# D3.3

## DNS-based Publication of Trust Schemes

| Document Identification | |
|---|---|
| **Date** | 27.02.2018 |
| **Status** | Final |
| **Version** | Version 1.0 |

| | | | |
|---|---|---|---|
| **Related WP** | WP 3 | **Related Deliverable(s)** | D3.1 |
| **Lead Authors** | FHG | **Dissemination Level** | PU |
| **Lead Participants** | FHG | **Contributors** | USTUTT, NLNET, TUG, G&D, GS, IBM |
| **Reviewers** | TIL, ATOS | | |

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

# 1. Executive Summary

This document provides the design of the DNS-based Publication of Trust Schemes in LIGHTest. The design of the conceptual description of trust schemes was already developed in the deliverable D3.1. This deliverable describes now the publication of Trust Schemes using the existing data elements and infrastructure of the DNS system.

The publication of trust schemes is only one part of publishing trust-related information within LIGHTest. Due the fact, that also Trust Translation and Trust Delegation require the publication of trust-related information, a consolidated approach to publishing trust-related information in the DNS in general is developed together with WP4 and WP5.  The consolidated approach is also presented in the corresponding deliverables D4.3 and D5.3. In each of the deliverables, the publication of trust-related information is specified accordingly for either the publication of Trust Schemes, or Trust Translation Lists, or Trust Delegations.

In this deliverable, the consolidated approach is specified for Trust Scheme memberships. For the discovery and verification of Trust Scheme memberships, the Trust Scheme Publication Authority (TSPA) is responsible (see D3.1). The concept of the TSPA consists of two components, a DNS Name Server with DNSSEC extension and a Trust Scheme Provider. Therefore, the publication of Trust Schemes requires additions in the DNS resource records (e.g. Pointers and URIs) as well as the publication of a signed Trust List on the Trust Scheme Provider side. The trust list should use or be conforming to existing standards, which involve a trust list and a trust list provider (e.g. ETSI TS 119 612).

In addition to the development of the consolidated approach to publishing trust-related information in the DNS for trust scheme membership, this deliverable demonstrates its applicability by showing several examples of usage for selected trust schemes. These are eIDAS, the Pan Canadian Trust Framework, STORK, the Electronic Signature Law of the People’s Republic of China, and FIDO. These examples demonstrate the required additions in the DNS resource records as well as examples for the publication of a signed Trust Lists. Hereby the publication of trust schemes is described for both already existing (e.g. eIDAS) as well as fictional (e.g. Pan Canadian Trust Framework, FIDO, STORK, Electronic Signature Law of the People’s Republic of China) signed Trust Lists. Hence, with this approach the publication of already existing as well as newly developed Trust Schemes in the LIGHTest infrastructure can be realized.

# 2. Document Information

## 2.1 Contributors

| Name | Partner |
|---|---|
| Heiko Roßnagel | FHG |
| Sven Wagner | USTUTT |
| Martin Hoffmann | NLNET |
| Olamide Omolola | TUG |
| Frank-Michael Kamm | G&D |
| Jesse Kurtto | GS |
| David Hixon | IBM |

## 2.2 History

| Version | Date | Author | Changes |
|---|---|---|---|
| V0.1 | 23.11.2017 | FHG,<br>USTUTT | Initial draft,<br>Table of Content |
| V0.2 | 10.01.2018 | FHG,<br>USTUTT | Scope, eIDAS demonstration added |
| V0.3 | 26.01.2018 | NLNET | Consolidated Approach |
| V0.4 | | <br>TUG<br>GS<br>IBM<br>G&D | Demonstration for<br>Electr. Sign. Law China<br>PCTF<br>STORK<br>FIDO |
| V0.5 | 09.02.2018 | FHG,<br>USTUTT | Compilation of final draft |
| V1.0 | 27.02.2018 | FHG | Integrating review suggestions, formatting changes |

# 3. Table of Contents

## 3.1 Table of Figures

## 3.2 Table of Tables

n.a.

## 3.3 Table of Acronyms

| | |
|---|---|
| ATV | Automatic Trust Verifier |
| DANE | DNS-based Authentication of Named Entities |
| 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 |
| LoA | Level of Assurance |
| PCTF | Pan Canadian Trust Framework |
| TLS | Transport Layer Security |
| TSPA | Trust Scheme Publication Authority |
| URI | Uniform Resource Identifier |

# 4. Scope of the Deliverable

This deliverable D3.3 describes the design of the publication of Trust Schemes using the existing data elements and infrastructure of the DNS system. The publication design is based on the conceptual description of trust schemes in D3.1. In D3.1, a concept for trust scheme publication was elaborated, which consists of two components: a DNS Name Server with DNSSEC extension and a Trust Scheme Provider, which provides a signed Trust List indicating that a Certificate Issuer is trusted under the scheme operated by the Trust Scheme Provider. The verification of the claims requires in addition fingerprints of the issuer certificate and of the trust scheme provider certificate in the corresponding DNS records.

Within LIGHTest, a publication of trust-related issues is required not only for Trust Schemes but also for Trust Translation Lists and Trust Delegation. Therefore, a consolidated approach to Trust-related Publications in general is developed together with WP4 and WP5, which can be specified for either the publication of Trust Schemes, or Trust Translation Lists or Trust Delegations.

Therefore, the deliverable first introduces this consolidated approach to publishing trust-related information in the DNS in Section 5 by specifying trust declarations and describing the publication, discovering, and authentication of trust declarations. This Section 5 is almost identical to Section 6 of D4.3 and Section 8 of D5.3. That leads to some duplication in the deliverables, however it provides for the readers standalone and complete documents.

In Section 6, the Trust Scheme Membership Publication is presented. This includes also a brief summary of the conceptual description of trust schemes in D3.1 with a special focus on the issues relevant to Trust Scheme Membership Publication (Section 6.1).

The applicability of the consolidated approach to publishing trust-related information in the DNS for trust scheme membership is demonstrated in Section7. In that section, by way of example, the framework is applied for selected trust schemes, which are described already in D3.1. These are eIDAS (Section7.1), the Pan Canadian Trust Framework (Section7.2), STORK QAA/AQAA (Section7.3), the Electronic Signature Law of the People’s Republic of China (Section7.4), and FIDO (Section7.5). In addition, examples for ordinal Trust Schemes (Section7.6) and the concept for tuple based Trust Schemes (Section7.7) is presented.

# 5. A Consolidated Approach to 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. This section looks at the basic properties these assurances exhibit independently of their concrete contents and introduces an underlying, fundamental framework.

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 extends 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.

## 5.1 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 a 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 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.

## 5.2 Publication of Trust Declarations

When using trust declarations for verifying an electronic transaction, a validator needs to construct a chain of trust declarations leading from any of the trusted entities to those entities appearing in the transaction. It does so by recursively finding and adding applicable trust declarations that have an originator that is either a trusted entity or the target of the declaration is already part of the chain. In order to do this in an unaided, automatic way, the validator needs a way to gain access to all declarations that are potentially usable in this process.

There are two fundamental strategies for the verifier to find declarations when they are needed: 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 or if the sender of the transaction doesn’t know the trusted entities the receiver will base their verification on, 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. The verifier can then decide itself which declarations it needs and try and find them as needed.

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 exists 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.

### 5.3 Discovering Trust Declarations

In order to build a chain of trust from published declarations, the verifier needs to be able to discover their existence. Given a transaction and a set of already trusted entities, this chain can be built from two sides: either the verifier starts with a trusted entity, tries to discover all the declarations that have this entity as their source, and repeats this process until it arrives at a declaration that includes the transactions as its target or runs out of declarations to apply. Alternatively, it can start with the transaction, attempts to discover all declarations that have the transaction as their target, and continues by recursively trying to find declarations that have the sources of already discovered declarations as their target until it arrives at an already trusted entity as a declaration's source or, again, runs out of declarations.

In both cases, the verifier needs a mechanism to search for the URI of a declaration based on a given entity. Such a mechanism will have to take some information that identifies that entity as input. Given that entities are typically identified by X.509 certificates, it should be possible to include such identifying information in the entity's certificate. Once of the possible options is to use a domain name as the entity's identifier. This has the advantage that the name can be used directly as a search input for the DNS, allowing to use the DNS as a global, highly available, distributed, and independently managed data store.

A domain name can be stored in a certificate either in the subject alternative name or issuer alternative name extensions. The subject alternative name indicates the domain name identifying the entity using the certificate while the issuer alternative name identifies the entity having issued the certificate.

Using this domain name as input, the URIs to the declarations for differing aspects should be stored in the DNS. As the DNS uses record types to distinguish between different types of information stored for a domain name, one option is to register an individual record type for each aspect of trust declarations. However, this would clutter the space of types. As the data stored is the same in every case – a URI – an alternative approach can be used whereby the aspect is encoded as a prefix to the domain name. This is already used for instance with the SRV record type for discovering the host name and port where a certain networking service is available for a domain. As such services sometimes are available over different transport protocols, a two-layer prefix of the form *_service._protocol* is used where the latter describes the transport protocol to be used and the former the networking service.

In keeping with this concept, LIGHTest proposes to use the DNS for providing pointers to all kinds of declarations via a pair of prefixes of the form *_aspect._application*. Here, the type of application for which a declaration is published is the second part of the prefix while the first part defines the particular aspect within that application. For LIGHTest itself, the application is, of course, trust-related declarations, which is identified via using the literal label *_trust* as the second part. Each of the aspects of trust declarations defines its own label to be used as the first part.

Under the name constructed by concatenating the prefix with the entity's own identifying domain name, the entity can now publish pointers to trust declarations of the corresponding aspect that relate to it. It can use the already existing URI resource record type for this purpose. This type is defined in section 4 of RFC 7553 [1]. Its record data contains exactly one URI. While section 5 of the RFC describes a different use of the record, this use is limited by a different set of prefixes, allowing its reuse for declaration publication based on the prefixes defined above.

If the entity in question is not the originator of a declaration it may not control the URI under which the declaration is published. In this case, it may be beneficial to only point to the originating entity rather than burden itself with tracking whether the originators URI has changed. This can be done easily if the originator is an entity identified by a domain name, too. In this case, instead of publishing URI resource records under the its domain name prefixed by the declaration aspect, the entity will publish PTR resource records. Such record types, part of the original DNS specification in RFC 1035 [2], contain another domain name as their record data. If such records are present, they instruct the verifier to continue discovery for declaration at the entity identified by these domain names. Note that as these names in the PTR record's data identify entities, they, too, need to be prefixed with the correct declaration prefix before continuing querying.

## 5.4 Authenticity of Trust Declarations

If a verifier retrieves a declaration from somewhere in the network, it needs to make sure that the data it received is indeed the declaration made by the originator. Since the URI resource records are stored under the domain name identifying the originator, it is reasonable to assume they are authentic if DNSSEC validation succeeds, guaranteeing that the URI is indeed the one intended by the originator.

In the next step, where the verifier contacts the server indicated in the URI, it needs to ensure that it communicates with the correct server. When using encrypted transport via the TLS protocol, the server will identify itself via a certificate. In order to deal with shortcomings of the traditional method of certificate verification, a protocol called DNS-based Authentication of Named Entities or DANE for short allows a server operator to publish information about the certificates used in DNS.

Since, however, the server is not necessarily operated under authority of the originator of the declaration – for instance because the declarations are hosted by a third party that provides better availability, this does not guarantee that the declaration received is indeed the one that the originator intended.

This final link can be provided if the declaration itself is a signed document. The originator can then publish the certificates that it uses for signing declarations of a certain aspect using a slightly adapted version of the DANE protocol.

To do so, it adds SMIME resource records under the same domain name it placed the URI records pointing to the declarations. These records define conditions a certificate has to fulfill to be accepted. By placing these records, the originator declares that all documents retrieved via the pointers have to verify considering these conditions.

# 6. Trust Scheme Membership Publication

## 6.1 Concept for Trust Scheme Publication

The concept of Trust Scheme Publication was already developed in D3.1 [3]. This section provides a brief summary of this concept with a special focus on the issues relevant to Trust Scheme Membership Publication.

The role of the Trust Scheme Publication Authority (TSPA) in LIGHTest is to enable discovery and verification of trust scheme memberships. Therefore, the concept of the TSPA consists of two components: a DNS Name Server with DNSSEC extension and a Trust Scheme Provider. The DNS Name Server enables discovery of the Trust Scheme Provider that operates a Trust Scheme. The Trust Scheme Provider provides a signed Trust List, which indicates that a certificate Issuer is trusted under the scheme operated by the Trust Scheme Provider. In addition, it provides the Tuple-Based representation of a Trust Scheme. This is depicted in Figure 1, which is an updated version of Figure 4 in [3] and also published in [4].

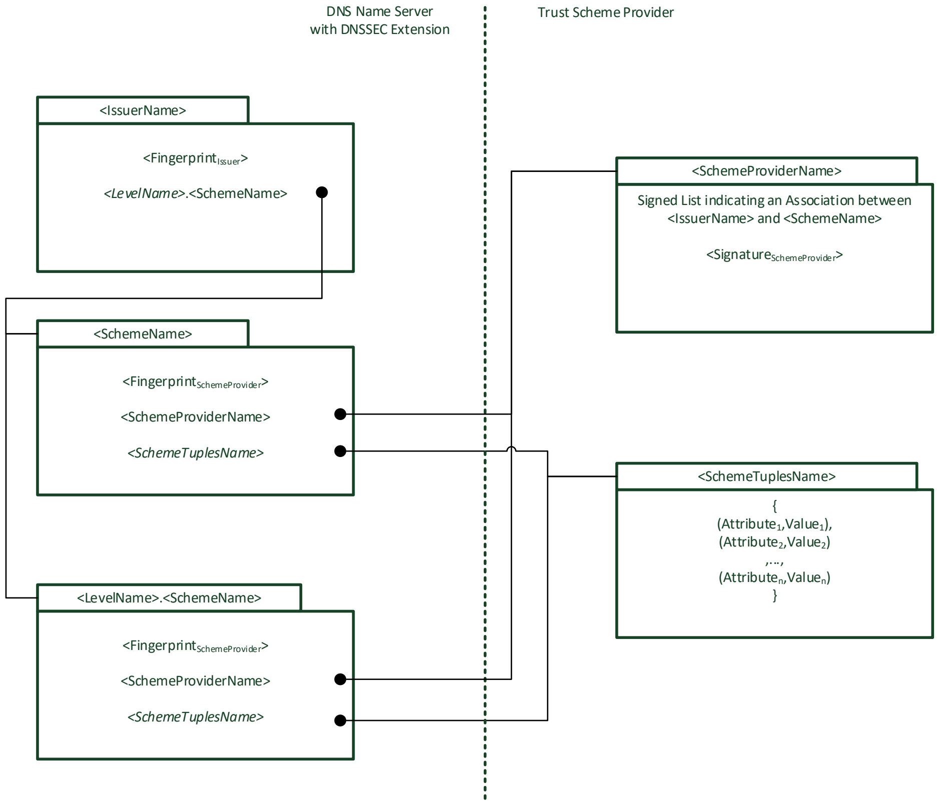

*Figure 1: Representation of Trust Scheme Publications in the TSPA [3]*

For the Trust Scheme Publication different types of Trust Scheme representation need to be considered. A Trust Scheme can be published as a boolean trust scheme publication (e.g. [5]), and an ordinal trust scheme publication (e.g. [6]). Boolean trust scheme publications indicate the entities that comply with the requirements of the trust scheme, and thus are a member of the trust scheme. Ordinal trust scheme publications indicate the entities that comply with the requirements of an ordinal aspect (e.g. a level of assurance) of the trust scheme. However, boolean and ordinal trust scheme publications do not provide any information on the requirements of the trust scheme, or the ordinal value (e.g. Level of Assurance) of the trust scheme that is represented by the trust scheme publication. In order to fill this gap, tuple-based trust scheme publications provide the requirements of a trust scheme in the form of attributes and values.

For the representation of Trust Scheme Associations in the Trust Scheme Provider existing standards should be used. ETSI TS 119 612 [7] provides such a standard. It is described in detail in Section 6.3.2 in D3.1. In summary, a Trusted List in ETSI TS 119 612 provides the association of an Issuer with a Trust Scheme, while enabling authenticity of the claim by enabling verification via the Trust Scheme Provider certificate. This standard is also used in the European Regulation No 910/2014 on “electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC”, which is in the following referred to as eIDAS regulation [8]. Figure 2, which is taken from D3.1 indicates the contents of a Trusted List as described in ETSI TS 119 612 and which is mainly used in this deliverable in the examples in Section 7.

For Trust Scheme Publication, the concept of the TSPA with the two components entails additions in the DNS resource records (e.g. Pointers and URIs) as well as the publication of a signed Trust List on the Trust Scheme Provider side. For example, the verifications of the claims from the Trust List require fingerprints of the issuer certificate and of the trust scheme provider certificate in the corresponding DNS records (see Figure 1). Furthermore, it is difficult to clearly distinguish publication and discovering of Trust Schemes with this approach, as several components e.g. additions in the DNS resource records, are used for both, publication and discovering.

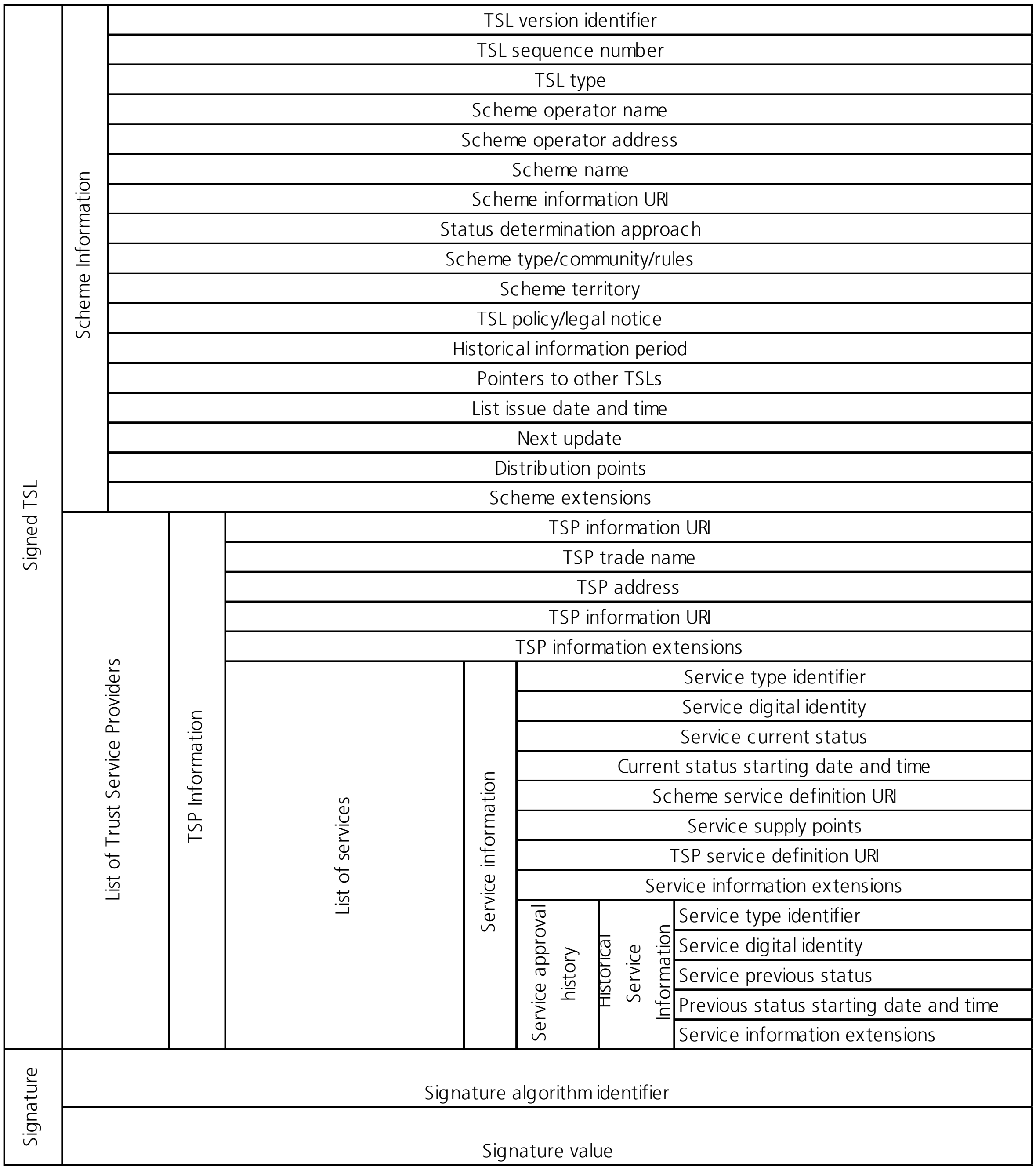

*Figure 2: Overview on TSL contents as defined in ETSI TS 119 621. Adapted from [7], see also D3.1*

## 6.2 Publication of Trust Scheme Membership

A declaration for trust scheme membership certifies that a trust service follows the rules and conditions laid out by a trust scheme for the services it provides. In a Trust Membership Declaration, the trust scheme is both the source and originator of the declaration while the trust service that is declared a member of the scheme is the target (see also Section 5.1 for more details).

The trust declaration document is a trusted list that indicates the membership of the trust service with the referred to trust scheme. This trust list should use or be conforming to existing standards, which involve a trust list and a trust list provider (e.g. ETSI TS 119 612 [7]). In addition, it may be necessary to extend that standard to allow for inclusion of the domain name identifying the trust service.

A trust scheme publishes its trust membership declaration under the ***_scheme._trust*** prefix using the URI record type containing a URI pointing to the trust declaration document.

A trust service can point to the trust schemes it claims to be a member of by publishing PTR records with the trust scheme's domain name identifier under the ***_scheme._trust*** prefix.

The prefix ***_scheme._trust*** indicates within LIGHTest trust membership declarations and claims.

A trust service includes its own domain name identifier, its certificates in the subject alternative name extension and, optionally, in the certificates issued in its role as a certificate issuer in the issuer alternative name extension.

By way of example, the publication of trust scheme membership is demonstrated in the following Section 7 for selected trust schemes.

# 7. Demonstration for Selected Trust Schemes

To demonstrate the consolidated approach to publishing trust-related information in the DNS for trust scheme membership, which is described in Section5 and Section6, this chapter provides several examples of usage for selected trust schemes.

## 7.1 eIDAS

These examples show the trust verification for a certificate issued by an eIDAS qualified trust service. For clarity, real domain names of the players involved but in highly speculative ways are used.

### 7.1.1 Germany

It is assumed that the electronic transaction is simply a signed document. It is signed with a certificate issued by the German D-Trust GmbH, which is a qualified trust service provider for Germany. Either the certificate contains the Issuer Alternative Name extension with a domain name value of *d-trust.net* or the issuer certificate used for signing the certificate contains the Subject Alternative Name with that domain name value.

In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name *_scheme._trust.d-trust.net*:

```
;; QUESTION SECTION:
;_scheme._trust.d-trust.net.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.d-trust.net.   IN  PTR  nrca-ds.de.
```

This indicates to the verifier that D-Trust claims a membership with a trust scheme identified as *nrca-ds.de* (which is the domain used by the German Bundesnetzagentur for their eIDAS trusted list). The verifier will have to discover the trust list for that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.nrca-ds.de.   IN  URI

;; ANSWER SECTION:
_scheme._trust.nrca-ds.de.   IN  URI  https://www.nrca-
ds.de/st/TSL-XML.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list.

Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
    ;; QUESTION SECTION:
    ;_scheme._trust.nrca-ds.de.   IN  SMIMEA

    ;; ANSWER SECTION:

    _scheme._trust.nrca-ds.de.   IN  SMIMEA  <SMIMEA record data>
```

Assuming that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *nrca-ds.de*. Sadly, this isn't something that appears in the certificate of the trust service provider.

As an example for eIDAS Trust Service Status Lists, the current (January 2018) Trust Service Status List of Germany was downloaded (*https://www.nrca-ds.de/st/TSL-XML.xml*) and a shortened version for of the Trust Service Status Lists and for one Trust Service Provider are added in the Appendix in Section11.

#### 7.1.1.1 Trust Translation within eIDAS

To demonstrate the full potential of the eIDAS trust scheme, a short subsection for trust translation is included here. More details and a general description of the publication of trust translation in LIGHTest can be found in the Deliverable 4.3 "DNS-based Publication of Trust Translation Schemes" [9].

In the eIDAS example from above, the verifier can try its luck with trust translation as next step. The scheme *nrca-ds.de* may publish claims to being an equivalent scheme. The verifier conducts another DNS query:

```
    ;; QUESTION SECTION:
    ;_translate._trust.nrca-ds.de.  IN  PTR

    ;; ANSWER SECTION:
    _translate._trust.nrca-ds.de.   IN  PTR esig.ec.europa.eu
```

It does indeed claim to be an equivalent scheme to a scheme named *esig.ec.europe.eu* which, as it happens, is mentioned in the trust policy. The verifier needs to check this claim by locating the trust translation declaration:

```
    ;; QUESTION SECTION:
    ; _translate._trust.esig.ec.europa.eu.  IN  URI

    ;; ANSWER SECTION:
    _translate._trust.esig.ec.europa.eu.   IN  URI \
          https://ec.europa.eu/information_society/policy/esignature/\
          trusted-list/tl-mp.xml
```

The verifier downloads the document and finds *nrca-ds.de* mentioned there.[1] It needs to check the certificates used for signing the document, so:

```
;; QUESTION SECTION:
; _translate._trust.esig.ec.europa.eu.  IN  SMIMEA

;; ANSWER SECTION:
_translate._trust.esig.ec.europa.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. In other words, the German scheme is equivalent to (and therefore complies with) the EU level trust scheme for qualified trust services.

### 7.1.2 Spain

It is assumed that the electronic transaction is simply a signed document. It is signed with a certificate issued by the Spanish Firmaprofesional, S.A., which is a qualified trust service for Spain.

In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name *scheme._trust.firmaprofesional.es*:

```
;; QUESTION SECTION:
;_scheme._trust.firmaprofesional.es.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.firmaprofesional.es.   IN  PTR  sede.minetur.gob.es.
```

This indicates to the verifier that Firmaprofesional, S.A. claims a membership with a trust scheme identified as *sede.minetur.gob.es* (which is the domain used by the Spanish Ministry of Energy, Tourism and Digital Agenda for their eIDAS trusted list). The verifier will have to discover the trust list for that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.sede.minetur.gob.es.   IN  URI

;; ANSWER SECTION:
scheme._trust.sede.minetur.gob.es.   IN  URI
https://sede.minetur.gob.es/prestadores/tsl/tsl.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list.

[1] Note that the current trusted list format does not actually contain a field to note the trust scheme name and would need a small modification in order to work.

Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
; _scheme._trust.sede.minetur.gob.es.   IN  SMIMEA

;; ANSWER SECTION:

_scheme._trust.sede.minetur.gob.es.   IN  SMIMEA  <SMIMEA record data>
```

Assuming, that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *sede.minetur.gob.es*.

### 7.1.3 Austria

It is assumed that the electronic transaction is simply a signed document. It is signed with a certificate issued by the Austrian A-Trust, which is a qualified trust service for Austria.

In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name *_scheme._trust.a-trust.at*:

```
;; QUESTION SECTION:
;_scheme._trust.a-trust.at.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.a-trust.at.   IN  PTR  signatur.rtr.at.
```

This indicates to the verifier that A-Trust claims a membership with a trust scheme identified as *signatur.rtr.at* (which is the domain used by the Austrian Regulatory Authority for Broadcasting and Telecommunications for their eIDAS trusted list). The verifier will have to discover the trust list for that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.signatur.rtr.at.   IN  URI

;; ANSWER SECTION:
_scheme._trust.signatur.rtr.at.   IN  URI
https://www.signatur.rtr.at/currenttl.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list.

Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
;_scheme._trust.signatur.rtr.at.   IN  SMIMEA
```

```
;; ANSWER SECTION:

_scheme._trust.signatur.rtr.at.   IN  SMIMEA  <SMIMEA record data>
```

Assuming, that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *signatur.rtr.at*.

### 7.1.4 Further qualified trust services

In the previous sections, the electronic transaction was a signed document, signed with a certificate issued by a qualified trust service of a Member State. For other qualified trust services of eIDAS, the procedure for the trust verification of the certificate is similar. Therefore, we refer for the procedure for trust verification to Section7.1 and focus here on trust service providers and qualified trust services within eIDAS.

In each country of the European Union, there are several trust service providers, which provide one or more qualified trust services. The *Trusted List Browser* (https://webgate.ec.europa.eu/tl-browser/#/) is an existing tool, which enables to browse the national Trusted Lists and the European List of Trusted Lists (LOTL).

Further qualified trust services of eIDAS are:

- QCert for ESig: qualified certificate for electronic signature;
- QCert for ESeal: qualified certificate for electronic seal;
- QWAC: qualified certificate for website authentication;
- QTimestamp: qualified time stamp;
- QeRDS: qualified electronic registered delivery service;

For the examples from above, the German trust service provider D-Trust GmbH, provides the following trust services: qualified certificate for electronic signature; qualified certificate for electronic seal; qualified certificate for website authentication; and qualified timestamp. In Spain, the trust service provider Firmaprofesional, S.A., provides the following trust services: qualified certificate for electronic signature; qualified certificate for electronic seal; qualified certificate for website authentication; and qualified timestamp. In Austria, the A-Trust GmbH provides only qualified certificate for electronic signature. For a qualified timestamp, the trust service provider e-commerce monitoring GmbH can be chosen for example.

In order to discover the trust scheme(s) this trust service is a member, the workflow is the same as described above in section 7.1. The reason for this is that the national trusted list of each Member State includes all information related to the qualified trust service providers, which are supervised by the issuing Member State.

For example, the eIDAS trusted list for Germany, which is publicly available via the URI *https://www.nrca-ds.de/st/TSL-XML.xml* contains for the trust service provider D-Trust

GmbH entries for all trust services it provides, i.e. qualified certificate for electronic signature; qualified certificate for electronic seal; qualified certificate for website authentication; and qualified timestamp.

## 7.2 Pan Canadian Trust Framework

Despite being launched in 2016, the Pan-Canadian Trust Framework (PCTF) seems to be still at a very early stage or even dead. Their last press release was released in January 2017, and no published documents contain the terms “DNS”, “trust list” or “trust scheme”.

In addition, DIACC's attention (and funding) seems to have moved towards block chains. As no further documentation is publicly available, the process described below is an educated guess of a possible implementation.

In Canada, CIRA (Canadian Internet Registration Authority) manages the list of authorised .ca registrars. A fictional service provider, Maple Finance, could then utilise any of the authorised, DNSSEC-compliant registrars to host their own domain and trust scheme, while Canada Trust, a fictional trust provider, would issue certificates used by others participants to digitally sign the published trust lists. These operations are detailed at query level in Chapter 7.4.

## 7.3 STORK2.0

Each Member State (MS) of the European Union (EU) has their own way of verifying identity for electronic transactions (eID). With 28 countries in the EU, and potentially multiple types of electronic identification per country, that means there can be even more than 28 different ways of verifying the authenticity of an electronic transaction. This can make it difficult or impossible to verify eID’s in cross border transactions. Furthermore, it is unlikely that an agreement can be reached between the 28 MS for a single method of authenticating eID’s (nor would such an agreement be in line with current EU policy on electronic identification).

The large scale pilot project STORK addressed this issue by enabling interoperability between MS eID authentication systems. So, STORK is not a trust scheme, it was a pilot project which created a network that tested eID interoperability approaches, which would ultimately feed into the eID trust model integrated in eIDAS. The basic architecture was that each MS should create a node in the STORK network that handled authentication and signature verification for the people in that MS[2]. Then whenever a cross border authentication (or verification) was required, the eID is passed to the node on the STORK network that owns that eID. For example, if a French citizen opens a bank account in Germany, their identification would be passed to the

[2] With the exception of Austria, which did not adopt the node based model.

French STORK node to be authenticated. Consequently, each Member State (MS) can continue to use their own methodology, yet cross border authentication and verification is possible.

eID's also have various levels of assurance ("quality of authentication"), with each MS defining the levels that they support. This causes additional interoperability problems if a particular level of assurance is required to verify a cross border transaction between two MS that use different systems to describe levels of assurance of their eID's. STORK also enabled interoperability of levels of assurance by studying the systems for describing levels of assurance in the 28 MS and creating a mapping between them. This is an example of "trust translation".

eID's can also contain or refer to "attributes", which extend verification to not only verify the identity of the person, but also verify things about them. For example, an attribute could be "isaLicensedMedicalDoctor", meaning that the person that owns the eID is licensed to practice medicine in the member state of which they are a citizen. STORK provides an extension mechanism called "Attribute Providers" that allows 3rd parties in a MS to plugin to the STORK network and augment the eID authentication and verification process. For the medical doctor example, the legal entity that licenses doctors to practice medicine in a particular MS could create an attribute provider plugin to the STORK network that would augment the eID's of that MS with an attribute indicating whether the eID identifies a person licensed to practice medicine in that MS.

Development of STORK stopped in 2015 with the adoption of the eIDAS Regulation. Some of the ideas of STORK became input to the technical specifications to eIDAS. The technical realisations of STORK are still around and interoperate with eIDAS via the stable eIDAS node releases[3].

Hence, the publication of STORK is demonstrated with a fictional example, e.g. a French citizen opens a bank account in Germany from above. This requires a passing to the French STORK node for authentication. It is assumed that the citizen claims a membership with a node of the STORK network identified as *stork-eu.example*. For the discovery of the STORK network, the verifier will have to discover the trust list via a DNS query.

```
;; QUESTION SECTION:
;_scheme._trust.stork-eu.example.     IN   URI

;; ANSWER SECTION:
_scheme._trust.stork-eu.example.    IN   URI   https://www.stork-
eu.example/ct/TSL-XML.xml
```

It will now download that list and see if the node appears on that list. In this list all nodes of the STORK network are listed. For the publication of this list, the existing standard ETSI TS 119 612 can be used.

[3] See https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Implement+or+operate+an+eIDAS+Node+-+Overview

Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
;_scheme._trust.stork-eu.example.   IN  SMIMEA

;; ANSWER SECTION:

_scheme._trust.stork-eu.example.   IN  SMIMEA  <SMIMEA record data>
```

As an alternative, and this is probably the more realistic scenario for the application of STORK, the publication could be done via the eIDAS/STORK plug-in[4] and then follow the eIDAS publication as described in Section 7.1.

## 7.4 Electronic Signature Law of the People's Republic of China

The People's Republic of China does not have a public trust list published under any domain presently. Therefore, the process described here is from an educated guess on how it should be accessed if it was published on a domain. We use the eIDAS format as a template for China also.

It is assumed that the electronic transaction is simply a signed document. It is signed with a certificate issued by the Chinese Trust (A fictional trust provider), which is a qualified trust service for China. Either the certificate contains the Issuer Alternative Name extension with a domain name value of *ctrust.cn* or the issuer certificate used for signing the certificate contains the Subject Alternative Name with that domain name value.

In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name *_scheme._trust.ctrust.cn*:

```
;; QUESTION SECTION:
;_scheme._trust.ctrust.cn.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.ctrust.cn.   IN  PTR  miit.gov-cn.example.
```

This indicates to the verifier that Chinese Trust claims a membership with a trust scheme identified as *miit.gov-cn.example* (which is the fictional domain used by the Chinese for their trusted list owned by Ministry of Industry and Information Technology in China). The verifier will have to discover the trust list for that via another DNS query:

[4] See https://www.esens.eu/content/eidasstork-plug-now-available

```
;; QUESTION SECTION:
;_scheme._trust.miit.gov-cn.example.    IN  URI

;; ANSWER SECTION:
_scheme._trust.miit.gov-cn.example.   IN  URI  https://www.
miit.gov-cn.example/ct/TSL-XML.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list.

Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
;_scheme._trust.miit.gov-cn.example.   IN  SMIMEA

;; ANSWER SECTION:

_scheme._trust.miit.gov-cn.example.   IN  SMIMEA  <SMIMEA record data>
```

Assuming, that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *miit.gov-cn.example*.

Note that a fictitious example of the publication of the trust list that could be owned by the Ministry of Industry and Information Technology in China is provided in the Appendix in Section11.

## 7.5 FIDO

The FIDO (Fast Identity Online) scheme is an open specification for strong authentication which has been outlined in more detail in deliverable D 3.1. To overcome existing authentication silos, the FIDO alliance has developed a cross-industry specification for a strong client authentication towards a relying party web service. The scheme itself is pseudonymous but can be combined with identity applications like a mobile ID or identity federation.

Regarding the choice of authenticators the FIDO specification defines an open ecosystem of client devices or software components that functionally comply with the FIDO standard but differ with respect to implementation details, security level (hardware or software based security) and user experience. Since this imposes the challenge for a relying party to determine which type of authenticator has been used, FIDO also introduces an attestation scheme where the registration data of the authenticator is signed with a private key residing inside the authenticator [10].

The existence and use of an attestation scheme is directly related to the topics of trust and integration into trust schemes, as addressed by LIGHTest. If a relying party is part of a trust scheme or a regulatory scheme requiring links to an existing trust scheme (e.g. for payment applications) then it needs to be able to ensure the policies regarding the authenticator strength as defined by the trust scheme or regulatory scheme. In other words, the relying party needs to be able to determine if a specific authenticator of an end user is compliant to the trust/regulatory scheme requirements.

FIDO addresses this topic partially by introducing an authenticator metadata definition. These metadata describe some basic implementation and security properties of an authenticator to allow the relying party to enforce security policies and to restrict the use of authenticators to certain basic types. For example, metadata include information about the type of user verification like the minimum length of a PIN or password, the number of retries or the False Acceptance Rate (FAR) in case of biometric authenticators. In addition, the metadata also include a reference to the attestation certificate used for authenticator attestation (`attestationRootCertificates` and `attestationCertificateKeyIdentifiers`). Further details of FIDO metadata can be found in [11]. Metadata can be published by the FIDO metadata service [12], directly by the authenticator vendor or by a third party metadata service. Among others, the metadata service publishes the `aaid` of an authenticator, the corresponding `attestationCertificateKeyIdentifiers` and the URL to the actual metadata.

While this scheme of metadata and authenticator attestation is sufficient for general purpose applications in the private sector, a more trust and security sensitive application will require further trust information. The main question for the relying party is whether a specific authenticator is trustworthy and acceptable by a specific trust scheme. This scheme could also be an application-specific subdomain of a trust scheme, e.g. the national banking association scheme for payment applications in a specific EU member country as part of the European regulated market for financial services.

Therefore, the task of the relying party is to query, whether a certain authenticator claims compliance to the relevant trust scheme. The trust anchor for this purpose would be the attestation certificate. It is issued by the vendor of the authenticator and referenced in the metadata statement as `attestationRootCertificates`. This reference points to the root of the certificate chain which can either be at the authenticator vendor (self-issued certificates) or could be a root of a service claiming conformance to a certain trust scheme. The relying party now has to verify that the root certificate is part of a specific trust scheme.

The query will now look very similar to the eIDAS example in section 7.1. The relying party performs a DNS query for PTR records at the name *_trustscheme._decl.vendor.com*:

```
;; QUESTION SECTION:
;_scheme._trust.vendor.com.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.vendor.com.   IN  PTR  banking-ts.example.
```

In this example the vendor of the authenticator would claim a membership in the (hypothetic) trust scheme of a national banking association, qualifying this authenticator for banking transactions in that country. The relying party now has to discover the trust list for the banking trust scheme that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.banking-ts.example.    IN   URI

;; ANSWER SECTION:
_scheme._trust.banking-ts.example.    IN   URI  https://www.banking-
ts.example/tsl/TSL-XML.xml
```

After downloading the list the relying party can now verify if the attestation root certificate is on the list of trusted authenticators. Since the list is signed, it will also check whether the certificate used for signing is valid:

```
;; QUESTION SECTION:
;_scheme._trust.banking-ts.example.    IN   SMIMEA

;; ANSWER SECTION:

_scheme._trust.banking-ts.example.    IN   SMIMEA   <SMIMEA record data>
```

The relying party can now be sure that the corresponding FIDO authenticator used by the end user for registration is acceptable for banking applications in this member state since its root certificate for attestation is part of the trust list of the *banking-ts.example* trust scheme. The verification of the signed registration data with the corresponding certificate (up to the root certificate) is part of the standard FIDO protocol anyway.

With this approach, the relying party queries conformance of a specific authenticator from a specific vendor. As a consequence, the corresponding trust scheme would have to list all the acceptable authenticators of all known vendors. Since this list may change frequently this would impose a significant maintenance effort for the trust scheme administration. It may therefore be more efficient to publish the minimum required metadata properties of an acceptable authenticator, rather than the certificate of a specific authenticator. For example, a banking trust scheme may require hardware protection of the authentication credentials, a PIN length of at least 4 digits and a retry counter with a maximum of three retries.

This can be achieved by publishing the `Policy` dictionary in the trust list, which contains the list of acceptable or disallowed authenticator criteria [10]. When performing a registration, the relying party needs to retrieve the `Policy` dictionary from the corresponding trust list and provide this to the FIDO server. The server will then forward these policy requirements to the client and follow the standard FIDO protocol. With this approach, the trust scheme can enforce an authenticator policy without having to include individual authenticator models.

The most simple way of publishing the FIDO policy with an ETSI TS 119600-based trust list is to use the service information extension fields (chapter 5.5.9 of [7]), especially the field <AdditionalServiceInformation> (chapter 5.5.9.4 of [7]) to publish a URI of the actual FIDO `Policy` dictionary as JSON file. The corresponding entry looks like:

```
< AdditionalServiceInformation >

      <OtherInformation>

            <URI xml:lang="en"> http://banking-ts.example/fido/uaf/v1/banking-
            ts_policy.json </URI>

      </OtherInformation>

</ AdditionalServiceInformation>
```

An example policy looks like [10]:

```
{

  "accepted":

  [

      [

        { "userVerification": 2, "authenticationAlgorithms": [1, 2, 5, 6],
"assertionSchemes": ["UAFV1TLV"]},

        { "userVerification": 16, "authenticationAlgorithms": [1, 2, 5, 6],
"assertionSchemes": ["UAFV1TLV"]}

      ]

  ]

}
```

In this example the policy is matching the combination of a fingerprint-based and a face recognition-based authenticator.

A relying party can thus directly retrieve the accepted `Policy` of the trust scheme and can provide this to the own FIDO server which in turn will embed this policy into the FIDO registration request message. The FIDO client will then verify if an authenticator (ASM) with the corresponding `Policy MatchCriteria` is available on the user device. If yes, the user can perform the registration with this authenticator or can choose between several authenticators, if available.

By using this approach the trust scheme administrator does not have to specify individual authenticators (a list that would need to be updated frequently) but rather one or several classes of authenticators with certain minimum requirements. In future these requirements could also include the FIDO security level once the corresponding certification scheme has been established by the FIDO alliance.

### 7.6 Ordinal trust scheme

As described in Section6.1 and in more detail in D3.1 [3] a Trust Scheme can be published as a boolean trust scheme publication, and as an ordinal trust scheme publication. As the previous examples have demonstrated the publication of Boolean trust schemes, this section will show how ordinal trust schemes can be published assuming two Level of Assurances.

This is a fictional example for ordinal trust schemes using eIDAS for non-qualified trust services. *D-Adv-eSig-Service1* is a trust service for advanced electronic signatures. In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name _scheme._trust.d-adv-eSig-Service1:

```
;; QUESTION SECTION:
;_scheme._trust.d-adv-eSig-Service1.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.d-adv-eSig-Service1.  IN  PTR  TS-d-adv-eSig-de.example.
```

This indicates to the verifier that D-Adv-eSig-Service1 claims a membership with a trust scheme identified as *TS-d-adv-eSig-de.example* (which is the fictive domain used in Germany for the eIDAS trust service for advanced electronic signatures). The verifier will have to discover the trust list for that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.TS-d-adv-eSig-de.example.   IN  URI

;; ANSWER SECTION:
_scheme._trust.TS-d-adv-eSig-de.example.   IN  URI
https://www.TS_d-adv-eSig-de.example/TSL-XML.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list. Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
;_scheme._trust.TS-d-adv-eSig-de.example.   IN  SMIMEA

;; ANSWER SECTION:

_scheme._trust.TS-d-adv_eSig-de.example.  IN  SMIMEA <SMIMEA record data>
```

Assuming, that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *TS-d-adv-eSig-de*.example.

This is part 2 of the fictional example for ordinal trust schemes using eIDAS for non-qualified trust services. D-Adv-eSig-Service1 is a trust service for **non-**advanced electronic signatures. In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records at the name *_scheme._trust.d-**non**-adv-eSig-Service1*

```
;; QUESTION SECTION:
;_scheme._trust.d-non-adv-eSig-Service1.  IN  PTR

;; ANSWER SECTION:

_scheme._trust.d-non-adv-eSig-Service1.  IN  PTR  TS-d-non-adv-eSig-
de.example.
```

This indicates to the verifier that *D-Adv-eSig-Service1* claims a membership with a trust scheme identified as *TS-d-non-adv-eSig-de.example* (which is the fictive domain used in Germany for the eIDAS trust service for non-advanced electronic signatures). The verifier will have to discover the trust list for that via another DNS query:

```
;; QUESTION SECTION:
;_scheme._trust.TS-d-non-adv-eSig-de.example.   IN  URI

;; ANSWER SECTION:
_scheme._trust.TS_d-non-adv_eSig-de.example.   IN  URI
https://www.TS-d-adv-eSig-de.example/TSL-XML.xml
```

It will now download that list and see if the issuer certificate from the electronic transaction appears on that list. Since the list is signed, it will also check whether the certificate used for signing the trusted list is valid. Again, DNS:

```
;; QUESTION SECTION:
;_scheme._trust.TS-d-non-adv-eSig-de.example.   IN  SMIMEA

;; ANSWER SECTION:

_scheme._trust.TS-d-non-adv-eSig-de.example.  IN  SMIMEA  <SMIMEA record
data>
```

Assuming, that all of this checks out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme *TS-d-non-adv-eSig-de.example*.

## 7.7 Tuple-based trust scheme

As described in Section6.1 and in more detail in D3.1 [3] boolean and ordinal trust scheme publications do not provide any information on the requirements. Therefore, tuple-based trust scheme publications are required, which provide the requirements in the form of attributes and values.

The principle for the publication of tuple-based trust schemes is similar to the publication of boolean and ordinal trust schemes described above. In addition, the tuples with the attributes and values needs to be published. To do so, there are two possibilities. First, the signed trust list using ETSI standard TS119612 needs to be extended by the tuples, i.e. the tuples are added in the XML file of the trust list. Second, an extra document, which lists all the tuples with the attributes and values is created using an own standard. In addition, this requires a pointer from the signed trust list to this document, which also should be signed with the same key as the trust list. For the pointer, the field <*AdditionalServiceInformation*> which belongs to the service information extensions of ETSI TS119612 (see Chapter 5.5.9 in [7]) could be used in the signed trust list to publish a URI identifying additional information (see also the FIDO example in Section7.5). Furthermore, it is also required that the ATV knows the format and can read and process this extra document.

The basis for this tuple based publication is the data model, which is already introduced in D3.1 [3]. However, it will be developed further and finalized in D3.2, which is due in M24. Therefore, the publication of the data model will be described in more detail in D3.2. It is intended that for the tuple based publication the ETSI standard TS119612 is used where it is applicable and for the remaining ones an extra document with a self-defined and ATV readable standard is used.

# 8. Summary and Conclusion

The deliverable 3.3 outlines the publication of Trust Schemes within LIGHTest based on the conceptual description of the Trust Scheme Publication Authority (TSPA) in D3.1. The concept of the TSPA with the two components entails for the publication additions in the DNS resource records (e.g. Pointers and URIs) as well as the publication of a signed Trust List on the Trust Scheme Provider side.

The developed consolidated approach to publishing trust-related information in the DNS in general was specified for trust scheme memberships and demonstrated on several examples. Hereby the publication of trust schemes of both already existing (e.g. eIDAS) and fictional (e.g. Pan Canadian Trust Framework, FIDO, Electronic Signature Law of the People's Republic of China) of signed Trust Lists is described.

This approach enables the publication of already existing as well as newly developed trust schemes in the LIGHTest infrastructure.

# 9. References

[1] O. K. P. Falstrom, The Uniform Resource Identifier (URI) DNS Resource, RFC 7553, Internet Engineering Task Force, June 2015.

[2] P. Mockapetris, Domain names – implementation and specification, RFC 1035, Internet Engineering Task Force, November 1987.

[3] The LIGHTest Project, D3.1 - Conceptual Framework for Trust Schemes (1), Project Deliverable, 2017.

[4] S. Wagner, S. Kurowski, U. Laufs und H. Rossnagel, „A Mechanism for Discovery and Verification of Trust Scheme Memberships: The LIGHTest Reference Architecture,“ in *L. Fritsch et al. (Eds.): Open Identity Summit, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik*, Bonn, 2017.

[5] European Standard ETSI EN 319 401, Electronic Signatures and Infrastructures (ESI); General Policy Requirements for Trust Service Providers, ETSI, Sophia Antipois Cedex, France, 2016.

[6] ISO/IEC, Information technology -- Security techniques -- Entity authentication assurance framework, Geneva, CH, 2013.

[7] ETSI TS 119 612, „Electronic Signatures and Infrastructures (ESI);Trusted Lists,“ European Telecommunications Standards Institute; Technical Specification, Sophia Antipolis Cedex, V2.1.1 (2015-07), 2015.

[8] European Parliament, Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, European Parliament, Brussels, Belgium, Regulation 910/201, 2014.

[9] The LIGHTest Project, D4.3 - DNS-based Publication of Trust Translation Schemes, Project Deliverable, 2017.

[10] FIDO Alliance, „FIDO UAF Protocol Specification,“ 02 02 2017. [Online]. Available: https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-protocol-v1.1-id-20170202.html. [Zugriff am 21 06 2017].

[11] FIDO Alliance, „FIDO Metadata Statements,“ 02 02 2017. [Online]. Available: https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-metadata-statement-v1.1-id-

20170202.html. [Zugriff am 091 01 2018].

[12] FIDO Alliance, „FIDO Metadata Service,“ 02 02 2017. [Online]. Available: https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-metadata-service-v1.1-id-20170202.html. [Zugriff am 09 01 2018].

# 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.

The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.

# 11. Appendix

### eIDAS Trust Service Status List

As an example for eIDAS Trust Service Status Lists, this is the current (January 2018) Trust Service Status List of Germany. It is downloaded from https://www.nrca-ds.de/st/TSL-XML.xml. The content is partially shortened and the corresponding changes are marked in green, e.g. German text or X509 certificates. In addition, the TrustServiceProviderList is shortened (in total there are 22 trust service providers). One example of a trust service provider is also shown below in the appendix.

```
<?xml version="1.0" encoding="UTF-8"?><TrustServiceStatusList
xmlns="http://uri.etsi.org/02231/v2#"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns3="http://uri.etsi.org/02231/v2/additionaltypes#"
xmlns:ns4="http://uri.etsi.org/01903/v1.3.2#"
xmlns:ns5="http://uri.etsi.org/TrstSvc/SvcInfoExt/eSigDir-1999-93-EC-
TrustedList/#"
xmlns:ns6="http://uri.etsi.org/01903/v1.4.1#" Id="TrustServiceStatusList-1"
TSLTag="http://uri.etsi.org/19612/TSLTag">
<SchemeInformation>
<TSLVersionIdentifier>5</TSLVersionIdentifier>
<TSLSequenceNumber>61</TSLSequenceNumber>
<TSLType>http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUgeneric</TSLType>
<SchemeOperatorName>
<Name xml:lang="en">Federal Network Agency</Name>
<Name xml:lang="de">Bundesnetzagentur</Name>
</SchemeOperatorName>
<SchemeOperatorAddress>
<PostalAddresses>
<PostalAddress xml:lang="en">
<StreetAddress>Canisiusstr. 21</StreetAddress>
<Locality>Mainz</Locality>
<PostalCode>55122</PostalCode>
<CountryName>DE</CountryName>
</PostalAddress>
<PostalAddress xml:lang="de">
<StreetAddress>Canisiusstr. 21</StreetAddress>
<Locality>Mainz</Locality>
<PostalCode>55122</PostalCode>
<CountryName>DE</CountryName>
</PostalAddress>
</PostalAddresses>
<ElectronicAddress>
<URI xml:lang="en">mailto:eIDAS@bnetza.de</URI>
<URI xml:lang="en">http://www.bundesnetzagentur.de</URI>
</ElectronicAddress>
</SchemeOperatorAddress>
<SchemeName>
<Name xml:lang="en">DE:Trusted list including information related to the
qualified trust service providers which are supervised by the issuing
Member State, together with information related to the qualified trust
```

```
services provided by them, in accordance with the relevant provisions laid down in Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.</Name>
<Name xml:lang="de">DE:Vertrauensliste mit Angaben zu den qualifizierten Vertrauensdiensteanbietern, ... .</Name>
</SchemeName>
<SchemeInformationURI>
<URI xml:lang="en">https://www.nrca-ds.de/en/tsl_e.htm</URI>
<URI xml:lang="de">https://www.nrca-ds.de/tsl.htm</URI>
</SchemeInformationURI>
<StatusDeterminationApproach>http://uri.etsi.org/TrstSvc/TrustedList/StatusDetn/EUappropriate</StatusDeterminationApproach>
<SchemeTypeCommunityRules>
<URI xml:lang="en">http://uri.etsi.org/TrstSvc/TrustedList/schemerules/EUcommon</URI>
<URI xml:lang="en">http://uri.etsi.org/TrstSvc/TrustedList/schemerules/DE</URI>
</SchemeTypeCommunityRules>
<SchemeTerritory>DE</SchemeTerritory>
<PolicyOrLegalNotice>
<TSLLegalNotice xml:lang="en">The applicable legal framework for the present trusted list is Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.</TSLLegalNotice>
<TSLLegalNotice xml:lang="de">Der für diese Vertrauenslisten geltende Rechtsrahmen ist die Verordnung (EU) Nr. 910/2014 ...</TSLLegalNotice>
</PolicyOrLegalNotice>
<HistoricalInformationPeriod>65535</HistoricalInformationPeriod>
<PointersToOtherTSL>
<OtherTSLPointer>
<ServiceDigitalIdentities>
<ServiceDigitalIdentity>
<DigitalId>
<X509Certificate>MIID/abcdefgh0123456789</X509Certificate>
</DigitalId>
</ServiceDigitalIdentity>
<ServiceDigitalIdentity>
<DigitalId>
<X509Certificate>MIID/bcdefghi0123456789</X509Certificate>
</DigitalId>
</ServiceDigitalIdentity>
<ServiceDigitalIdentity>
<DigitalId>
<X509Certificate>cdefghij0123456789</X509Certificate>
</DigitalId>
</ServiceDigitalIdentity>
<ServiceDigitalIdentity>
<DigitalId>
<X509Certificate>defghijkl0123456789</X509Certificate>
</DigitalId>
```

```
</ServiceDigitalIdentity>
</ServiceDigitalIdentities>
<TSLLocation>https://ec.europa.eu/information_society/policy/esignature/trusted-list/tl-mp.xml</TSLLocation>
<AdditionalInformation>
<OtherInformation>
<TSLType>http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUlistofthelists</TSLType>
</OtherInformation>
<OtherInformation>
<SchemeTerritory>EU</SchemeTerritory>
</OtherInformation>
<OtherInformation>
<ns3:MimeType>application/vnd.etsi.tsl+xml</ns3:MimeType>
</OtherInformation>
<OtherInformation>
<SchemeOperatorName>
<Name xml:lang="en">European Commission</Name>
</SchemeOperatorName>
</OtherInformation>
<OtherInformation>
<SchemeTypeCommunityRules>
<URI xml:lang="en">http://uri.etsi.org/TrstSvc/TrustedList/schemerules/EUlistofthelists</URI>
</SchemeTypeCommunityRules>
</OtherInformation>
</AdditionalInformation>
</OtherTSLPointer>
</PointersToOtherTSL>
<ListIssueDateTime>2017-12-01T01:00:00Z</ListIssueDateTime>
<NextUpdate>
<dateTime>2018-05-27T00:00:00Z</dateTime>
</NextUpdate>
<DistributionPoints>
<URI>https://www.nrca-ds.de/st/TSL-XML.xml</URI>
</DistributionPoints>
</SchemeInformation>
<TrustServiceProviderList>
<TrustServiceProvider>
<TrustServiceProvider>
<TrustServiceProvider>
<TrustServiceProvider>
<TrustServiceProvider>
...22 <TrustServiceProvider>
</TrustServiceProviderList>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="id-177987bd9b07ee0b175c455e00260704"><ds:SignedInfo><ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/><ds:Reference Id="xml_ref_id" Type="" URI=""><ds:Transforms><ds:Transform
```

```
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/><ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><ds:DigestValue>v3oGrIOMp
0EbGmzlh
cQIbsT7WplPxw5IGPs0QRTFEwE=</ds:DigestValue></ds:Reference><ds:Reference
Type="http://uri.etsi.org/01903#SignedProperties"
URI="#xades-id-177987bd9b07ee0b175c455e00260704"><ds:Transforms><ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><ds:DigestValue>mWNqN6WjC
Iib6jTGB
NZ+ZxrXnzqVhTAPbF5V5Y7tpt4=</ds:DigestValue></ds:Reference></ds:SignedInfo><ds
:Signatu
reValue
Id="value-id-
177987bd9b07ee0b175c455e00260704">efghijkl0123456789</ds:SignatureValue><
ds:KeyInfo><ds:X509Data><ds:X509Certificate>fghijklm0123456789</ds:X509Certifi
cate></d
s:X509Data></ds:KeyInfo><ds:Object><xades:QualifyingProperties
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
Target="#id-177987bd9b07ee0b175c455e00260704"><xades:SignedProperties
Id="xades-id-
177987bd9b07ee0b175c455e00260704"><xades:SignedSignatureProperties><xades
:SigningTime>2017-12-
01T13:12:59Z</xades:SigningTime><xades:SigningCertificate><xades:
Cert><xades:CertDigest><ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><ds:DigestValue>4dsvmQtp3S
V5/ysk+d
pRX2eP4QQ=</ds:DigestValue></xades:CertDigest><xades:IssuerSerial><ds:X509Issu
erName>C
N=14R-CA
1:PN,O=Bundesnetzagentur,C=DE</ds:X509IssuerName><ds:X509SerialNumber>1151</ds
:X509Ser
ialNumber></xades:IssuerSerial></xades:Cert></xades:SigningCertificate></xades
:SignedS
ignatureProperties><xades:SignedDataObjectProperties><xades:DataObjectFormat
ObjectReference="#xml_ref_id"><xades:MimeType>application/octet-
stream</xades:MimeType
></xades:DataObjectFormat></xades:SignedDataObjectProperties></xades:SignedPro
perties>
</xades:QualifyingProperties></ds:Object></ds:Signature></TrustServiceStatusLi
st>
```

### eIDAS Trust Service Provider

From the example Trust Service Status Lists for Germany from above, one (randomly chosen) of the 22 Trust Service Provider are shown here. This Service provides electronic timestamps compliant with eIDAS. The content is also downloaded from https://www.nrca-ds.de/st/TSL-XML.xml, partially shortened and the corresponding changes, e.g. the X509 certificate, are marked in green.

```
<TrustServiceProvider>
<TSPInformation>
<TSPName>
<Name xml:lang="en">exceet Secure Solutions GmbH</Name>
</TSPName>
<TSPTradeName>
<Name xml:lang="en">VATDE-215384696</Name>
<Name xml:lang="en">NTRDE-HRB 78770 Duesseldorf</Name>
</TSPTradeName>
<TSPAddress>
<PostalAddresses>
<PostalAddress xml:lang="en">
<StreetAddress>Rethelstr. 47</StreetAddress>
<Locality>Duesseldorf</Locality>
<PostalCode>40237</PostalCode>
<CountryName>DE</CountryName>
</PostalAddress>
</PostalAddresses>
<ElectronicAddress>
<URI xml:lang="en">http://www.exceet-secure-solutions.de</URI>
<URI
xml:lang="en">mailto:info@exceet-secure-solutions.de</URI>
</ElectronicAddress>
</TSPAddress>
<TSPInformationURI>
<URI
xml:lang="en">http://www.exceet-secure-solutions.de/en/it-security
/electronic-timestamps-compliant-with-eidas/</URI>
<URI
xml:lang="de">http://www.exceet-secure-solutions.de/it-security/el
ektronische-zeitstempel-nach-eidas/</URI>
<URI
xml:lang="en">http://www.exceet-secure-solutions.de/en/it-security
/governance-risk-compliance/</URI>
<URI
xml:lang="de">http://www.exceet-secure-solutions.de/it-security/go
vernance-risk-compliance/</URI>
</TSPInformationURI>
</TSPInformation>
<TSPServices>
<TSPService>
<ServiceInformation>
<ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/TSA
/QTST</ServiceTypeIdentifier>
```

```
<ServiceName>
<Name xml:lang="en">exceet TSA</Name>
<Name xml:lang="de">exceet TSA</Name>
</ServiceName>
<ServiceDigitalIdentity>
<DigitalId>
<X509Certificate>abcdefgh0123456789</X509Certificate>
</DigitalId>
<DigitalId>
<X509SubjectName>CN=exceet trustcenter CA2, O=exceet Secure Solutions GmbH, C=DE</X509SubjectName>
</DigitalId>
<DigitalId>
<X509SKI>o7AmghzxQnnQGqwp3XwFlw/e3ZE=</X509SKI>
</DigitalId>
</ServiceDigitalIdentity>
<ServiceStatus>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</ServiceStatus>
<StatusStartingTime>2016-10-26T10:00:00Z</StatusStartingTime>
</ServiceInformation>
</TSPService>
</TSPServices>
</TrustServiceProvider>
```

### Electronic Singature of Law, China

A sample trust list from the Ministry of Industry and Information Technology in China is provided below.

```
<?xml version="1.0" encoding="UTF-8"?><TrustServiceStatusList
xmlns="http://uri.etsi.org/02231/v2#"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns3="http://uri.etsi.org/01903/v1.3.2#"
xmlns:ns4="http://uri.etsi.org/02231/v2/additionaltypes#"
xmlns:ns5="http://uri.etsi.org/TrstSvc/SvcInfoExt/eSigDir-1999-93-EC-
TrustedList/#" xmlns:ns6="http://uri.etsi.org/01903/v1.4.1#"
TSLTag="http://uri.etsi.org/19612/TSLTag">
    <SchemeInformation>
        <TSLVersionIdentifier>5</TSLVersionIdentifier>
        <TSLSequenceNumber>1</TSLSequenceNumber>

<TSLType>http://uri.etsi.org/TrstSvc/TrustedList/TSLType/CNlist</TSLType>
        <SchemeOperatorName>
            <Name xml:lang="en">Ministry of Industry and Information
Technology</Name>
        </SchemeOperatorName>
        <SchemeOperatorAddress>
            <PostalAddresses>
                <PostalAddress xml:lang="en">
                    <StreetAddress>Kina Street;Postboks 2193</StreetAddress>
                    <Locality>Beijing</Locality>
                    <PostalCode>100037</PostalCode>
                    <CountryName>CN</CountryName>
                </PostalAddress>
                <PostalAddress xml:lang="en">
                    <StreetAddress>Kina Street;Postboks 2193</StreetAddress>
                    <Locality>Beijing</Locality>
                    <PostalCode>100037</PostalCode>
                    <CountryName>CN</CountryName>
                </PostalAddress>
            </PostalAddresses>
            <ElectronicAddress>
                <URI xml:lang="en">mailto:info@miit.gov.cn</URI>
                <URI
xml:lang="en">http://www.miit.gov.cn/Servicemenu/English</URI>
            </ElectronicAddress>
        </SchemeOperatorAddress>
        <SchemeName>
            <Name xml:lang="en"> CN:Trusted list including information related
to the qualified trust service providers which are supervised by the issuing
republic of China.</Name>
        </SchemeName>
        <SchemeInformationURI>
            <URI
xml:lang="en">http://www.miit.gov.cn/Servicemenu/English/Digitisation/Digital-
Signature/Trusted-List-China.aspx</URI>
        </SchemeInformationURI>
```

```
<StatusDeterminationApproach>http://uri.etsi.org/TrstSvc/TrustedList/StatusDet
n/CNdetermination</StatusDeterminationApproach>
        <SchemeTypeCommunityRules>
            <URI
xml:lang="en">http://uri.etsi.org/TrstSvc/TrustedList/schemerules/CN</URI>
        </SchemeTypeCommunityRules>
        <SchemeTerritory>CN</SchemeTerritory>
        <PolicyOrLegalNotice>
            <TSLLegalNotice xml:lang="en">The applicable legal framework for
the present trusted list is a regulation of the republic of
China.</TSLLegalNotice>
        </PolicyOrLegalNotice>
        <HistoricalInformationPeriod>65535</HistoricalInformationPeriod>
        <ListIssueDateTime>2017-09-25T01:00:00Z</ListIssueDateTime>
        <NextUpdate>
            <dateTime>2018-03-25T00:00:00Z</dateTime>
        </NextUpdate>
        <DistributionPoints>
            <URI>http://www.miit.gov.cn/~/media/Files/It-
loesninger/NemID/TLCN.xml</URI>
            <URI>http://www.miit.gov.cn/~/media/Files/It-
loesninger/NemID/tlcnxml.sha2</URI>
        </DistributionPoints>
    </SchemeInformation>
    <TrustServiceProviderList>
        <TrustServiceProvider>
            <TSPInformation>
                <TSPName>
                    <Name xml:lang="en">Chinese Trust</Name>
                </TSPName>
                <TSPTradeName>
                    <Name xml:lang="en">VATCN-30808460</Name>
                    <Name xml:lang="en">TRUST2408</Name>
                </TSPTradeName>
                <TSPAddress>
                    <PostalAddresses>
                        <PostalAddress xml:lang="en">
                            <StreetAddress>Kina Street 10</StreetAddress>
                            <Locality>Beijing</Locality>
                            <StateOrProvince>Beijing</StateOrProvince>
                            <PostalCode>100037</PostalCode>
                            <CountryName>CN</CountryName>
                        </PostalAddress>
                    </PostalAddresses>
                    <ElectronicAddress>
                        <URI xml:lang="en">mailto:cvmio@ctrust.cn</URI>
                        <URI xml:lang="en">mailto:mamch@ctrust.cn</URI>
                        <URI xml:lang="en">https://www.ctrust.cn/en/</URI>
                    </ElectronicAddress>
                </TSPAddress>
                <TSPInformationURI>
                    <URI xml:lang="en">https://www.ctrust.cn/en/om-
nemid/historien_ctrust/oces-standarden/oces-
certifikatpolitikker/index.html</URI>
```

```
                    <URI
xml:lang="en">https://www.ctrust.cn/en/about_ctrust/citizen/conditions/</URI>
                </TSPInformationURI>
            </TSPInformation>
            <TSPServices>
                <TSPService>
                    <ServiceInformation>

<ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceTypeI
dentifier>
                        <ServiceName>
                            <Name xml:lang="en">Root Certification Authority
of TRUST2408 OCES Primary CA</Name>
                        </ServiceName>
                        <ServiceDigitalIdentity>
                            <DigitalId>
<X509Certificate>{{X509 certificate}}</X509Certificate>
                            </DigitalId>
                        </ServiceDigitalIdentity>

<ServiceStatus>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/recognisedatn
ationallevel</ServiceStatus>
                        <StatusStartingTime>2016-06-
30T22:00:00Z</StatusStartingTime>
                        <ServiceInformationExtensions>
                            <Extension Critical="true">
<AdditionalServiceInformation>
    <URI
xml:lang="en">http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignature
s</URI>
</AdditionalServiceInformation>
                            </Extension>
                        </ServiceInformationExtensions>
                    </ServiceInformation>
                    <ServiceHistory>
                        <ServiceHistoryInstance>

<ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/CA/PKC</ServiceType
Identifier>
                            <ServiceName>
<Name xml:lang="en">Root Certification Authority of TRUST2408 OCES Primary
CA</Name>
                            </ServiceName>
                            <ServiceDigitalIdentity>
<DigitalId>
    <X509SubjectName>CN=TRUST2408 OCES Primary CA, O=TRUST2408,
C=CN</X509SubjectName>
</DigitalId>
<DigitalId>
    <X509SKI>{{X509 SKI}}</X509SKI>
</DigitalId>
                            </ServiceDigitalIdentity>

<ServiceStatus>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/accredited</S
erviceStatus>
```

```
                            <StatusStartingTime>2010-03-
02T23:00:00Z</StatusStartingTime>
                        </ServiceHistoryInstance>
                    </ServiceHistory>
                </TSPService>
            </TSPServices>
        </TrustServiceProvider>
    </TrustServiceProviderList>
```