D5.2

Conceptual Framework for Delegations (2)

Document Identification	
Date	30.08.2018
Status	Final
Version	Version 1.0

Related WP	WP2	**Related Deliverable(s)**	D2.1, D2.14
Lead Authors	TUG	**Dissemination Level**	PU
Lead Participants	TUG, DTU, FHG, TUBITAK, USTUTT, GS, IBM	**Contributors**	See Table
Reviewers	G+D, 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 a conceptual framework for the Delegation Publisher in the LIGHTest project. The framework provides the basis of the work within WP 5. It was used as input to the deliverables D5.3 [1], and D5.4 [2]. Those two deliverables required preliminary results.

Delegations already have been used in schemes regarding electronic signatures and are used as a foundation for this deliverable. Chapter 6 gives an overview about the schemes, which are taken from the inventory deliverable [1].

In Chapter 7 two scenarios for publication and revocation of delegations are given. These examples show how a workflow could look like. Further, they extend the use cases given in the architecture [2] with a view from the delegation publishers side.

Chapter 8 elaborates the general concepts of delegations and gives an overview on delegation in academic papers. The general description provides an explanation on all basic delegation types and how they differ from each other. From this basic delegation types, the common fields for delegations are derived in section 8.1.2. Section 6.4 provides an overview on delegation in academic literature and summarizes them.

The conceptual view for the delegation publisher is presented in Chapter 9. The delegation publisher presented is a web application with a RESTFUL API. The API provides the user with methods to publish, search, download, and revoke delegations. The API is used by the Mandator, Intermediary, Proxy, and ATV. It provides the required data for the delegations to those actors. Further, this chapter offers information on the possible file formats, and describes the revocation process of a delegation.

Last, a short discussion and example on how to use the Trust Policy Language (TPL) to verify a delegation is given in Chapter Delegation in TPL10. In this chapter, we discuss the use of the TPL based on a purchasing example and provide a practical example on how an expert would write a trust policy for the verification of a delegation.

2. Document Information

2.1 Contributors

Name	Partner
Olamide Omolola	TUG
Georg Wagner	TUG
Peter Lipp	TUG
Edona Fasllija	TUBITAK
Sven Wagner	USTUTT
Sebastian Kurowski	FHG
Jesse Krutto	GS
Sebastian Mödersheim	DTU

2.2 History

Version	Date	Author	Changes
0.1	23/07/2018	Sebastian Mödersheim	Delegation in TPL
0.2	23/07/2018	Edona Fasllija	STORK Delegations
0.3	23/07/2018	Sven Wagner	Relationships update
0.4	24/07/2018	Georg Wagner	Update on the concept

3. Table of Contents

3.1 Table of Figures

3.1 Table of Tables

3.2 Table of Acronyms

AQAA	Attribute QAA, Quality of Attributes
ATV	Automatic Trust Verifier
CA	Certification Authority
CertID	Certifiate Identification
CRL	Certificate Revocation List
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DP	Delegation Publisher
ISI	Institute for Scientific Information
LoA	Level of Assurance
OCSP	Online Certificate Status Protocol
SME	Small and Medium Enterprise
SP	Service Provider
TCP	Transmission Control Protocol
TPL	Trust Policy Language
TSPA	Trust Scheme Publication Authority
TSP	Trust Service Provider
TTA	Trust Scheme Translation Authority
UDP	User Datagram Protocol
UTC	Universal Time Code
XML	eXtended Markup Language

4. Scope of the Deliverable

This deliverable documents the initial design of a conceptual framework for delegations in the LIGHTest architecture.

Delegations are an integral part of the LIGHTest architecture. This deliverable studies the delegations available in chosen trust schemes. It also elaborates on the relationship with other work packages such that there is concrete information on how delegations affect the other work packages in particular.

There is a wide range of scenarios where delegations are necessary and this deliverable evaluates them in a clear and concise manner.

There are several ways to implement electronic mandates depending on the organization or trust scheme. This deliverable studies the general concepts of electronic mandates. It also provides the data format / representations for electronic mandates and proposes tools that are used for both publication and detection of delegation.

This deliverable does not cover procedures on how delegations can be carried out. Such procedures are organization or trust scheme specific. We, however, focus on how to discover, authenticate and publish delegations using the LIGHTest architecture.

5. Terminology

This section contains the terminology used throughout the deliverable. It is important, that all readers of this document have the same understanding of the concepts. Furthermore, it makes communication between partners and participants easier. Parts of the terminology are reused from D2.14 [2] and D5.1 [3] to provide a comprehensive terminology dictionary in this document.

5.1 Delegation Provider

The Delegation Provider is a web component, which is used to publish delegations. It is used by other components by querying it. The Delegation Provider contains a repository for delegations and one for revoked delegations.

5.2 Verifier

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

5.3 Automated Trust Verifier

LIGHTest component used by the Verifier to verify documents. This component queries the delegation provider to verify the existence of the delegation and verifies that the delegation has not been revoked.

5.4 Mandator

Person or entity empowering another person or entity to act on behalf of itself. The Mandator is the creator of delegations and publishes delegations at the Delegation Provider of his/her choice.

5.5 Proxy

Person or entity empowered by a Mandator to act on behalf of another person or entity. The Proxy is the person or entity executing the delegation.

5.6 Intermediary

Person or entity empowered by a Mandator to find another person or entity to act on behalf of the Mandator. The intermediary acts as a link between Mandator and Proxy in the selection process. The parameters of the delegation are provided by the Mandator.

5.7 Electronic Mandate

Delegation which a Proxy has received from a Mandator and can act upon.

6. Existing and Relevant Delegation Schemes

This section provides information on existing schemes and their understanding of mandates as well as delegations in the academic literature. We have chosen the three schemes from the inventories deliverable [1] that represent different ways to issue mandates and are relevant in the European context of LIGHTest. These are STORK and its AQAA scheme, the Austrian Mandate Service, and the Finish tax system KATSO.

6.1 STORK – AQAA

The STORK2.0 Attribute Quality Authentication Assurance scheme (AQAA) has been extended over STORK QAA to enable quality levels to be assigned to attribute assertions in addition to the eID solutions. Service providers and Attribute/mandate providers are guided to identify the quality (AQAA level) of the services and attributes they provide respectively.

A mandate can be expressed as yet another attribute set provided by a mandate provider in equivalence to an attribute provider in the context of AQAA. Therefore, service providers use STORK2.0 infrastructure to verify the delegation of a legal entity (e.g. a person, organization…) by a mandate holder (the person/entity that represents the other entity) in the same way as attribute provisioning.

The data model of a mandate used in STORK2.0 AQAA is a simplified version of the Austrian mandate model. The model is comprised of the following:

I. The identity of the representative entity.
II. The identity of the represented entity.
III. The authorizations granted by the represented entity to the representative.
IV. Restrictions to the authorizations granted (e.g. time restrictions and transaction limits).

This model is defined by the following attributes, with mandate being one attribute that is composed by 3 underlying attributes as described in Table 1.

Table 1: Attributes of a mandate

Name	Type	Description
Representative	XML	The entity which received the permission to act on behalf of the represented entity.
Represented	XML	The entity granting the authorisations directly or indirectly to the representative so as to act on its behalf.

Name	Type	Description
MandateContent	XML	The authorizations granted by the represented entity to the representative entity along with restrictions.

There are three main processes defined in the STORK2.0 AQAA scheme that are related to persons acting on behalf of others, namely *authentication on behalf of* and *powers for signature validation,* where natural persons act on behalf of other persons like SMEs , and the *powers validation*, which involves the validation of powers stored at service providers. A brief description of these three main processes follows.

Authentication on behalf is the process that authorizes a representative to access privileged data of the represented person. At the end of this process, the representative and the represented person are fully authenticated, their eID data is transferred to the SP, and this SP recognizes this user as a representative of a known person.

Powers (for digital signature) is the process of verifying that a representative has enough power to represent the represented person. This happens in the case where a service provider has received the digital signature of the representing person on behalf of the represented person. This process is quite similar to the *Authentication on behalf* process, the main difference being the initiating action from the SP side.

Powers Validation is the process of verifying the validity of representation powers. This process involves the Service Provider that needs to verify that the powers are still valid and not revoked. In case the powers are still valid, the mandate data is returned.

A sample use case is depicted below.

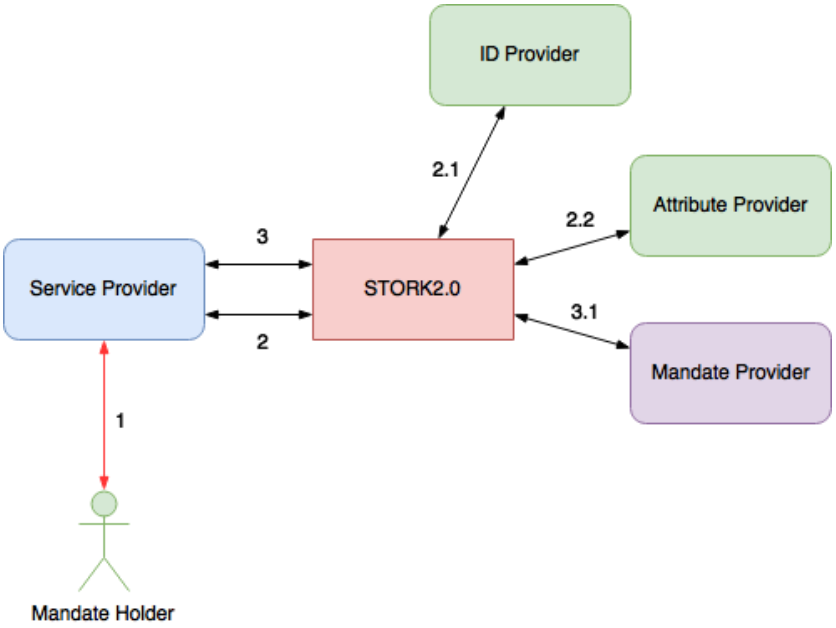

Figure 1 A sample STORK2.0 delegation use case

The steps of the use case are as follows:

1. The Proxy requests a service on behalf of Mandator.
2. The service provider uses STORK2.0 infrastructure to check the eID of the Proxy and authenticate her.
   2.1. STORK2.0 queries the ID provider of the Proxy for eID information.
   2.2. STORK2.0 queries the Attribute provider of the Proxy in order to obtain attribute information.
   2.3. The user is authenticated.
3. The service provider queries and verifies the mandate information from STORK2.0
   3.1. STORK2.0 accesses the mandate provider of the entity being delegated, gets the mandate information and validates it.
   3.2. The necessary information is returned to the service provider.
4. The service provider decides to accept or reject the service request from the Proxy with respect to the AQAA quality level of the eID, attributes, the mandate information and the corresponding levels that the service provider itself requires for that specific service.

The service provider may choose to or not to store the mandate information for further efficiency of its services. However, the service provider is required to verify the validity of the mandate information for each service it provides.

6.2 Austria MOA/MOA-ID

SourcePIN Register Authority, i.e. the data protection commission, operates the Austrian mandate service. The figure below shows the process model of the online mandate service.

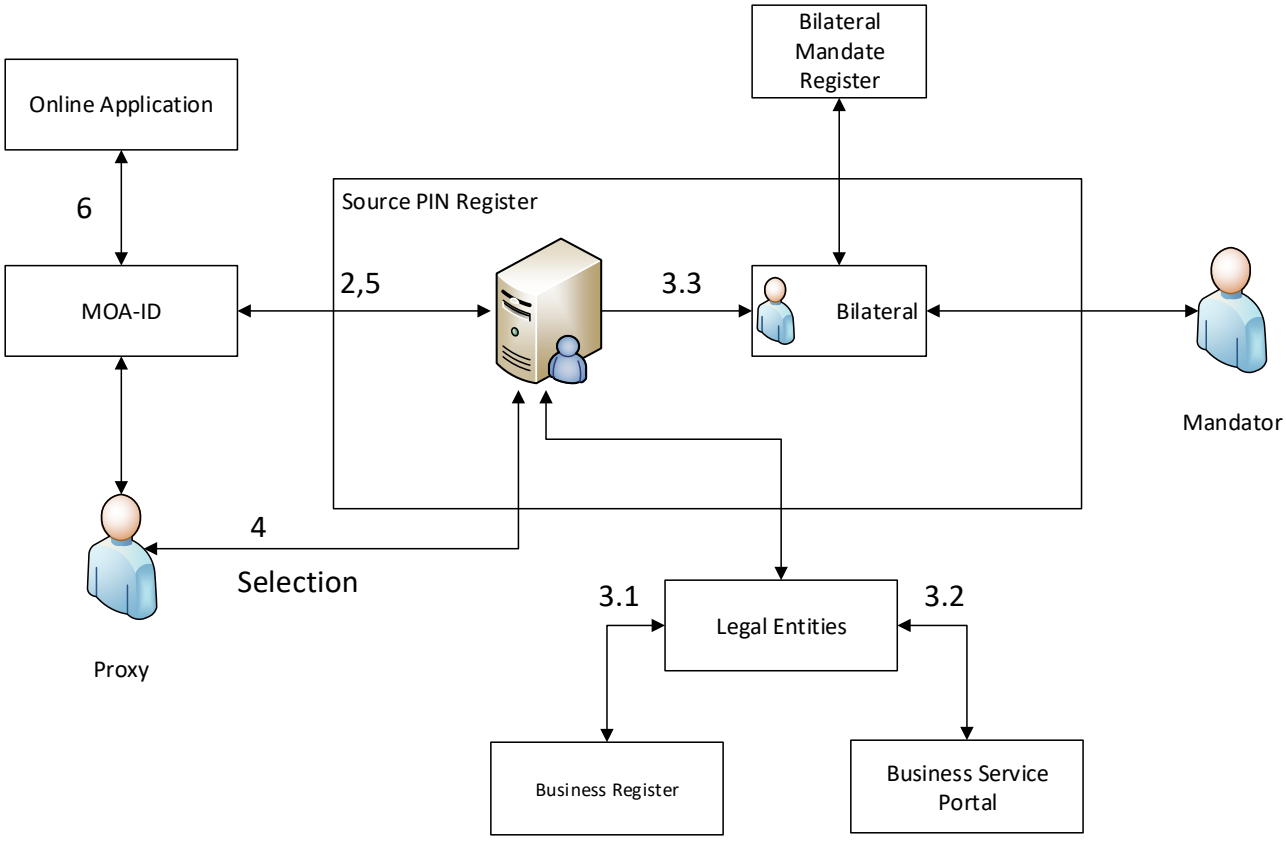

Figure 2: Architecture of the online mandates service. [6]

The essential steps of an authentication with the use of mandates are explained in the following list.

1. The authentication of an online application on behalf begins like a normal authentication to a service without having a mandate in place. In addition, however, the “authentication on behalf” option has to be selected to make use of the MOA-ID service.
2. In the case of a proxy, MOA-ID accesses the online mandate service of the SourcePIN Register Authority. The communication between MOA-ID and the online mandate service is handled by a SOAP-based WEB service, whereby the login data (identity link and certificate) of the representative is transferred to the online mandate service.
3. The online mandate service uses the information transferred by MOA-ID to obtain proxy information from different registers.
    1. Business register for legal powers of sole representation of
        - companies registered in the Austrian Register of Company Names
        - associations registered in the central Register of Associations

- legal entities registered in the Supplementary Register for Others.

2. Business service portal for arbitrary mandates of legal entities (e.g. for Lawyers, etc.)
3. Bilateral mandate register of the SourcePIN Register Authority. The SourcePIN Register Authority operates a service for the registration of bilateral mandates between private individuals. Persons (proxies) can login to the bilateral register by means of a Citizen Card and grant other persons special mandates.

4. Afterwards, the representative is forwarded to the online mandate service where the representative makes the choice of online mandate.
5. After choosing the online mandate, it is transferred to MOA-ID where MOA-ID checks the electronic signature of the online mandate and concludes the authentication process.
6. In the last step, the authentication data and the selected online mandate is transferred to the online application.

An important application case involves authorized professional representatives or alternatively legal professionals authorized for representation such as lawyers, notaries or civil engineers. Due to their professional qualifications, they can act on behalf of clients, whether they are private individuals or legal entities. In this case, the capacity of the authorized professional representative or legal professional authorized for representation is checked by the online mandate service, whereby the certificate conveyed by MOA-ID contains the capacity and must be valid at the time of the check. Afterwards, the authorised professional representative or legal professional authorized for representation can intervene for any private individual or legal entity.

The Austrian MOA/MOA-ID system does not support electronic revocation of mandates. [4] argues, that a revocation is already sufficient if publicly announced. Similar to paper based delegations, the proxy needs to destroy the copies of the delegation. This is already hard to prove with paper-based delegations and becomes even harder with electronic delegations as the proxy can create arbitrary instances of the delegation itself. The mandator needs to trust the proxy in all cases that the proxy destroys all the delegation material available at the proxy side. However, the delegations provided by MOA/MOA-ID provide a validity date and thus are only valid for a certain period.

6.3 KATSO

KATSO federates business entity identities across the public sector in Finland. Besides companies registered in Finland, The Finnish Tax Office enables registration of foreign nationals via an online service.

The legal signatory of a Finnish Company (or authorized user on their behalf, acting via a mandate granted to him/her) may not be a Finnish citizen, may not live in Finland and may not have a Finnish ID number. Using KATSO, they are able to automate the registration process and still provide cost-effective two-factor authentication using printed one-time password lists.

Foreign companies must file an income tax return or give a report of their operations, if they receive a tax return form or a request from the Tax Administration to provide such information.

The required form is still available as a paper version, but the goal is to eventually migrate to 100% e-service operation. If the company has operations in the building, construction or installation sectors, it is required to report on them every year, regardless of whether it has to file a Finnish income tax return or not.

Mandates are an important part of KATSO. Without a legally binding Power of Attorney mandate, the primary company representative would have to personally manage all of the company's e-service use. The workflow starts by first checking the roles and period of validity that the Mandator has specified in the new Power of Attorney to be issued. Then the representative should either accept or reject the Power of Attorney. Acceptation is done via an online signature. To go over the details of the new Power of Attorney, the representative should confirm his or her identity (via bank ID or a government-issued smart card) or visit a tax office personally. The online Power of Attorney will then become effective as soon as the relevant authority has confirmed the representative's right to sign the company name from the business register.

Revoking a delegation in KATSO can be done in two ways. The first method is to visit a tax office dealing with KATSO registration and place a request to revoke the delegation. The second method is to delete the delegation from the KATSO management interface online. To achieve this on the KATSO management interface, the user logs in to the system using the appropriate credentials and navigates to the pages where the delegations are listed. The delegation can be deleted without any further process.

6.4 Delegation in academic publications

In addition to the overview of existing, relevant delegation schemes in this section, this section gives a short overview on delegation schemes published in academic journals. For this purpose, a literature review using ISI Web of Knowledge with the keywords "delegation" and "identity management" was conducted and by the time of writing 20 results were found. The results indicate that different types of delegation exist. The delegation types and most relevant publications are listed in Table 2 and the key findings of these publications are summarized below.

Table 2: Delegation Types and their most relevant publications

Delegation Type	References
Delegation of Authentication	Raja and Razak (2015): Analysis of Security and Privacy in Public Cloud Environment. International Conference on Cloud Computing (ICCC), p53-58 Ma and Woodhead (2006): Authentication delegation for subscription-based remote network services. Computers & Security, p371-378

Delegation Type	References
Delegation of Authority	Karp at al (2010): From ABAC to ZBAC: The Evolution of Access Control Models. 5th International Conference on Information Warfare and Security, p202-211 Cho et al (2007): A unified user consent acquisition and delivery mechanism for multi-source user data integrated service. IEEE International Symposium on Consumer Electronics, p309-315 Hulseboch (2003): Privacy in content distribution networks - A framework description. 18th International Conference on Information Security, p435-440
Delegation of Identity	Zhang and Chen (2011): A Delegation Solution for Universal Identity Management in SOA. IEEE Transcations on Services Computing, vol. 4, p70-81 Garcia and Oliva (2010): Improvements of pan-European IDM Architecture to Enable Identity Delegation Based on X.509 Proxy Certificates and SAML. 4th International Workshop on Information Security Theory and Practice, p183-198 Garcia and Oliva (2009): Solving Identity Management and Interoperability Problems at Pan-European Level. On the Move Confederated International Conference and Workshops, p805-809 Golodoniuc et al (2015): PID Service - an advanced persistent identifier management service for the Semantic Web. 21st International Congress on Modelling and Simulation (MODSIM2015), p767-773
Delegation of personal data	Wohlgemuth et al (2010): On Observable Delegation of Personal Data by Watermarking. 6th IEEE Consumer Communications and Networking Conference, p1143-+
Delegation of Privacy-Relevant Actions	Hansen et al (2009): Delegation for Privacy Management from Womb to Tomb - A European Perspective. 5th IFIP WG 9 2, 9 6/11 7, 11 4, 11 6/PrimeLife International Summer School, vol. 320, p18-+
Delegation of Rights	Prochazka et al (2014): Perun - Modern Approach for User and Service Management. IST-Africa Conference and Exhibition Proceedings Bussard et al (2008): An Approach to Identity Management for Service Centric Systems. 1st European Conference ServiceWave, p254-

Delegation Type	References
	Wohlgemuth and Mueller (2006): Privacy with delegation of rights by identity management. International Conference on Emerging Trends in Information and Communication Security, p175-190 Kruk et al (2006): D-FOAF: Distributed identity management with access rights delegation. 1st Asian Semantic Web Conference, p140-154

For the delegation of authentication, Raja and Razak (2015) emphasize that anonymous authentication, revocation, unlinkability and delegation of authentication for multiple cloud services are obligatory user privacy parameters that require to be addressed through identity management services in the cloud. Ma and Woodhead (2006) describe a system that provides controlled access to subscription-based remote network services through a browser and plug-ins are used to provide an authentication-delegation service and a policy-based authorization service.

For the delegation of authority, Karp at al (2010) have developed a scalable authorization based access control (ZBAC) approach with general SOA security and inter-domain trust based on authority delegation and the use of trust anchors between communities. Cho et al (2007) propose a unified user consent acquisition and delivery mechanism for multi-source user data integrated service by introducing a delegation authority (DA) for user consent acquisition and delivery. Hulseboch (2003) propose a framework for secure delivery of personalized content in the presence of a transparent intermediary while the privacy of the user is being preserved. The framework includes delegation of authority to the intermediary content distributor allowing him to act on behalf of the content provider.

For the delegation of identity, Zhang and Chen (2011) propose a practical delegation solution for universal identity management using pseudonym-based signature scheme. A proxy signature is presented with the pseudonyms as public keys where delegation can be achieved through certificate chains. Garcia and Oliva (2010) propose an architecture based on X.509 Proxy Certificates and SAML assertions to enable delegation in provision of services in the complex and heterogeneous by public institutions in the EU. Golodoniuc et al (2015) develop a web service offering advanced persistent identifier management PID Service. It enables stable identification of digital objects for Semantic Web and Linked Data applications including automatic data harvesting and digital entity identification.

For the delegation of personal data, Wohlgemuth et al (2010) investigate on digital watermarking in order to observe the enforcement of obligations for a delegation of personal data without a trusted third party.

For the delegation of privacy-relevant actions, Hansen et al (2009) elaborate on delegation of personal data under a lifelong perspective and point out possible legal, technological, and organizational measures to appropriately take up the arising challenges.

For the delegation of rights, Prochazka et al (2014) introduce the identity and access management system Perun, which provides functionality covering management of the whole user life cycle in nowadays e-Infrastructures. The Perun system supports management of virtual organizations, rights delegation, group management and enrolment management for making flexible user management easy to use. Bussard et al (2008) address the problem of identity management for service-centric systems and proposes a novel approach based on an abstract delegation framework supporting different access control mechanisms. The abstract delegation framework is designed to give control and clarity to the user consuming applications based on service composition. Wohlgemuth and Mueller (2006) propose a generic privacy-preserving protocol with delegation extension for sharing identifying attributes as credentials with others. Kruk et al (2006) present a distributed identity management with access rights delegation, which deploys social networks.

The comparison of this literature review for delegation in academic journals with the proposed scenarios for delegations within LIGHTest, publishing delegation process (see Chapter 7.1) and revocation of a delegation process with archive (see Chapter 7.2) shows the following. There is no direct application or transfer of one of the described methods in literature to the proposed scenarios for delegations. However, the described delegation types also occur in the proposed scenarios and these publications provide a helpful and useful overview about the importance and possibilities of delegation processes.

7. Scenarios for Delegations

This section contains scenarios from the Delegation Publishers point of view. It tries to give a brief overview on how the architecture can handle delegations and which steps are necessary to create and revoke delegations.

The following scenarios do not cover the usage of a delegation and the expiration of a delegation. The usage of a delegation during the verification process has been covered in detail in D2.14 [4]. Expiration of a delegation is a simple process where the ATV has to check a field inside the delegation data and take a decision if the delegation is valid or not. Expired delegations do not necessarily need to be removed the Delegation Provider. Expired delegations can still be queried. Thus, the ATV handles expired delegations in the same way as valid ones.

7.1 Publishing Delegation Process

This scenario focuses on the creation and deployment of delegations. It shall provide a clear picture how the handling of delegations within the processes of creation and deployment is done.

Situation:

A company wants to empower an employee to do purchasing tasks on behalf of the company. The employee shall only have the allowance to do purchases up to a certain amount. All purchases above that amount require the authorisation of the CEO of the company.

Further, assume that the employee is within the same trust scheme as the company, so that no trust translation is required. Furthermore, delegations themselves do not require translation themselves. In case of translation always the certificate, signature, etc. are translated.

In this scenario, the employee receives the delegation for a special purpose, as the employee will be able to do purchases on behalf of the company. For the ATV to find the delegation, the delegation information has to be included in the transaction itself. The Proxy provides the delegation information for the transaction. To do so, the Proxy needs to download the delegation from the Delegation Provider and during the creation of the transactions signature container embed it there.

During the creation, the Proxy embeds the delegation information in the signature container. The signature container contains the delegation information in a fixed position. This is necessary for the ATV to discover the delegation in the signature container. The ATV will look for a file named delegation.xml in the container. In this file, the ATV can find all delegation information and the address of the Delegation Provider. With this address, the ATV can later check if the delegation has been revoked since the delegation was issued.

This scenario shows how a delegation is published at the Delegation Provider and provides an overview of the publication process. The full publication process is described in D5.3 [5].

Information flow in the Architecture:

To support the understanding of the information flow, Figure 3 shows the parties and parts required and how they interact. A detailed description can be found in the following description.

1. The Company starts the delegation process with the Delegation Provider.
2. The Company requests the certificate from the Employee, if it is not yet stored somewhere.
3. The Employee provides the certificate.
4. The Company creates the delegation with the following data:
    a. Delegation data in this particular case covers all actions the Employee will be able to do. In this case the delegation will cover all expenses for purchases up to a limit of €1M.
5. The Company signs the delegation.
6. The Company creates encryption keys for the delegation.
7. The Company encryptes the delegation and encryption keys.
8. The Company uploads it to the Delegation Provider
9. The Delegation Publisher issues a receipt to the company.
    a. The issued receipt contains all information about the delegation for the Company.

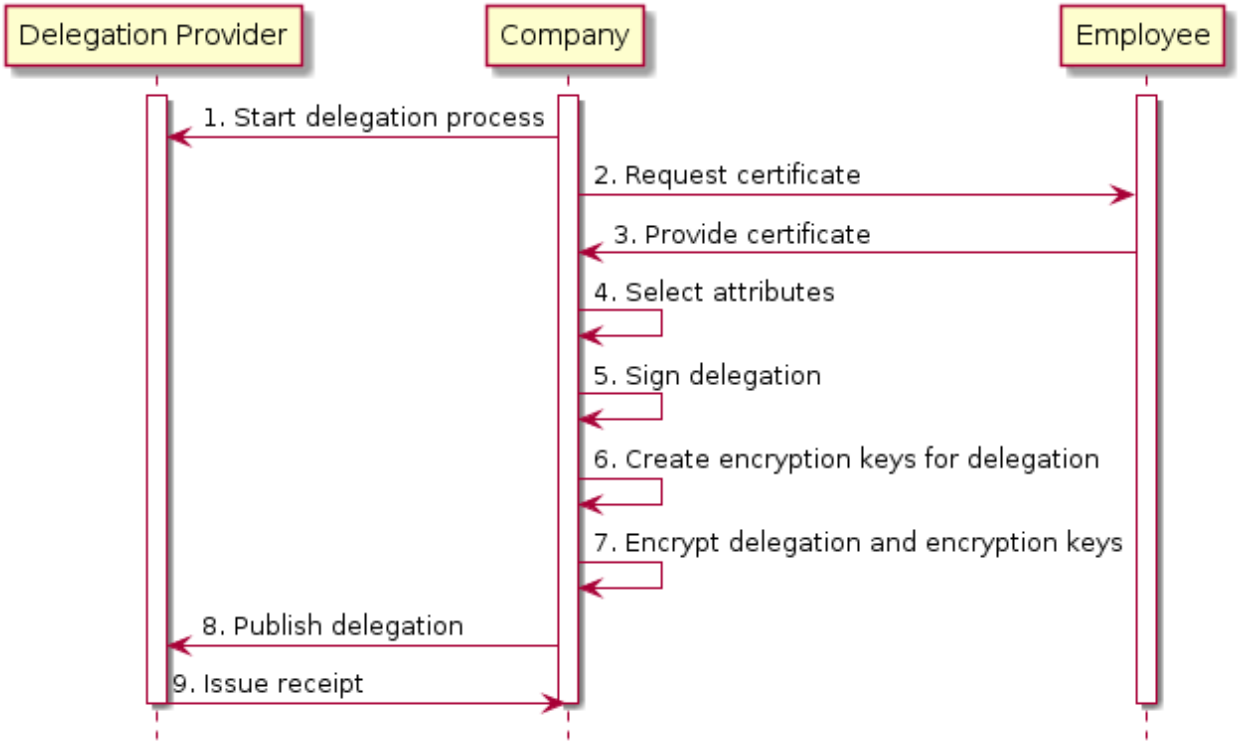

Figure 3: Creation process of a delegation

7.2 Revocation of a Delegation Process

This scenario has a focus on the revocation of a delegation. It shall provide a clear picture on how a possible process for the revocation of a delegation can look like and provide an overview on the actors involved in this project.

Situation:

An employee leaves the company. The employee holds a delegation for purchasing purposes. Now the company has to remove the delegation because the employee does not have an allowance to do purchases on behalf of company anymore.

This scenario assumes that an employee has a working contract with a company and a delegation to purchase on behalf of the company. In any case the delegation is not deleted, but is put on a revocation list. This revocation list can be queried using OCSP. OCSP is successfully used within CAs for revocation of certificates, and can thus be easily adopted for the cause of LIGHTest.

Information Flow in the Architecture:

To support the information flow, Figure 4 shows the parties and their interactions in the process of the revocation of a delegation. A detailed description can be found in the following description.

1. The Company decides to revoke the delegation for an employee.
2. The Company starts the revocation process.
3. The Company searches for the correct delegation details at the Delegation Provider
4. The Delegation Provider collects all the required data for the revocation.
    a. Data to be collected could be reasons why the employee got the delegation revoked.
    b. Archiving period needs to be set. Some data may only be required for a certain amount of time. This could be done by setting the records validity time to the end of the archiving period.
5. The Delegation Provider prepares the revocation list for the delegation.
6. The Delegation Provider issues a receipt to the company.

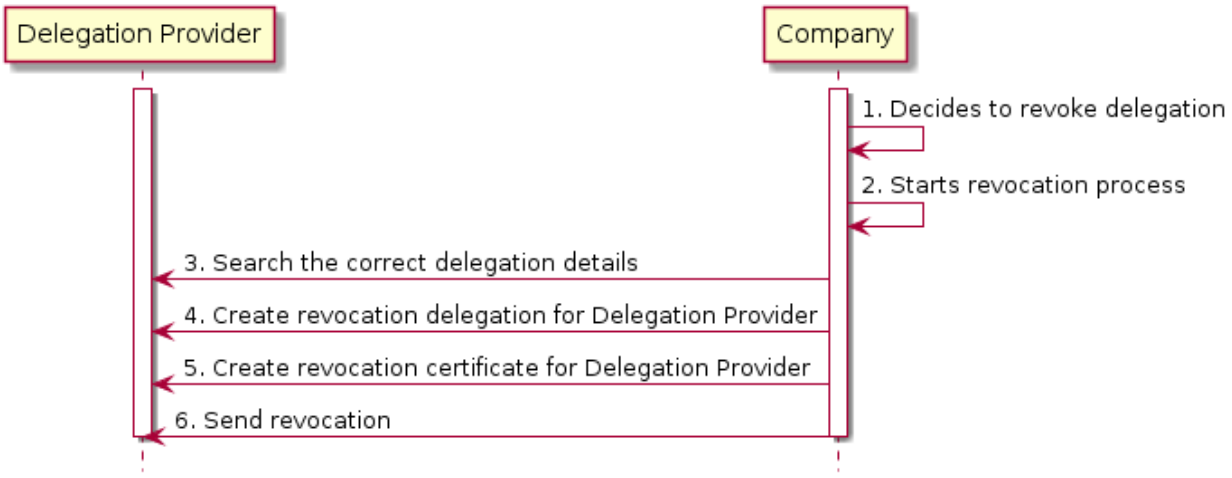

Figure 4: Revocation process of a delegation

8. Conceptual Framework for Delegations

8.1 Concepts

This section describes the general concepts related to electronic mandates/delegations that LIGHTest aims to support. The main actors in an electronic transaction that involves an electronic mandate are described first, followed by the types electronic mandate LIGHTest aims to support. Furthermore, the concepts related to the content of an electronic mandate and four main usage scenarios are analysed.

An **electronic mandate** is a credential issued by a Mandate Authoritative Source asserting that a representative is empowered to act on behalf of a mandatory in electronic transactions bound to a certain scope.

The main actors in an electronic mandate transaction are as follows:

1. The Mandator, who is the original possessor of the rights and delegates these rights to the representative to act on behalf of him/her.
2. The Proxy, which is the authorized person/entity these rights are transferred to.
3. The Intermediary, which is the person/entity that may act as a proxy for the establishment of a mandate between the Mandator and the Representative.
4. The Mandate Authoritative Source, which issues an electronic mandate, and provides means for its verification/revocation.

The mandate types involving the aforementioned actors can be one of the following combinations:

(i) A natural person can represent a legal entity.
(ii) A natural person can represent another natural person.
(iii) A legal person/entity can represent a legal person/entity.
(iv) A legal person/entity can represent a natural person.

These examples only sketch what mandates can do. By chaining multiple mandates together it is possible to create even more complex situations, but all complex scenarios can be reduced to these four simple combinations. At the end of each chain the Proxy is always a natural person taking the decisions. Thus, a natural person can represent a legal person through a chain of mandates and can represent a natural person through a chain of mandates.

8.1.1 Types of Representations

There are three main mandate types, namely bilateral, substitution, and delegation. This section explains how those mandate types work and what they can be used for.

Figure 5: Bilateral Type Representation

A bilateral mandate is the most basic type. Here a Mandator empowers a Proxy to act on behalf of the Mandator. This is also a direct representation. This representation is shown in Figure 5.

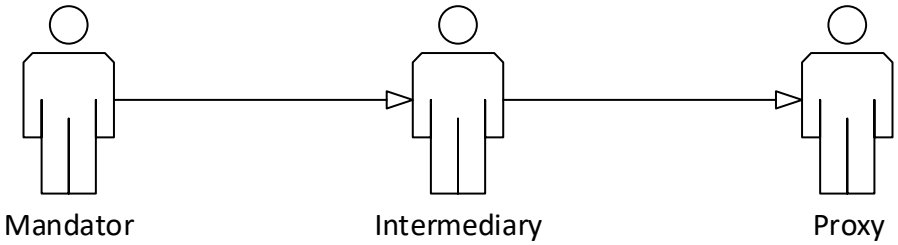

Figure 6: Substitution Type Representation

A substitution mandate is an indirect representation, which involves an Intermediary actor. The first part of the delegation from the Mandator to the Intermediary is a bilateral type delegation. To create the indirect representation, the Intermediary must have the consent from the Mandator to do so. The first condition together with the allowance for substitution creates the precondition for this type of delegation. The Intermediary then chooses a substitute for himself and allows the substitute to act as Proxy for the Mandator. The Mandator empowers the Intermediary to represent and gives the permission to add a substitute instead of the Intermediary. The substitute is then empowered to represent the Mandator and becomes the Proxy of the Mandator.

The substitution type delegation usually causes chains of mandates. In LIGHTest this type of delegation is allowed.

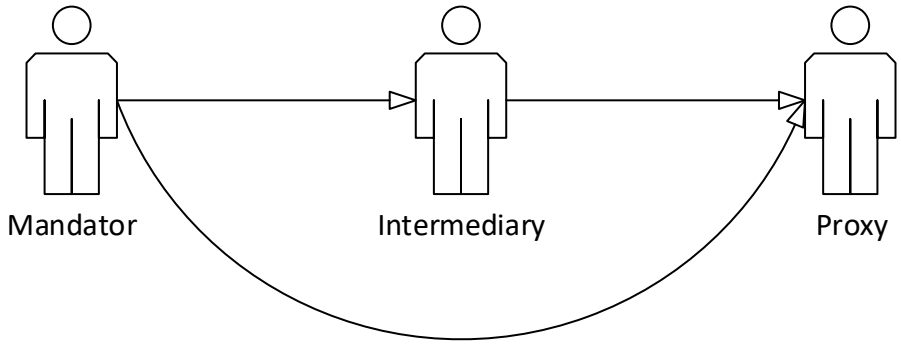

Figure 7: Delegation Type Representation

A Delegation Type mandate is composed of an indirect relationship establishment step as in the Substitution type mandate, followed by a direct relationship establishment between the Mandator and the Proxy, by eliminating the Intermediary. In the first step, the Mandator creates a bidirectional delegation to the Intermediary. Similar to the substitution type, the Intermediary now empowers the Proxy to act on behalf of the Intermediary. So far this seems like a normal substitution type. The main difference between the substitution type and the delegation type mandate is, that once the connection between Mandator and Proxy has been established the Intermediary disappears. The Intermediary is only needed to create the connection between the Mandator and the Proxy.

8.1.2 Content of a Delegation

The content of a mandate is split into two groups to in order to use them for multiple purposes. LIGHTest does not only have one special purpose and the technology behind LIGHTest can be used for many different applications. This requires mandate to be flexible. Figure 8 provides a schematic representation of data stored in delegations.

The first group of fields are mandatory as they must exist in every delegation as without them no delegation is possible. Mandatory fields are there to define the rough boundaries of a delegation and provide information about the Mandator. The content of a Mandate can be comprised of the following fields:

1. Identity information of the Representative, Mandator and Intermediary, together with their level of assurance
2. Date of Issuance
3. Validity time
4. Domain Specific (Scope of the empowerment), which defines the boundaries of how and where the mandate can be used, and is dependent on the business domain of the electronic transaction
5. General Restrictions regarding the usage of the mandate, such as disallowing further sub-delegations

6. Mandate enrolment Level of Assurance, which depends on the mandate issuing process of the Mandate Authoritative Source, the LoA of the eID, etc.
7. Secure Container showing that the content of this mandate has been issued by a Mandate Authoritative Source, for example, the mandate is signed by the Authoritative Source.

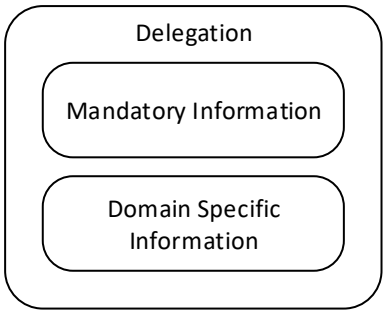

Figure 8: Schematic representation of the data saved in a delegation

The second group of a delegation is domain specific. Just the mandatory information does not limit the proxy at all except for the time in which the proxy is allowed to act on behalf of the Mandator. Domain specific information is required to limit the power of the Proxy. A full description of the fields in the delegation can be found in Table 3.

Unfortunately, domain specific information heavily depends on the area where LIGHTest is used and a taxonomy of powers is required in order to provide the domain specific information.

8.1.3 Use Cases of a Delegation

There are four main use cases with electronic mandates. Namely, creation, usage, expiration, and revocation. The creation use case describes the scenario of the person/entities applying for an electronic mandate to a Mandate Authoritative Source. The usage scenario is where the person/entity uses the delegation to sign transactions. The expiration use case describes the normal aging of a delegation. After a certain date, given by the Mandator, the Delegation becomes invalid and cannot be used anymore. The revocation use case describes the revocation of a delegation. If the Mandator does not require the Proxy anymore, the delegation can be revoked and thus becomes invalid and cannot be used anymore.

9. Conceptual View on Delegations

A delegation publisher operates a web based server component with an REST API, which does the main work. The Delegation Publisher therefore provides the capability for the ATV to look up the association of a Proxy with a Mandator, and the properties of the given (Figure 9).

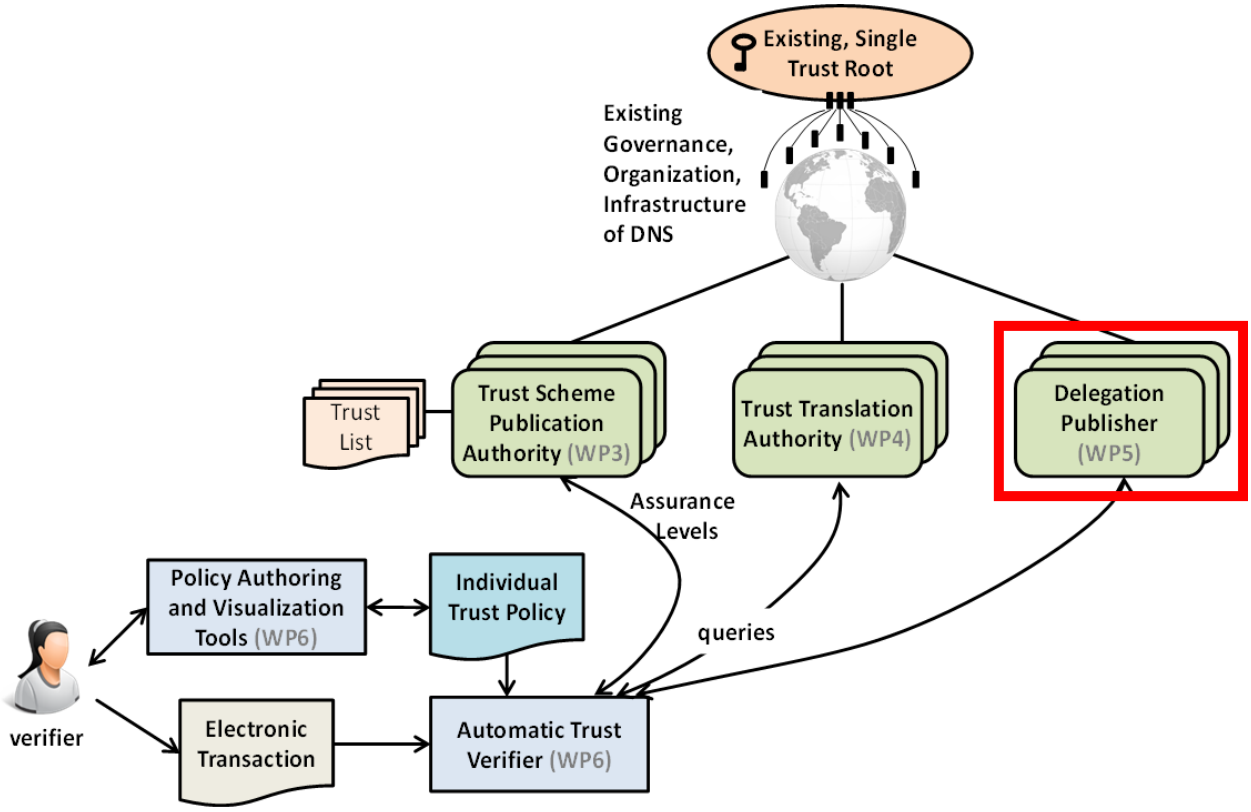

Figure 9: Delegation Publisher location in LIGHTest

The Delegation Publisher consists of a web based component with a RESTFUL API. With this API the Delegation Provider provides the functionality to publish, search, download, and revoke delegations. Each delegation is treated as an individual data set, and are stored as unique individual data sets. This storage method can be used to publish the different delegation types as seen in section 8.1.1.

The Delegation Publisher provides the ability to create the most basic types of delegations, namely (i) bilateral type, (ii) substitution type, and (iii) delegation type.

9.1 Querying the Delegation Publisher during Verification

The LIGHTest Reference Architecture [2] introduces scenarios in which the Delegation Publisher is contributing to the automated trust verification. In this scenario, the Delegation Publisher is

queried with the hash of the delegation. The Delegation Publisher then provides the required information to the verifier to complete the verification.

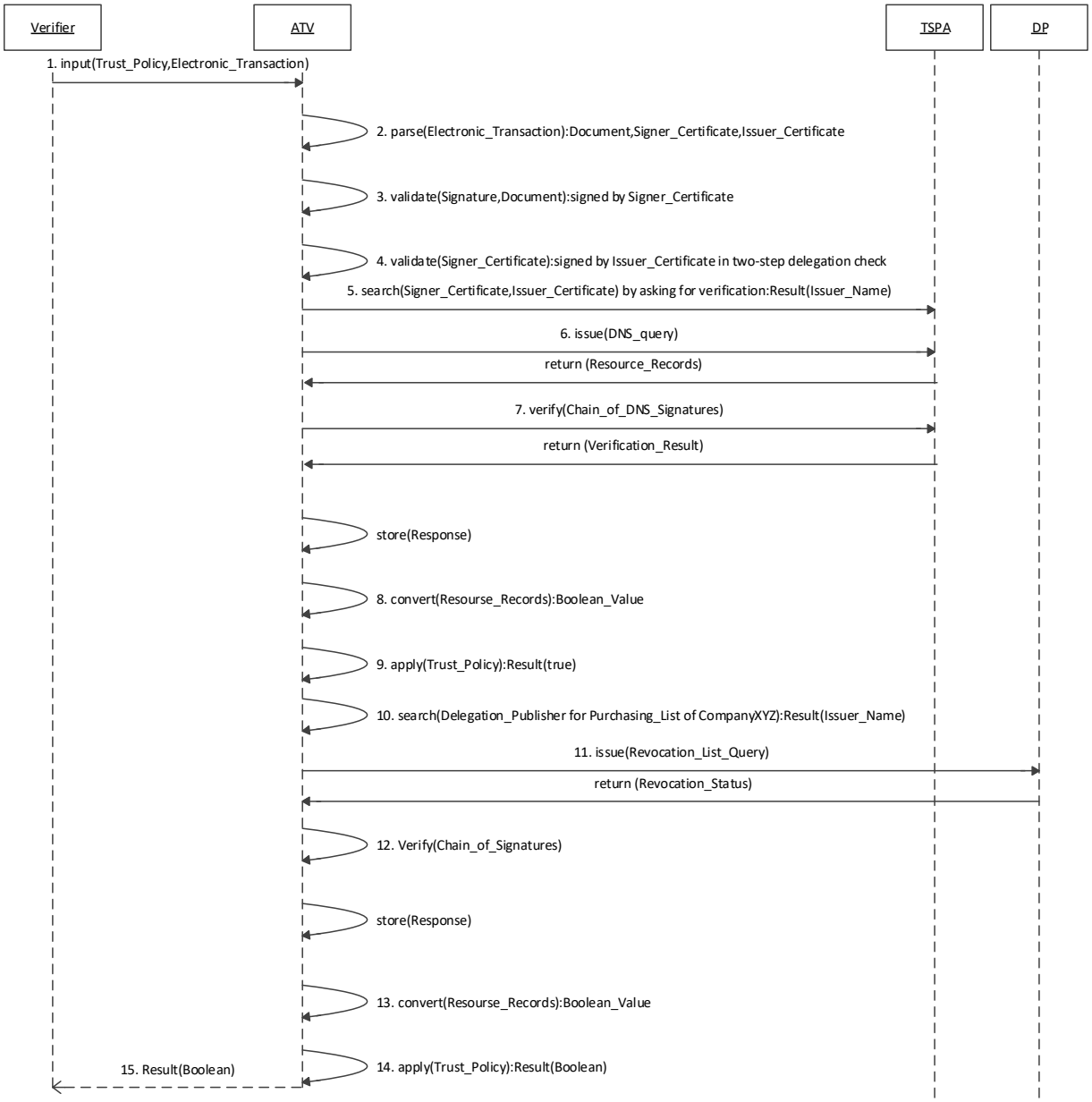

Figure 10: Sequence Diagram for Trust Delegation Scenario (from [2])

In the provided scenario from the reference architecture [2] (see Figure 10), the ATV extracts the certificates contained in the transaction (Step 2) and validates the signatures obtained in the previous step together with the document (Step 3 and Step 4). Next the discovery information for the issuer is extracted and the TSPA is queried (Steps 5 to 9). Next, a discovery of the delegation publisher is done (Step 10).

At Step 11, the ATV knows the Issuer Name, but not if a delegation exists. Therefore, a discovery of an existing delegation is required. The following notation aims at indicating the exchanges between the ATV and the Delegation Publisher in Alice and Bob notation.

1. ATV: URL of the Delegation Provider
2. ATV $\rightarrow$ DP: ATV contacts the DP and provides the SHA256 hash of the delegation as an OCSP request.
3. DP $\rightarrow$ ATV: DP searches for the revocation if it exists, and signs the response that is sent to the ATV.
4. ATV: Verifies in case of a revocation, if the revocation is valid. Therefore, the ATV verifies the signature on the revocation.

The content of the delegation is evaluated by the ATV and it is not part of this deliverable to explain how the ATV has to handle the detailed information, if more information is required please refer to D6.2 [8]. However, the specification on how the detailed content looks like is given in the following sections of this deliverable.

9.2 Publication of Delegations

An overview on the concept for the publication on the delegation publisher is given in Figure 11. The Delegation Publisher receives and stores the signed and encrypted delegation from the Mandator and stores the delegation in the internal database. The Delegation Publisher also receives the key for the delegation and some other meta-information. The key for the delegation is also encrypted. This ensures, that the Delegation Publisher has no knowledge about the internals of the delegation, and just stores the data. Later, the Proxy can use his public key to search for, and download delegations.

The communication between the Mandator and the Delegation Provider is done via a RESTFUL API. Within the RESTFUL API, the Mandator can use the HTTP commands GET, PUT, and POST to receive and send data. The data needs to be send to the right endpoint on the server, so that it can be saved in the right location.

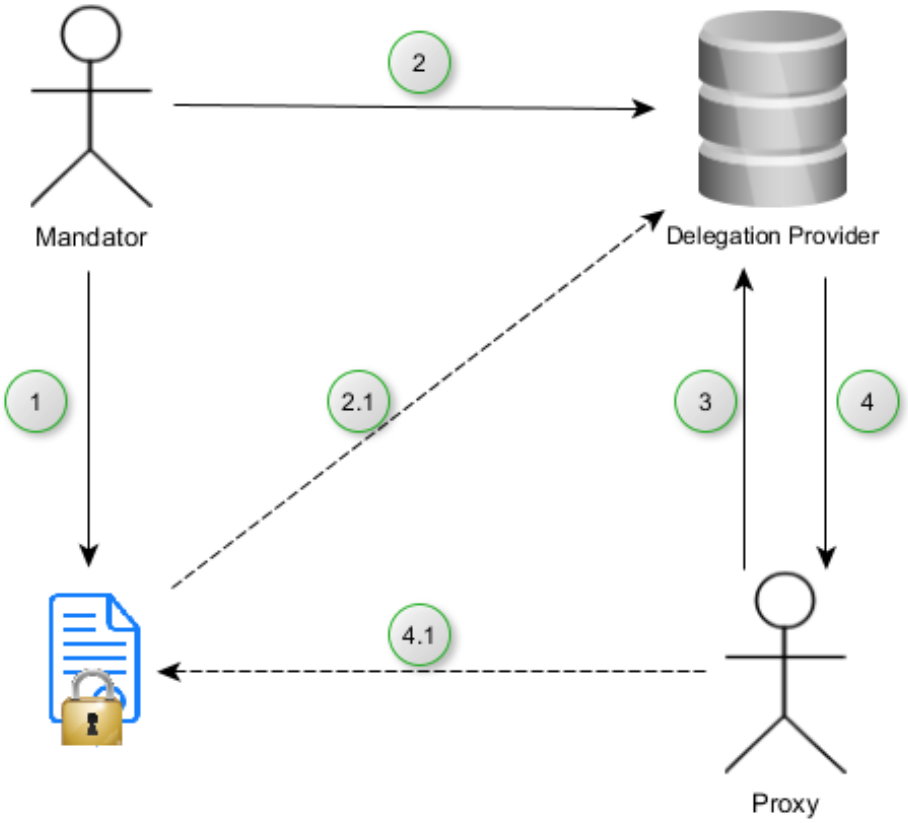

Figure 11: Creation, Publication, and Discovery of a delegation (from [5])

9.3 Representation of Delegations in the Delegation Provider

The information saved in the Delegation Provider is stored encrypted. However, the encrypted data holds the information in an XML format. The format is based on the ETSI 119 621 [9], and the file format presented in [6].

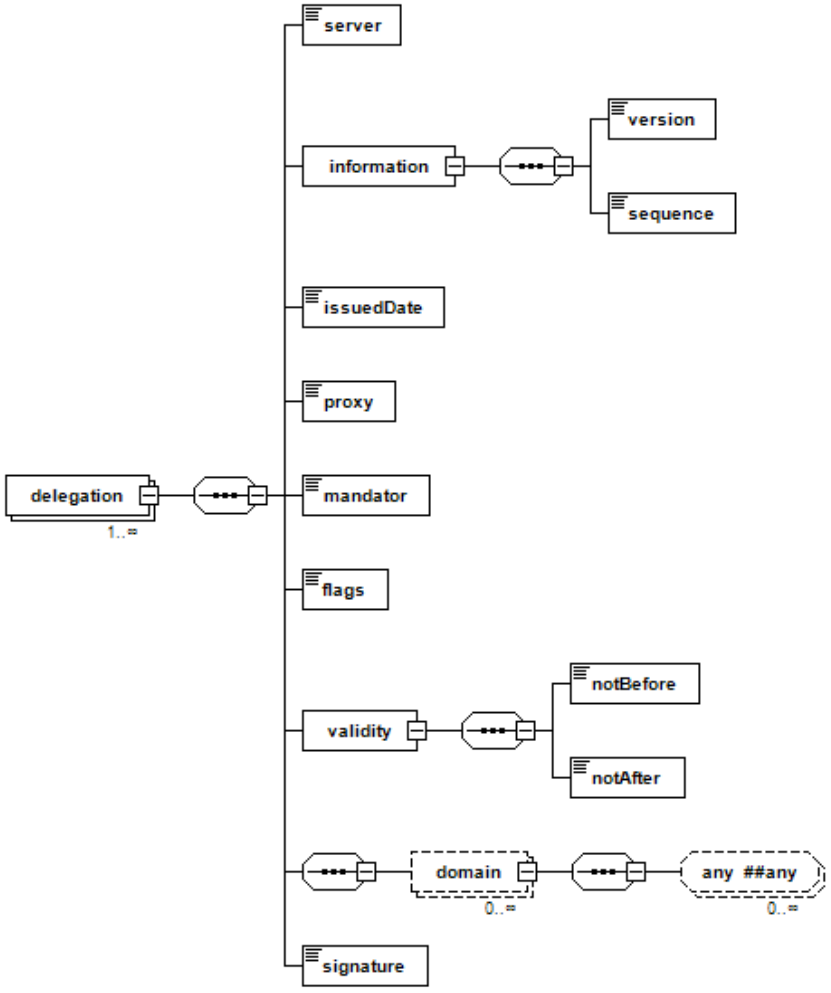

Figure 12: XML style sheet for delegations.

The file can contain one or more delegations. One delegation is a simple bilateral delegation, while with multiple delegations, more difficult and sophisticated delegation patterns can be build, like the substitution or delegation type of delegation. Each block in the delegation is signed individually by the Mandator who issued it. This way the delegation can be customized. During publication and verification, the delegation needs to be verified, if the data in the delegation are valid. The verification process needs to check if the original delegation (normally the first one), is not violated. A violation can occur, if an Intermediary changes one of the parameters and changes it to a value that was never allowed by the original delegation.

Each Delegation Information contains a Version Identifier, Sequence Number and List Issue Date and Time field. The Delegation is divided into General and Metadata. The General part contains the Proxy; certificate of the delegate, and the Mandators information. The Metadata

field is divided into Mandatory; containing a Date of Issuance, Validity Time, General Restrictions and a Scope of Empowerment; and Specific; which contains the Domain Specific Fields. Detailed information on the content of the fields is found in Table 3.

Table 3: Description of the fields of a delegation

Field	Presence	Format	Description	Value
Version Identifier	Yes	Integer	Specifies the version of the list	Fixed Value of 1
Sequence Number	Yes	Integer	Specifies the Sequence Number of the published list	1 at start and incremented for every iteration of the list
IssuedDate	Yes	DateTime	It specifies the date and time when the delegation was issued	Universal Time Code (UTC) when the delegation was issued
Proxy	Yes	Unicode String	Certificate of the Delegate	Electronic certificate of the delegate
Mandator	Yes	Unicode String	Domain name of the delegation issuer; required to verify complex delegations	Domain Name of the delegation issuer
validity	yes	Group	This group contains fields that identify the time period when the delegation is valid	
notAfter	Yes	Date	Date when the delegation expires	UTC by which the delegation is expired
notBefore	Yes	Date	Date when from when on the delegation is valid	UTC by which the delegation is valid
flags	yes	Enumeration	The flags define which kind of delegation is further allowed besides the bilateral type.	SubstitutionAllowed DelegationAllowed

Field	Presence	Format	Description	Value
server	yes	URL	Contains the URL of the delegation provider where the delegation is saved.	URL

9.4 Revocation Mechanisms for Delegations

From time to time, it may become necessary to revoke a delegation. Without revoking the delegation, the Proxy is still able to execute the delegation. As an example, an employee leaving a company must not be able to sign contracts in the name of the company. Depending on the notAfter date in the delegation, the Proxy may still have some time to execute the delegation.

One possible solution to handle revocations is the issuance similar to Certificate Revocation Lists (CRLs). They contain all revoked delegation hashes Depending on the number of revoked delegations, the CRL may get very big. It must be transmitted for each query, because a new entry may have been added since the last query. Thus, the list alone may cause the verifiers network to slow down significantly and a better solution should be used.

Another solution is to handle revocations with something like an Online Certificate Status Protocol (OCSP). In contrary to the CRL, OCSP can handle queries for specific certificates, and only provide answers for the requested certificates, reducing the transmission overhead of the unwanted revocations dramatically. For the purpose of revoking a delegation the existing OCSP needs to be adapted slightly.

9.4.1 Revoking a Delegation

To revoke a delegation, the Mandator must know which delegation is to be revoked first. The Mandator can search the Delegation Provider for the correct delegation. The Mandator has all the information about the delegation available as well and can send a request to the Delegation Provider to revoke the delegation. Therefore, the Mandator searches the Delegation Provider for the delegation. If the delegation exists, the Mandator can send a revoke command to the Delegation Provider. The Delegation Provider then adds the selected delegation to the revocation list.

To revoke a delegation, the Mandator creates a revocation delegation for the Delegation Provider. The Mandator has to grant the Delegation Provider the right to revoke this delegation. In contrary, OCSP uses an OCSP responder to sign the responses of revocation queries. The revocation delegation, that the Delegation Provider receives, also requires a certificate, that is signed by the Mandator. This certificate is then used by the Delegation Provider to sign the response to a status request. For a better understanding, the process is depicted in Figure 13.

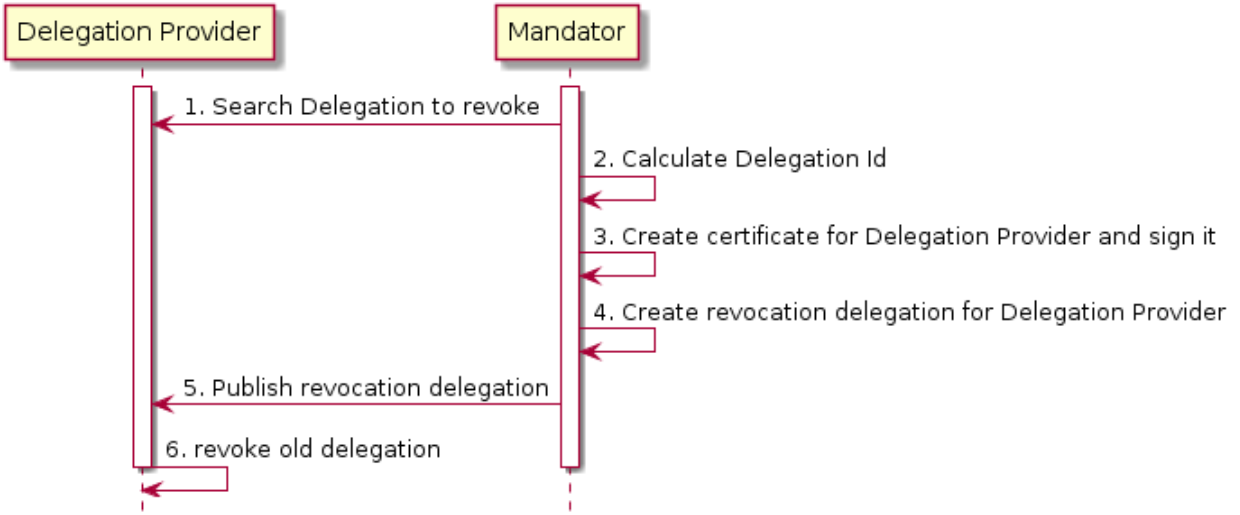

Figure 13: Revocation Process.

If a delegation is revoked the Delegation Provider stores the revocation time with the revocation. This revocation time is later sent if the status of a delegation is queried.

9.4.2 Querying the Status of a Delegation

To query the online revocation list of the Delegation Provider, an adapted version of the OCSP [7] is used. As the Delegation Provider does not have a Certificate we can rely on, the Mandator issues one for the Delegation Provider during the revocation process. This certificate is then used to sign the status response of the protocol. The Verifier then has to verify the signature and the delegation that has been issued with it. The Verifier can also check if the signature has been done with the same key as the revoked delegation. The Verifier should be able to build a chain to the Mandators certificate which provides proof of the correctness of the Delegation Providers answer. Figure 14 contains the details of the status querying process.

In the request for the status, a CertID sequence needs to be sent. This sequence contains the certificate serial number. Since delegations don't have a serial number, the hash value of the whole delegation is used as a serial number. The verifier has all the necessary information for the calculation available as the delegation data are included in the transaction.

One limitation of this protocol is that we only allow one query for one delegation at the time. This should not be any big limitation for LIGHTest. Depending on the transaction, the ATV probably has to query multiple Delegation Providers anyway. To query the status of multiple delegations, the Delegation Provider would need to sign the response with a valid signature. As the certificate for this signature is issued by the Mandator, the Delegation Provider would have a wide variety of certificates to choose from. Therefore, we omit this option of multiple queries at once.

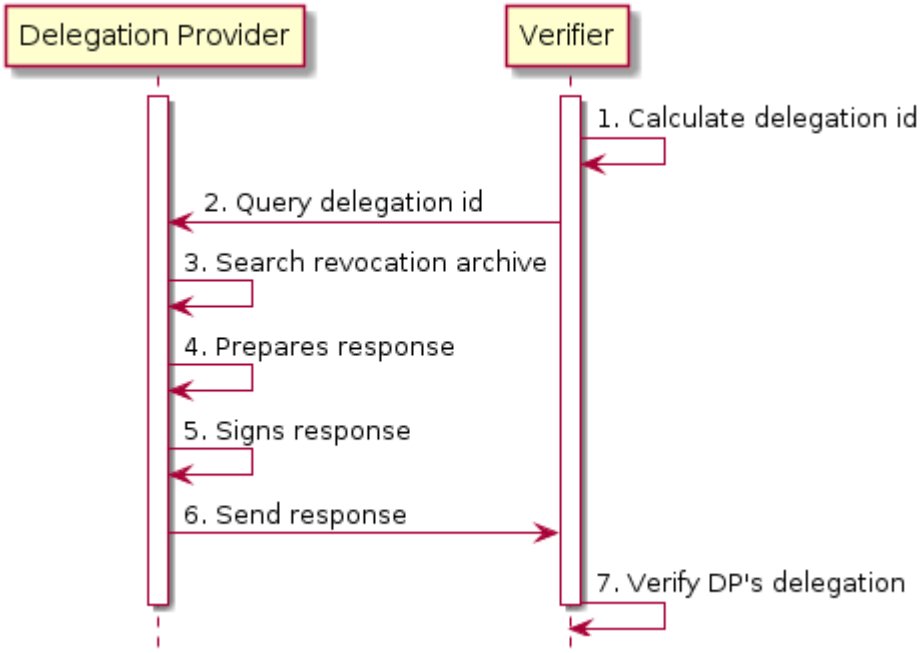

Figure 14: Querying for a revocation

10. Delegation in TPL

Finally, we would like to briefly illustrate how to express delegations in TPL. For this purpose, let us now consider a simple delegation scenario: we have a `Document` that contains a purchase, signed by the proxy of a company. To this end, the company has issued a `Mandate` containing at least the following information:

- A reference for the Mandator, in this case we assume this is a pointed to a trust membership claim of eIDAS. This may however be more indirect, e.g. a certificate of the Mandator that was issued by an authority that is on the eIDAS trust list. Only for simplicity we assume here the mandator is itself on the trust list.
- The public key of the proxy, so the signature of the proxy on a transaction can be verified. Strictly speaking, one does not need the identity of the proxy here. Alternatively, the proxy could also prove its identity using an eIDAS trust scheme. Using a public key, however, gives pseudonymity for the proxy (while several transactions made with respect to the same mandate are of course linkable).
- The `purpose`, here purchase. This may be more fine grained, e.g., allowing purchases only up to a certain limit.
- The authoritative `delegationProvider DP`: the ATV is required to check that the `DP` indeed has a valid entry containing (amongst others) a hash of the delegation. This check is to ensure that the delegation has not been revoked by the mandator. Also, this must be part of the signed delegation, so there is a specified server `DP` that has the authority over the validity of the mandate. The entry contains the entire mandate, but in encrypted form plus a hash of the mandate (so only the proxy of the mandate can read it, but ATV who has received the mandate can check it with this hash).

The necessary checks to be performed can then be described by the following TPL specification:

```
checkQualSigDeleg(Document,Mandate) :-
  % Check the Mandate fits the document:
  extract(Mandate,format,delegation),
  extract(Mandate,proxyKey,PkSig),
  verify_signature(Document,PkSig),
  extract(Mandate,purpose,purchase),
  % Check the mandator key (wrt Trust list):
```

```
extract(Mandate,issuer,Mandator),
trustscheme(Mandator,eIDAS_qualified)
lookup(Mandator,TrustListEntry),
extract(TrustListEntry, pubKey, PkIss),
verify_signature(Mandate,PkIss),
% Check the mandate is still valid:
extract(Mandate,delegationProvider,DP),
lookup(DP,DPEntry),
extract(DP,fingerprint,HMandate),
verify_hash(Mandate,HMandate).
```

11. Conclusion

This document shows the concepts of delegations and their representation in LIGHTest. It starts with an analysis of existing projects, shows scenarios on how delegation scan be created and revoked. Further, it provides a summary of the concepts of delegations and finally brings everything together in the conceptual form.

The existing projects provide an overview on the different requirements for using delegations. The three chosen examples; namely STORK 2.0, Austrian MOA/MOA-ID, and KATSO; are used to provide this overview on the different requirements for delegations.

The conceptual representation shows, that DNS is not necessarily required for the functionality of the Delegation Provider. This allows us to creaqte our own format for the representation of delegations with the https part. Due to the lack of a representation in the DNS, we only have one web based component. This component handles all the requests from the creation until the revocation of the delegation.

12. References

[1] The LIGHTest Project, D5.3 - DNS-based Publication of Delegations, 2018.

[2] The LIGHTest Project, D5.4 - Discovery of Delegations, 2018.

[3] The LIGHTest Project, „D2.1 - Inventories," *Project Deliverable,* 2017.

[4] The LIGHTest Project, „D2.14 - Architecture and Technical Coordination," *Project Deliverable,* 2017.

[5] The LIGHTest Project, „D5.1 Conceptual Framework for Delegations," 2017.

[6] EGIZ, „Vollmachten Service," [Online]. Available: https://www.egiz.gv.at/en/e-government/6-Online-Vollmachten . [Zugriff am 27 08 2018].

[7] T. Rössler, „Empowerment through Electronic Mandates -- Best Practice Austria," in *Software Services for e-Business and e-Society: 9th IFIP WG 6.1 Conference on e-Business, e-Services and e-Society, I3E 2009, Nancy, France, September 23-25, 2009. Proceedings*, Nancy, France, 2009, pp. 148-160.

[8] The LIGHTest Project, „D6.2 - Requirements and Design of a Conceptual Framework for Trust Policies," 2018.

[9] ETSI, „ETSI Technical Specification 119 612: Requirements for Trusted Lists (EU version of ETSI TS 102 231)," 2013. [Online]. Available: http://uri.etsi.org/19612/v1.2.1/. [Zugriff am 01 2017].

[10] G. Wagner, O. Omolola und S. More, „Harmonizing Delegation Data Formats," in *Open Identity Summit 2017*, Karlstad, Sweden, 2017.

[11] IETF, RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, 2013.

[12] The LIGHTest Project, „D3.1 - Design of a Conceptual Framework for Trust Schemes," *Project Deliverable,* 2017.

[13] S. Wohlgemuth, G. Muller, N. Sonehara und I. Echizen, „On observable delegation of personal data by watermarking," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, 2009.

[14] S. Wohlgemuth und G. Müller, *LNCS 3995 - Privacy with Delegation of Rights by Identity Management.*

[15] A. S. Raja und S. A. Razak, „Analysis of Security and Privacy in Public Cloud Environment," in *Cloud Computing (ICCC), 2015 International Conference on*, 2015.

[16] M. Prochazka, S. Licehammer und L. Matyska, „Perun—Modern approach for user and service management," in *IST-Africa Conference Proceedings, 2014*, 2014.

[17] R. J. Hulsebosch, „Privacy in Content Distribution Networks," in *Security and Privacy in the Age of Uncertainty*, Springer, 2003, pp. 435-440.

[18] M. Hansen, M. Raguse, K. Storf und H. Zwingelberg, „Delegation for Privacy Management from Womb to Tomb--A European Perspective," in *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, 2009.

[19] S. S. García und A. G. Oliva, *LNCS 5872 - Solving Identity Management and Interoperability Problems at Pan-European Level.*

[20] S. S. Garcı́a und A. G. Oliva, „Improvements of pan-European IDM architecture to enable identity delegation based on X. 509 proxy certificates and SAML," in *IFIP International Workshop on Information Security Theory and Practices*, 2010.

[21] Y. Cho, S. Cho und S.-H. Jin, „A Unified User Consent Acquisition and Delivery Mechanism for Multi-Source User Data Integrated Service," in *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*, 2007.

[22] L. Bussard, E. Di Nitto, A. Nano, O. Nano und G. Ripa, „An approach to identity management for service centric systems," in *European Conference on a Service-Based Internet*, 2008.

[23] G. e. al, „PID Service - an advanced persistent identifier management service for the Semantic Web," in *21st International Congress on Modelling and Simulation (MODSIM2015),, p767-773*, 2015.

13. Project Description

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

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

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

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

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

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