:

- Authenticator Version: 1.5
- Authenticator ID (AAID): "13456"
- Attestation Certificate: "zakdjdgk458496982"
- User verification method such as fingerprint biometrics: "myVerif"
- Whether keys are protected by Trusted Execution Environment (TEE) or Secure Element (SE): "TEE"
- Whether biometrics are protected by TEE: true

And the other FIDO U2F authenticator, not so new, features:

- Authenticator Version: 1.1
- Authenticator ID (AAID): "09876"
- Attestation Certificate: "jdlwoqkvjje82"
- User verification method such as fingerprint biometrics: "anotherVerif"
- Whether keys are protected by Trusted Execution Environment (TEE) or Secure Element (SE): "SE"
- Whether biometrics are protected by TEE: true

Relating to the eID card emitted and recognized in a given eIDAS infrastructure (note that this trust scheme is ordinal –high, substantial, low --, not tuple-based), we are going to suppose the FIDO Alliance and the eIDAS government agreed that such identification documents with High level are valid for the attestation process. However the government will not allow the use of that FIDO authenticator (so, the agreement is only in one direction).

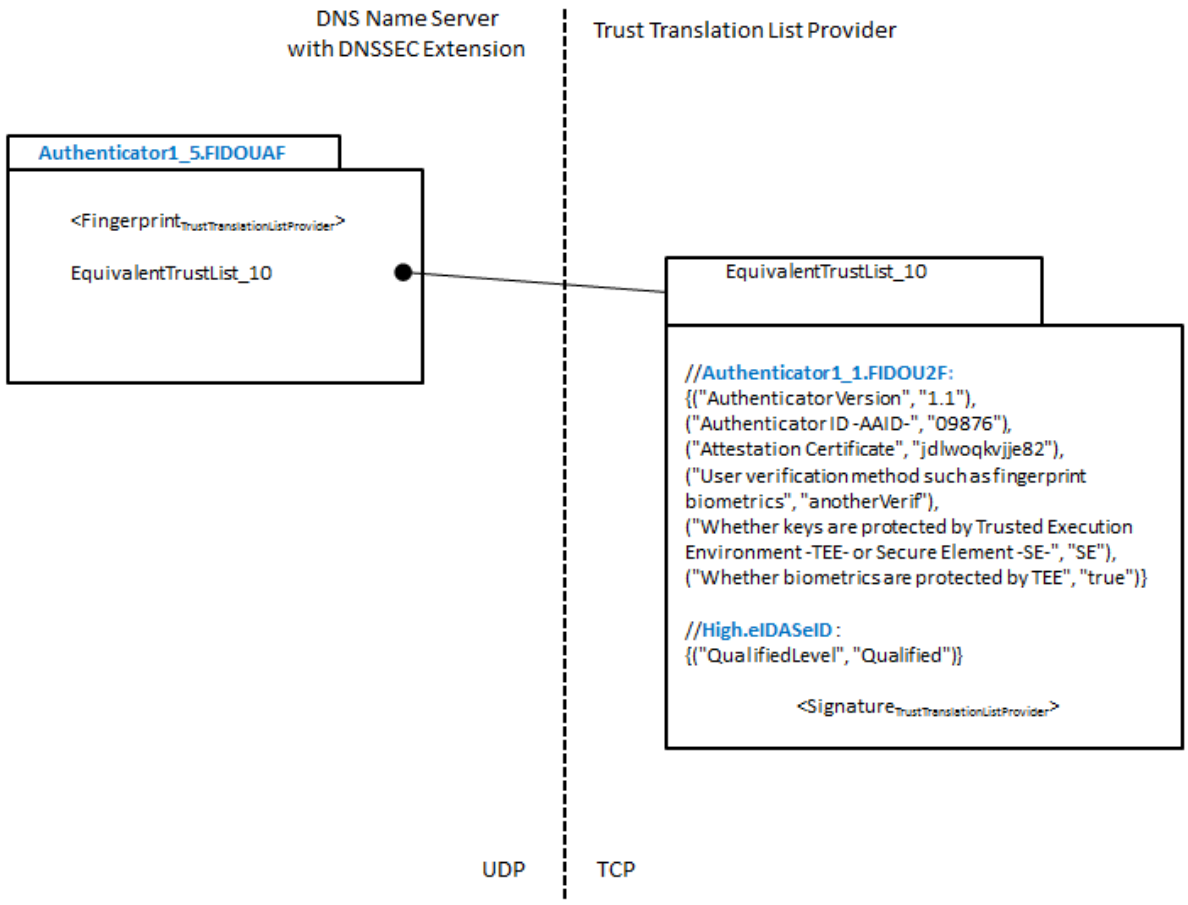

Figure 5: Sample of tuple-based trust scheme being translated

In the Figure 5, the equivalence list of an imaginary level of trust of a FIDO UAF authenticator (tuple-based scheme) is shown. Two equivalent levels of trust schemes *represented* as tuples are listed: another FIDO authenticator level, and an eIDAS level.

Using the TPL again, the above translation would be as follows:

```
translate_identity(EIDAS,FIDO_UAF_1_5) :-
    attrib(EIDAS,    schemename,eidas),
    attrib(FIDO_UAF,schemename,fido_uaf_1_5),
    translate_qual(EIDAS,FIDO_UAF_1_5).

translate_identity(FIDO_U2F_1_1,FIDO_UAF_1_5) :-
    attrib(FIDO_UAF,     schemename,fido_uaf_1_5),
    attrib(FIDO_U2F_1_1,schemename,fido_u2f_1_1),
    translate_qual(FIDO_U2F_1_1,FIDO_UAF_1_5).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Translations of Attribute                        %%%
%%% eIdentity qualification                          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

translate_qual(EIDAS,FIDO_UAF_1_5) :-
    attrib(EIDAS,    eIdentity_level,qualified),
    attrib(FIDO_UAF_1_5, authenticator_version, "1.5"),
    attrib(FIDO_UAF_1_5, aaid, "13456"),
    attrib(FIDO_UAF_1_5, attestation_certificate, "zakdjdgk458496982"),
    attrib(FIDO_UAF_1_5, user_verification_method, "myVerif"),
    attrib(FIDO_UAF_1_5, keys_protected_by_TEE_SE, "TEE"),
    attrib(FIDO_UAF_1_5, biometrics_protected_by_TEE, true).

translate_qual(FIDO_U2F_1_1,FIDO_UAF_1_5) :=
    attrib(FIDO_U2F_1_1, authenticator_Version, "1.5"),
    attrib(FIDO_U2F_1_1, aaid, "09876"),
    attrib(FIDO_U2F_1_1, attestation_certificate, "jdlwoqkvjje82"),
    attrib(FIDO_U2F_1_1, user_verification_method, "anotherVerif"),
    attrib(FIDO_U2F_1_1, keys_protected_by_TEE_SE, "SE"),
    attrib(FIDO_U2F_1_1, biometrics_protected_by_TEE, true),

    attrib(FIDO_UAF_1_5,authentication_version,"1.5"),
    attrib(FIDO_UAF_1_5, aaid, "13456"),
    attrib(FIDO_UAF_1_5, attestation_certificate, "zakdjdgk458496982"),
    attrib(FIDO_UAF_1_5, user_verification_method, "myVerif"),
    attrib(FIDO_UAF_1_5, keys_protected_by_TEE_SE, "TEE"),
    attrib(FIDO_UAF_1_5, biometrics_protected_by_TEE, "true").
```

Figure 6: TPL specifying a hypothetical translation for the FIDO authenticator version 1.1

Again, no agreements' conditions over the above equivalences are specified for simplicity. We will see the whole data model in the next section.

7.3. Description of data model

(Following the description done in previous D4.1 [8], and looking for a hint in D3.1 [9]&D5.1 [10] data model description)

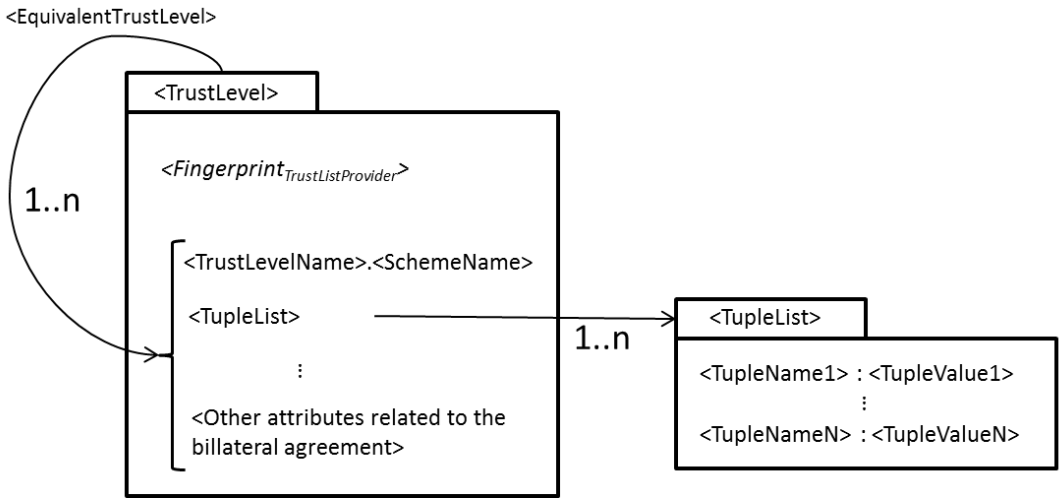

Figure 7: Conceptual View of the Data Model

The TTA is managing every trust level like a trust level in a tuple-based trust scheme, since every trust scheme can be expressed as a tuple-based trust scheme. In this way the translation process is more ideal.

However, the TSPA is not publishing its trust schemes in the same way than the TTA manages its trust level translations since boolean, ordinal and tuple-based trust schemes are distinguished by the TSPA. Thus, a correspondence between trust levels by the TTA and trust levels by the TSPA has to be defined. Such relationship will be established within the TTA.

The following table tries to show how the previous examples of translation can be stored in the TTA module. We have decided to represent all the trust levels like *tuple-based*, the most generic representation:

Table 3: Translation data in the TTA module

TTA_ID	TRUSTLEVELNAME_ID	TUPLELIST (*)	LIST OF EQUIVALENT TTA_ID (**)
1	Advanced.eIDASeSeal	{("AdvancedLevel", "Advanced")}	4, 5, 9
2	Qualified.eIDASeSeal	{("QualifiedLevel", "Qualified")}	10, 101, 203
…			
7	Qualified.eIDASeTimestamp	{("QualifiedLevel", "true")}	22
…			

10	High.eIDASeID	{("HighLevel", "High")}	
…			
12	Authenticator1_1.FIDOUAF	{("Authenticator Version", "1.1"), ("Authenticator ID -AAID-", "13456"), ("Attestation Certificate", "zakdjdgk458496982"), ("User verification method such as fingerprint biometrics", "myVerif"), ("Whether keys are protected by Trusted Execution Environment - TEE- or Secure Element -SE-", "TEE"), ("Whether biometrics are protected by TEE", "true") }	14
13	Authenticator2_1.FIDOUAF	…	15
14	Authenticator1_5.FIDOU2F	{("Authenticator Version", "1.5"), ("Authenticator ID -AAID-", "09876"), ("Attestation Certificate", "jdlwoqkvjje82"), ("User verification method such as fingerprint biometrics", "anotherVerif"), ("Whether keys are protected by Trusted Execution Environment - TEE- or Secure Element -SE-", "SE"), ("Whether biometrics are protected by TEE", "true") }	10, 12
15	Authenticator2_5.FIDOU2F	…	13

(*) The levels of trust of a given trust scheme are unique within the LIGHTest framework: although there are two hierarchies stored in the LIGHTest DNS server, one for the TTA and one for the TSPA, the names of the trust levels in a trust scheme are uniquely defined.

Note the trust levels managed and returned by the TTA *have to exist* in the TSPA namespace.

(**) Conditioned by related bilateral agreements.

The data model designed for the TTA module is depicted in the Figure 8:

Trust Level	Trust Level Name			
	Trust Scheme Kind			
	Tuple-based representation	{Attribute name, Attribute value}		
	Trust Level Equivalence List	{Trust Level Name, Agreement}	Agreement	Trusted Trust Scheme
				Recognized Trust Scheme
				Unique/Double direction
				Agreement date
				Duration
				Status
				Conditions: what kind of conditions there could be
				Activation date
				Entry date (?)
				Leaving date (?)
	Fingerprint			

Figure 8: TTA data model

The field contents are explained in Table 4:

Table 4: Data fields in the TTA model

Field	Mandatory	Format	Description	Value
Trust Level Name	yes	String	It specifies the DNS* entry to the TTA module. Its meaning is a given trust level in its corresponding trust scheme.	"trustLevel_ trustScheme"
Trust Scheme Kind	yes	String	Although every trust scheme can be expressed like a tuple-based format, we need to store the original nature of the trust scheme, in order to return the equivalent list in boolean/ordinal trust levels whenever it is required.	One of the following values: “boolean”, “ordinal”, “tuple”.

Tuple-based representation	yes	List of string pairs (attribute name, attribute value)	Every trust level in a trust scheme is doable of being expressed in tuple-based format.
			Thus, boolean and ordinal trust levels will be expressed as a list of tuples.
				{
				(“attribute1", "att1Value"),
				("attribute2", "att2Value"), …
				("attributeN", "attNValue")
				}

Trust Level Equivalence list	yes		It is a list of equivalent trust levels with their correspondent agreements. A trust level is published by the TTA if and only if a trust level equivalence list exists, although it has one element, one translation. The equivalent trust level is the trust level name published by the **TSPA** module. Each equivalent trust level has also attached a pointer to the related agreement between the trust schemes (the trusted one, and the recognized one).	

Field	Mandatory	Format	Description	Value
Agreement	yes	Data structure	Key information about the bilateral agreement signed between two trust schemes in relation to make equivalents some trust levels of each trust scheme.	See below the related fields.
A. TrustedTrustScheme	yes	String	It specifies the DNS* entry to the TSPA module. It references to the source trust scheme (the trusted one).	"trustScheme"
A. RecognizedTrustScheme	yes	String	It specifies the DNS* entry to the TSPA module. It references to the target trust scheme (the recognized one).	"trustScheme"
A.DoubleOrUnique	yes	Boolean	If true, the agreement is applied in both directions. If false, the agreement is only valid in one direction.	true/false
A.AgreementDate	yes	Date	The date of the signature of the agreement.	A date value: dd/mm/yyyy
A.Duration		Numeric	The duration of the agreement.	Number of days, months, years??
A.Status	TBD	String	The state of the agreement	Values To Be Defined
A.Conditions (?)	?	?	Any condition to be taken into account?	
A.ActivationDate	yes	Date	Date of activation of this agreement within the LIGHTest framework.	date value: dd/mm/yyyy
A.EntryDate	yes	Date	Date of entry into the LIGHTest framework.	date value: dd/mm/yyyy
A.LeavingDate	yes	Date	Date of leaving from the LIGHTest framework.	date value: dd/mm/yyyy

Field	Mandatory	Format	Description	Value
Fingerprint	yes	String	The signature of the TTA module?	The string of the signature.

7.4. Functional architecture

As explained before, the Trust Translation Authority consist of two functions; feeding and adapting data (performed by the Trust Translation Provider), and discovery and publishing, the following is a brief description of the functional architecture to attend these functions.

Attending to the Architectural principles described in D2.14 [4] about barriers minimisation and reuse of existing technologies, an off-the-shelf DNS Name Server with DNSSEC/DANE extensions will be used to perform the Discovery service of the TTA. This DNS will present a structure of domain names such that, for debugging purposes, also let to hierarchically navigate from a trust scheme name, in a descending manner, to its services and get its corresponding levels in order to obtain the equivalent levels in other trust schemes (see example in section 7.2 of this document).

Regarding the data TTA has to publish, which is a List of equivalent Trust Scheme Levels, there is no bound to the number of Trust Scheme Levels resulting of the translation and hence there is no limit to the extension in bytes needed. This is against DNS capabilities, a DNS can manage messages of 512 bytes or 1280 bytes as a maximum. Therefore, rather than the data themselves, the DNS will provide pointers to the data that will be hosted by a different service. These pointers will be expressed as Universal Resource Identifiers (RFC 7553). See D2.7 [5], section 8.3[i].

To cope with the publication function the TTA counts with two different services; a service to provide an interface to the person (authority) in charge of maintaining the Trust Translation data, the **Management API**, and other service to manage the data and make it publicly available, the **Data Management Service**.

The Management API will provide the required functions to add, modify and remove Trust Translation data while ensuring the authenticity of the data provided by the authority. Such authority would correspond to an administrator role in the LIGHTest framework.

The Data Manager Service will transform the Trust Translation Data into the appropriate formats and will make it publicly available. This consists in building the Trust Translation List according to the information provided by the authority and writing these lists into files (xml files). Then the data manager service will update the structure of domains and pointers in the DNS name server so that ATV can discover these files.

These two services are integrated in the functional block named Trust Translation Provider (D4.1 [1]) and therefore deployed as a single module.

Whereas the DNS Name Server is unique, the TTA can deploy as many Trust Translation Provider instances as needed in order to fulfil the needs of different organizations. For instance

two countries should be able of managing their own Trust Agreements and so to manage the Trust Translation Data published.

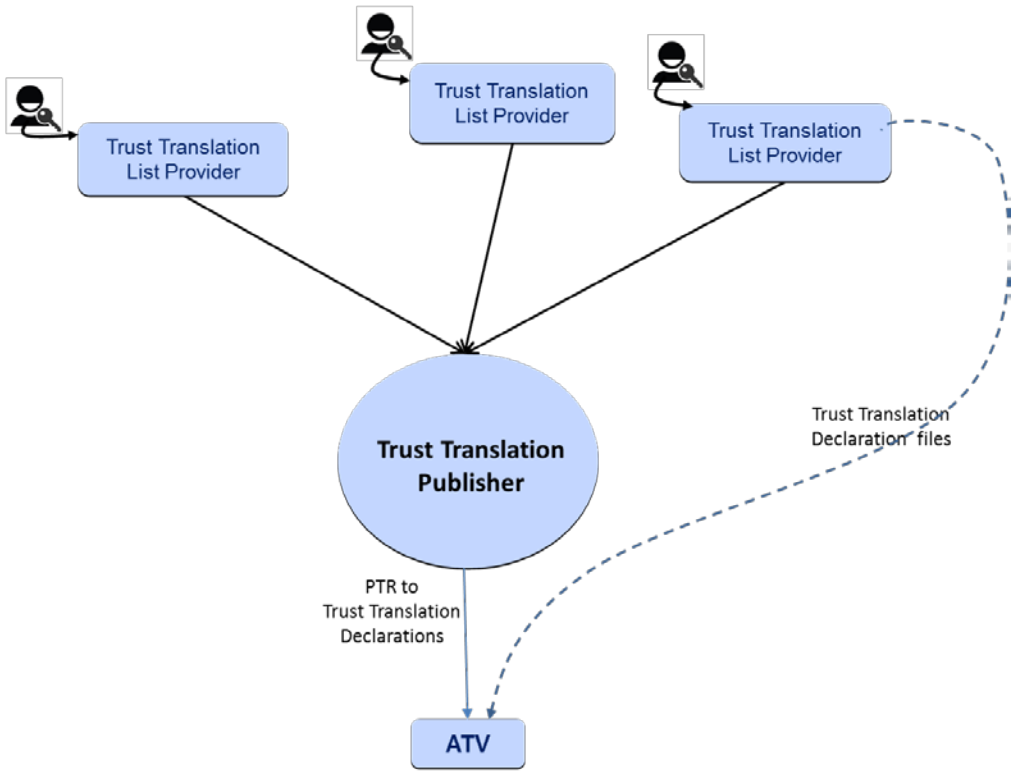

Figure 9: Functional Architecture diagram

8. Use cases publication of mobile ID for trust translation

In order to illustrate the concept of trust translation with a real-world use case, an example of trust translation for the mobile ID use case as defined and developed in WP 7 shall be discussed here.

As was already discussed in deliverable D 7.1 [11], the derived mobile ID landscape can become quite complex if the issuer of the primary ID and the secondary ID issuer of the (derived) mobile ID operate in different trust schemes. Theoretically, the relying party that consumes the mobile ID could also be located in a third trust scheme. In most practical implementations however this will be very unlikely and it will therefore not be discussed here. Technically, the concept of trust translation between the primary and secondary ID issuer simply needs to be mirrored between the secondary ID issuer and the relying party trust schemes. For convenience, the relationship of parties in the mobile ID scheme is shown again in Figure 10.

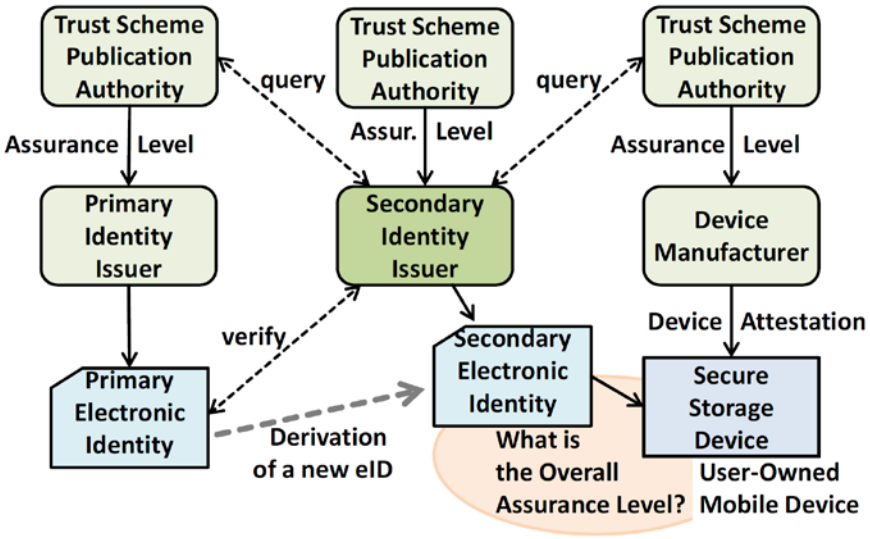

Figure 10: Relation of trust schemes in the case of derived mobile IDs. (Source: DoW)

In a concrete example, a U.S. citizen with a primary ID issued under a U.S. trust scheme (e.g. PIV or NIST, see D 4.1 [1]) could register for a European e-government service that is subject to the eIDAS trust scheme. For registration, the citizen needs to provide an identification with a certain eIDAS trust level (e.g. at least “substantial”). The e-government service uses the FIDO-based mobile ID scheme, as outlined in deliverable D 7.2 [12]. The example is illustrated in Figure 11.

Within this scheme, the citizen needs to register a user device as FIDO authenticator, thus creating a public/private key pair. After registration the citizen will be forwarded to an ID provider (secondary ID issuer, see Figure 10) that cooperates with the e-government service. In this example, the ID provider is also operating under the eIDAS trust scheme.

The U.S. citizen now needs to identify towards the ID provider with a U.S. issued ID card (e.g. a PIV card). Since the secondary ID issuer will finally need to issue a certificate over the FIDO public key, it does not only have to verify the validity of the U.S. issued identity but also needs to query the translation of trust levels from the PIV assurance levels to the eIDAS levels. For this query, the ID provider can use the methodology and infrastructure as described in chapters 6 and 7.

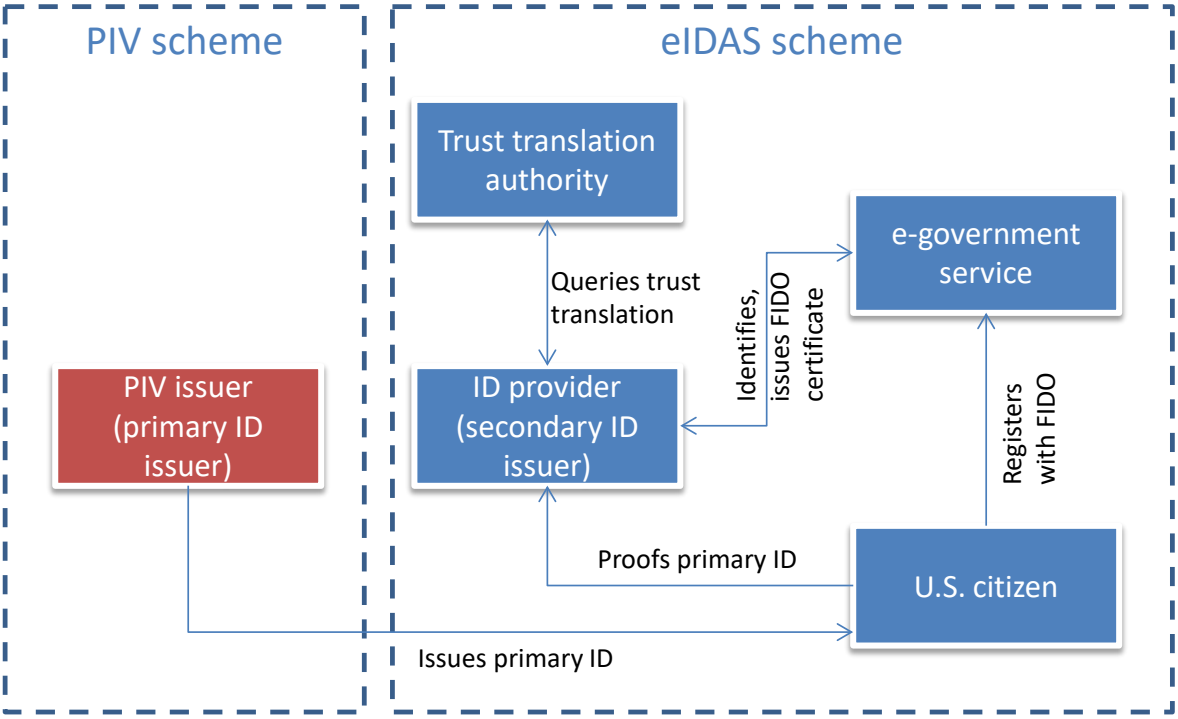

Figure 11: Relation of acting entities and trust schemes for the example of trust translation in the LIGHTest mobile ID scheme.

After querying and translating the U.S. trust levels into the eIDAS levels the ID provider issues a certificate over the FIDO public key and optionally also performs a proof-of-possession of the FIDO private key. The certificate is then exported to the e-government service which can verify its validity and can store the certificate in the FIDO server for verification of future authentications.

Depending on the requirements of the e-government service it may even be required to query several IDs from different trust schemes. In this example, the e-government service could require the proof of name and date of birth of the U.S. citizen and a proof of possession of a professional degree or qualification. While the former two attributes can be derived from the U.S. issued ID, the proof of professional qualification could require to verify a certificate issued under a different trust scheme, for example if the U.S. citizen has received the professional degree in Canada. If the e-government service also requires a proof of citizenship of the European city for which it operates, even a third query might be required, thus in this case without trust translation.

In summary, the methodology developed in chapters 6 and 7 is a flexible tool to realize even complex mobile ID use cases across several trust schemes.

9. References

[1] “LIGHTest. D4.1 Conceptual Framework for Trust Scheme Translation (1),” 2017.

[2] “LIGHTest. D3.3 DNS-based Publication of Trust Schemes,” 2018.

[3] “LIGHTest. D5.3 Publication of Delegations,” 2018.

[4] “LIGHTest. D2.14 Architecture and Technical Coordination,” 2017.

[5] “LIGHTest. D2.7 DNSSEC Expertise and Building Blocks,” 2017.

[6] “ http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG,” 2014.

[7] “LIGHTest. D2.4 Formal Description and Analysis of Concepts,” 2017.

[8] “LIGHTest. D4.1 Conceptual Framework for Trust Scheme Translation (1),” 2017.

[9] “LIGHTest. D3.1 Conceptual Framework for Trust Schemes,” 2017.

[10] “LIGHTest. D5.1 Conceptual Framework for Delegations,” 2017.

[11] “LIGHTest. D7.1 Definition of Requirements for derivation and attestation,” 2017.

[12] “LIGHTest. D7.2 Definition of Device System Architecture and Derivation Scheme of mobile IDs,” 2017.

10. 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 Globalsign (FI).
The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.