THE LIGHT*est* COOKBOOK

A Practical Guide to deploying a Lightweight Infrastructure for Global Heterogeneous Trust Management

LIGHT*est* has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321

Foreword

Trading has always involved trust and risk. Whether it is barter, or for promissory notes. Before modern communication, when trading with someone you did not know, you would take up references from someone you did trust. The longer the distance, the more difficult that was. Long distance trading sometimes involved family members, traveling and becoming established in those distant locations.

In the Twenty-First Century, sophisticated methods to provide that trust between counterparties have been developed, but still rely on someone to provide the references. These now take the form of 'Trusted Third Parties' who may be regulated from within the industry itself, or through legislation, to provide 'certification'. So, how do you obtain the details of the certifications that you need to satisfy acceptable risk and build trust?

LIGHT^{est} empowers counterparties to establish trust and awareness of risk through recognising the 'Chain of Trust' and associated certification. It can also provide automatic verification of that certification for every transaction.

LIGHT^{est} helps you manage your trust and risk.

This Cookbook will guide you step by step through the LIGHT^{est} infrastructure and tools that can support you in your business as well as demonstrate existing use cases and examples of use of LIGHTest.

1. Table of Contents

1. Table of Contents 1
1.1 Table of Acronyms 2
1.2 Some Useful Definitions about Trust 3

2. Benefits using LIGHT^{est} 5

3. An Introduction to LIGHT^{est} in Action 6
3.1 The Halloween Pumpkin Oil Company 6
3.2 How Trust Lists Actually Work 9

4. Overview of infrastructure 10
4.1 Example: Pumpkin Oil Company 11

5. Installation instruction and examples for end-users and TSPs 12
5.1 ATV Installation 12
5.1.1 Create and verify an electronic transaction using the REST API 12
5.1.2 Find previously verified electronic transaction using the API 13
5.2 TSPA Installation 13
5.2.1 Create or Update a Trust List 14
5.2.2 Delete a Trust List 14
5.2.3 Create or Update a Trust Scheme 14
5.2.4 Delete a Trust Scheme 14
5.3 TTA Installation 15
5.3.1 Create an agreement of Translation 17
5.3.2 Retrieve names of available agreements of translation 18
5.3.3 Retrieve details of an agreement of translation 18
5.3.4 Delete an agreement of Translation 18
5.3.5 Get information about agreements by Trust Scheme 18
5.3.6 Get Trust Schemes 19
5.3.7 Get Trust Scheme details 19
5.3.8 TTA file downloading service 19
5.4 DP Installation 20

6. Policy Tool and ATV 21
6.1 Creating a trust policy 21
6.2 Creating a transaction 23
6.3 Verifying a transaction with a trust policy 23

7. Use Cases and Examples 28
7.1 Use case: Open PEPPOL eProcurement 28
7.1.1 Access Point Verification use case 28
7.1.2 eProcurement pre-award Service Provider Trust Establishment 29
7.2 Use case: Correos Digital Services 30
7.2.1 My Mailbox - eDelivery use case 31
7.2.2 My Notifications - eNotifications use case 33
7.3 LIGHT^{est} Mobile ID scheme using FIDO 35

7.4 Examples of use of LIGHT^est 38

8. Beyond LIGHT^est 40

1.1 Table of Acronyms

Acronym	Description
AP	Access Point
API	Application Program Interface
AS4	Applicability Statement 4
ASiC	Associated Signature Containers
ATV	Automatic Trust Verifier
B2B	Business to Business
DNS	Domain Name System
DNSSEC	Domain Name System SECurity extensions
eIDAS	Electronic Identification, Authentication and trust Services
GUI	Graphical User Interface
HTTP(S)	Hypertext Transfer Protocol (Secure)
LoA	Level Of Assurance
NIST	National Institute of Standards and Technology
PEPPOL	Pan European Public Procurement On-Line
PKI	Public Key Infrastructure
REST	REpresentational State Transfer
SP	Service Provider
TIA	Transport Infrastructure Agreement
TP	Trust Policy
TPL	Trust Policy Language
TSL	Trust Service Status List
TSP	Trust Service Provider
TSPA	Trust Scheme Publication Authority
TTA	Trust Translation Authority
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language

1.2 Some Useful Definitions about Trust

The following provides a description of terms that are used within this cookbook.

Electronic Identification, Authentication and Trust Services
The EU Trust Services Regulation, eIDAS, provides a European Trust Scheme for Electronic Identification, Authentication and other Trust Services for use within the EU. It is applicable in law for relationships between EU Member State governments, and their citizens and businesses, but is widely be adopted as a general use Trust Scheme within the EU.

Entity
An Entity is a person, organization, or thing that is enrolled in a Trust Scheme and certain attributes of which are certified by a Trust Scheme Authority.

Level of Assurance
It is the category given to a trust service in order to classify the grade of trust provided by such service. For example, according to the eIDAS regulation assurance levels of electronic identification schemes are low, substantial and high.

Trust Authority
A Trust Scheme Authority operates a Trust Scheme and comprises the organizational, regulatory/legal, and technical measures to operate in that role, in a way that is agreed by national or international governing bodies.

Trust List
A Trust List is a specific data file of a specific format that is certified by the issuing authority (e.g., via electronic signature). It provides a list of all the enrolled entities. An existing and widely accepted standard for Trust Lists is ETSI TS 119 612.

Trust Policy
A Trust Policy is a recipe, expressed in a Trust Policy Language, that takes an Electronic Transaction and potentially multiple Trust Schemes, Trust Translation Schemes and Delegation Schemes as input and creates a single Boolean value (trusted [y/n]) and optionally an explanation (e.g., why not trusted) as output. In LIGHTest, a Trust Policy is evaluated by the Automatic Trust Verifier component.

Trust Policy Language
A Trust Policy Language (TPL) is a formal language with well-defined semantics that is based on a mathematical formalism and is used to express the recipe of a trust policy. For the LIGHT^{est} TPL the logic programming language Prolog that is based on Horn clauses is used.

Trust Scheme
A Trust Scheme is operated by a Trust Scheme Authority and comprises the organizational, regulatory/legal, and technical measures to assert trust-relevant attributes about enrolled Entities in a given domain of trust. A Trust Scheme operates in a given Trust Domain and typically has a declared or implied purpose.

Trust Scheme Provider
A Trust Scheme Provider operates a Trust Scheme. It decides, whether an Entity is associated with its Trust Scheme. The Trust Scheme Provider in LIGHT^{est} provides the Trust Lists for a Trust Scheme.

Trust Scheme Publication
A Trust Scheme Publication is always operated by a Trust Scheme Provider and associated with a Trust List which indicates that an Entity operates under the Trust Scheme, which the Trust Scheme Publication corresponds to. A Trust Scheme Publication in LIGHT^{est} can be a Boolean, Ordinal and Tuple-Based Trust Scheme Publication.

Trust Scheme Publication Authority
A Trust Scheme Publication Authority (TSPA) enables discovery of the Trust Scheme Provider. The TSPA defines how Trust Schemes are published making use of the existing infrastructure and global trust anchor of the Domain Name System (DNS).

Trust Service Provider
A Trust Service Provider (TSP) is defined as a natural or a legal person who provides one or more trust services (e.g. qualified signature, seal, timestamp, etc.).

Trust Translation Authority
This Entity provides the translation between trust levels, by enabling Trust Scheme Providers to indicate which trust levels are recognized as trustworthy by their Trust Scheme.

2. Benefits using LIGHT^{est}

The main benefits using LIGHT^{est} infrastructure are as follows:

- **Usage of a global Trust Infrastructure**
  LIGHT^{est} builds a global trust infrastructure where arbitrary authorities can publish their trust information. User organisations can use this to publish lists of qualified trust services, as business registrars and authorities 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.
  LIGHT^{est} infrastructure also provides tools to use a global trusted communications mechanism – the DNS – for the discovery, validation and translation of certain trust information. This means companies, administrations, and citizens can use LIGHT^{est} open source software to easily query this trust information to verify trust in simple signed documents or multi-faceted complex transactions.

- **Global Acceptance of the Approach Beyond Europe**
  LIGHT^{est} addresses this challenge by embedding its technical innovations into an inclusive and collaborative strategy that positions LIGHT^{est} from the start as a global initiative, open to extra-European collaboration.

- **Support for Heterogeneous Trust Models**
  LIGHT^{est} supports heterogeneous models of trust by moving the decision point for who is trusted to the verifier's trust policy. It typically selects and combines few existing large-scale trust schemes (such as that of EU qualified signature) and can further personalize it with local black- and white-lists.

- **Automatic Handling of Subsidiarity Principle in Trust Schemes**
  LIGHT^{est} uses the native and massively proven DNS mechanism to delegate the management of sub-domains to third parties.

- **Access to Trust Schemes based on Human-Readable Names**
  LIGHT^{est} uses DNS domain names to identify trust schemes. Using the existing DNS, this name can then be used by software to locate and access the data that is contained in the named trust scheme.

- **Use of a Single Trust Root to Replace a Multitude of Trust Anchors**
  LIGHT^{est} applies the existing, unique, and globally accepted trust root of the DNS. The standard mechanism of the DNS (with DNSSEC extension) allows to derive trust in trust scheme data from this single trust root and the (domain) name of the trust scheme.

- **Integration of Multiple Types of Trust Schemes in a Single Infrastructure**
  LIGHT^{est} uses a very generic model of trust scheme and supports an open number of trust schemes to coexist concurrently.

3. An Introduction to LIGHT^est in Action

3.1 The Halloween Pumpkin Oil Company

In this chapter, we present an example, which will guide you through the document and the processes where LIGHT*est* infrastructure and tools can support you in your business.

Meet Alice…

Alice is the Managing Director of the "Halloween Pumpkin Oil Company". Let's call it "HPOC". HPOC is a growing company located in Austria selling processed pumpkin oil to supermarkets and restaurants. HPOC is seeking to expand its business internationally, in and outside of Europe. But of course, it cannot afford to make mistakes when choosing business partners abroad.

As HPOC wishes to accept orders from all parts of the world, Alice needs to know that the business partner that she has chosen is trustworthy, in particular financially, because usually HPOC delivers the order before the buyer pays. Hence, HPOC needs to be careful and wants to be certain that the buyers are able to pay for the produce upon delivery.

Therefore, HPOC has set some requirements for logistics and trustworthiness:

- Logistics: International orders for Pumpkin seed oil are only accepted if the amount of Pumpkin seed oil is more than 10 Gallons
- Logistics: Local orders for Pumpkin seed oil are only accepted if the amount of Pumpkin seed oil requested is less than 100 Gallons
- Trustworthiness: All orders are signed, and the signer certificate is signed by an accepted issuer certificate

Luckily for Alice, her company is part of the "Pumpkin Oil Federation" (POF). They have the local Pumpkin Oil Industry's welfare in mind and have set up a flexible way to ensure that certifications are current, and to support the networking between producers and consumers around the world.

POF uses LIGHT*est* to ensure only genuine product is traded between trustworthy business partners (producers and buyers).

So how does it work?

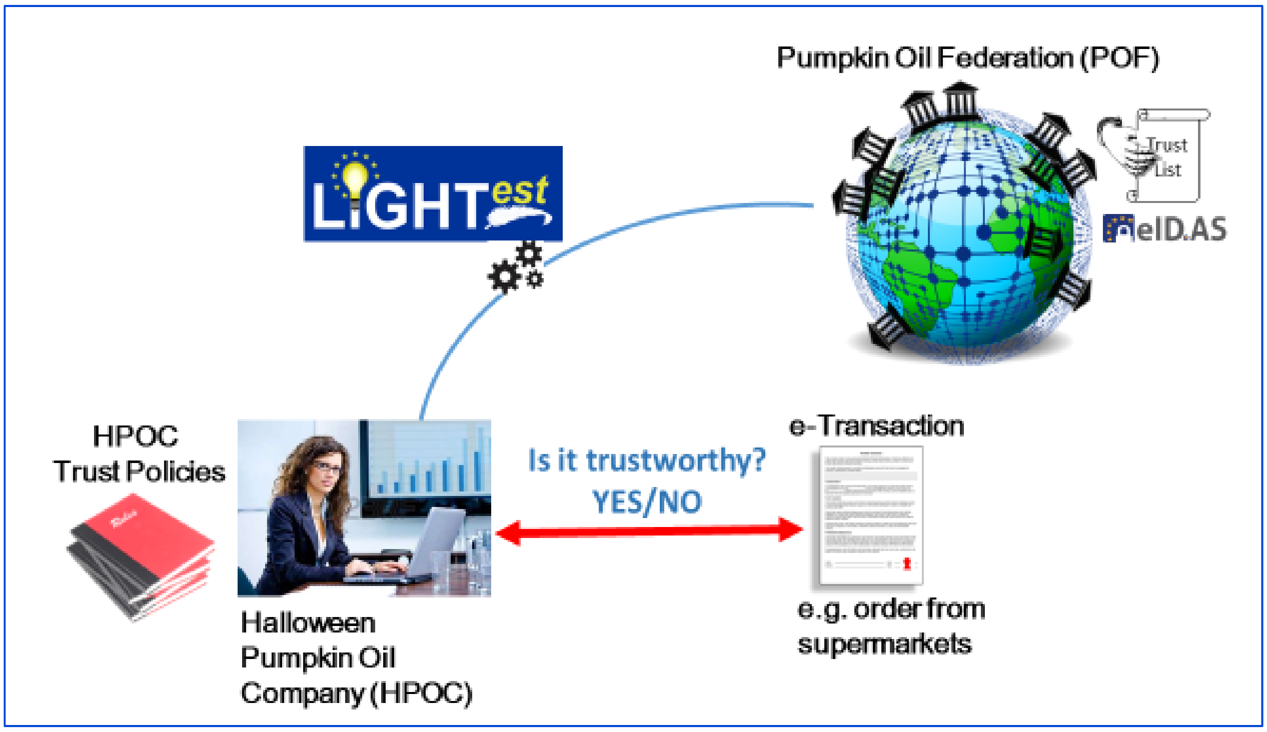

Pumpkin Oil Federation (POF)
Trust List
eIDAS
LIGHTest
HPOC Trust Policies
Rules
Is it trustworthy? YES/NO
e-Transaction
e.g. order from supermarkets
Halloween Pumpkin Oil Company (HPOC)

Figure 1: Overview "Halloween Pumpkin Oil Company" - Example

- POF operates a Trust Scheme where the rules are defined which needs to be fulfilled by the trusted entities.
- POF maintains a list of trusted entities (producers and buyers of pumpkin oil) around the world.
- This list is called a **"Trust List"**.
- So now, when Alice from HPOC receives an order for pumpkin oil from an unknown buyer, via an electronic transaction she can ask if that buyer is certified, and then can check that certification through the POF, and the chain of trust down to that new buyer using LIGHT^{est} infrastructure.
- HPOC can trust the new buyer because it is certified by an organisation, that itself is trusted by POF.
- Each time HPOC trades with that organisation, it can check whether the certifications are still valid using LIGHT^{est}, and the Trust Lists.

In addition to using the Trust List of POF to verify the trustworthiness of its business partner, Alice and HPOC can also decide that they trust entities of other existing trust schemes, e.g. the eIDAS regulation on electronic identification and trust services for electronic transactions in the internal market.

So now, Alice can place a purchase order knowing that the counterparty has a current certification that can be trusted and is vouched-for by her industry federation.

LIGHT^{est} provides all the components necessary to establish the trust between buyers and sellers in a highly scalable and consistent manner.
LIGHT^{est} components can provide this transparency easily, without the need for special communications.
LIGHT^{est} can verify the trust automatically for Alice when each transaction takes place.

And LIGHT^{est} can do a lot more!

LIGHT^{est} can help delegate responsibilities. One of the LIGHT^{est} components provides information on delegated responsibilities inside an organisation. So, if the director of a buyer of pumpkin oil chooses to employ a Purchasing Manager, then that information can be declared via LIGHT^{est} and in addition his/her authority can be limited, e.g. for purchases up to 50 Gallons.
So now, using LIGHT^{est,} the seller of the raw pumpkin oil can be sure that the purchase order received is authentic and authorised.

LIGHT^{est} helps everyone trade successfully!

The uses for LIGHT^{est} are endless. The above is a simple example to demonstrate the use of LIGHT^{est} and its components in a common trading situation.

3.2 How Trust Lists Actually Work

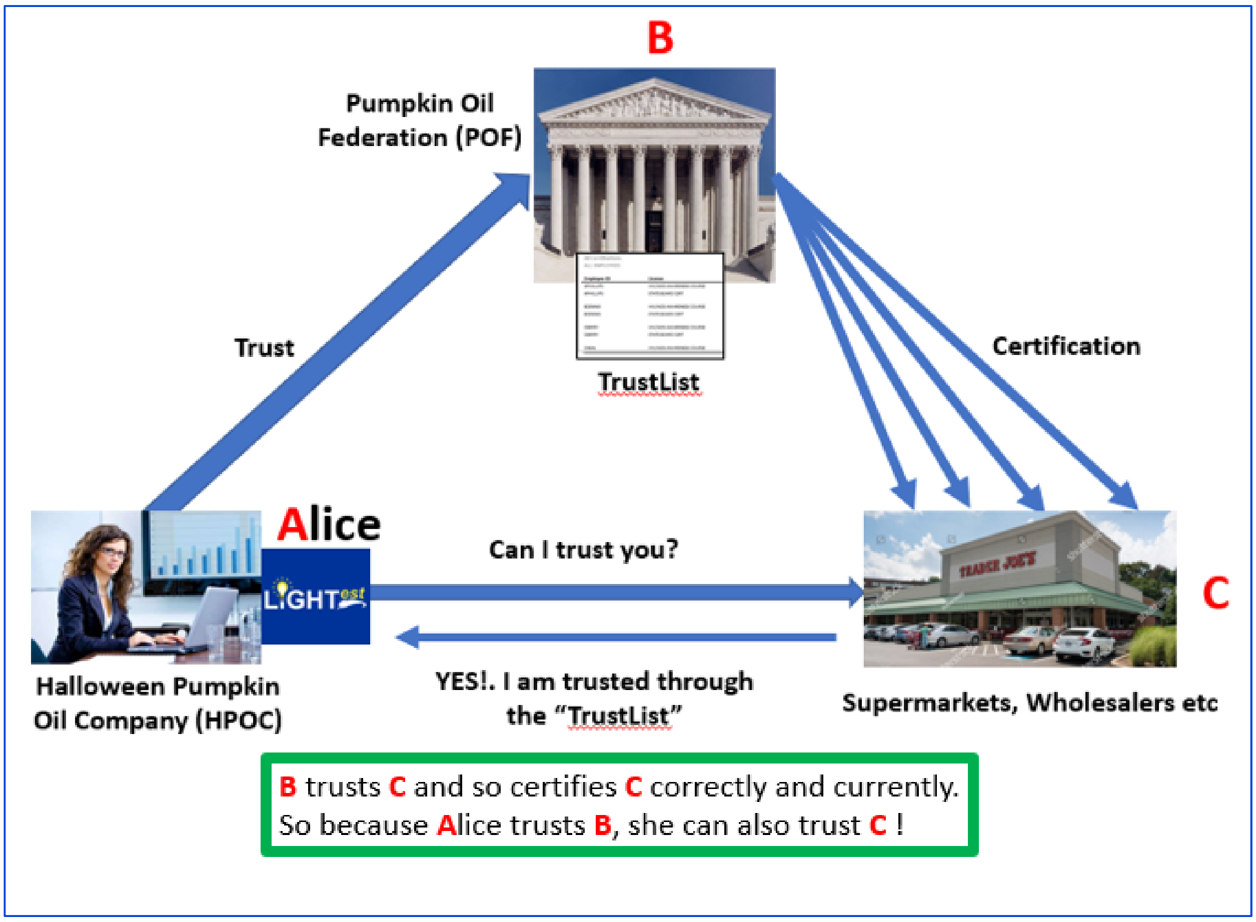

B
Pumpkin Oil Federation (POF)
Trust
TrustList
Certification
Alice
Can I trust you?
YES!. I am trusted through the "TrustList"
C
Halloween Pumpkin Oil Company (HPOC)
Supermarkets, Wholesalers etc
B trusts C and so certifies C correctly and currently.
So because Alice trusts B, she can also trust C !

Figure 2: How TrustLists work

Trust Lists help trust one another without specific personal knowledge.

It goes as follows: **"I trust you, and you trust them. Therefore, I trust them too!"**

- In the case above B trusts C and certifies that in its TrustList.
- Alice trusts B and so accepts its TrustList in which C is included.
- Therefore, Alice can now trust C.

The beauty of TrustLists is that they are scalable and can extend out so that Alice could trust many more entities than should/could possibly know directly. Also, the TrustList at B can also list other TrustLists, so the number of trusted entities can grow easily.

It does depend on the TrustLists being trustworthy themselves, so it can get more complicated.

4. Overview of infrastructure

This section gives an overview of the LIGHTest reference architecture. It defines the macroscopic design of the LIGHTest infrastructure as well as the overall system's components, their functionality and their interaction on a high-level view.

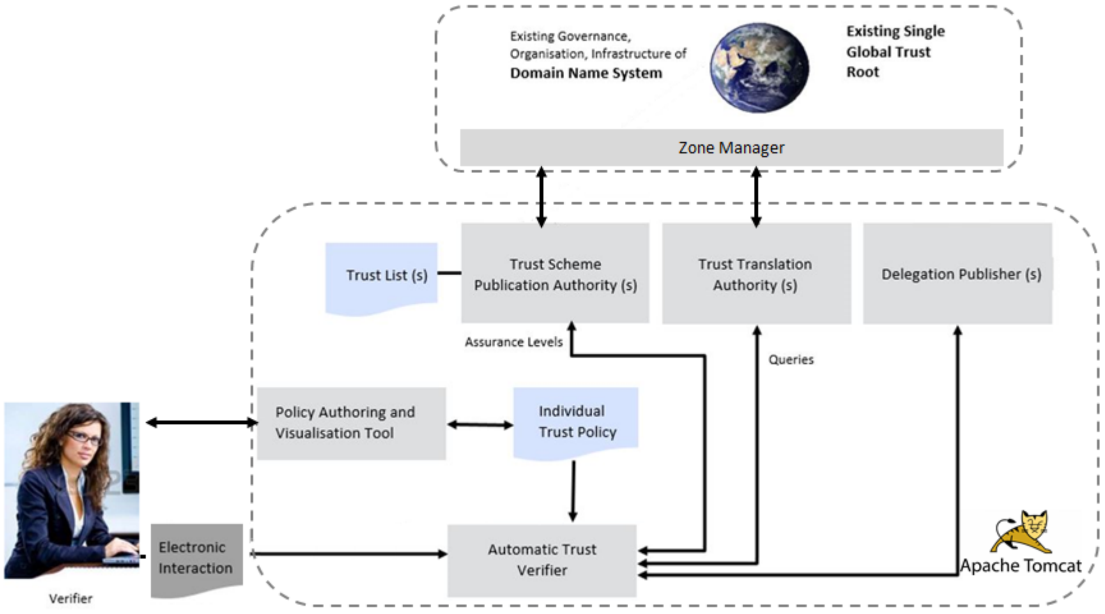

Existing Governance, Organisation, Infrastructure of Domain Name System
Existing Single Global Trust Root
Zone Manager
Trust List (s)
Trust Scheme Publication Authority (s)
Trust Translation Authority (s)
Delegation Publisher (s)
Assurance Levels
Queries
Policy Authoring and Visualisation Tool
Individual Trust Policy
Electronic Interaction
Automatic Trust Verifier
Apache Tomcat
Verifier

Figure 3: Technical components of LIGHTest

Figure 3 shows the LIGHTest reference architecture with all the major software components. It illustrates how a verifier can validate a received Electronic Transaction based on her individual Trust Policy and queries to the LIGHTest reference trust infrastructure.

The Verifier interacts with the Policy Authoring and Visualization Tools (e.g. desktop or web applications). These tools also facilitate non-technical users the visualization and editing of trust policies, which can be individual and specific for each transaction. The role of the trust policy is the provision of formal instructions for the validation of trustworthiness for a given type of electronic transaction.
The Automatic Trust Verifier (ATV) takes the electronic transaction and trust policy as input and provides as output if the electronic transaction is trustworthy or not. In addition, the ATV may provide an explanation of its decision, in particular if the transaction was considered as not trustworthy. To have it running, it is necessary to install an Apache Tomcat, which is a web server, or application server (HTTP), that provides an environment to run Java code (https://tomcat.apache.org/download-80.cgi). For all human interaction, a Graphical User Interface (GUI) is going to be used, meaning a Java 8 or higher installation on the local computer is needed to be installed as well.
The Trust Scheme Publication Authority (TSPA) operates an off-the-shelf DNS Name Server with DNSSEC extension and one or more Trust Scheme Providers. A server publishes multiple trust lists under different sub-domains of the authority's domain name. The TSPA enables discovery and verification of trust scheme memberships.
The Trust Translation Authority operates a standard DNS Name Server with DNSSEC extension. A single authority could publish Trust Translation Lists under different sub-domains of the authority's domain name. Trust Translation Lists express which levels from other Trust Domains are trusted.

The Delegation Publisher operates an off-the-shelf DNS Name Server with DNSSEC extension and a web server application. A server publishes multiple delegations via a REST API operated on the Delegation Provider itself.
As these three components need a web server to be deployed, the same Apache Tomcat could be used to install the complete LIGHT^{est} platform. The LIGHT^{est} DNS server component has two different modules: the DNS itself and a front-end named ZoneManager. The Zone Manager offers a simple REST-like API to the TTA and TSPA that allows them to add, alter, and remove information through HTTP requests.
For the verification of the authenticity of these trusted lists, the trusted lists are signed using public-key cryptography. In the hierarchical form, there is always a trust anchor required on top. At the trust anchor the key pair needs to be known as correct without further evidence. By utilizing the existing single trust root of the DNS, LIGHT^{est} can significantly simplify the provisioning of trust anchors. With these trust anchors any Trust Service Provider in the hierarchical structure can be found and verified.

4.1 Example: Pumpkin Oil Company

For the example defined in section 3.1 the actors using LIGHT^{est} infrastructure and the corresponding LIGHT^{est} components are as follows:

LIGHT^{est}	**Pumpkin Oil Company Example**
Verifier	Alice from the Halloween Pumpkin Oil Company (HPOC)
Policy Authoring and Visualization Tool	Tool that supports Alice to define the trust policy which business partners to trust
Individual Trust Policy	Formal instructions for validation of trustworthiness: - Orders are signed - Signer certificate is signed by an accepted issuer (e.g. POF, eIDAS) - Logistics requirements (e.g min./ max amount per order)
Electronic Transaction	Signed order for pumpkin oil from an unknown buyer
ATV	Tool that verifies the electronic transaction using the individual trust policy
TSPA	The Pumpkin Oil Federation (POF) operates a TSPA; POF also operates its own Trust Scheme
Trust List	POF maintains and publishes the list of trusted business partners
TTA	Provides the translation of any trust scheme level from eIDAS and TR to POF, based on a previous bilateral agreement
Zone Manager	API, which allows the TSPA and TTA to add, alter, and remove information in DNS
Delegation Publisher	Delegation is not considered in this demo example. One application could be a delegation that a Purchasing Manager has the permission for purchases up to 1000€.

5. Installation instruction and examples for end-users and TSPs

We describe the instructions for installation based on the Halloween Pumpkin Oil Company example presented in section 3.1. The Halloween Pumpkin Oil Company demonstrator is found at: https://github.com/H2020LIGHTest/PumpkinSeedOilDemo.

5.1 ATV Installation

The Automatic Trust Verifier (ATV) can be accessed using a REST API and a Graphical User Interface (GUI). This gives the ATV flexibility and it can be used in different ways, e.g. on a server, where clients can connect to verify a transaction, locally on a desktop for a single user to verify transactions or on a mobile platform to verify transactions on tablets/smart phones. For the example in this document, the GUI will be used primarily for verification.

Both, the REST API (https://github.com/H2020LIGHTest/AutomaticTrustVerifier-API) and GUI (https://github.com/H2020LIGHTest/AutomaticTrustVerifier-GUI) are implemented as self-contained jar archives. For the REST API an Apache Tomcat server is required, while the GUI just needs a Java 8 or higher installation on the local computer. To get the ATV running change the "atv.configuration" file. As default, DNSSEC and DANE validation is turned on, and the path to the DNSSEC root key is given.

The ATV accepts the electronic transactions in the ASIC-S and ASIC-E container format conforming to the ETSI TS 103 174 V2.2.1 specification. As the verifier may require different requirements for the verification, the ATV gets support from a trust policy. To define a trust policy, a Trust Policy Authoring Tool (TPAT) is included with the ATV that uses a Trust Policy Language (TPL) to define a policy.

In order to verify a transaction via the REST API the */addInstance* endpoint needs to be called. The endpoint takes the transaction and the policy as parameters. The ATV will now verify the transaction. In the GUI case it will provide feedback instantly if the transaction could be verified, together with an audit log. The REST API requires a call to the */findInstance* endpoint to retrieve the result.

5.1.1 Create and verify an electronic transaction using the REST API

Name	Resource	Method	Body	Description
Create and Verify Electronic Transactions	\$api_url/api/v1/addinstance\$	GET	application/zip	CREATE and VERIFY electronic transaction
Example	get https://atvapi.tug.do.nlnetlabs.nl/atvapi/api/v1/addinstance			
Answer	200 OK	{ "verificationResult": "\$string\$", "result": \$integer\$ }"		

5.1.2 Find previously verified electronic transaction using the API

Name	Resource	Method	Body	Description
Find previously verified transactions	$api_url/api/v1/findinstance$	POST	text/text	FIND previously verified electronic transaction
Example	POST https://atvapi.tug.do.nlnetlabs.nl/atvapi/api/v1/findinstance			
Answer	200 OK	{ "verificationResult": "$string$", "result": $integer$ }"		

More information on how to use the REST API can be found in the test section of the repo i.e., https://github.com/H2020LIGHTest/AutomaticTrustVerifier-API/tree/master/src/test/java/at/tugraz/iaik/lightest/atv/api/controller/v1.

5.2 TSPA Installation

The Trust Scheme Publication Authority (TSPA) contains a REST API that is used to manage the tasks of publishing, updating, and deleting trust schemes, and trust lists. TSPA uses a ZoneManager component that is responsible from creating the corresponding DNS records for trust scheme publication and discovery.

ZoneManager must run on the DNS nameserver with DANE and DNSSec validation enabled that should be deployed. And, the ZoneManager requires nsd to be configured. Nsd is the domain name server that it can interface with. In the ZoneManager, the FQDN of domain namer server should be defined according to the domain name and scheme name definitions.

But, TSPA can be deployed on a different server. TSPA is a web application that is delivered as a jar archive and requires Apache Tomcat or a similar server to run. It needs to be placed in the configured folder from Apache Tomcat. Further, the “config.properties” file, that ships with the web application needs to be changed. The entries for “nameserver.address”, and “nameserver.token” need to be changed to the settings of the ZoneManager.

The TSPA API uses the following parameters, which are the very similar for all calls.

Parameter	Type	Value	Description
api_version	Fixed	v1	API Version that should be accessed by the TSPA. This version is currently "v1". If in the future different endpoints are to be accessed, this version needs to change as well.
scheme_name	Variable		Name of the scheme (e.g. eIDAS) followed by the FQDN of the domain name server (e.g. lightest.nlnetlabs.nl).

A parameter can either be fixed or variable. A fixed parameter will contain the same value for all calls, while the variable changes for different calls.

Important: if a scheme_name does not contain the FQDN, it will be rejected by the TSPA. In the following description, we use our demo setup to publish trust lists and schemes.

5.2.1 Create or Update a Trust List

Name	Resource	Method	Body	Description
Publish Trust List	api_version/scheme_name/trust-list	PUT	application/json, application/xml, text/plain	CREATE or UPDATE an trust list entry
Example	PUT https://tspa.tug.do.nlnetlabs.nl/tspa/api/v1/federation.pof-demo.lightest.nlnetlabs.nl/trust-list			
Answer	200 OK	{ "url\":\" https://tspa.tug.do.nlnetlabs.nl/testschemes/Pumpkin_Demo_TS_v0.3-signed.xml\", "certificate\": [] }"		

This call publishes a trustlist called federation.pof-demo, which is given as the scheme_name parameter in the PUT request. In the JSON body, the trust scheme claims to be part of Pumpkin_Demo_TS, and publishes the certificate for this trust scheme. The URL in the JSON identifies the position where the trust-list is stored.

5.2.2 Delete a Trust List

Name	Resource	Method	Body	Description
Delete Trust List	api_version/scheme_name/trust-list	DELETE	None	Deletes the trust list
Example	DELETE https://tspa.tug.do.nlnetlabs.nl/tspa/api/v1/federation.pof-demo.lightest.nlnetlabs.nl/trust-list			
Answer	200 OK			

The trust list federation.pof-demo, as specified in the scheme_name parameter is deleted from the nameserver.

5.2.3 Create or Update a Trust Scheme

Name	Resource	Method	Body	Description
Publish Trust Scheme	api_version/scheme_name/schemes	PUT	application/json or text/plain	CREATE or UPDATE an trust scheme
Example	PUT https://tspa.tug.do.nlnetlabs.nl/tspa/api/v1/company-ca.pof-demo.lightest.nlnetlabs.nl/schemes			
Answer	201 OK	{ "schemes" : [" federation.pof-demo.lightest.nlnetlabs.nl"] }		

This example creates updates of a trust scheme. The current trust scheme company-ca.pof-demo claims that it is part of the federation.pof-demo scheme. Therefore, the owner of company-ca.pof-demo publishes a list of claims in the body of the request.

5.2.4 Delete a Trust Scheme

Name	Resource	Method	Body	Description
Delete Trust Scheme	api_version/scheme_name/schemes	DELETE	application/json or text/plain	DELETE an trust scheme
Example	DELETE https://tspa.tug.do.nlnetlabs.nl/tspa/api/v1/company-ca.pof-demo.lightest.nlnetlabs.nl/schemes			
Answer	200 OK			

This call deletes the trust scheme company-ca.pof-demo from the server.

5.3 TTA Installation

The Trust Translation Authority (TTA) contains a REST API that is used to manage the tasks of trust translations. TTA uses a ZoneManager component that is responsible from creating the corresponding DNS records for trust translation publication and discovery.

ZoneManager must run on the DNS nameserver with DANE and DNSSec validation enabled that should be deployed. And, The ZoneManager requires nsd to be configured. Nsd is the domain name server that it can interface with. In the ZoneManager, the FQDN of domain namer server should be defined according to the domain name and scheme name definitions.

But TTA can be deployed on a different server. TTA is a web application that is delivered as a jar archive and requires Apache Tomcat or a similar web server to run. It needs to be placed in the configured folder from Apache Tomcat. Further, TTA provides a config API:

Name	Resource	Method	Body	Description
Get configuration	/ cfg/conf	GET		Get a configuration of TTA
Example	https://tta-lightest.eu:8443/unittesting/ttaFM/mng/cfg/conf			
Answer	200 OK	{ "tta-configuration": [..] }		

Name	Resource	Method	Body	Description
Set param	/trustScheme	PUT	{"name": "sign", "value": "false" }	Set value of a configuration parameter
Example	https://tta-lightest.eu:8443/integration/ttaFM/mng/cfg/param			
Answer	200 OK			

TTA enables to differentiate the trust schemes by the value of the “name” field in the “trustSchemeNames” and “trust-scheme-list”. Then, the representation of a Trust Scheme within the TTA tool, can have different definitions depending of the value of all its properties.

For instance, in the case of eIDAS, we can set as Trust Scheme name *eidas_<service>*, so TTA will manage eidas_eSignature and eidas_eSeal as different Trust Schemes, the information retrieved with the trustScheme resource API will be:
GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme

```
{
   "trustSchemeNames": [
      "eidas-eseal",
      "eidas-esignature" ]
}
```

GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme/eidas-eseal

```
{
   "trust-scheme-list": [
      {
         "level": "advanced",
         "provider": "eu",
         "name": "eidas-eseal"
      }
   ]
}
```

Other alternative is to set *eidas* as the name of the Trust Scheme adding the service name as a param; in this case TTA will manage *eidas* as a Trust Scheme with two definitions. This enables TTA to manage Tuple-based Trust Schemes, where a Trust Scheme can have several definitions depending on the attributes and their values. So, TTA should consider that the name of a trust scheme is formed in the way: **TrustSchemeName_Service_Level**.

GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme

```
{
   "trustSchemeNames": [
      " eidas" ]
}
```

GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme/eidas

```
{
   "trust-scheme-list": [
      {
         "name": "eidas",
         "provider":"eu",
         "params": [
            { "level": "qualified },
            { "service": "esignature" } ]
      },
      {
         "name": "eidas",
         "provider":"eu",
         "params": [
            { "level": "advanced" },
            {"service": "eseal" } ]
      }
   ]
}
```

5.3.1 Create an agreement of Translation

Name	Resource	Method	Body	Description
Create translation	/translation	POST	See body example 1	Create an agreement of translation
Example	POST https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/translation			
Answer	200 OK	{ "agreename":"eidas-esig_tr1", "xmlFile":"tr1.xml", "tplFile":"tr1.tpl.p7s", }		

Body example 1:

```
{
"agreement": {
"name": "eidas-esig_tr1",
"status": "active",
"creation-date": "2019-08-10",
"leaving-date": "2020-08-10",
"activation-date": "2019-08-10",
"source": {
"name": "eidas-esignature",
"level": "advanced",
"provider": "eu"

},
"target": {
"name": "tr-esig",
"provider": "tr",
"params": [{
"name": "param1",
"value": "value1"
},
{
"name": "param2",
"value": "value2"
} ] } } }
```

Where,

- agreement->name: (mandatory) it is the name of the translation agreement.
- agreement->status: (mandatory) *internal use*, set to “active” by default.
- agreement-> *-date: (optional) format YYYY-MM-DD
- agreement->source: (mandatory) on whom trusts originally.
- agreement->target: (mandatory) on whom trusts as result of the agreement.
- source, target:
  - name: (mandatory) name of a trust scheme.
  - level: (optional) level of a trust scheme.
  - provider: (optional)
  - params: (optional) list of params representing the tuples in a tuple-based trust scheme.
  - params->name: name of the tuple.
  - params->value: value of the tuple.

5.3.2 Retrieve names of available agreements of translation

Name	Resource	Method	Body	Description
get translations	/translation	GET		Retrieve a list of names of all available agreements of translation
Example	GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/translation			

Answer	200 OK	{
		    "agreementNames": [
		        "eidas-esig_tr1"
		    ]
		}		

5.3.3 Retrieve details of an agreement of translation

Name	Resource	Method	Body	Description
get agreement	/translation/<name>	GET		Retrieve details of an agreement of translation
Example	GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/translation/eidas-esig_tr1			
Answer	200 OK	The body of the answer is the one used to create the agreement of translation. See Body example 1 above.		

5.3.4 Delete an agreement of Translation

Name	Resource	Method	Body	Description
Delete translation	/translation/<name>	DELETE		Delete an agreement of translation
Example	DELETE https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/translation/eidas-esig_tr1			
Answer	200 OK			

5.3.5 Get information about agreements by Trust Scheme

Name	Resource	Method	Body	Description
Get agreements by trust scheme	/getAgreementsRelatedToATrustScheme/<trust scheme name>	GET		It answers a list of agreement names in which the given trust scheme participates as source and as origin
Example	GET *https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/getAgreementsRelatedToATrustScheme/tr-esig*			

Answer	200 OK	{
		    "trustSchemeName": " tr-esig ",
		    "TargetOfTrustIn": [
		        "eidas-esig_tr1"
		    ],
		    "OriginOfTrustIn": []
		}		

5.3.6 Get Trust Schemes

Name	Resource	Method	Body	Description
Get trust scheme	/trustScheme	GET		Get a list of available Trust Schemes
Example	GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme			
Answer	200 OK	{ "trustSchemeNames": ["tr- esig ", "eidas-esignature"] }		

5.3.7 Get Trust Scheme details

Name	Resource	Method	Body	Description
Get trust scheme details	/trustScheme/<TSname>	GET		Get details of a Trust Scheme
Example	GET https://tta-lightest.eu:8443/integration/ttaFM/mng/rsc/trustScheme/tr-esig			
Answer	200 OK	{ "trust-scheme-list": [{ "provider": "tr", "name": "tr-esig", "params": [{ "param1": "value1" }, { "param2": "value2" }] }] }		

5.3.8 TTA file downloading service

TTA works as File Server to store Trust Translation files, and provides the corresponding file downloading service.

Name	Resource	Method	Body	Description
GET TTA file	/TrustTranslationDeclaration/ <trust_scheme_name>	GET		Download TT declaration file
Example	https://tta-lightest.eu:8443/integration/ttaFM/mng/TrustTranslationDeclaration/tr-esig			
Answer	200 OK			

This service takes into account the HTTP header “accept”: if Accept header contains *xml* the file provided will be an XML trust translation file; any other case the file provided will be TPL format.

5.4 DP Installation

The Delegation Provider (DP) contains a REST API that is used to manage the tasks of creating, publishing and checking revocation status of delegations.

DP is a web application that is delivered as a jar archive and requires Apache Tomcat or a similar server to run. It needs to be placed in the configured folder from Apache Tomcat.

DP is not considered in the Halloween Pumpkin Oil Company example. For further information we refer to the corresponding deliverables.

6. Policy Tool and ATV

6.1 Creating a trust policy

A trust policy contains instructions on how a transaction is verified. Within the scope of LIGHTest, a trust policy can be created with the aid of a trust policy authoring tool (TPAT).

Before creating a trust policy, a format is needed. A format is a representation of a standard transaction (in this case Pumpkin seed oil order form). Formats are created in XML and are expected to be widely known. An example of a Pumpkin seed oil format is given below:

```
<?xml version="1.0" encoding="UTF-8"?>

<form format="pumpkinSeedOil">

    <purchaser>
        <name>John Doe</name>
        <street>Dartmouth St.</street>
        <city>Midfarthington</city>
        <country>England</country>
    </purchaser>

    <item_origin>International</item_origin>
    <ammount>7</ammount>

</form>
```

Figure 4 Transaction Format

Moving on, we need to create the trust policy that the Pumpkin Seed Oil producer needs to verify an order that was received.

```
accept(Form) :-
  extract(Form, format, pumpkinSeedOil),

  extract(Form, purchaser, Buyer),
  extract(Buyer, name, BuyerName),
  print(BuyerName),

  authorize_order(Form),

  extract(Form, certificate, Certificate),
  extract(Certificate, format, x509cert),
  extract(Certificate, pubKey, PK),
  verify_signature(Form, PK),
  check_qualified(Certificate).


authorize_order(Form) :-
  extract(Form, item_id, ID),
   ID == 54678,
  extract(Form, amount, Amount),
   Amount  <= 10,
  print(authorized_Order_for_Item_54678).

authorize_order(Form) :-
  extract(Form, item_id, ID),
   ID == 42,
  extract(Form, amount, Amount),
   Amount  <= 10,
  print(authorized_Order_for_Item_42).

authorize_order(Form) :-
  extract(Form, item_id, ID),
   ID == 7,
  extract(Form, amount, Amount),
   Amount  <= 100,
  print(authorized_Order_for_Item_7).

authorize_order(Form) :-
  print(authorizing_order_failed),
  false().


check_qualified(Certificate) :-
  extract(Certificate, issuer, IssuerCertificate),

  trustschemeX(IssuerCertificate, pumpkin_Oil_Federation, TrustListEntry),
  extract(TrustListEntry, format, trustlist_entry),

  print(checking_Trust_Status),
  verify_service_type(TrustListEntry),

  extract(TrustListEntry, pubKey, PkIss),
  verify_signature(Certificate, PkIss).


trustschemeX(IssuerCert, TrustedScheme, TrustListEntry) :-
  print(trustschemeX_without_Translation_START),
  extract(IssuerCert, trustScheme, Claim),
  trustscheme(Claim, TrustedScheme),
  trustlist(Claim, IssuerCert, TrustListEntry),
  print(trustschemeX_without_Translation_DONE).

trustschemeX(IssuerCert, TrustedScheme, TrustedTrustListEntry) :-
  print(trustschemeX_with_Translation_START),
  extract(IssuerCert, trustScheme, Claim),
  trustlist(Claim, IssuerCert, TrustListEntry),
  encode_translation_domain(Claim, TrustedScheme, TTAdomain),
  lookup(TTAdomain, TranslationEntry),

  translate(TranslationEntry, TrustListEntry, TrustedTrustListEntry),
  print(trustschemeX_with_Translation_DONE).


verify_service_type(TrustListEntry) :-
  extract(TrustListEntry, serviceType, qualified_certificate_authority),
  print(verified_ServiceType_as_Qualified_CA).

verify_service_type(TrustListEntry) :-
  extract(TrustListEntry, serviceType, qualified_pof_buyer),
  print(verified_ServiceType_as_Qualified_Buyer).
```

Figure 5 Trust policy

The trust policy created above verifies that:

- All orders are signed by eIDAS qualified certificates
- International orders can order up to 10 gallons of pumpkin seed oil
- Local orders can order up to 100 gallons of pumpkin seed oil

The trust policy presented above can be created with the Trust Policy Authoring Tool (TPAT).

6.2 Creating a transaction

The reference architecture defines an example Associated Signature Container Extended (ASICE) transaction. In theory any form of transaction can be used but the necessary plugin for the transaction must be added to the ATV.

A transaction (in this case a pumpkin seed oil order form) will contain the order in the format specified above signed with a certificate. The transaction can be generated using standard means of creating that particular transaction mode.

6.3 Verifying a transaction with a trust policy

Verifying a transaction requires downloading the desktop GUI version of the ATV. This software can be found at: https://github.com/H2020LIGHTest/AutomaticTrustVerifier-GUI.

After downloading the software, the necessary documents i.e., the transaction and the trust policy are uploaded onto the transaction box frame and the trust policy box frame on the interface respectively as shown in Figure 7. The verification commences once the “Process Transaction” is clicked.

Figure 8 shows the status frame below the transaction and trust policy box frames. The status frame shows the process taken during the verification as well as the success or the failure of the verification.

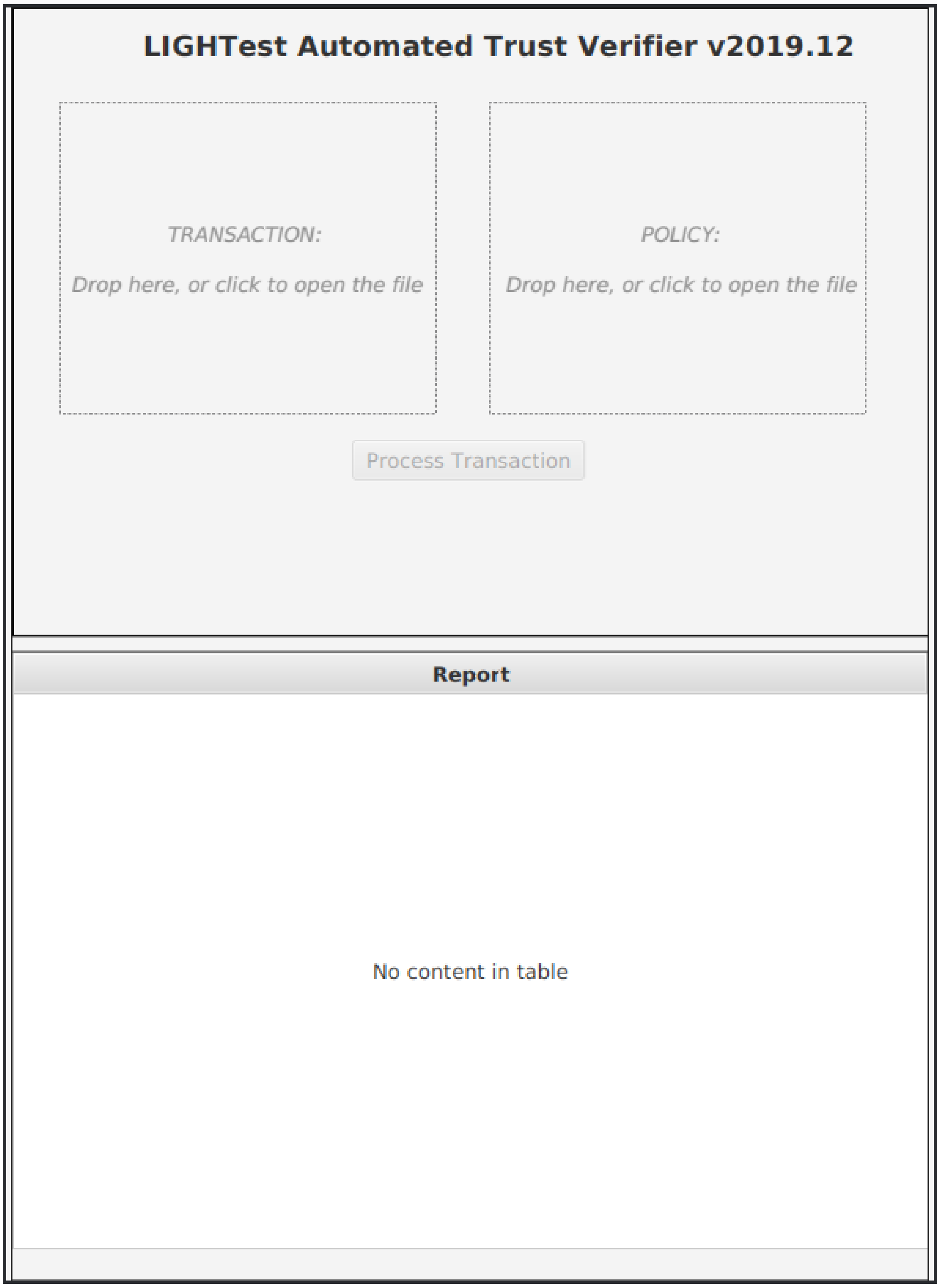
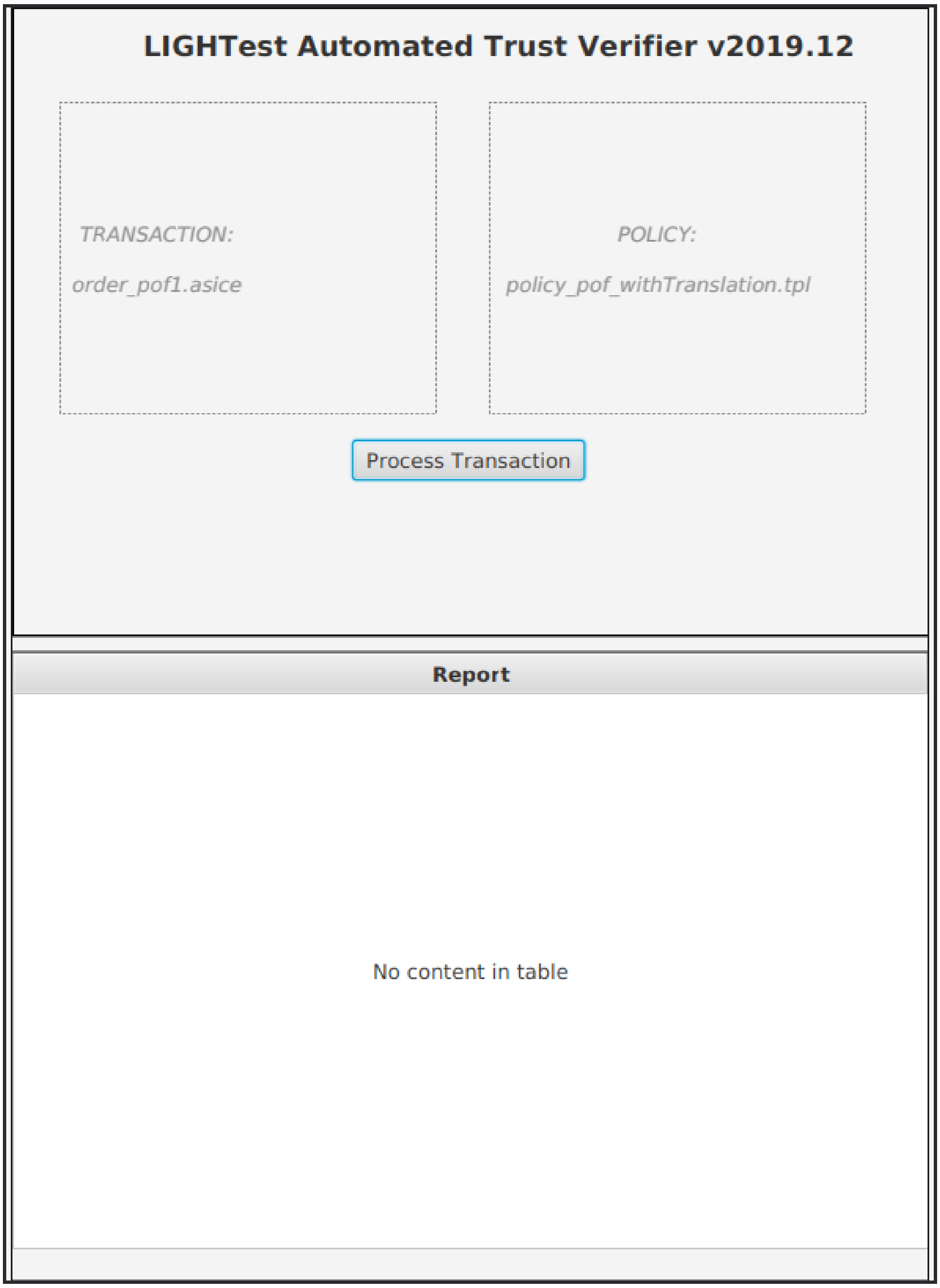

LIGHTest Automated Trust Verifier v2019.12
TRANSACTION:
Drop here, or click to open the file
POLICY:
Drop here, or click to open the file
Process Transaction
Report
No content in table

Figure 6 Start Screen


LIGHTest Automated Trust Verifier v2019.12

TRANSACTION:

order_pof1.asice

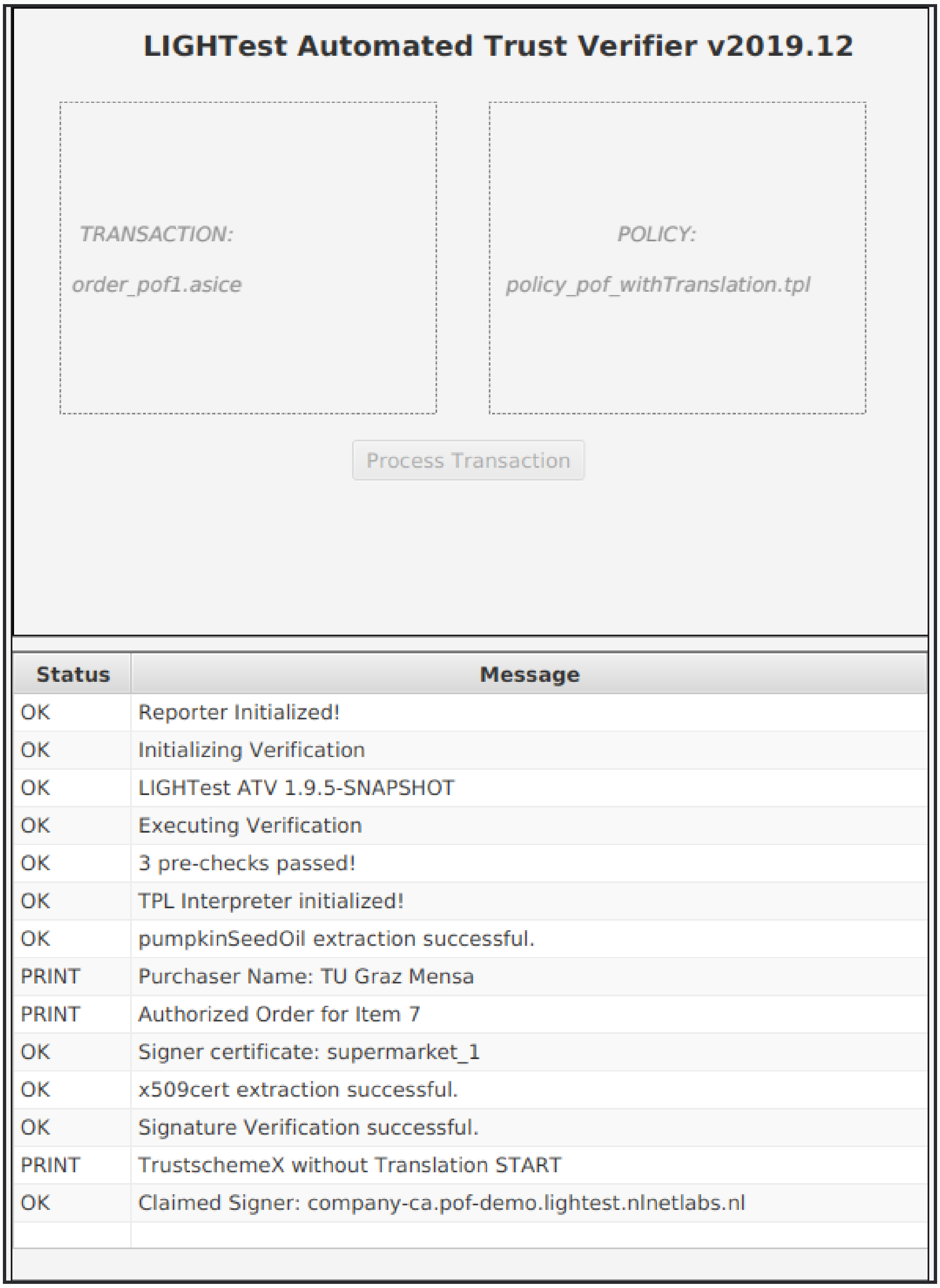
POLICY:

policy_pof_withTranslation.tpl

Report

No content in table

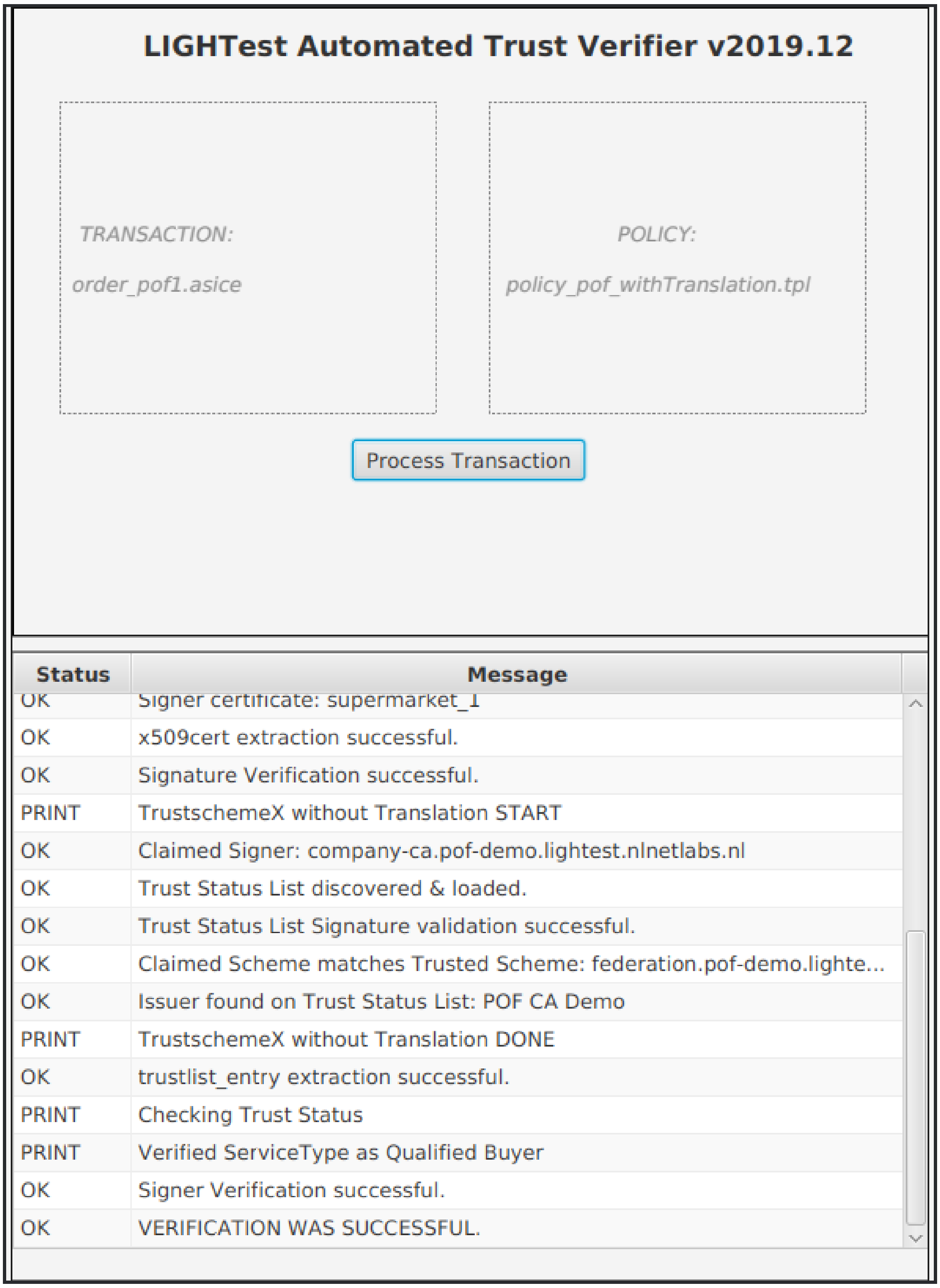
Figure 7 Upload transaction and trust policy



LIGHTest Automated Trust Verifier v2019.12

Status	Message
OK	Reporter Initialized!
OK	Initializing Verification
OK	LIGHTest ATV 1.9.5-SNAPSHOT
OK	Executing Verification
OK	3 pre-checks passed!
OK	TPL Interpreter initialized!
OK	pumpkinSeedOil extraction successful.
PRINT	Purchaser Name: TU Graz Mensa
PRINT	Authorized Order for Item 7
OK	Signer certificate: supermarket_1
OK	x509cert extraction successful.
OK	Signature Verification successful.
PRINT	TrustschemeX without Translation START
OK	Claimed Signer: company-ca.pof-demo.lightest.nlnetlabs.nl

Figure 8 Running Verification Process


LIGHTest Automated Trust Verifier v2019.12
TRANSACTION:
order_pof1.asice
POLICY:
policy_pof_withTranslation.tpl
Process Transaction

Status	Message
OK	Signer certificate: supermarket_1
OK	x509cert extraction successful.
OK	Signature Verification successful.
PRINT	TrustschemeX without Translation START
OK	Claimed Signer: company-ca.pof-demo.lightest.nlnetlabs.nl
OK	Trust Status List discovered & loaded.
OK	Trust Status List Signature validation successful.
OK	Claimed Scheme matches Trusted Scheme: federation.pof-demo.lighte...
OK	Issuer found on Trust Status List: POF CA Demo
PRINT	TrustschemeX without Translation DONE
OK	trustlist_entry extraction successful.
PRINT	Checking Trust Status
PRINT	Verified ServiceType as Qualified Buyer
OK	Signer Verification successful.
OK	VERIFICATION WAS SUCCESSFUL.

Figure 9 Verification Conclusion

7. Use Cases and Examples

7.1 Use case: Open PEPPOL eProcurement

The LIGHT^est infrastructure has a highly complex environment and a set of tools that may require developers of any industrial partner to gain knowledge about the LIGHT^est infrastructure first. This section aims to simplify the complexity of the integration by presenting use cases that effectively integrate LIGHT^est infrastructure in two OpenPEPPOL based message exchange processes on eProcurement transactions.

Two use cases have been identified and described in eProcurement that are subject of interest for the LIGHT^est project: Access Point (AP) Verification and Delegation of Document Signing to AP Provider. Each of these use cases require the publication of trust schemes. The first use case requires a publication of a trust scheme that concerns the APs, while the second use case requires a publication trust scheme that concerns the Service Providers (SPs) themselves. It should be noted that in the OpenPEPPOL infrastructure there is a distinction between the AP itself and the SP, thus the differentiation of the trust schemes.

These use cases use the LIGHT^est framework for the publication of schemes and by using the LIGHT^est tools validate the transactions against these schemes.

7.1.1 Access Point Verification use case

When AP providers join the PEPPOL eDelivery Network and sign the PEPPOL Transport Infrastructure Agreements (TIAs), they are provided with a PEPPOL Digital Certificate. This certificate contains the key information for validating all communications on the PEPPOL network. The certificate is valid as long as the TIA is valid and can be revoked if service providers are in breach of the Agreement. This ensures only known and trusted providers provide services on the eDelivery Network.

These certificates are used in the eDelivery Network of PEPPOL, where trust is established by using digital signatures in the messages exchanged between the APs, using secure and reliable protocols like OASIS AS4. By means of LIGHTest, the APs will use the ATV to validate the certificate used for the signature of the AS4 message against the published Trust Schemes. PEPPOL APs make use of all the features of AS4, including message signing using XML digital signature and XML Encryption using X.509 security tokens. These tokens also provide the trust establishment. Each AP performs message encryption and signing as part of its message submission process, and message decryption and signature validation as part of its reception process. For proper signature validation, each AP must have the PEPPOL PKI installed as a trusted PKI.

For the first use case, an emulation of an update of the OpenPEPPOL Public Key Infrastructure (PKI) is done. The purpose of this update is to strengthen the signing algorithm of the certificates from SHA-1 to the new SHA-2. By integrating the ATV into the AP software, the trust information can be moved from the AP software itself onto the LIGHT^est trust infrastructure. This has the benefit of easy and seamless update of this trust information when needed without modifying the AP software. In this PKI migration scenario, the APs will only have to update their certificates that they use for signing their message at their leisure.

The main system where verification takes place in an eDelivery transaction is inside the AP during the reception process. In LIGHTest, the Holodeck B2B AS4 AP (which is used as the AP software) is extended so that it will use the ATV to properly validate the certificate used for signature, by querying the LIGHTest infrastructure. This extension is effectively an ATV Adapter (as shown in Figure 10) which will override the default behaviour of trust verification within the AP software, replacing it with calls to ATV library methods. If the certificate verification in the ATV succeeds, the adapter will then mark the certificates as trusted and the trust verification for the transaction will succeed. Otherwise, a failure message along with an explanation of all the steps the ATV has taken and on what step the trust verification has failed will be provided back to the Holodeck B2B and the message will not be accepted.

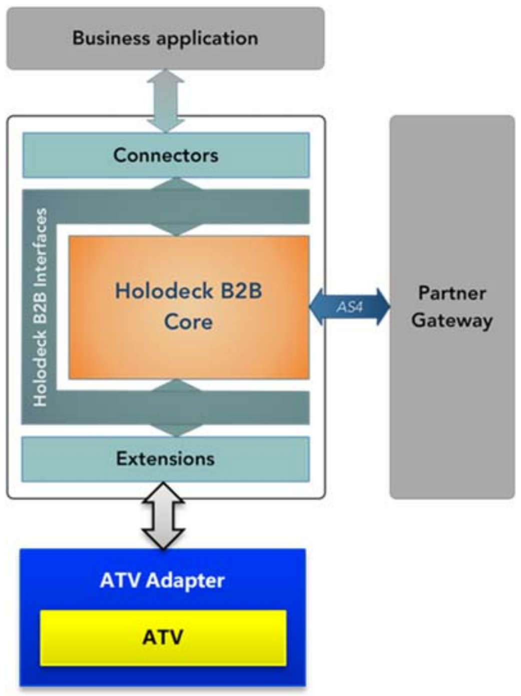


Figure 10 Holodeck B2B Component Architecture

7.1.2 eProcurement pre-award Service Provider Trust Establishment

The second use case demonstrates the use of LIGHTest technology in a cross-border, healthcare scenario. There will be a fictitious sender named German Medical Device Manufacturer in Germany that will send an invoice to French Hospital, a hospital in France for 200 insulin pumps over the PEPPOL network, using the PEPPOL specifications.

The overall goal of the second use case is to transmit a signed and encrypted cross-border ASiC Container, which typically contains the tender offer, through the PEPPOL network, issued by a German Medical Device Company to a French Hospital. Both of these entities are fictitious as is the transaction between them. In addition, the French Hospital will reply with an acknowledgment document, called the tender receipt. Because of the complexities behind the tender submission, the second use case demonstrates the use of the LIGHTest's Delegation Publisher. Economic Operators (EOs) contract with pre-award SPs to handle the complexities of business-to-business message processing. In the PEPPOL BIS, the ASiC containers used, are complex to prepare and are expected to be signed by the owner of the documents inside the container. With delegation the EOs, in our case the German Medical Device Company, can delegate the signing of ASiC containers to the pre-award SP. The receiver of the ASiC container can use a delegation provider to verify that the pre-award SP had the authority to sign the package.

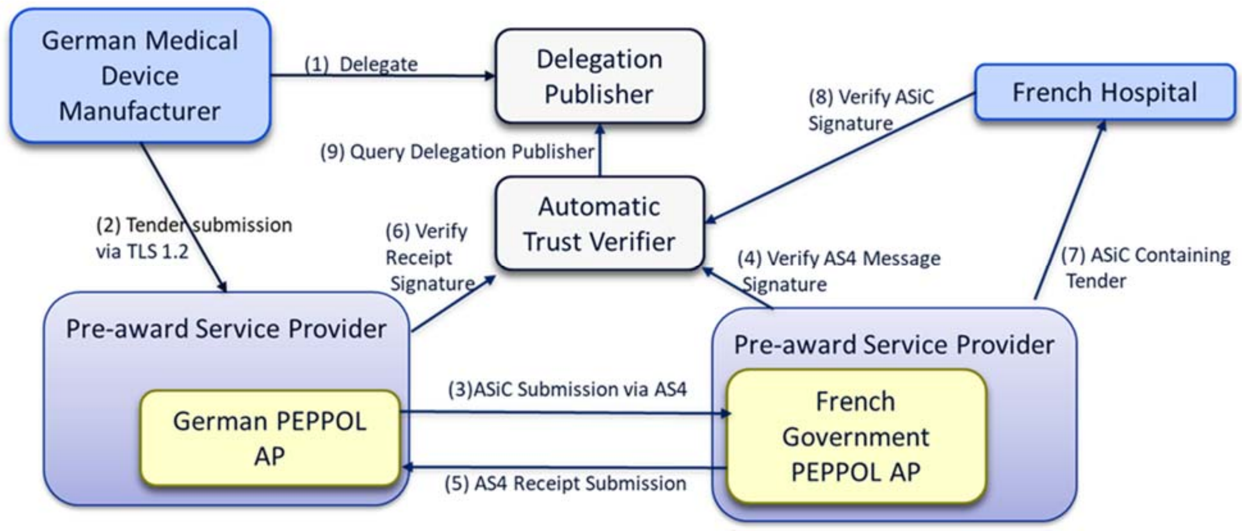

German Medical Device Manufacturer
(1) Delegate
Delegation Publisher
(8) Verify ASiC Signature
French Hospital
(9) Query Delegation Publisher
(2) Tender submission via TLS 1.2
(6) Verify Receipt Signature
Automatic Trust Verifier
(4) Verify AS4 Message Signature
(7) ASiC Containing Tender
Pre-award Service Provider
German PEPPOL AP
(3)ASiC Submission via AS4
(5) AS4 Receipt Submission
Pre-award Service Provider
French Government PEPPOL AP

Figure 11 PEPPOL eProcurement use case conceptual flow

This is a multi-step process involving five distinct message flows and their respective services built for the scenario. All the steps have trust operations and all trust operations other than TLS 1.2 authentication will be done using LIGHTest technology. The following flow describes the full process and the respective services built in favour of the pilot's scenario (see Figure 11).

1. Initially, the German Medical Device Manufacturer creates a bilateral delegation between it and its pre-award SP, and publishes it to the Delegation Provider using the LIGHTest Delegation Tools.
2. The German Medical Device Manufacturer creates and submits the tender details to its Pre-award SP for packaging, signing and submission to the French hospital.
3. The pre-award SP packages and signs the tender details into an ASiC-E and submits the tender using the PEPPOL AP.
4. The submitting PEPPOL AP creates, signs and submits an AS4 message to the receiving PEPPOL AP.
5. The receiving AP receives the AS4 message, verifies the message signature and validates the certificate used for signing using the ATV. If everything is OK, it then creates a signed AS4 receipt and sends it back to the submitting AP of the German Medical Device Manufacturer.
6. The submitting AP verifies the receipt signature and validates the certificate used for signing using the ATV.
7. The receiving AP pushes the tender ASiC-E container to the French Hospital.
8. The French Hospital verifies the signature of the ASiC-E using the ATV, including the delegation verification.
9. The ATV performs a query to the Delegation Provider, to validate the delegation of the signature and returns the result back to the French Hospital. For simplification reasons, this step will be conducted in the receiving AP on behalf of the French Hospital.

7.2 Use case: Correos Digital Services

The specific use cases of the integration of Correos digital products with LIGHTest that will be proposed on this section, include integration as APIs calling at any point of the service. Specifically, it'll be called at moments where it would add value to any business transaction. This pilot is mainly focus on market-check of the ATV access door as way to present LIGHTest as a trust service. Moreover, LIGHTest is capable to offer a quick and easy check with any due Level of Assurance (LoA) policy and give some kind of assurance of the check to infrastructure customer. It was proposed that this assurance could be shown as a "badge" of LIGHTest checked.

Within all the Correos Digital services they were chosen *My Mailbox* and *My Notifications*:

- **My Mailbox** ("Mi Buzón") is a digital service for citizens, companies and governments enabling them to send and receive documentation. Information is stored in cloud with all legal guarantees and high security standards. Sender and receiver are validated and uniquely identified by Correos. Individuals subscribe to any verified business/government agency to start receiving trusted information. LIGHTest ATV would be used to inform users about document eDelivery LoA for intra-European entities, and LoA translation in the case of non-European entities.

- **My Notifications** ("Mis Notificaciones") is a digital service foreseeing centralization and management of governmental eNotifications for one or several individuals or legal entities. In such secured and trusted communications, it would be useful to offer value by double-checking with a certified entity in Europe that such communication is done accordingly to current legislation.

7.2.1 My Mailbox - eDelivery use case

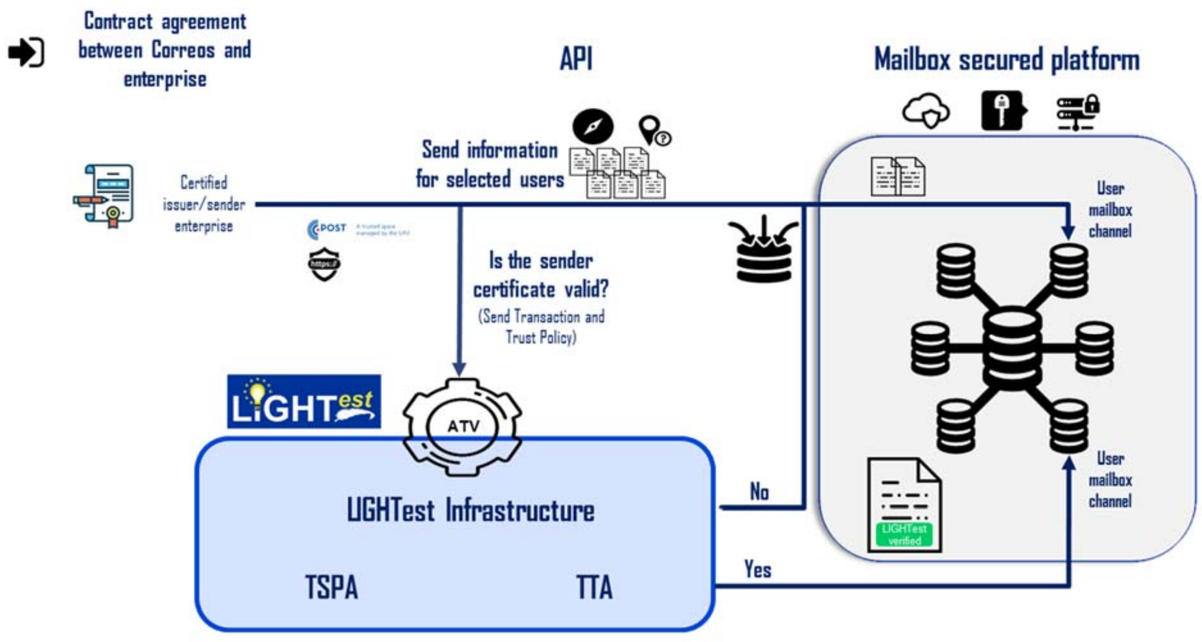

Contract agreement between Correos and enterprise
API
Mailbox secured platform
Send information for selected users
Certified issuer/sender enterprise
Is the sender certificate valid?
(Send Transaction and Trust Policy)
User mailbox channel
ATV
LIGHTest Infrastructure
TSPA
TTA
No
Yes
LIGHTest verified
User mailbox channel

Figure 12 : My Mailbox functional design with LIGHTest

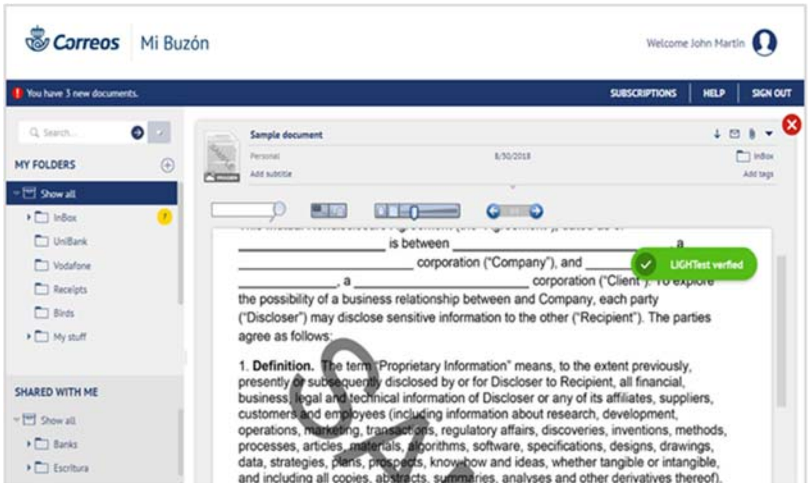
By means of LIGHTest, Correos service will validate that the entity sending a document operates under a known Trusted Scheme (eIDAS eSignature, eIDAS eSeal, NIST electronic signature, etc.); querying, through the ATV:

- First, the TSPA will fetch the metadata of the Trusted Scheme claimed and validate the trust on the sender.

- Second and optional, query the TTA to determine the trust scheme level for the case that the trust on the sender could not be verified by the TSPA only.

LIGHT^{est} ATV would be used to verify the LoA of the transaction based on a Trust Scheme and possible LoA translation in the event the senders under trust schemes are different from those accepted for the transaction.

Currently, *My Mailbox* service allows users to visualize stored or received documents in the account as well as associated metadata: Title, date, tags, description, etc. The result of the electronic transaction validation will be shown by adding a watermark to the document received in *My Mailbox* platform. Such watermark will enable the user of the service to clearly identify if the document transaction is compliant with the chosen eIDAS Trust Scheme (i.e: eSeal, eSignature, eTimestamp, etc.), visually showcasing the added value on using LIGHTest.
Once a document is received by the *My Mailbox* service, it is shown in the system as follows (including the watermark):



________________ is between ________________, a ________________ corporation ("Company"), and ________________, a ________________ corporation ("Client"). To explore the possibility of a business relationship between and Company, each party ("Discloser") may disclose sensitive information to the other ("Recipient"). The parties agree as follows:

1. **Definition.** The term "Proprietary Information" means, to the extent previously, presently or subsequently disclosed by or for Discloser to Recipient, all financial, business, legal and technical information of Discloser or any of its affiliates, suppliers, customers and employees (including information about research, development, operations, marketing, transactions, regulatory affairs, discoveries, inventions, methods, processes, articles, materials, algorithms, software, specifications, designs, drawings, data, strategies, plans, prospects, know-how and ideas, whether tangible or intangible, and including all copies, abstracts, summaries, analyses and other derivatives thereof).

Figure 13 LIGHT^{est} view for My Mailbox

Trust policies have to be defined to help LIGHT^{est} understand what has to be checked in order to evaluate information within its Trust Scheme. Once the trust policy is created in natural language, it will be converted into TPL language using the Trust Policy Authoring Tool GUI.

The Data sent must be signed/sealed by an advanced eSignature/eSeal according to eIDAS regulation. If this is not fulfilled, check possible translations for eIDAS eSignature/eSeal.

The basic electronic transaction (containing the electronic signature) created for the My Mailbox scenario has to comply with the ASiC container standards. This ASiC container also includes the trust policy, as the ATV is only able to interact through ASiC containers.

The following two use cases can help clarify My Mailbox general use case and its applicability in cases of business use:

- **Use case 1 - intra-European communication**: Correos (as a verified sender within My Mailbox service) sends a document to a citizen that requires a security "double-check" because of its business or legal value. So, it will be validated if the communication (understanding it as the act of sending the document) is compliant with the definition and requirements of eSeal or eSignature within eIDAS regulation. It is an example of use related to the TSPA.

- **Use case 2 - non-European communication**: Similar to previous one, but the document would be sent by a country outside of EU (e.g. Turkey) instead, where eIDAS requirements would not apply. In this example, the compliance against eSeal or eSignature and its translation from a non-EU scheme would also be checked having an example related to the TSPA and the TTA.

7.2.2 My Notifications - eNotifications use case

My Notifications is a digital service foreseeing centralization and management of governmental eNotifications for one or several individuals or legal entities. In such secured and trusted communications, would be useful to offer value by double-checking with a certified entity in Europe that such communication is done accordingly to current legislation.

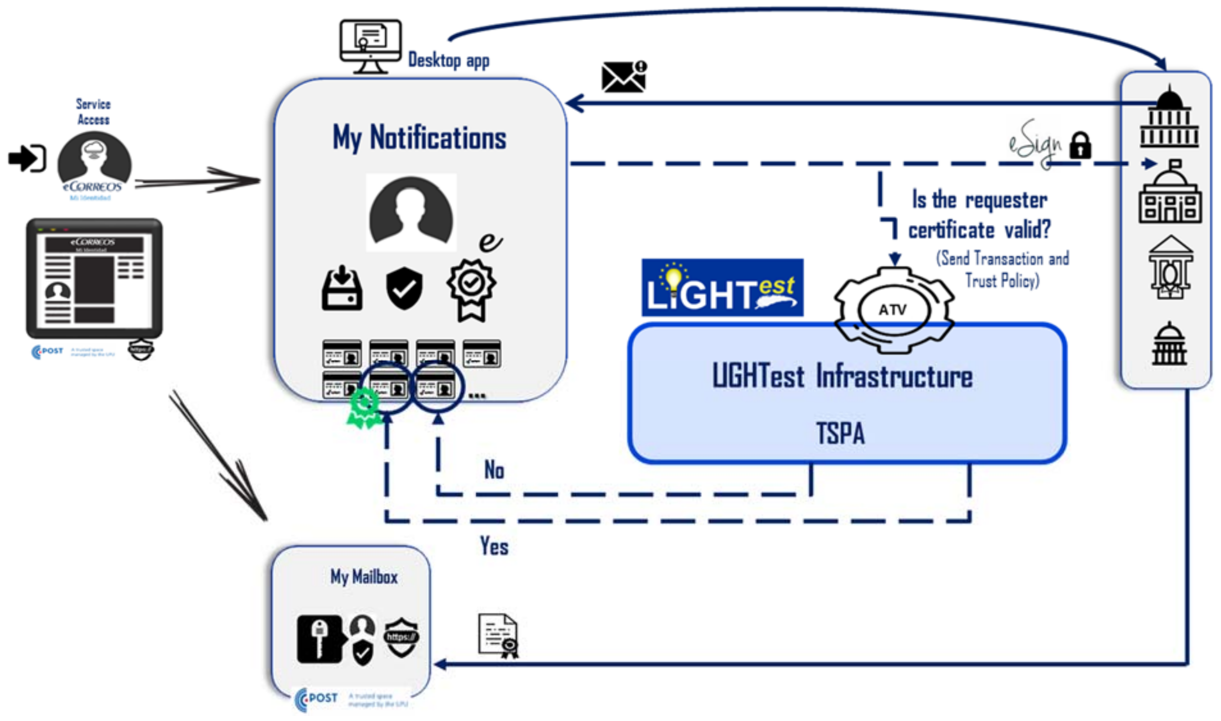
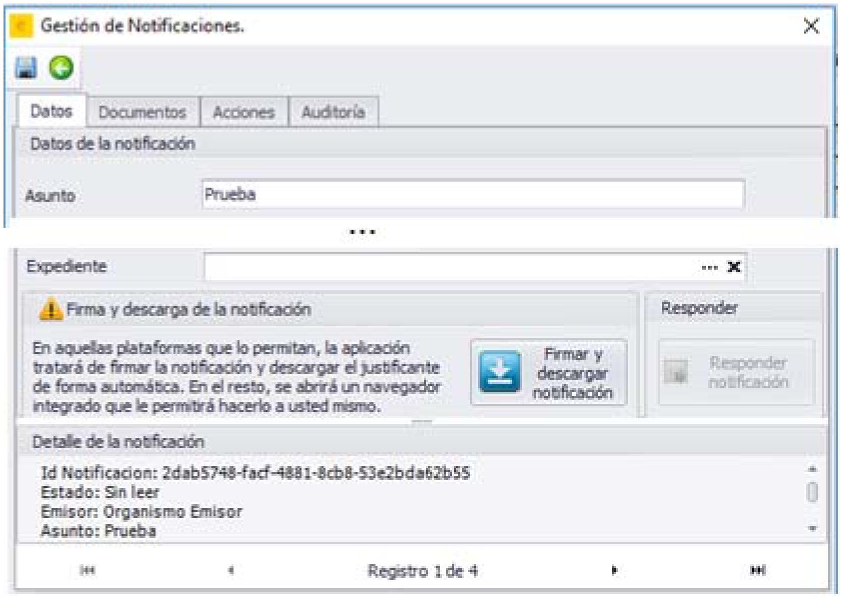


Figure 14 My Notifications functional design with LIGHTest

By means of LIGHTest, Correos service will make their notifications request more secure and trusted, providing value by double-checking with a European certified entity that such request is done according to current eIDAS regulation and validating Trusted Scheme eSignature / eSeal. Trust Policies must be defined to help LIGHT^{est} understand what has to be checked to evaluate information within its Trust Scheme. Once more, natural language is used to present the Correos *My Notifications* trust policy and then converted into TPL language using the Trust Policy Authoring Tool GUI.

Data request to a government organization must be signed/sealed by qualified eSignature/eSeal of eIDAS.

The basic electronic transaction (containing the electronic signature) created for *My Notifications* scenario must comply with ASiC container standards. This ASiC container also includes the Trust Policy aforementioned, as the ATV is only able to interact through ASiC containers. Once the user of the *My Notifications* service chooses to download the notification by clicking on "*Firmar y descargar notificación*" ("Sign and Download Notification"), all actions defined for the scenario will be carried out. During the execution of this notification request, the *My Notifications* service will check the request through LIGHT^{est} ATV with eIDAS eSignature or eSeal Trust Schemes.



En aquellas plataformas que lo permitan, la aplicación tratará de firmar la notificación y descargar el justificante de forma automática. En el resto, se abrirá un navegador integrado que le permitirá hacerlo a usted mismo.

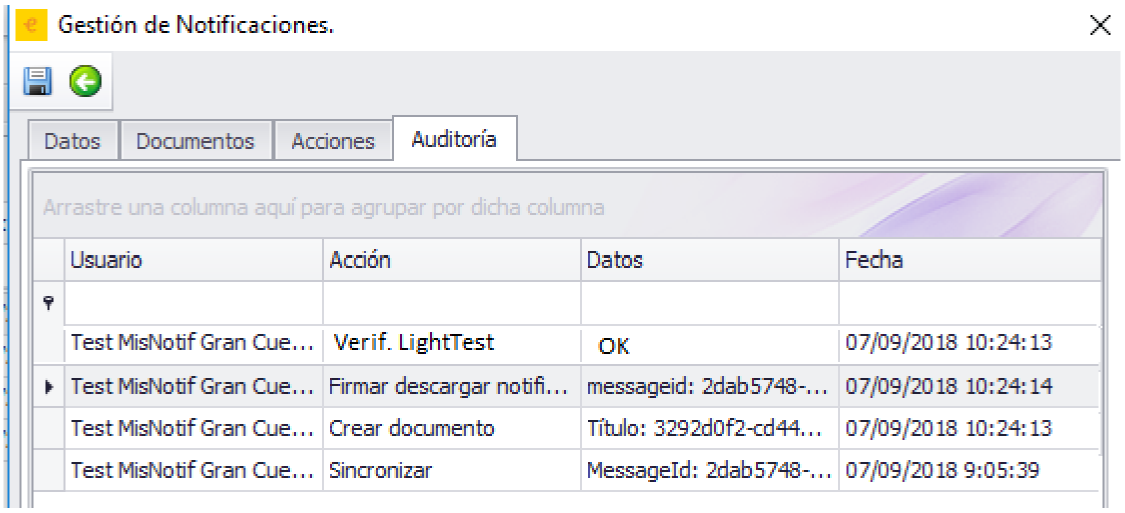
Id Notificacion: 2dab5748-facf-4881-8cb8-53e2bda62b55
Estado: Sin leer
Emisor: Organismo Emisor
Asunto: Prueba

Figure 15 My Notifications - example of current notification details

Furthermore, the document associated with the notification will be downloaded and stored at My Mailbox. Along with the request for these documents, My Notifications will need to generate an ASiC container with the transaction (information of the request for notification) and the previously created specific policy to be checked. This ASiC container is sent to LIGHT^{est} ATV for verification and service will manage the response.

If verification is successful, a watermark will be added to the request details and information, so the user can confirm a request has been issued under some specific eIDAS Trust Scheme (whether eSeal or eSignature).

The audit register stores the ATV response and the information of the validation performed against LIGHT^{est} ATV:


Gestión de Notificaciones.
Datos
Documentos
Acciones
Auditoría
Arrastre una columna aquí para agrupar por dicha columna

Usuario	Acción	Datos	Fecha
Test MisNotif Gran Cue…	Verif. LightTest	OK	07/09/2018 10:24:13
Test MisNotif Gran Cue…	Firmar descargar notifi…	messageid: 2dab5748-…	07/09/2018 10:24:14
Test MisNotif Gran Cue…	Crear documento	Titulo: 3292d0f2-cd44…	07/09/2018 10:24:13
Test MisNotif Gran Cue…	Sincronizar	MessageId: 2dab5748-…	07/09/2018 9:05:39

Figure 16 My Notifications - audit log

Currently, the notification associated with this testing agency is in the form of a pdf document. An XML document containing the certification of notification request will be created (plus a module developed within the *My Notifications* architecture). Within this XML, the transaction content sent to LIGHT^{est} ATV will be included as well as the Trust Policy checked. Once the Trust Policy used in this example is created / generated, it is retrieved when LIGHT^{est} ATV is called.

7.3 LIGHT^{est} Mobile ID scheme using FIDO

As the use of mobile devices is more and more dominating the landscape of electronic transactions one of the key applications of LIGHT^{est} is a system of mobile electronic identities that can be used for a large group of use cases like identification, authentication and signing of transaction data. Since these identities are typically related to a primary identity (e.g. a government issued eID card) the focus is on derived mobile IDs that have been generated after a process of initial identification and a procedure of deriving credentials. These credentials are linked to the primary identity and can be securely stored on mobile devices. The primary ID in this scenario acts as a root of trust for the derived credentials.
One of the key building blocks of such a use case is a mobile ID scheme that can propagate trust information from the original source of the primary ID to the actual relying party of the mobile ID. Within LIGHTest, the reference implementation of mobile ID is based on a FIDO scheme with extended identity information. The goal of the Fast Identity Online (FIDO; https://fidoalliance.org/) alliance is to define an interoperable specification for mobile authentication and to overcome existing fragmentation and silos. More than 330 products have already been certified according to the FIDO specifications and several large roll-outs have been done, involving multi-million numbers of end users.

For the LIGHT^{est} demo implementation, a concrete use case has been selected as reference:

- An end user wants to subscribe to an e-government service on a mobile device. With an initial, pseudonymous registration, the user can register to the service using strong authentication (i.e. 2-factor authentication based on FIDO). With this pseudonymous registration the user can already access some basic services. For advanced and personalised services, the user has to be identified with a known and potentially pre-defined LoA.
- To provide identification, the user is redirected to an external identity provider that is able to verify the primary ID (e.g. an ID card) of the user and to issue an attestation over the ID attributes.
- With the attestation of identity attributes, the proof of possession of authentication credentials, and the attestation of the authenticator type, the demonstrator backend queries the corresponding trust scheme using the LIGHT^{est} infrastructure. With this query the total LoA of the mobile ID is determined, taking into account the strength of the primary ID and the authenticator LoA.
- The demonstrator backend now connects the identity attestation and LoA attestation with the corresponding FIDO authentication credentials by issuing a certificate over the FIDO public key.
- With these certified credentials the user can now access the enhanced and personalized services of the e-government platform.

The architecture of the demonstrator solution is designed in a way to provide a maximum of modularity. This is especially relevant for real-world use cases where a relying party might already have a relationship with an ID provider and where an independent standard FIDO authentication may be in use. With the modular architecture it will be possible to add a LIGHT^{est} integration on top of a running system and to easily exchange individual components if they provide the corresponding standards. An overview of the architecture is shown in Figure 17.

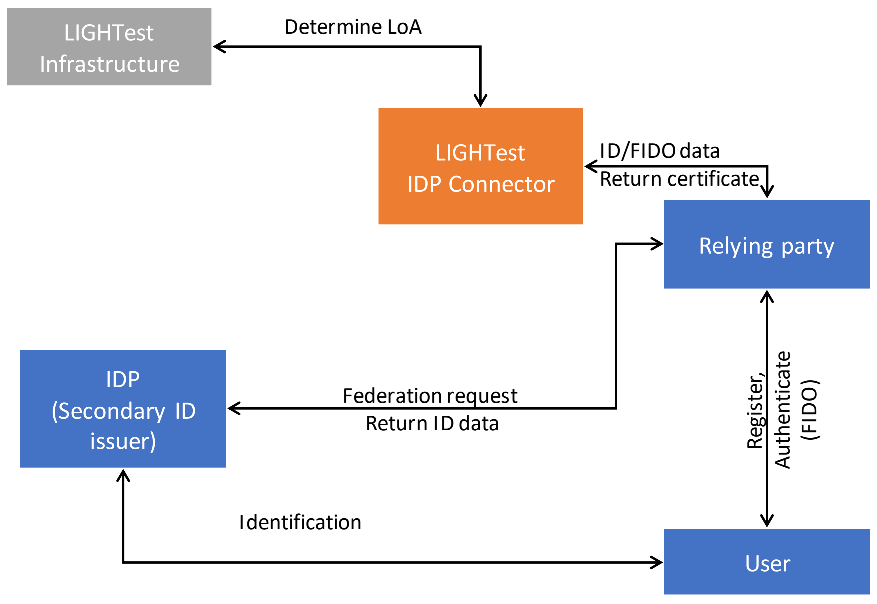


Figure 17 : Architecture overview of the involved demonstrator application components.

The app flow starts with the initial registration of the FIDO user if the user is not already registered with this device (see sample screenshots in Figure 18). In a first step the user has to provide a username which can be pseudonymous. According to the FIDO flow the user can now choose between different available authenticators that fulfil the policies defined by the service provider. In the demo application the user can choose between a fingerprint authenticator and an EMV-card based authenticator that was developed in the LIGHT^{est} sister-project FutureTrust. The choice of the authenticator type will influence the overall LoA.

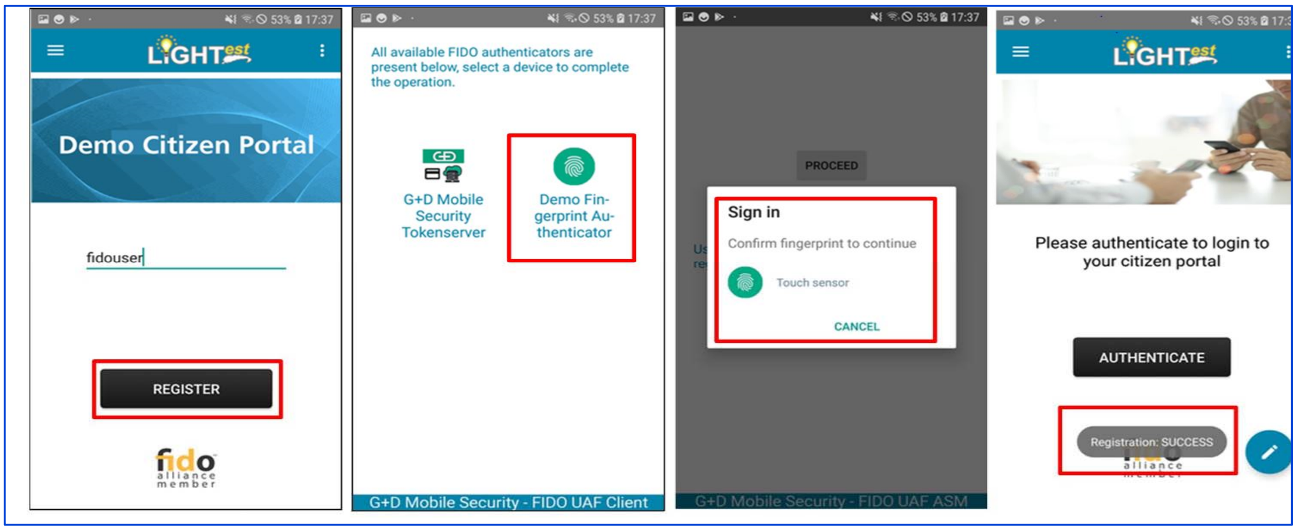
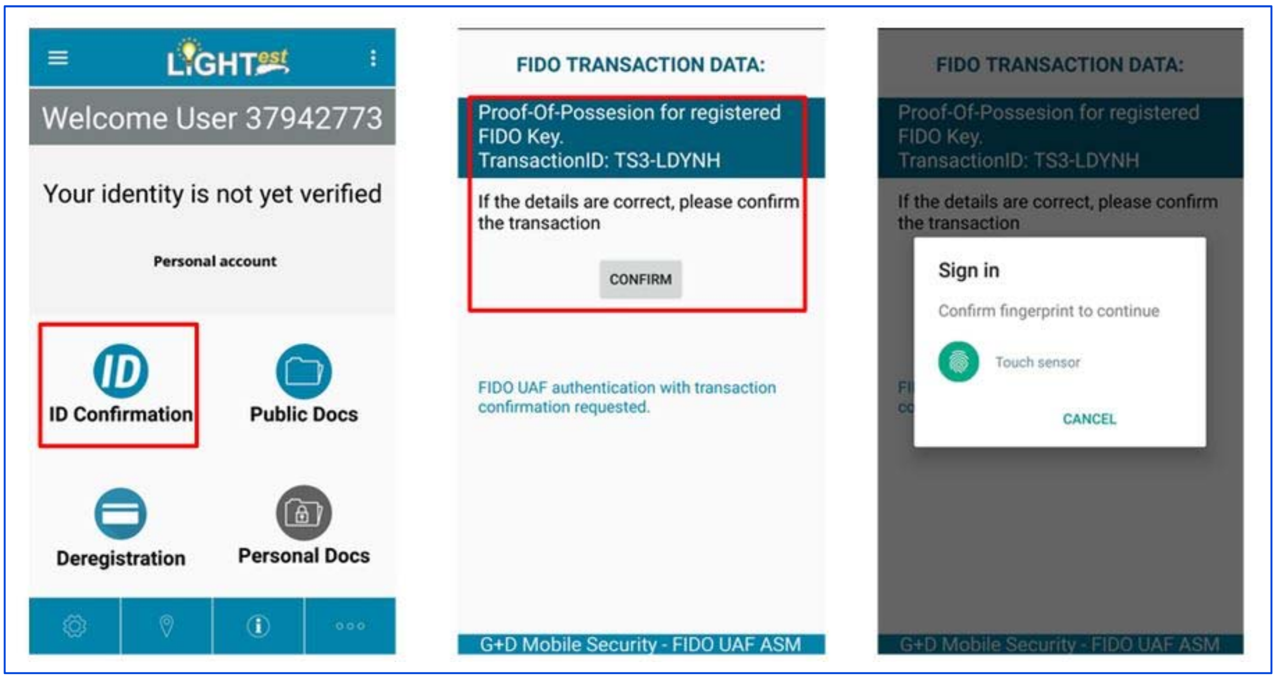

Demo Citizen Portal
fidouser
REGISTER
All available FIDO authenticators are present below, select a device to complete the operation.
G+D Mobile Security Tokenserver
Demo Fingerprint Authenticator
G+D Mobile Security - FIDO UAF Client
PROCEED
Sign in
Confirm fingerprint to continue
Touch sensor
CANCEL
G+D Mobile Security - FIDO UAF ASM
Please authenticate to login to your citizen portal
AUTHENTICATE
Registration: SUCCESS

Figure 18: Screen shots of the user registration step within the demo application.

After this initial registration step the user is registered but still not identified, which allows access only to some basic services. By selecting the “ID Confirmation” button, the user is asked to provide the proof-of-possession of the private key by confirming the transaction request with the transaction ID retrieved from the IDP connector. When the user confirms the transaction, a signed transaction is available, and the user is re-directed to the ID provider. The corresponding screen shots are shown in Figure 19.



Proof-Of-Possesion for registered FIDO Key.
TransactionID: TS3-LDYNH

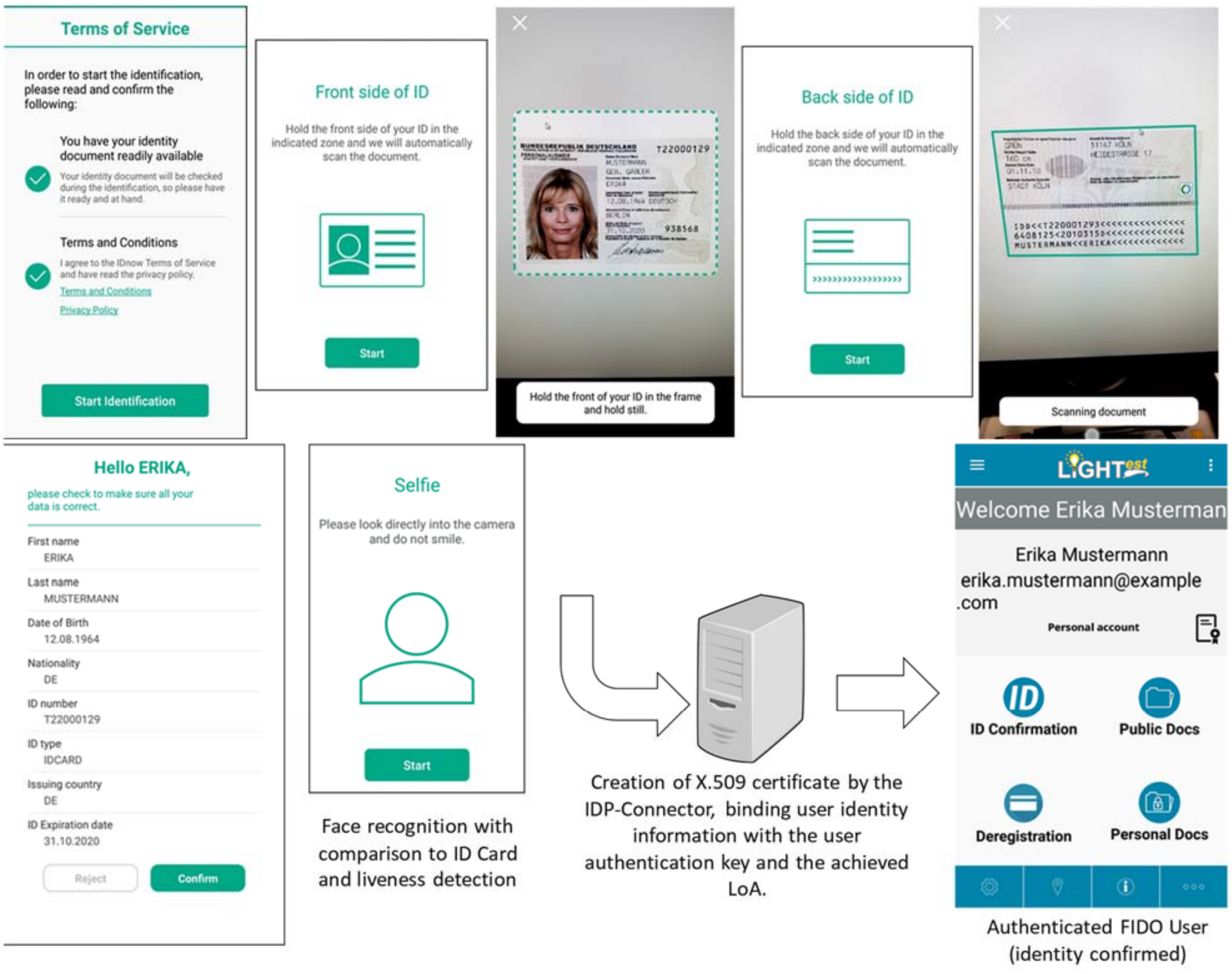
If the details are correct, please confirm the transaction

FIDO UAF authentication with transaction confirmation requested.

Figure 19: Screen shots of the proof-of-possession step during identification of the user.

Now the actual identification process starts. This is generally handled by the external ID provider. In case of the demo application this is the demo version of the commercial identification service IDNow. The service comprises of a client SDK that is integrated into the demo application and which handles a self-service flow in which a user scans his/her ID document and makes a self-portrait image. This data is sent to the IDNow demo backend and is verified automatically. The server backend then returns the verified ID data with a corresponding attestation.

With the ID data and the proof-of-possession FIDO data the API of the IDP connector is called for creation of the X.509 certificate. The IDP connector then calls the LIGHTest demo infrastructure to determine the LoA of the provided ID/authenticator combination. This is invisible to the end user and is completely handled by the app and the backend servers without user interaction. In the end, the user receives the confirmation that he/she is now registered as an identified user. Now the previously locked services are available, in this demo case the “Personal Docs” button. The identification process is shown in Figure 20.



In order to start the identification, please read and confirm the following:

You have your identity document readily available

Your identity document will be checked during the identification, so please have it ready and at hand.

Terms and Conditions

I agree to the IDnow Terms of Service and have read the privacy policy.

Terms and Conditions

Privacy Policy

Hold the front side of your ID in the indicated zone and we will automatically scan the document.

Hold the front of your ID in the frame and hold still.

Hold the back side of your ID in the indicated zone and we will automatically scan the document.

please check to make sure all your data is correct.

First name	ERIKA
Last name	MUSTERMANN
Date of Birth	12.08.1964
Nationality	DE
ID number	T22000129
ID type	IDCARD
Issuing country	DE
ID Expiration date	31.10.2020

Please look directly into the camera and do not smile.

Face recognition with comparison to ID Card and liveness detection

Creation of X.509 certificate by the IDP-Connector, binding user identity information with the user authentication key and the achieved LoA.

Welcome Erika Musterman

Erika Mustermann
erika.mustermann@example.com

Authenticated FIDO User (identity confirmed)

Figure 20: Screen shots of self-service identification process

The successful integration shows that the concept of a mobile ID scheme with known LoA and trust propagation from the secondary ID issuer to the relying party is possible and that all developed components interact as planned. As a consequence, the demonstrator application shows that the advantages of the FIDO authentication system, of various ID provider solutions and of the LIGHT^{est} ecosystem can be combined to provide a powerful tool for trustworthy mobile ID applications within a defined trust domain.

7.4 Examples of use of LIGHT^{est}

The different three technical pillars of LIGHT^{est} (TSPA, TTA and DP) services can be applied in a multitude of different ways, being some examples the following:

For Trust Services, LIGHT^{est} can be used to verify published information on Trust Lists. The published information could be names, certificates, objects, things, basically anything and that list of items could be verified. For instance, it could be possible to verify that person X has a valid driver license in country Y using LIGHT^{est} to check the published list 'List of valid driver's license in Y' that was published by the official authority of transport of that country. This function uses the Trust Publication Authority (TSPA).

For Trust Translations, LIGHT^{est} can be used in a cross-borders or translation scenario where some items may transfer into a different kind of item in other scenarios. Simply said, this would imply that object A on List 1 would be translated to object 2 on List B, with a previous legal agreement that establishes that relation. For example, one European company and one Asian company sign an agreement where the signature of European CEO will be equivalent to a high level certificate submitted by Asian company. In terms of validation, the European company will check the certificate to accept it as their CEO signature. This function uses the Trust Translation Authority (TTA).

For Trust Delegation, LIGHT^{est} could be used in a variety of delegation scenarios or simply to check or verify the delegation itself. An example of this function would be to check if an employee has the delegation rights from a company to purchase items in the companies' name. LIGHT^{est} would check if the employee is on the Companies published Delegation list of Employees with Purchasing Rights. This function uses the Delegation Authority (DA).

8. Beyond LIGHT^{est}

LIGHT^{est} allows you to use a global, known and trusted infrastructure to: Retrieve declared policy and certification details from partners; verify those declared policies and certifications from Trust Lists; determine trust assurances behind partners, and so facilitate your own decision making. As such LIGHTest does not provide an alternative to eIDs, business registers or certifications, neither does it allow you to outsource trust decisions. LIGHT^{est} is intentionally flexible and addresses an increasing need to establish ad hoc trusted relationships between counterparties.

So, to what uses could LIGHT^{est} be applied in the future, meeting the varying needs of differing relationships?

LIGHT^{est} has already been incorporated into its initial use-cases. In particular, this is in the context of certification- based policies such as Public Key Infrastructure. An additional use-case that has been examined is in the context of higher education and the trust in qualifications and educational standards.

However, arguably the largest potential market is for compliance with contract-based policies. For example, in a business-to-business relationship, the counterparties may have particular requirements regarding child-labour, or health-and-safety procedures. These requirements may not have any associated formal certifications. So how can these be as easily incorporated into LIGHT^{est} as the certified policies, without a Trusted Third Party cannot be checked with formal authority?

This problem has already been addressed in many industries, especially in the business-to-consumer world like applying online for car insurance. Here the consumer has to declare many details of driving and claim history which are not easily verifiable. The fundamental is that these declarations, and any subsequent issuance of a car insurance policy are governed by contract law, rather than legislation. The outcome is that in the event of a false declaration, a subsequent claim will be void, and the contract will be null. In this case some of the declarations are verified by trusted third parties, like credit rating agencies, and others by government agencies, but the more complex, subjective (such as health) and expensive to verify declarations are not verified.

In the case of LIGHT^{est}, a counterparty may appoint a trusted third party, such as a provider of accredited management systems certification to provide the required assurance and certification.

This contractual relationship works well in business-to-business transactions. For Industry 4.0, we need Integrity and transparency above and beyond business searches, for software driven supply chains and ad hoc suppliers and customers. Dynamic factors to consider with the partners are: Quality Mandates; Safety; Ethics; Environmental; Documentation / Compliance; Security (Cyber and Physical); Resilience; Access Control; and Authorisation.

Below (Figure 20) is an example of a set of possible requirements for a fictitious business-to-business relationship:

Quality Mandate - certification of services	Y/N
Safety - do staff health conditions comply to…?	Y/N
Ethics - is there a business ethics policy?	Y/N
Environment - is there a carbon reduction strategy?	Y/N
Compliance – does the company comply to ISO…….?	Y/N
Security (Cyber / Physical) - is there a dedicated CSO?	Y/N
Resilience – is there a resilience strategy implemented?	Y/N
>Off-site data backup?	Y/N
Access Control	
>Biometrics (fingerprint)?	Y/N
>Offboarding Process?	Y/N
Authorisation / Delegation Policies	
>Purchasing Manager Grade C < €50,000	Y/N

Figure 21 Commercial Requirements in example Industry 4.0

In this particular case, there are a number of issues that can only be self-declared. Using LIGHT*^{est}*, these can become part of each and every transaction, and if they are later found to be false, and there is a loss that has been suffered, there may be a strong case for legal action and compensation.

LIGHT*^{est}*, in the medium term, and with some development might be seen as a declaration mechanism in the business-to-business environment, which can extend across many different issues which may be difficult to certify by Trusted Third Parties.

LIGHT*^{est}*, in the longer term could even be considered as an interlocuter between two artificial entities, who interacting. Consider this in bank trading platforms where automated trading might take place in a more ad hoc market, where declarations (such as liquidity attested by a central bank) could be used in a vast global open bourse.

LIGHT*est* is flexible and extensible, and has a potential only limited by imagination.

THE LIGHT*est* COOKBOOK

LIGHT*est*, initially funded by the European Commission is 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. User organisations 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 LIGHT*est* open source software to easily query this trust information to verify trust in simple signed documents or multi-faceted complex transactions.

This is a simple step-by-step guide to establishing a framework demonstrating the basic rules of LIGHT*est* in an everyday ICT environment.

https://lightest-community.org

LIGHT*est* has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321