iSHARE scheme v0.1

Welcome to Confluence and welcome to the iSHARE scheme!

The iSHARE scheme is a collaborative effort to improve conditions for data-sharing for organisations involved in the Dutch logistics sector. Within two years the project aims to establish a fully functional "scheme" which manages a set of agreements made between involved organisations. The scope of the iSHARE scheme focusses on topics of authentication, authorization and identification. In January 2018 the iSHARE scheme will be ready to open up to the market after two years of building and adjusting agreements to improve the conditions for sharing data.

On Confluence, we co-create the iSHARE scheme "on the go". Here, we bring the v0.1 version or "startdocument" to a v1.0 scheme document that is ready for implementation. What iSHARE is - and how we will co-create exactly - is described in the Introduction. The Key features, guiding principles & assumptions and the Roles & Responsibilities are described in separate chapters. All remaining Legal, Operational, Functional and T echnical topics of the iSHARE scheme are described in detail in separate chapters as well.

Introduction

The iSHARE project was initiated by the Neutral Logistics Information Platform (NLIP) through a tender project. NLIP asked market companies to present plans to lower barriers for more efficient data exchange in the Dutch logistics sector. The combination of the companies Innopay and Maxcode eventually won the tender with their plan to set-up a scheme of multilateral agreements instead of, for instance, a more technology centric approach to build a software platform. Since June 2016 the iSHARE project team worked towards the realisation of this scheme which is scheduled to go live in January 2018.

The establishment of the iSHARE scheme knows four phases:

- Phase 1: (Jun 2016 - Jan 2017): Preparatory phase, results in startdocument v0.1 which provides the preliminary scope for the iSHARE scheme based on identified challenges and use cases of involved organisations.
- Phase 2: (Jan 2017 - Jun 2017): Co-creation phase, during this phase involved organisations work collaboratively towards iSHARE scheme v1.0 which contains the first full set of agreements for improved data sharing conditions. Involved organisations work towards a full set of agreements in four working groups: Legal, Operational, Functional and Technical. The set of agreements will be realised and tested in the iSHARE reference implementation that will be developed alongside.
- Phase 3: (Jun 2017 - Jan 2018): Soft launch phase, during this phase the involved organisations organise how the iSHARE scheme's integrity and sustainability are kept in check. This involves setting up procedures for accession to the scheme and/or establishing/designating an organisation entrusted with the responsibility to safeguard the integrity of the iSHARE scheme.
- Phase 4: (Jan 2018 and onwards): iSHARE live. iSHARE opens up to any party interested and willing to abide by the agreements as set out by involved organisations.

This document is, at the time of writing in January 2017, the iSHARE Scheme v0.1 "startdocument" and the result of phase 1. The startdocument serves as the output of phase 1 and the input for phase 2, during which working group members will take ownership of the document and make it evolve towards iSHARE scheme v1.0. The document contains and proposes a number of topics which need to be detailed further by the iSHARE co-creation working groups. The document serves as a discussion starter and is by no means meant as a prescription. Working groups are free to propose additions, removals or modifications to the topics in this document (read more on the purpose of the startdocument).

The remainder of this document contains all the topics that need to be detailed by workgroup members. This chapter further describes the context of the iSHARE project and provides background information (read more about the goals and scope of the iSHARE scheme, read more about the c o-creation in working groups). The chapters that follow provide insight into what key features, guiding principles and assumptions are considered for the iSHARE scheme, which roles and responsibilities are foreseen, and what Legal, Operational, Functional and Technical agreements are needed.

Goals and scope of the iSHARE scheme

The iSHARE scheme is a collaborative effort to improve data-sharing of organisations involved with the Dutch logistics sector.

The ambition of the iSHARE project is to take away barriers in the way of sharing data, to empower new forms of collaboration in chains and to help scale up existing initiatives that aim to improve conditions for data exchange. The underlying assumption is that if we are able to improve our common skill to handle data in a smart and efficient way, this will lead to a more efficient use of infrastructure, less carbon emissions and a more competitive logistics sector.

The iSHARE scheme's scope focuses on three main topics that are of importance in any data exchange context:

1. Identification
2. Authentication
3. Authorization

These three aspects are considered crucial in any communication between parties, also in the context of exchanging logistical data. Within the iSHARE scheme, agreements are made on these three topics with the aim of working towards a more uniform, straightforward and controlled way of exchanging data on a bigger scale than is possible right now.

- Uniform: one way of working which is compatible with all types of modalities, big and small organisations, public or private organisations, suppliers or receivers of data or their softwarepartners, etc. iSHARE aims to create new possibilities for efficiency improvements, time gains and cost savings.
- Straightforward: Easy to connect with new, existing and third-party business partners throughout the sector, more certainty on trustworthiness of parties you exchange data with, a building block which is easy to implement by your software partners or your IT department, an addition that empowers your existing solutions.

- Controlled: The basic principle within iSHARE is that the owner of the data stays in control at all times; the owner decides with whom what data is exchanged for how long.

These three aims can only be reached when a variety of perspectives is considered during the establishment of the scheme. To this end, a variety of organisations are involved in defining the agreements for iSHARE. During the co-creation phase of the iSHARE project, the involved organisations invested in the iSHARE scheme in terms of expertise. To read more about the co-creation process, we refer to the chapter on co-creation in working groups.

Co-creation in working groups

The iSHARE scheme is established by its participating organisations. Through the iSHARE co-creation process, the collective expertise of participants will lead to a practical and widely applicable scheme. This process is fueled by the belief that a practical solution is the result of dialogue and deliberation: participants have to collaboratively think of a generic solution which solves both their own challenges but also those of other participants. It is important to note that at the beginning of the co-creation process there is no clear description of what the eventual scheme must look like: what the iSHARE scheme entails or doesn't entail is the result of the co-creation process and the agreements made by the participants.

The co-creation process is structured in the following ways:

- There are four topics with dedicated working groups: Legal, Operational, Functional and Technical (LOFT). The assumption is that for a fully functional scheme, at least these topics need to be discussed and organised.
- The working groups start with input in the form of the "startdocument". This document provides an overview of relevant topics that will be detailed by the working groups.
- The regular meeting of working groups and the agreements made within the working groups are facilitated by the chairman and secretary of the working groups.

The participants of the co-creation process have a variety of backgrounds: private and public organisations, bigger and smaller organisations, (serving) different modalities, both providers and receivers of data. The variety of organisations ensures that the iSHARE scheme will be widely applicable.

Purpose of this document

This document (of which version 0.1 is known as "startdocument") contains the current state of agreements within the iSHARE scheme. Version 1.0 and up of this document will contain a full set of agreements and relevant standards as decided within the working groups of the co-creation process. This document therefore is a growing document to which additions and changes are constantly made. The following working groups bear the responsibility to add detail to this document:

- Legal
- Operational
- Functional
- Technical

All named working groups focus on their respective topic within the iSHARE scheme.

Version 0.1 of this document is meant as a discussion starter for the working groups. This startdocument provides a first draft of topics that should be addressed and detailed during the co-creation phase of iSHARE. Next to a preliminary table of contents, some topics contain descriptions of possible solutions to be considered. The startdocument aims to be complete in scope, but is by definition not complete in detail.

This document is present in the online environment of "Confluence", which allows for collective editing/commenting. All participants are encouraged to comment on the topics addressed within this environment so that all relevant arguments are considered.

Please note: Any statements in this document are in no way intended to favour a certain solution; all statements (even firmly stated texts) should be seen as open to discussion.

Notational conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 (http://www.ietf.org/rfc/rfc2119.txt).

Other conventions might be added later.

Glossary

Authentication

Authorization

Authorization Registry

Certificate Authority

Co-creation

Data Consumer

Data Provider

Identification

Levels of Assurance

Public Key Infrastructure (PKI)

PKI Root

PKI Trusted List

Scheme

Trust Framework

Use case

Authentication

Authentication is the process of validating the identity attributes presented by the user during the identification process. The goal of the authentication is to check the authenticity of the presented identity attributes before the user is allowed to enter the third and last step in the data request of the iSHARE transaction which is authorization. Authentication can be achieved by asking the user to enter credentials ("something you know") to prove that they are the legitimate holder of the presented identity.

The picture below is the official iSHARE logo for authentication (used in presentations and external communication) and illustrates the user that gets a checkmark after presenting his/her identity attributes during the identification process. The checkmark symbolises the successful outcome of the validation of the identity attributes.

Authorization

Authorization is the process of validating authorisation attributes and policies before giving users access to an environment for data, services and other functionalities. The owner of the environment (Data Provider) can decide to perform the authorization management and validation process in ternally or to rely on an Authorization Registry for that. The Data Provider decides the granularity of the authorisation and which authorisation attributes have to be presented by the user to pass the third and last step of the iSHARE transaction which is the authorisation before getting access to the data.

The picture below is the official iSHARE logo for authorisation (used in presentations and external communication) and illustrates the user that is authorised to access an environment (for data/services/functionalities). The opened lock symbolises the authorised access of a user to the environment which is the last required step in the data request of the iSHARE transaction.

Identification

Identification is the process of claiming one's identity ("prove that you somebody") at an authority with the goal to enter the authority's environment by presenting identity attributes defined and accepted by the authority. In the case of iSHARE, it is proposed to reuse existing identity solutions from identity providers in the Dutch market such as eHerkenning and iDIN, and once expanding to other countries, international identity solutions. Identification is achieved by asking the user to present their identity attributes ("something they are") such that they can be validated within the second step in the data request of the iSHARE transaction which is authentication.

The picture below is the official iSHARE logo for identification (used in presentations and external communication) and illustrates the user that presents his/her identity card. The identity card symbolises the identity attributes that the user presents at the identity provider to log in to the environment for further steps (access requests for data).

Levels of Assurance

The table below describes the three levels of assurance according to the eIDAS regulation. The first column states the level of assurance, the second column briefly explains the degree of confidence one can have in the assurance level and the third column states the associated risk with the assurance level.

Under the table, the link to the levels of assurance in eHerkenning are added.

Level of Assurance	Confidence degree in identity	Risk degree of identity
1 - Low assurance	Limited confidence in the identity of the signer	Reference to technical specifications, standards and procedures, including technical controls, the purpose of which is to **decrease the risk of misuse or alteration of the identity**

2 - Substantial assurance	Limited degree of confidence in the claimed identity of the signer	Reference to technical specifications, standards and procedures, including technical controls, the purpose of which is to **decrease substantially the risk of misuse or alteration of the identity**
3 - High assurance	High degree of confidence in the claimed identity of the signer	Reference to technical specifications, standards and procedures, including technical controls, the purpose of which is to **prevent misuse or alteration of the identity**

eHerkenning levels of assurance

As the Dutch identity solution eHerkenning is often referred to in the course of the iSHARE working groups, the link to the eHerkenning assurance levels is added on this page.

PKI root

A PKI root is another term for root certificate, and stands for an unsigned or self-signed public key certificate that identifies the Certificate Authority , the party who is trusted by all members in the trust framework. The most common type of PKI certificates are based on the X.509 standard and normally include the digital signature of the Certificate Authority. The certificate authority issues digital certificates to all members in the trust framework.

Note that message encryption/decryption is not the same as message signing/validation. In message encryption/decryption, messages are encrypted with a public key and decrypted with a private key, meaning that the message is kept secret and can be read only by the single holder of the private key. For digital signatures, it is reverse. Messages are signed with a private key and validated by a public key. With a digital signature, you are trying to prove that the document signed by you and that the message is sent out by you. You want to prove the authenticity of the message by signing it with your private key to proof to the whole world (publicly) that the message originated from you.

Digital signatures are verified using a 'chain of trust'. All certificates in the chain of trust are signed with the Certificate Authority's private key. So when members in the trust framework exchange messages including their digital certificates, the authenticity of the messages can be validated by all other members in the trust framework as they hold the public key which they can use to validate digital signatures and therefore be sure of the authenticity of the sender (to be a member of the trust framework).

Public Key Infrastructure (PKI)

A Public Key Infrastructure (PKI) is an infrastructure that consists of an architecture, organisation & technology and roles, policies & procedures to manage digital certificates and public-key encryption. The purpose of a PKI is to ensure secure digital communication and the trustful digital exchange of data to enable electronic (online, digital) services.

Digital certificates are issued and revoked by a Certificate Authority which is a role within a public key infrastructure (PKI).

Scheme

A scheme can have different meanings but what we mean here is a collaborative effort of organisations to achieve a common goal which can be different for every scheme. In page Goals and scope of the iSHARE scheme the goals and scope of the iSHARE scheme are described.

Another example is a card scheme. Examples for card schemes are Visa, MasterCard, American Express etc. which are all payment networks linked to payment cards with different payment products (credit, debit, pre-paid). Banks and other financial institutions can become a member of a card scheme with the goal to receive licenses to issue payment products and process payment transactions of the payment networks.

Trust framework

A trust framework consists of a group of participants who all work with combined efforts towards the same goal, namely building a system that works and that all participants trust. And by trust we mean: willing to participate in and rely on. To achieve both goals, all possible risks have to be addressed by technical, functional and operational specifications and legal rules.

Technical, functional and operational specifications are needed to ensure

- The system's processes, policies, procedures, performance rules and requirements, assessment criteria, etc.
- Make it work
- Make it trustworthy

Legal Rules are needed to ensure

- Existing law
- Contractual obligations
- Regulate technical, functional and operational specifications

- Make technical, functional and operational specifications legally binding on the participants
- Define and govern the legal rights and responsibilities of the participants

Versioning

Unique version numbers will be assigned to unique states of the iSHARE scheme. The v0.1 version of the scheme is called the "Startdocument" - one of the deliverables of iSHARE Phase 1.

The Startdocument is co-developed into new versions in the four working groups, with a v0.8 version ready at the start of Q2 2017 and a v1.0 version at the end of Q2.

A new version of the iSHARE scheme is shared with the Steering Committee whenever it meets - a total of six times in H1 2017.

Legal notices

Note: in the course of 2017 it will be decided under what terms these pages will be governed and a final position on intellectual property rights will be established.

Other legal notices might be added later.

Key features, guiding principles & assumptions

This section provides a high level overview of the features and requirements that the iSHARE scheme aims to support:

- Key features
- Guiding principles
- Assumptions

Key features

Based on the inventory of use cases taken during Phase 1, the iSHARE scheme should at least support the following:

- Provide trust framework for PKI certificates
- Provide flexibility in authorization
- Allow for management of consent
- Support multiple interaction models

Please note: in line with the iSHARE guiding principles, these key features might be realised by (re)using existing standards or initiatives.

Provide trust framework for PKI certificates

The iSHARE scheme requires public key encryption for the following purposes:

- Proof of origin of data
- Authenticity of identities
- Protection of data against unauthorized access

A PKI is required, in order to:

- Publish public keys (through digital certificates)
- Certify that public keys are tied to the right individuals or organizations
- Verify the validity of public keys

The iSHARE scheme should provide a PKI root list that contains trusted PKI roots that meet the iSHARE requirements. Trusted PKI roots within the iSHARE PKI root list can be (and should be) trusted by every iSHARE participant. The term "PKI root" is otherwise known as Certificate Authority (read more on Public Key Infrastructure (PKI)) .

It is assumed that existing PKIs are sufficient to meet all iSHARE requirements. If, during the course of phase 2, this assumption turns out to be false, an additional iSHARE specific PKI can be created (Read more on what iSHARE's own PKI might entail).

Provide flexibility in authorization

The iSHARE scheme envisions a world in which (access) authorizations are flexible in three ways:

- **Authorization scope**
  The authorization scope refers to the objects or resources (most of the times data) from a specific party, to which authorizations need to be assigned. The scope can include many or all resources (e.g. all data), or only some resources (e.g. specific data fields). Either way, the scope is always governed by a formal agreement and implemented by technical means.

- **Authorization granularity**
  The authorization granularity refers to the characteristics of both the data and the rules (policies, conditions0 that apply. Authorizations to data can be coarse-grained (e.g. someone has access to all data in a certain data scope) or fine-grained (e.g. someone has access to

only data with a low sensitivity level). The rules (policies, conditions) that control the authorizations can be fine-grained as well, meaning that many different types of rules can apply, such as time of day, location, organisation, role, and competence level.

- **Authorization source**
  The authorization source refers to the location of the rules (policies, conditions) and the attributes (e.g. subject attributes, object attributes) that govern the authorizations. These can be located near the data, at a dedicated source, or a combination thereof.

The final architecture will be dependent on requirements such as data ownership, formal agreements, communication and security.

Allow for management of consent

For appropriate recognition of authorizations a mechanism to manage consent is required. This mechanism should support both rule based consent (e.g. based on information already residing in a company's ERP system) or case by case consent given by a natural person (e.g. through some sort of digital signature on a mobile device).

Any form of consent should be subject to a management procedure allowing Data Owners to modify or withdraw certain rights.

Support multiple interaction models

To cater for different user scenarios, the iSHARE scheme supports several interaction models:

- Both machine to machine (M2M) and human to machine (H2M) interfaces should be supported. Possible human to human (H2H) interfaces like Peer2Peer might be included as well.
  - **Machine-to-machine (M2M)** communication is used for data transmission between electronic devices using both wireless communication such as Wi-Fi, RFID, and wired communication such as FTTx.
  - **Human-to-machine (H2M)** communication is used for data transmission between a human (user) and a device and vice versa. A prerequisite is an interface that allows the input of the user to be translated into signals that the device understands, and allows the device to provide the required result to the human.
- Both request-response and publish-subscribe models are supported.

Guiding principles

To achieve the goals of the iSHARE scheme, it is paramount to stay close to certain guiding principles. The following principles must be kept in mind at all times during the development of the iSHARE scheme:

1. **Generic building block for entire sector**
   a. The iSHARE scheme should be applicable anywhere in the logistics sector;
   b. The iSHARE scheme should be extensible to cater for more situation/sector specific use cases that go beyond the generic nature of the scheme.
2. **Scope limited to identification, authentication and authorization**
   The iSHARE scheme should only apply to the topics of identification, authentication and authorization in the context of finding and exchanging data. An important part of the discussion in the Functional working group should focus on the legal framework - i.e. on what can(not) be done with provided data.
3. **Reuse existing buildings blocks where possible**
   a. The iSHARE scheme should build on existing initiatives that fit with the goals of iSHARE where possible;
   b. The iSHARE scheme should only draft a new solution when no existing initiatives or standards are sufficient to serve the goals of the iSHARE scheme.
4. **Ready for international use**
   The iSHARE scheme should, as far as possible, build on international open standards and best practices to cater for application in - and compatibility with - the international context.
5. **Agnostic towards nature and contents of data**
   The iSHARE scheme should be agnostic to the nature and contents of the data exchanged within iSHARE.
6. **Authorization principles**
   Authorizations should always be governed by to the following principles:
   a. Need to know: a subject should only have access to the information that is needed and nothing more
   b. Least privilege: only the minimum level of authorizations should be assigned to a subject
7. **iSHARE environment is based on existing standards**
   The iSHARE environment will be build upon existing standards and specifications listed in the section about relevant technical standards.

Assumptions

The iSHARE scheme starts from the following assumptions. If these assumption turn out to be false this has to be addressed. This is not necessarily a task of the iSHARE project.

1. **Data ownership**
   The Data Provider is assumed either to be the Data Owner or have knowledge on who the Data Owner is. Only in this way the Data Provider can decide with whom to share data based on the explicit consent of the Data Owner. Note that the iSHARE scheme will not try to resolve any debates on data ownership.
2. **Data formats and semantics**
   In order to be able to share data a mutual understanding of the meaning of data and the way data is structured is required. It is assumed this mutual understanding exists and data sharing can therefore commence and be meaningful.
3. **Data requests and responses**
   Related to formats, but separated here because of the relation with iSHARE. In order to share data an interface should be defined. In this

interface both a request for data and the resulting response should be defined. Since iSHARE is data agnostic it is assumed that these definitions either exist or are created during the course of implementing specific data sharing cases.

4. **Data classification**
   The classification of data in categories is an important pre-requisite for the authorization. Data can be classified in categories defining their type, location, sensitivity and protection level. Authorization depends on the access rights of the data consumer that are checked as part of the data requesting process. Clustering the data in categories does not only simplify the authorization process, it also provides a clear overview to the Data Provider over their data and lowers the risk of sharing sensitive data with unauthorized Data Consumers.
   A risk analysis is part of the data classification process.

Roles & Responsibilities

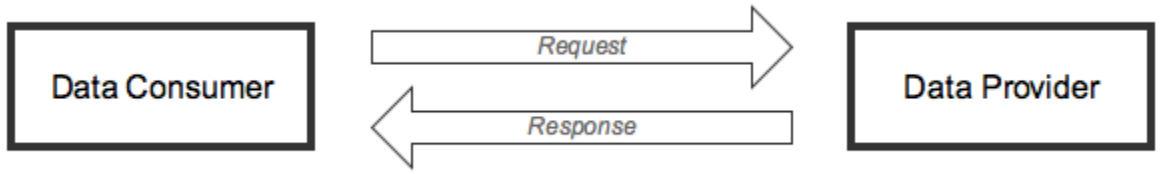
This section describes the roles that are described as part of the iSHARE scheme and their general responsibilities. A more detailed explanation of their functional behaviour and interaction between roles is described in Functional.

As mentioned before, the market for the iSHARE network will focus on the entire national logistics sector. Since logistics is crossing borders, all communication will be in English and therefore it will be taken into account that the iSHARE network can be applicable in other countries.

First the two sided market model of iSHARE will be discussed, with the Data Provider and the Data Consumer. Besides these primary roles the supporting roles will be further explained.

Two-sided 'market' of data sharing

The 'market' for data sharing is two-sided. In this two-sided market there are two main roles that have their own distinct needs and behaviours. In every iSHARE exchange within the network there will *at least* be a Data Consumer and a Data Provider. These roles can be retaken each time by users of the iSHARE network - a Data Consumer in one exchange can be a Data Provider in the next. The basic relation between the two roles is as follows:



The iSHARE scheme, and it's roles, will be applicable for the entire logistics sector. It will be set up for European wide use, although the initial emphasis will lie in the Netherlands. Logistical parties who want to share data with (other) logistical parties will be able use the iSHARE scheme to facilitate their needs.

Data Consumer

The **Data Consumer** is the party that requires data from the Data Provider.

Most likely the initiator of an exchange in the iSHARE context. This is the party that is in need of the information the Data Provider

Data Provider

The **Data Provider**:

- Is the Data Owner, or;
- Has knowledge of who the Data Owner is and therefore is the source of the required data for the Data Consumer. Can decide with whom to exchange data based on the explicit consent of the Data Owner.

Supporting roles

Next to the main roles of Data Consumer and Data Provider present in every exchange made within the iSHARE context, the iSHARE scheme describes the following supporting roles that are optionally present:

Certificate Authority

Description

A **Certificate Authority (CA)** is:

- An entity that issues digital certificates;
- A trusted party, and;
- Responsible for the binding to a specific entity of the certificate (registration & issuance).

A digital certificate certifies the ownership of a public key by the named subject of the certificate, so other parties can rely upon signatures or assertion made about the private key that corresponds to the certified public key.

A **Registration Authority** (aka sub CA) verifies the identity of entities requesting their digital certificates to be stored at the CA. Makes sure of the validity and correctness of the registration. Accepts digital certificates & authenticating.

A **Validation Authority** provides entity information on behalf of the CA.

Relevance

The CA can act as a gate keeper for exchanges in an iSHARE context. The Data Consumer needs a certificate to identify itself to the Data Provider. The Data Provider can check (through the Authorization Registry) if this certificate is revoked or not. When the certificate is revoked, the exchange will not be completed.

Note: it is only possible to check revocation status of a certificate when a Data Provider initiates an exchange. In the situation where the revocation status of a certificate will not be checked with every exchange, e.g. due to the high number of exchanges, there is a possibility that a certificate has been revoked while exchanges were still on going.

Authorization Registry

Description

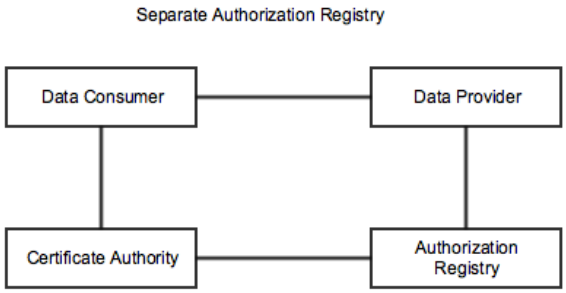
An **Authorization Registry**:

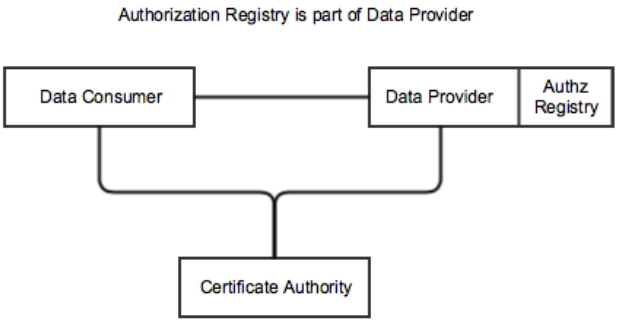
- Manages records of authorizations of users/entities within the scheme;
- Checks on the basis of the registered permission, or a representative of a company, if an entity is entitled to take delivery of the requested data, and;
- Confirms the established powers towards the Data Provider.

The Authorization Registry can also be referred to as Mandate Provider. It can be either a separate entity within the iSHARE scheme or a part of the Data Provider, depending on the size and preferred choice in architecture - as depicted below.

In the situation of high speed/high volume exchanges it is possible to establish a 'session'. In this case the Authorization Registry checks the Authorization and sets a time or amount of exchanges-limit for the next check. Therefore the processes can run smoothly without interruption by entities who are familiar to each other.

Depiction





Relevance

The Authorization Registry works together with the Certificate Authority to make sure whether the Data Consumer is authorized to fulfill the requested exchange. Without an authorization from the Authorization Registry an exchange will not take place.

Identity Provider

Description

An **Identity Provider (IdP)**:

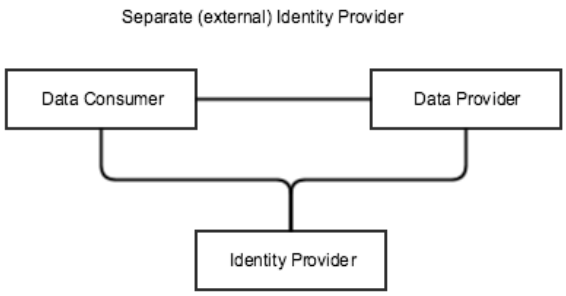
- Provides identifiers for human users looking to interact with a system;
- Asserts to such a system that such an identifier presented by a user is known to the provider, and;
- Possibly provides other information about the user that is known to the provider.

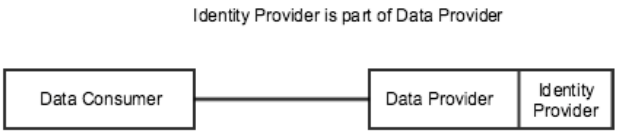
An Identity Provider can be either a separate entity within the iSHARE scheme or a part of the Data Provider, depending on the size and preferred choice in architecture - as depicted below.

In the iSHARE scheme, several states and proces flows can be identified. The Identity Provider will play important roles when a (new) entities will be on boarded into the iSHARE scheme and when exchanges of data will take place. From an transaction point of view the Identity Provider will be called upon every time when an entity needs to be verified before an exchange is initiated.

Note: It is possible that the Identity Provider will not play a role in every exchange within the iSHARE scheme (e.g. for scaling reasons in low-risk situations).

Depiction





Relevance

Exchange

The Identity Provider plays a role in the iSHARE scheme when human persons have to interact with the network, e.g. in the situation where the size/weight of a truckload needs to be edited at the source of the Data Provider. When the identity of the user is not valid or known, the exchange will not be completed.

Registered identities

All human users of the iSHARE scheme need to have a registered identity to be able to interact with the system.

Broker

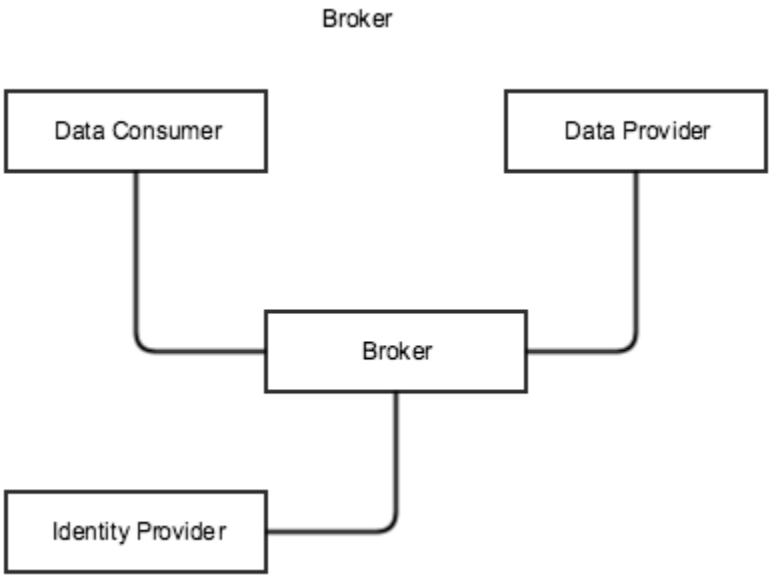
Description

A **broker**:

- Can provide (depending on the final technical details of the role) the technical or logical relation between users of the iSHARE scheme, e.g. between the Data Consumer and the Data Provider
- Will therefore be linked to several different parties

If the Data Consumer and the Data Provider do already have a direct relationship with each other, they *can* choose to make a data exchange without the use of a broker.

Depiction



Relevance

The iSHARE scheme will consist of various organisations that will take roles as Data Consumer and Data Provider. To keep the scheme clear and effective, there will not be a direct connection from every DP to every DC and all the other necessary roles. The broker will be the solution to link these parties to each other. Besides that, in the spirit of freedom of choice, the iSHARE scheme should support that several parties can offer similar services. Also, the broker makes it possible to broker exchanges with data providers who are yet unknown by a data consumer, but can be reached through iSHARE standards.

Signing service

...

Processes

The iSHARE scheme supports the following processes:

- Encryption
- Hashing
- Signing

Encryption

...

Hashing

...

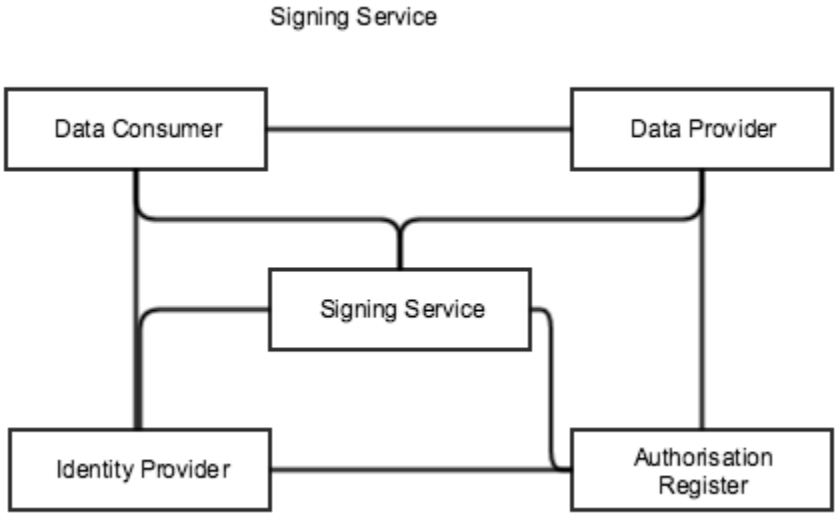
Signing

Description

Signing is the process of signing a message (or document, or transaction) which enables a receiver to confirm authenticity of the message.

As part of the service offered by Service Providers, Service Providers are required to sign messages authorized by the Data Provider.

Depiction



Relevance

The Signing Service provides the security that the transaction is valid from the Data Consumer perspective and that the transaction is authorized by the Data Provider.

Legal

This section covers the relevant Legal topics of the iSHARE scheme:

- Relevant Rules & Regulations
- Possible operational business models
- Required contracts: what are the contracts that bind the different roles to the iSHARE scheme?
- Branding & licensing

These topics (and possibly others that arise during Phase 2) are detailed by the Legal working group.

Relevant Rules & Regulations

About relevant rules an regulation the following can be stated: the solution should comly:

- The (legal and professional) standards for information security
- Dutch legislation and regulations insofar as they are applicable
- European regulations insofar as they are applicable, notably eIDAS regulation and GDPR

A detailed inventory of the relevant laws and regulations must still places available.

The scheme should pay attention to the current legal framework, in order to verify whether the scheme can lead to a full and proper arrangement of iSHARE. The scheme must guarantee that iSHARE can be used for the processes of private and public organisations, without this leading to legal or security problems. Within the scheme there should be a focus on these two situation in which:

- Unauthorized access get wrongly authorized
- Authorized users wrongly don't het access

Legal Framework

The iSHARE network should be arranged according to the privacy rules by design principles. This means that personal data may not be processed more often than is necessary for the purpose for which the personal data is obtained. This is in accordance with the Data Protection Act.

- There must be compliance with legal and professional standards for information security.
- There must be according Dutch laws and regulations as applicable.
- There must be complied with European legislation where applicable.

Electronic Access Services are focused on providing "trust". Clear legal frameworks contribute to this as well as a well-organized control system based on clear roles and responsibilities detailed in "Structure & Roles". Moreover, to provide legal requirements regarding reliability of iSHARE services, regarding identification, authentication and authorization of importance for the understanding and the development of the iSHARE Network.

The eIDAS regulation

The eIDAS regulation & trusted list of service providers and services

The eIDAS regulation obliges EU Member States to establish, maintain and publish trusted lists about qualified trust service providers (including trusted certificate authorities) and qualified trust services provided by the trust service providers.

Note that we include the eIDAS regulation of trusted list of service providers and their services because we might want iSHARE to support international PKI roots.

The trust service providers have to cover the following list of trust services:

- Time stamping: The date and time on an electronic document which proves that the document existed at a point-in-time and that it has not changed since then
- Electronic seal: The electronic equivalent of a seal or stamp which is applied on a document to guarantee its origin and integrity
- Electronic delivery: A service that is provided in the digital world through the internet or by means of other information and communication technologies (i.e. opening a bank account, transferring money etc. which used to be provided by people in the physical world)
- Legal admissibility of electronic documents to ensure their authenticity and integrity
- Website authentication: Trusted information on a website (e.g. a certificate) which allows users to verify the authenticity of the website and its link to the entity/person owning the website

Here is the link to the website: https://ec.europa.eu/digital-single-market/en/trust-services-and-eid

Possible operational business models

This section will describe the possible operational business models of the iSHARE scheme - it will be detailed by the Legal working group.

Required contracts

This section will describe the required contracts of the iSHARE scheme - it will be detailed by the Legal working group.

Branding & licencing

Note: both general terms and conditions and dispute management will be detailed by the Legal working group later.

Entrance criteria

All participants in the iSHARE network must meet the general entry requirements. The entry requirements are set for a number of reasons. The main one is the knowledge that the network will only be able to function as recipients of services have sufficient confidence in the scheme and its effects. Confidence in the scheme and the Recognition Services provided in connection with the scheme requires trust in the individual participants. The scheme must provide clear content requirements on which participants may join (and exit).

1. Entry Requirements - All participants in the scheme must meet the general entry requirements. The entry requirements are set for a number of reasons. The main one is the knowledge that the network will only be able to function as recipients of services have sufficient confidence in the scheme. Confidence in the scheme and the use of the network, supplied in connection with the scheme requires trust in the individual participants.
2. Surveillance, monitoring and control - Proper adherence to the agreed system is essential for confidence in the network (for Electronic Access Services). Monitoring and maintaining compliance is the responsibility of the regulator. The management organization responsible for the control and monitoring of compliance with the scheme agreements on behalf of the owner and for the regulator. Enforcing compliance with system arrangements is the task of the regulator.

Liability

Within the scheme, each participant is responsible for his own actions and / or omissions in the role he plays. The liability is subject to the general rules of Dutch/EU law regarding the content and extent of liability to pay damages. Participants may and may not derogate from these general

rules. How these rules work out in a particular case depends on the facts and circumstances of the case.
The participant may limit his liability in the contract which he concludes with a Data Consumer or a Data Provider. In addition, he remains subject to the general rules of the Dutch/EU law on liability and compensation.

Although the Certificate Authority (CA) may not be a member of the iSHARE network, the CA plays a vital role within the network of the scheme in the CA domain. For the CA registry is that it is liable for its own acts and / or omissions in the role it plays. The service intermediary is liable for its own acts and / or omissions.

Level of assurance

Determining the level of assurance (LoA) for a particular service/request is determined by the Data Provider. The Data Provider must ensure compliance with the AWB (General Administrative Law Act) to give substance to the standard of a reliable and confidential communications.
The Data Provider will therefore continue with the provision of a service/data and should carry out a risk assessment and must consider what measures should be taken to allow electronic communication sufficiently reliable and confidential. This includes an option for the required level of assurance for a particular service or data that will be used. In addition to determining the LoA chosen by the Data Provider to the service provider will have to take other measures to reliably electronically to provide a service in accordance with the requirements of the AWB. The additional measures to be taken are dependent on the confidence level.
Where Electronic Access Services is used for e-services outside the government (B2B and B2C) specific AWB naturally do not apply requirements. In the case of B2B and B2C services is that means publishers permission registers and the respective service providers an "information society service" and / or a 'remote service' (as defined in the Civil Code) offer. These parties are responsible to meet the associated information obligations and duties regarding the establishment of a legal agreement, as contained in the Civil Code.

Operational

This section covers the relevant Operational topics of the iSHARE scheme:

- Service Level Agreements
- Audits
- Incident Management
- Change Management
- Governing Body

These topics (and possibly others that arise during Phase 2) are detailed by the Operational working group.

Service Level Agreements

This document describes the service level agreements that apply to participants of the iSHARE network. It is a description of the minimum service level which should provide the participants with each other and their customers service and minimum service level management that the governing body provides to its participants/users. A service level agreement (SLA) is a contract between a service provider (either internal or external) and the end user that defines the level of service expected from the service provider. SLAs are output-based in that their purpose is specifically to define what the customer will receive.

- Up-time
- Response time
- Maintenance reports
- Monitoring
- Logging
- Archiving
- Reporting

Up time / Response time / Maintenance reports/windows

This section will describe the performance of the service that will be provided as agreed in the SLA.

Monitoring / Logging / Archiving / Reporting

This section will describe the behaviour of the background of the service that will be provided as agreed in the SLA.

Customer Support / Helpdesk

This section will describe the how problems reported by users will be handeled as agreed in the SLA.

Up-time

Up-time is a measure of the time a machine, in this case the network and it's servers, has been working and available. Uptime is the opposite of

downtime.

The times which are issued by participants and the management organisation guaranteed the availability the iSHARE network.

Response time

Response time is the time it takes for a device, network or service, when subjected to a change in input signal, to change its state by a specified fraction of its total response to that change. In the iSHARE environment the response time will be for the user the time it takes to proces a request and return a signal.

Purpose of setting performance standards is to ensure a good user experience, especially at peak times.

The norm for processing of messages for participants

1. 95% of messages MUST be returned within 2 seconds
2. 99% of the messages MUST be returned within 5 seconds
3. Each participant MUST be able to process at least 100 simultaneous messages while still meet the performance requirements

Maintenance reports

Maintenance reports are intended to monitor the growth of the network and it's service level agreements within the network. To be able to keep track on the growth number, guarantee it's uptime and service and be able to take action if it exceeds it's possible usage.

The participants and the management organisation collect personal information management reporting period (which runs from the first day of a calendar month 0:00 pm till the last day 24:00).

Each participant must reach the 5th of each month, provide reporting on the previous reporting period, the management organisation for 24:00. To this end, the participant must use the reporting tool made available by the management organisation. The management organisation will aggregate information sharing within 5 working days with all the participants and service providers.

Monitoring

The **monitoring** of the agreements made in the service level will be performed by the management organisation. The management organisation will use the analysis of the reports delivered by the participants as input for the monitoring. Other input will also be used like sample testing.

Logging

Logging is the proces that records events that occur in the iSHARE network, and/or messages and communication between different users of the iSHARE network.

Archiving

Archiving is the process of moving iSHARE data that is no longer actively used to a separate storage device for long-term retention.

Reporting

Data **reporting** is the process of collecting and submitting data to authorities entrusted with compiling statistics. Accurate data reporting gives rise to accurate analyses of the facts on the ground; inaccurate data reporting can lead to vastly uninformed decisions based on erroneous evidence. When data is not reported, the problem is known as underreporting; the opposite problem leads to false positives.

Audits

An **audit** is a systematic and independent examination of records that inform about performed actions by a system to check if the system safeguards the assets, maintains data integrity and operates effectively to achieve the predefined goals. Audits offer a great opportunity to periodically check the effectivity of implemented functionalities and is therefore recommended to put into place.

In the context of incident management, audits should be performed as security measure on executed data transactions to spot fraudulent and unauthorised actions and the instances who are accountable for that.

The scope and process of audits will be determined in the course of the iSHARE functional workshop.

Incident Management

The goal of the process **Incident Management** is to settle different types of incidents within the iSHARE network - in a structured way. Disruption of the service(s) should be (as) limited (as possible).

An **incident** is every event that is not part of iSHARE's standard operation and that has (potential) impact or risk with respect to the quality, availability, integrity and/or confidentiality of (information within) the iSHARE network. Incidents could include:

- Disruptions: events that lead to (parts of) the iSHARE service(s) being partially or entirely unavailable;
- Information security incidents: events such as the loss of a USB stick, laptop, harddrive but also signals of attempts of hacking, attempts to enter the iSHARE network or malware;
- Fraud or the presumption of fraud by, for example, an employee or a hacker.

Who is responsible for Incident Management, and how the Incident Management process is setup will be established in the Operational working group.

Change management

The process **Change Management** structures changes in:

- Scheme documentation
- Scheme implementations

It will be detailed by the Operational working group.

Governing body

The iSHARE scheme is an initiative with a long-term ambition to improve data sharing circumstances for the logistics sector. To operationalise this long-term ambition, iSHARE needs to become a sustained endeavour which is constantly improved by its stakeholders. To organise the constant improvement, a **governing body** needs to be shaped. Which form this governing body needs to take to optimally support the long-term ambitions needs to be discussed and decided upon within the iSHARE project together with involved stakeholders.

The governing body could take any shape, of which the most evident options would be:

- Establish a new governing organisation, either in the form of an association or a company depending on what is deemed most appropriate for the scheme
- Bestow governing responsibilities upon an existing association or company. This option is plausible when the existing organisation's capabilities and mandate are aligned with iSHARE goals and when the organisation enjoys the support of a significant majority of iSHARE stakeholders.

The responsibilities of the governing body will exist out of some or all of the following activities (non-exhaustive):

- Organise regular processes to constantly improve iSHARE scheme specifications with stakeholders;
- Develop, maintain and improve relevant core documents and standards for the iSHARE scheme;
- Define, maintain and execute certification procedures for organisations that want to participate or need to adhere to the iSHARE scheme rules;
- Develop, maintain and improve software or testing environments that facilitate the iSHARE scheme (e.g. testing suite, certification tools, software libraries, directory services or incident notification portals);
- Report on scheme performance where possible and where necessary to relevant stakeholders;
- Facilitate dispute management procedures;
- Facilitate incident management procedures;

Depending on the results of the co-creation phase and the direction of the iSHARE scheme at the end of the co-creation phase, the form of the future governing body for the iSHARE scheme can be determined.

Functional

This section covers the relevant Functional topics of the iSHARE scheme:

- Primary use cases
- Secondary use cases
- Detailing key features
- Functional requirements per role
- User interface requirements
- Identifiers

These topics (and possibly others that arise during Phase 2) are detailed by the Functional working group.

Primary use cases

In this section we elaborate on the key features that iSHARE should at least support (see key features) into more detail. We do this by drawing up the sequence diagrams belonging to the use cases gathered during interviews with stakeholders in Phase 1.

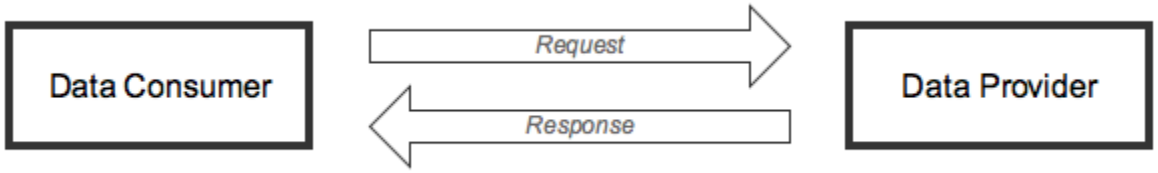
Walkthrough of the use cases

The use cases will be first explained through a description of the events. This will be follow by the roles which will be involved in the specific use case, these will also be shown in a diagram. In the following part the events will be shown in a sequence diagram, to give more detail on the interaction of the users and which events will follow each other up. To give a good detailed insight every step in the sequence diagram of the use case, an explanation of every step will be described. It starts with the initial state of the specific use case and follows every step, and possible exception of the step, until the final state is reached. These three elements (description, roles and sequence diagram) will give a good detailed insight of the specific use case.

The basic use case of an iSHARE transaction

The basis involves 2 roles, the Data Consumer and the Data Provider. The basic use case starts with a request, for a transaction, from the Data Consumer with it's certificate, to the Data provider. The Data Provider checks the validity of the certificate and the authorization of the Data Consumer to get access to the requested data. When the certificate is valid and the Data Consumer is authorised for the requested information,

the transaction can take place, signed by the Data Provider, as identity proof for the Data Consumer.



The primary use cases revolve around the use of iSHARE and the aspect of sharing data where various constellations of roles and constructions are taken into consideration.

- Interaction models
- 1. Share data bilaterally (M2M)
- 2. Share data involving an Authorization Registry (M2M)
- 3. Share data based on delegation (M2M)
- 4. Share data based on delegation and involving an Authorization Registry (M2M)
- 5. Share data (H2M)
- 6. Share data involving a Broker (H2M)
- 7. Share data involving a Broker and an Authorization Registry (H2M)

Note:

- M2M stands for Machine-to-Machine
- H2M stands for Human-to-Machine
- DP stands for Data Provider
- DC stands for Data Consumer
- CA stands for Certificate Authority
- AR stands for Authorization registry

Interaction models

Hereunder will the two types of interaction models be explained: Machine to Machine and Human to Machine. In short Machine to Machine means and interaction between two machines, where Human to Machine an actual human being is communicating with the iSHARE network.

Machine to Machine

M2M stands for any technology that enables the automated exchange of information and the performance of actions between electronic devices without requiring the assistance of humans. In some M2M applications, the electronic devices exchange their information with a central control unit/application which processes the information for humans.

To exchange (send and receive) information (in the form of electronic signals), a communication network or channel is required such as i.e. a telecommunication network, the internet (Wifi, 3/4G), radio-frequency identification (RFID), Bluetooth, infrared radiation, sensors (measuring temperature or other signals), autonomic competing (functioning of computer systems without input from a human) etc.

Examples of M2M technologies for sending and receiving signals are radio waves and telemetry/telephone lines.

The newest M2M wireless technology: the internet giving rise to the "internet of things" where heating units, electric meters, watches and other devices become "smart" and "connecting" using a chip to measure & collect data and the internet to send them to the receiving unit

Within iSHARE we foresee the following M2M scenario's (non-exhaustive):

- If M2M technology is going to be used within iSHARE, Data Providers (DP's) would not need to authorise Data Consumers (DC's) actively whenever DC's do data requests. The authorization requests would be automatically handled by an Authorization Registry (AR) where the authorization policies are registered, validated and executed.
  If M2M technology is going to be used within iSHARE and certain criteria are met by the Data Consumer (DC) and Data Provider (DP), data requests could be sent out automatically by the DC and automatically approved by the DP.
  - In the RWS use case, if a ship (DP) is within a certain distance from the port (DC), the port automatically could receive the rights to access the data of the ship.
    - Criterium for the port (DC) to send out data request automatically: ship is within a certain distance from port location.
    - Ship (DP) could authorise the port automatically to share shipment-specific data with the port based on authorization policies that are registered, validated and executed by the authorization register.

Human to Machine

H2M stands for any technology where human and machine functions are integrated into one system.

Even though the term H2M can be used in a much broader sense (see "Note" hereunder) we mean the human-computer interaction where

humans and computers interact through a user interface and perform activities for each other. This includes software (i.e. what is visible to the human on the computer monitor) and hardware (i.e. the mouse, keyboard and other devices).

Thus, in the context of iSHARE we mean the human interaction with the iSHARE system where a person actively performs actions by i.e. giving consent or entering data.

Within iSHARE we foresee the following H2M scenario's (non-exhaustive)

- Data Providers could be asked to actively give consent for the authorization of Data consumers.
- Data Consumers could be able to actively send out requests to Data Providers to edit data in their information system in case the DC finds that the provided data by the DP needs modification (in the 'give rights to edit data' use case)

Note that the term H2M can be used in a broader context focusing on any tool, object, robot that can interact with humans and can integrate human functions.

- A subtopic of H2M is the integration of human body functions (motions of arms & legs) in a machine, for example a robot with six robotic legs that are controlled by the leg and hand movements of its pilot
- Human-machine-systems are often use in the media and science fiction stories & movies such as the Terminator and Robocop.

1. Share data bilaterally (M2M)

Description

This is a generic use case explaining the basis for the sharing of data between two parties.

In this use case data a data transaction takes place between the Data Provider and the Data Consumer on initiation of the Data Consumer. The Data Consumer initiates the data transaction with a request for specific data from the Data Provider. The Data Provider needs to check the Data Consumer's certificate issued by a designated Certificate Authority. After the validation of the certificate, the Data Provider can check at the internal Authorisation Registry whether the Data Consumer holds authorization rights for the requested information. When the certificate is valid and the Data Consumer is authorized for the requested data, the data will be sent to the Data Consumer, signed by the certificate of the Data Provider, as identity proof to the Data Consumer.

Practical examples

All Functional working group-members are invited to add practical examples of this use case in the comment section.

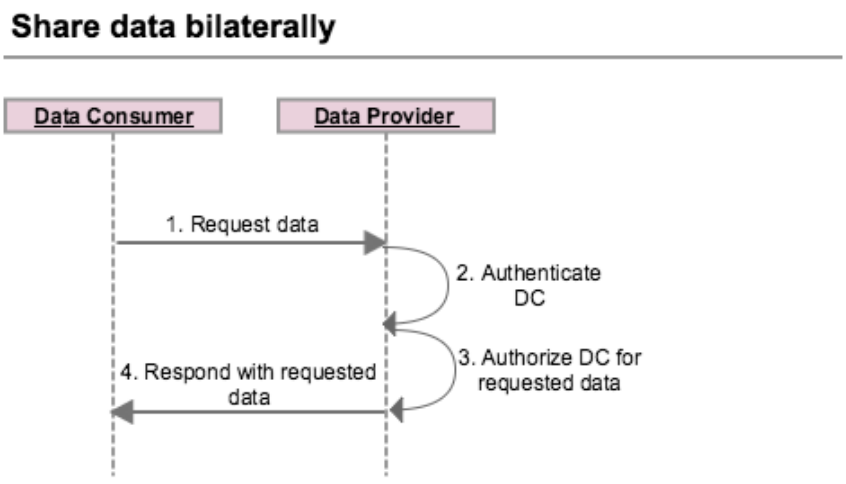
Roles

The roles which will be used in this use case are the following:

1. Data Consumer
2. Data Provider
3. Certificate Authority

Sequence Diagram

Hereunder you see the sequence diagram belonging to the use case "Share data bilaterally".



Footnote:

[1]: Message is signed with digital certificate issued by the Certificate Authority as a means of identity proof.

The different actions taking place in use case "Share data bilaterally" are described in more detail in the table hereunder.

Step no.	Action	Description
	Initial state	The Certificate Authority previously issued certificates to both the Data Provider and Data Consumer. The Data Provider and the Data Consumer made a bilateral agreement with regards to access rights and shared data.
1.	The Data Consumer sends a request for data to the Data Provider	The Data Consumer sends a data request to the Data Provider and signs it with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Consumer to the Data Provider.
2.	The Data Provider authenticates the Data Consumer	The Data Provider authenticates the Data Consumer based on the response of the Certificate Authority that the certificate from the Data Consumer is valid. The Data Provider has a trust relation ship with Certificate Authority and needs the check result to trust the genuineness of the certificate from the Data Consumer. If the check result is negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the authentication result.
3.	The Data Provider authorizes the Data Consumer to receive the requested data.	The Data Provider checks the authorization of the Data Consumer for the requested data. Optionally, the Data Provider can choose to rely on an external Authorization Registry for the validation of authorizations. In this case we assume that the Data Provider manages the registration and validation of the authorizations internally. The Data Provider authorizes the Data Consumer based on the positive result of the internal authorization check. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the successful authorization.
4.	The Data Provider sends a response including the requested data to the Data Consumer.	The Data Provider sends a response to the Data Consumer including the requested data. The requested data is encrypted within the response and can be only read by the Data Consumer. Even though the response is sent through the communication channel of iSHARE, the requested data is only accessible to the parties authorized by the Data Provider. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to the Data Consumer.
	Final state	The Data Consumer receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, the Data Consumer checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

2. Share data involving an Authorization Registry (M2M)

Description

This use case describes the data transaction involving an Authorization Registry. The difference with use case 1. Share data bilaterally (M2M) is that the authorization is now done by an external party instead of the Data Provider.

In this use case data a data transaction takes place between the Data Provider and the Data Consumer on initiation of the Data Consumer. After authenticating the Data Consumer (by checking the Data Consumer's certificate at the designated Certificate Authority), the Data Provider checks at the designated Authorization Registry if the Data Consumer has the authorization rights for the requested data. As soon as the Data Consumer is authorized for the requested data, the data will be sent to the Data Consumer, signed by the certificate of the Data Provider, as identity proof for the Data Consumer.

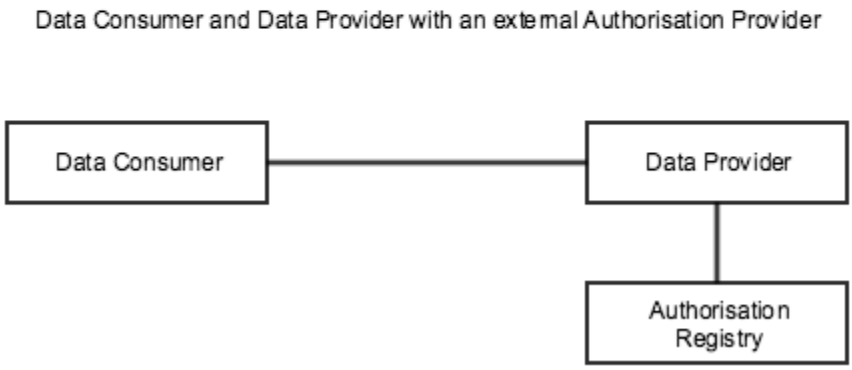
Practical examples

All Functional working group-members are invited to add practical examples of this use case in the comment section.

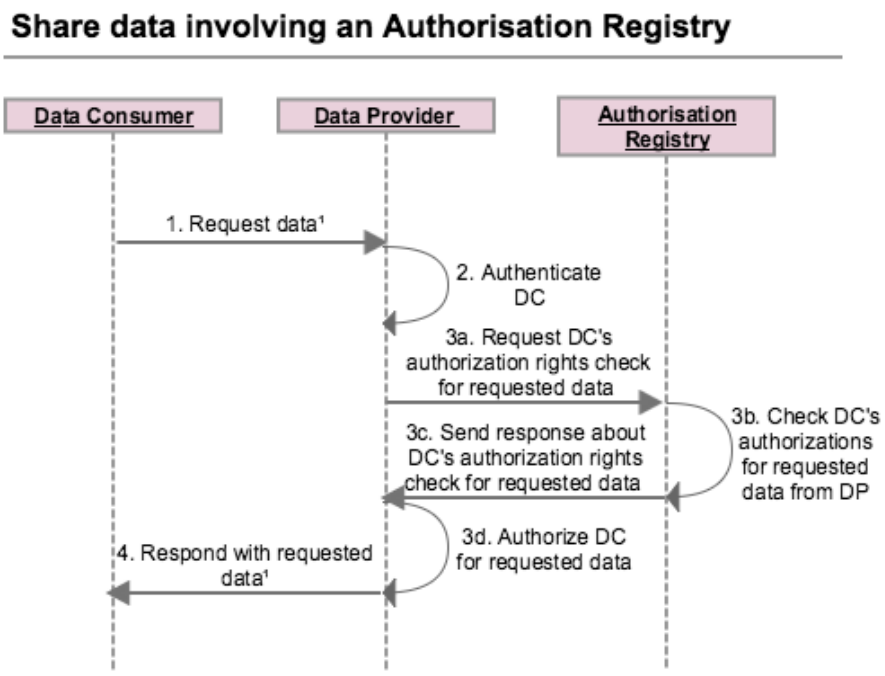
Roles

The roles which will be used in this use case are the following:

1. Data Consumer
2. Data Provider
3. Authorization Registry



Sequence Diagram



Footnote:

[1]: Message is signed with digital certificate issued by the Certificate authority as a means of identity proof.

The different actions taking place in use case "Share data involving an external Authorization Registry" are described in more detail in the table hereunder.

Step no.	Action	Description
	Initial state	The Certificate Authority previously issued certificates to both the Data Provider and Data Consumer. The Data Provider and the Data Consumer made a bilateral agreement with regards to access rights and shared data. The Data Provider relies on an external Authorization Registry for the validation of authorizations.
1.	The Data Consumer sends a request for data to the Data Provider	The Data Consumer sends a request for Data to the Data Provider and signs it with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Consumer to the Data Provider.
2.	The Data Provider authenticates the Data Consumer	The Data Provider authenticates the Data Consumer based on the response of the Certificate Authority that the certificate from the Data Consumer is valid. The Data Provider has a trust relation ship with Certificate Authority and needs the check result to trust the genuineness of the certificate from the Data Consumer. If the check result is negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the authentication result.
3a.	The Data Provider sends a request to the Authorization Registry to check the authorization of the Data Consumer for the requested data.	The Data Provider requests a check of the authorization of the Data Consumer for the requested data at the Authorization registry and signs it with the certificate issued by the Certificate Authority.

3b.	The Authorization Registry checks the authorization of the Data Consumer for the requested data from the Data Provider.	The Authorization Registry receives the request from the Data Provider to check the authorization of the Data Consumer for the requested data. The Authorization Registry first authenticates the Data Provider by checking the certificate (included in the authorization request) at the Certificate authority. The Authorization Registry then validates the authorization of the Data Consumer for the requested data from the Data Provider.
3c.	The Authorization Registry sends a check response about the authorization of the Data Consumer for the requested data to the Data Provider.	The Authorization Registry sends a response to the Data Provider including the check result of the authorization of the Data Consumer for the requested data.
3d.	The Data Provider authorizes the Data Consumer to receive the requested data.	The Data Provider authorizes the Data Consumer based on the positive result of the authorization check executed by the external Authorization Registry. If the check result was negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the authorization result.
4.	The Data Provider sends a response including the requested data to the Data Consumer.	The Data Provider sends a response including the requested data. The requested data is encrypted within the response and can be only read by the Data Consumer. Even though the response is sent through the communication channel of iSHARE, the requested data is not accessible for any other party than the Data Provider. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to the Data Consumer.
	Final state	The Data Consumer receives the response included with the requested data from the Data Provider. The response is signed with the certificate from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, the Data Consumer checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

3. Share data based on delegation (M2M)

Description

This is an extension from the basic use case of sharing data, now with the delegation of the rights from Data Consumer 1 to Data Consumer 2. In this specific use case, the rights for the data will be delegated from one party (Data Consumer 1) to another party (Data Consumer 2), so the other party (Data Consumer 2) will be able to receive the requested data from the Data Provider.

This use case is initiated by Data Consumer 1 who signs a delegation declaration with their own certificate and sends it to Data Consumer 2. The delegation declaration states that Data Consumer 1 delegates their own access rights for the data from the Data Provider to Data Consumer 2.

Data Consumer 2 follows up with a data request to the Data Provider, including their own certificate and the delegation declaration from Data Consumer 1. The Data Provider checks the certificate and delegation declaration for the requested data from Data Consumer 2. As soon as the certificate and delegation declaration are checked and valid, Data Consumer 2 is authorized for the requested data. The data will then be sent to Data Consumer 2, signed by the certificate of the Data Provider, as identity proof to Data Consumer 2.

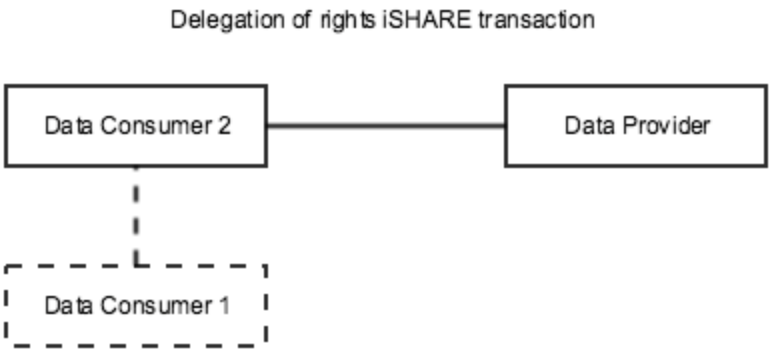
Practical examples

All Functional working group-members are invited to add practical examples of this use case in the comment section.

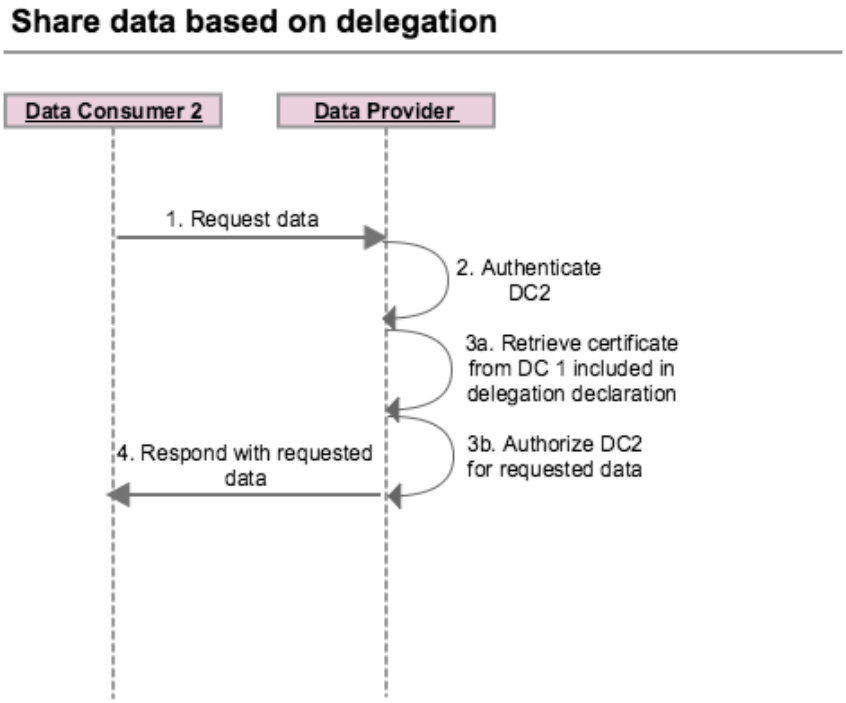
Roles

The roles which will be used in this use case are the following:

1. Data Consumer 1
2. Data Consumer 2
3. Data Provider



Sequence diagram



Step no.	Action	Description
	Initial state	The Certificate Authority previously issued certificates to the Data Provider, Data Consumer 1 and Data Consumer 2. The Data Provider and the Data Consumer 1 made a bilateral agreement with regards to access rights and shared data. Data Consumer 1 sent a message to Data Consumer 2 including a delegation declaration stating the transfer of access rights from Data Consumer 1 to Data Consumer 2. The message is signed with the certificate from Data Consumer 1 to guarantee the authenticity of the message to Data Consumer 2 and to proof the authenticity of the delegation declaration to the Data Provider.

1.	Data Consumer 2 sends a request for data to the Data Provider	The Data Consumer 2 sends a request for data to the Data Provider and signs it with the certificate issued by the Certificate Authority. The data request includes the delegation declaration signed with the certificate from Data Consumer 1. The digital signature included in the request is a means of identity proof for Data Consumer 2 to the Data Provider.
2.	The Data Provider authenticates Data Consumer 2	The Data Provider authenticates the Data Consumer based on the response of the Certificate Authority that the certificate from Data Consumer 2 is valid. The Data Provider has a trust relation ship with Certificate Authority and needs the check result to trust the genuineness of the certificate from Data Consumer 2. If the check result is negative, the Data Provider can decide to cancel the data request from Data Consumer 2. Optionally, the Data Provider can send a notification to Data Consumer 2 to inform about the authentication result.
3a.	The Data Provider retrieves the certificate from Data Consumer 1 included in the delegation declaration.	The Data Provider has no bilateral agreement with Data Consumer 2 and therefore the Data Provider cannot authorize Data Consumer 2 for the requested data. The Data Provider encounters the delegation declaration in the data request of Data Consumer 2. The delegation declaration includes a signature of Data Consumer 1. The Data Provider retrieves the certificate of Data Consumer 1 from the signature and requests a check from the Certificate Authority to prove the authenticity of the certificate.
3b.	The Data Provider authorizes Data Consumer 2 to receive the requested data.	Upon the positive check result of the Certificate authority for the certificate from Data Consumer 1, the Data Provider checks the authorization rights of Data Consumer 1 for the requested data. Based on the bilateral agreement between the Data Provider and Data Consumer 1 and the proven authenticity of the delegation declaration, the Data Provider authorizes Data Consumer 2 to receive the requested data. If the check result was negative, the Data Provider can decide to cancel the data request from Data Consumer 2. Optionally, the Data Provider can send a notification to the Data Consumer 2 to inform about the authorization result.
4.	The Data Provider sends a response including the requested data to Data Consumer 2.	The Data Provider sends a response to Data Consumer 2 including the requested data. The requested data is encrypted within the response and can be only read by Data Consumer 2. Even though the response is sent through the communication channel of iSHARE, the requested data is accessible only to the parties authorized by the Data Provider.. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to Data Consumer 2.
	Final state	Data Consumer 2 receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, Data Consumer 2 checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

4. Share data based on delegation and involving an Authorization Registry (M2M)

Description

This is another extension from the delegation of rights use case, this time involving an Authorization Registry that checks the authorization rights. The access rights for the data will be delegated from Data Consumer 1 to Data Consumer 2, such that Data Consumer 2 will be able to receive the data from the Data Provider after the authorization rights are checked by an Authorization Registry.

This use case is initiated by Data Consumer 1 who signs a delegation declaration with their own certificate and sends it to Data Consumer 2. The delegation declaration states that Data Consumer 1 delegates their own access rights for the data from the Data Provider to Data Consumer 2.

Data Consumer 2 follows up with a data request to the Data Provider, including their own certificate and the delegation declaration from Data Consumer 1. The Data Provider checks the certificate of Data Consumer 2 (as part of the authentication process) and forwards the delegation declaration for the requested data to the Authorization Registry. As soon as the delegation declaration is checked by the Authorization Registry and valid, Data Consumer 2 is authorized for the requested data. The data will then be sent to Data Consumer 2, signed by the certificate of the Data Provider, as identity proof to Data Consumer 2.

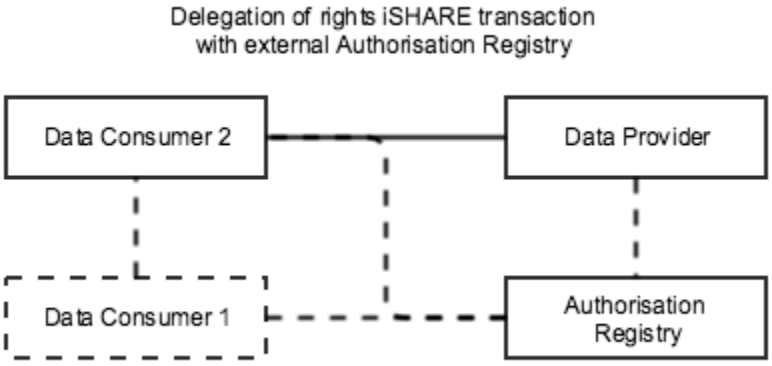
Practical examples

All Functional working group-members are invited to add practical examples of this use case in the comment section.

Roles

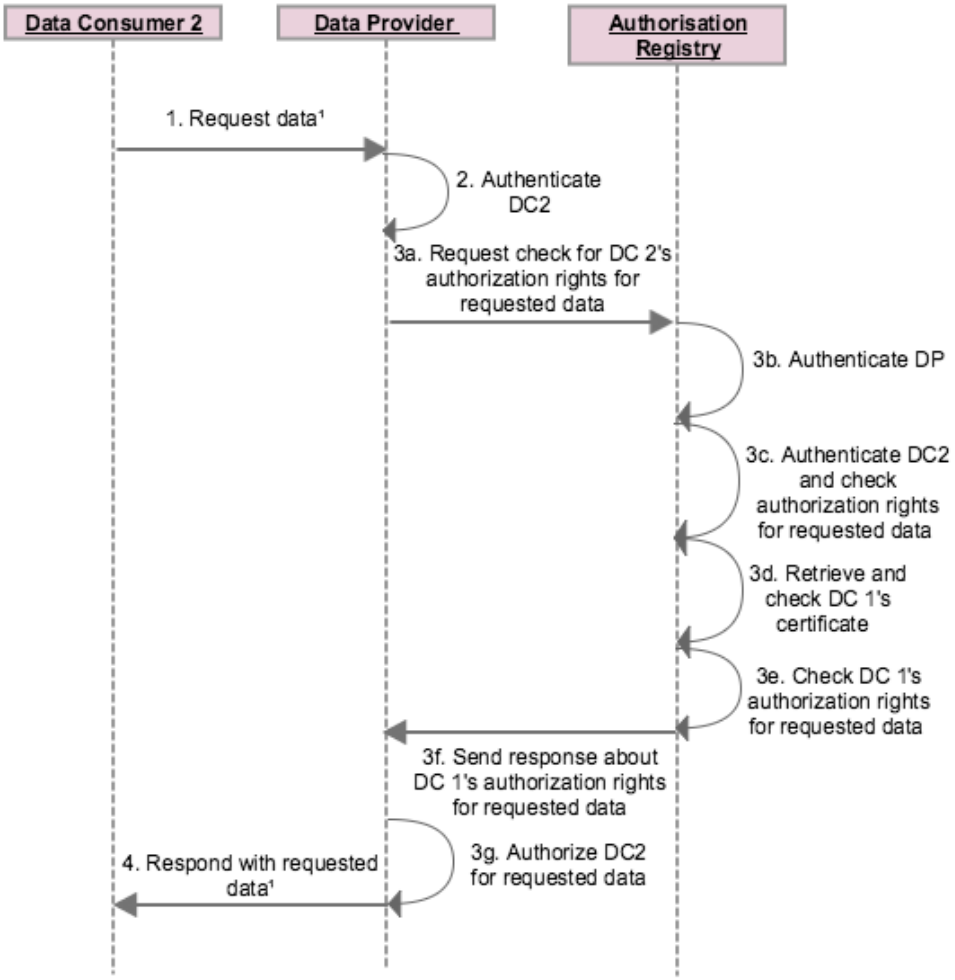
The roles which will be used in this use case are the following:

1. Data Consumer 1
2. Data Consumer 2
3. Data Provider
4. Authorization Registry



Sequence diagram

Share data based on delegation and involving an Authorisation Registry



Footnote:

¹: Message is signed with digital certificate issued by the Certificate authority as a means of identity proof.

Step no.	Action	Description
	Initial state	The Certificate Authority previously issued certificates to the Data Provider, Data Consumer 1 and Data Consumer 2. The Data Provider and the Data Consumer 1 made a bilateral agreement with regards to access rights and shared data. Data Consumer 1 sent a message to Data Consumer 2 including a delegation declaration stating the transfer of access rights from Data Consumer 1 to Data Consumer 2. The message is signed with the certificate from Data Consumer 1 to guarantee the authenticity of the message to Data Consumer 2 and to proof the authenticity of the delegation declaration to the Data Provider. The Data Provider relies on an external Authorization Registry for the validation of authorizations.

1.	Data Consumer 2 sends a request for data to the Data Provider	The Data Consumer 2 sends a request for data to the Data Provider and signs it with the certificate issued by the Certificate Authority. The data request includes the delegation declaration signed with the certificate from Data Consumer 1. The digital signature included in the request is a means of identity proof for Data Consumer 2 to the Data Provider.
2.	The Data Provider authenticates Data Consumer 2	The Data Provider authenticates the Data Consumer based on the response of the Certificate Authority that the certificate from Data Consumer 2 is valid. The Data Provider has a trust relation ship with Certificate Authority and needs the check result to trust the genuineness of the certificate from Data Consumer 2. If the check result is negative, the Data Provider can decide to cancel the data request from Data Consumer 2. Optionally, the Data Provider can send a notification to Data Consumer 2 to inform about the authentication result.
3a.	The Data Provider sends a request to the Authorization Registry to check the authorization rights of Data Consumer 2 for the requested data	The Data Provider sends a request to the Authorization Registry to check the authorization rights of Data Consumer 2 for the requested data. The authorization request includes the data request from Data Consumer 2 and the delegation declaration signed with the certificate from Data Consumer 1.
3b.	The Authorization Registry authenticates the Data Provider	The Authorization Registry authenticates the Data Provider based on the response of the Certificate Authority that the certificate from the Data Provider is valid. The Authorization Registry has a trust relation ship with Certificate Authority and needs the check result to trust the genuineness of the certificate from the Data Provider.
3c.	The Authorization Registry authenticates Data Consumer 2 and checks the authorization rights for the requested data.	The Authorization Registry authenticates Data Consumer 2 based on the response of the Certificate Authority that the certificate from Data Consumer 2 is valid. The certificate from Data Consumer 2 is included the in the authorization request from the Data Provider. The Authorization Registry checks the authorizations of Data Consumer 2 for the requested data and determines that that Data Consumer 2 is not allowed to access the requested data from the Data Provider.
3d.	The Authorization Registry retrieves the certificate from Data Consumer 1 and checks it	The Authorization Registry encounters the delegation declaration in the data request of Data Consumer 2. The delegation declaration includes a signature of Data Consumer 1. The Authorization Registry retrieves the certificate of Data Consumer 1 from the signature and requests a check from the Certificate Authority to prove the authenticity of the certificate.
3e.	The Authorization Registry checks the authorization rights of Data Consumer 1 for the requested data.	The Authorization Registry checks the authorizations of Data Consumer 1 for the requested data and determines that that Data Consumer 1 is allowed to access the requested data from the Data Provider. Based on the authentic delegation declaration signed by Data Consumer 1 and the bilateral agreement between the Data Provider and Data Consumer 1, the Authorization Registry concludes that Data Consumer 2 is authorized to access the requested data from the Data Provider.
3f.	The Authorization Registry sends a check response about the authorization rights of Data Consumer 2 for the requested data to the Data Provider	The Authorization Registry sends a response to the Data Provider including the check result of the authorization of Data Consumer 2 for the requested data.
4.	The Data Provider sends a response including the requested data to Data Consumer 2	The Data Provider sends a response to Data Consumer 2 including the requested data. The requested data is encrypted within the response and can be only read by Data Consumer 2. Even though the response is sent through the communication channel of iSHARE, the requested data is accessible only to the parties authorized by the Data Provider.. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to Data Consumer 2.
	Final state	Data Consumer 2 receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, Data Consumer 2 checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

5. Share data (H2M)

Description

This use case is about the authentication of a user through an Identity Provider. This time user will be seen as a person who directly interacts with the iSHARE network and the Data Provider is a machine answering to data requests from the user (Data Consumer). The use case is initiated by the user (Data Consumer) who logs into the system of the Identity Provider with user credentials. Upon the successful login, the Identity Provider issues authentication data to the Data Consumer.

As a next step, the user (Data Consumer) sends a request for specific data to the Data Provider and includes the authentication data from the Identity Provider. The Data Provider checks the authentication data from the user (Data Consumer) at the Identity Provider. If the authentication d ata from the user (Data Consumer) is valid, the Identity Provider returns this information to the Data Provider. Upon the successful authentication data validation from the Identity Provider, the Data Provider checks if the user (Data Consumer) holds the authorization rights to access the requested data. If the authorization rights from the user (Data Consumer) are valid, the requested data will be sent to the user (Data Consumer), signed by the certificate of the Data Provider, as identity proof to the Data Consumer.

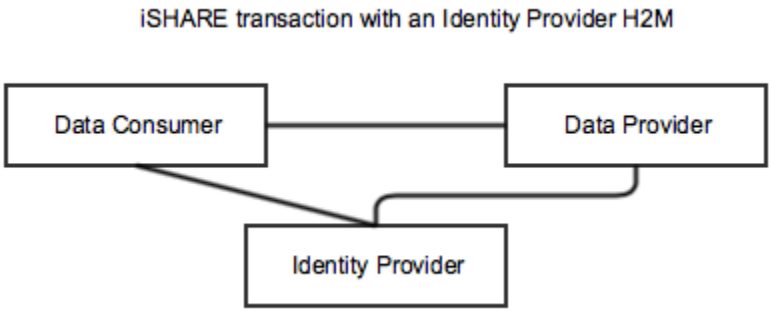
Practical examples

All Functional working group-members are invited to add practical examples of this use case in the comment section.

Roles

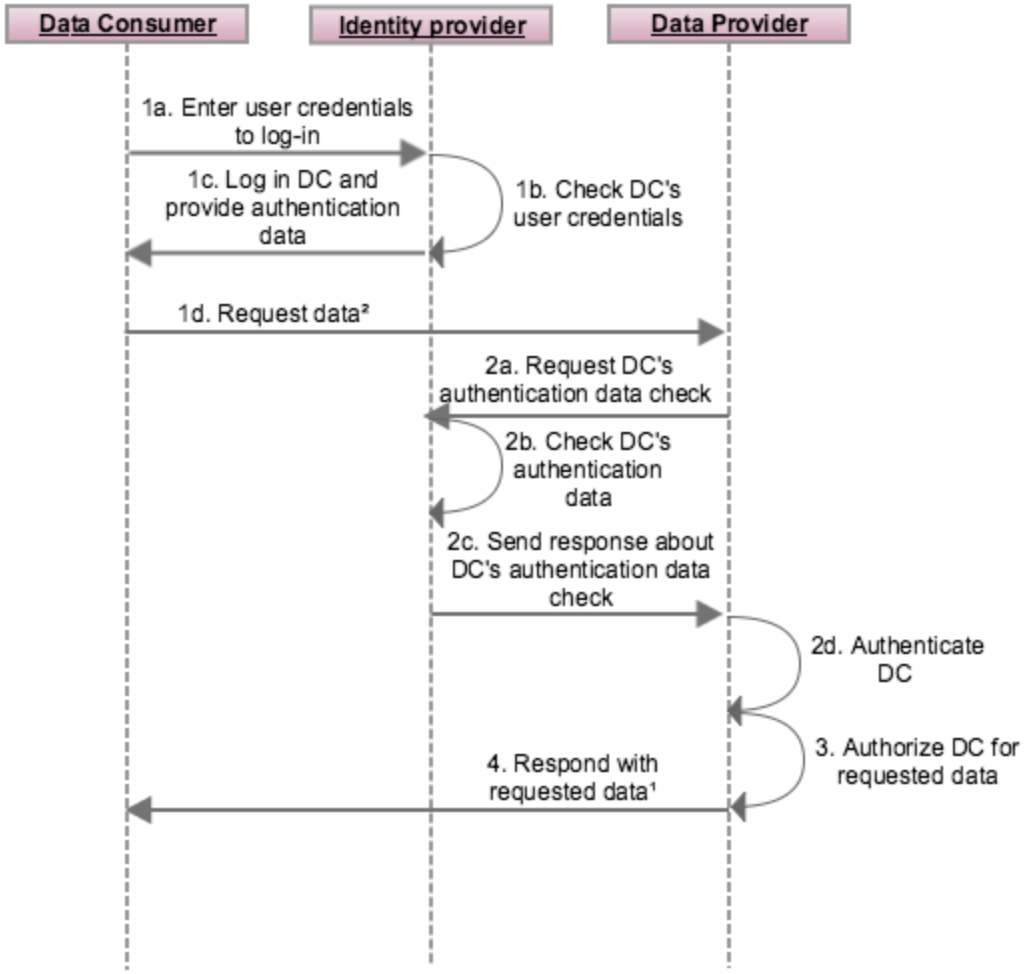
The roles which will be used in this use case are the following:

1. Data Consumer
2. Data Provider
3. Identity Provider



Sequence Diagram

Share data human-to-machine



Footnote:

[1]: Message is signed with digital certificate issued by the Certificate authority as a means of identity proof.

[2]: Message includes authentication data issued by the Identity Provider as means of identity proof.

Step no.	Action	Description
	Initial state	The Data Consumer is represented by a person operating a personal computer or mobile device. The Data Provider is represented by a machine that is programmed to automatically evaluate requests and give responses. The Identity Provider previously on-boarded the person representing the Data Consumer as user and issued log-in credentials. The Certificate Authority previously issued certificates to the Identity Provider, the Data Provider and Data Consumer. The Data Provider and the Data Consumer made a bilateral agreement with regards to access rights and shared data. The Data Provider accepts authentication data issued by the Identity provider.

1a.	The Data Consumer enters the user credentials into the system of the Identity Provider.	The Data Consumer types in the user credentials into the system of the Identity Provider. The system can be a browsed website or a mobile application.
1b.	The Identity Provider checks the user credentials of the Data Consumer.	Before the Data Consumer gets access to the system of the Identity Provider, the Identity provider first has to check and validate the user credentials provided by the Data Consumer as part of the log-in process.
1c.	The Identity Provider logs in the Data Consumer and issues authentication data to the Data Consumer.	The Data Consumer is logged in to the system of the Identity Provider and receives authentication data that can be included in data requests as identity proof for Data Providers.
1d.	The Data Consumer sends a request for data to the Data Provider	The Data Consumer sends a data request to the Data Provider and includes the authentication data issued by the Identity Provider. The authentication data included in the request is a means of identity proof for the Data Consumer to the Data Provider.
2a.	The Data Provider sends a request to the Identity Provider to check the authentication data of the Data Consumer.	The Data Provider requests a check of the authentication data of the Data Consumer from the Identity Provider. The Identity Provider first authenticates the Data Provider by checking the certificate (included in the authentication data check request) at the Certificate authority.
2b.	The Identity Provider checks the authentication data of the Data Consumer.	The Identity Provider receives the request from the Data Provider to check the authentication data of the Data Consumer. The Identity Provider validates the authentication data of the Data Consumer.
2c.	The Identity Provider sends a check response about the authentication data of the Data Consumer to the Data Provider.	The Identity Provider sends a response to the Data Provider including the check result of the authentication data of the Data Consumer.
2d.	The Data Provider authenticates the Data Consumer	The Data Provider authenticates the Data Consumer based on the response of the Identity Provider that the authentication data from the Data Consumer is valid. The Data Provider has a trust relation ship with Identity Provider and needs the check result to trust the genuineness of the authentication data from the Data Consumer. If the check result is negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the check result.
3.	The Data Provider authorizes the Data Consumer to receive the requested data.	The Data Provider checks the authorization of the Data Consumer for the requested data. Optionally, the Data Provider can choose to rely on an external Authorization Registry for the validation of authorizations. In this case we assume that the Data Provider manages the registration and validation of the authorizations internally. The Data Provider authorizes the Data Consumer based on the positive result of the internal authorization check. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the successful authorization.
4.	The Data Provider sends a response including the requested data to the Data Consumer.	The Data Provider sends a response to the Data Consumer including the requested data. The requested data is encrypted within the response and can be only read by the Data Consumer. Even though the response is sent through the communication channel of iSHARE, the requested data is only accessible to the parties authorized by the Data Provider. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to the Data Consumer.
	Final state	The Data Consumer receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, the Data Consumer checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to

which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

6. Share data involving a Broker (H2M)

Description

This use case is again about the authentication of a user through an Identity Provider involving a Broker that handles the authentication of the user and the communication between the Data Provider and the Identity Provider. Again, the user (Data Consumer) will be seen as a person who directly interacts with the iSHARE network and the Data Provider is a machine answering to data requests from the user (Data Consumer). The use case is initiated by the user (Data Consumer) who logs into the system of the Identity Provider with user credentials. Upon the successful login, the Identity Provider issues authentication data to the Data Consumer.

As a next step, the user (Data Consumer) sends a request for specific data to the Data Provider and includes the authentication data from the Identity Provider. The Data Provider needs to check the authentication data at the Identity Provider. Since there is no logical connection between the Data Provider and the Identity Provider, the Data Provider uses a Broker who can route the request to the matching Identity Provider.

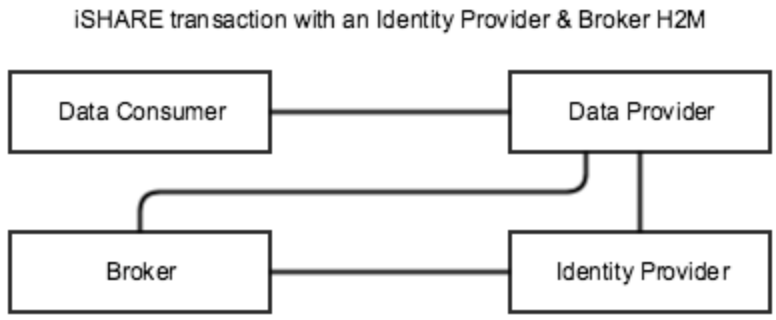
If the authentication data from the user (Data Consumer) is valid, the Identity Provider returns this information to the Broker who in turn will forward the successful authentication result to the Data Provider. Upon the receipt of the successful authentication data validation from the Identity Provider via the Broker, the Data Provider checks if the user (Data Consumer) holds the authorization rights to access the requested data. If the authorization rights from the user (Data Consumer) are valid, the requested data will be sent to the user (Data Consumer), signed by the certificate of the Data Provider, as identity proof to the Data Consumer.

Practical examples

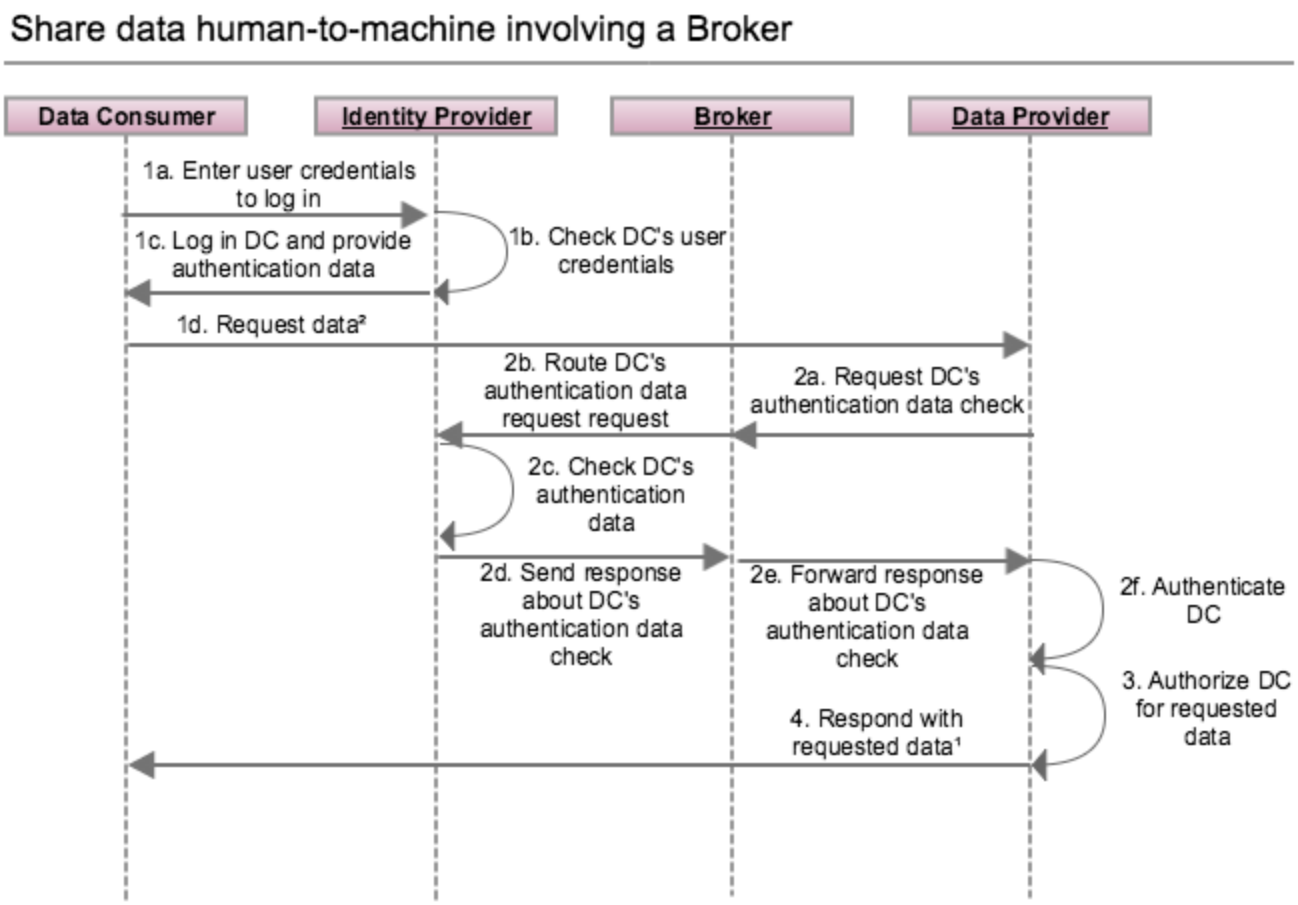
All Functional working group-members are invited to add practical examples of this use case in the comment section.

Roles

1. Data Consumer
2. Data Provider
3. Identity Provider
4. Broker



Sequence diagram



Footnote:

[1]: Message is signed with digital certificate issued by the Certificate authority as a means of identity proof.

[2]: Message includes an authentication data issued by the Identity Provider as means of identity proof.

Step no.	Action	Description
	Initial state	The Data Consumer is represented by a person operating a personal computer or mobile device. The Data Provider is represented by a machine that is programmed to automatically evaluate requests and give responses. The Identity Provider previously on-boarded the person representing the Data Consumer as user and issued log-in credentials. The Certificate Authority previously issued certificates to the Identity Provider, the Data Provider and the Data Consumer. The Data Provider accepts authentication tokens issued by the Identity provider. The Data Provider relies on a Broker for the routing to the correct Identity Provider and for the further handling of authentication tokens. The Data Provider and the Data Consumer made a bilateral agreement with regards to access rights and shared data.
1a.	The Data Consumer enters the user credentials into the system of the Identity Provider.	The Data Consumer types in the user credentials into the system of the Identity Provider. The system can be a browsed website or a mobile application.

1b.	The Identity Provider checks the user credentials of the Data Consumer.	Before the Data Consumer gets access to the system of the Identity Provider, the Identity provider first has to check and validate the user credentials provided by the Data Consumer as part of the log-in process.
1c.	The Identity Provider logs in the Data Consumer and issues authentication data to the Data Consumer.	The Data Consumer is logged in to the system of the Identity Provider and receives authentication data that can be included in data requests as identity proof for Data Providers.
1d.	The Data Consumer sends a request for data to the Data Provider	The Data Consumer sends a data request to the Data Provider and includes the authentication data issued by the Identity Provider. The authentication data included in the request is a means of identity proof for the Data Consumer to the Data Provider.
2a.	The Data Provider sends a request to the Broker to check the authentication data of the Data Consumer.	The Data Provider requests a check of the authentication data of the Data Consumer from the Broker. The Broker first authenticates the Data Provider by checking the certificate (included in the authorization request) at the Certificate authority.
2b.	The Broker forwards the check request for the authentication data of the Data Consumer to the Identity Provider.	The Broker routes the check request for the authentication data of the Data Consumer to the matching Identity Provider.
2c.	The Identity Provider checks the authentication data of the Data Consumer.	The Identity Provider receives the request from the Broker to check the authentication data of the Data Consumer. The Identity Provider first authenticates the Broker by checking the certificate (included in the authentication check request) at the Certificate authority. The Identity Provider validates the authentication data of the Data Consumer.
2d.	The Identity Provider sends a check response about the authentication data of the Data Consumer to the Broker.	The Identity Provider sends a response to the Broker including the check result of the authentication data of the Data Consumer.
2e.	The Broker forwards the check response about the authentication data of the Data Consumer to the Data Provider.	The Broker first authenticates the Identity Provider by checking the certificate (included in the authentication check response) at the Certificate authority. The Broker then routes the response of the check result of the authentication data of the Data Consumer from the Identity Provider to the matching Data Provider.
2f.	The Data Provider authenticates the Data Consumer	The Data Provider authenticates the Broker by checking the certificate (included in the authentication check response) at the Certificate authority. The Data Provider then authenticates the Data Consumer based on the response of the Broker that the access token from the Data Consumer is valid. The Data Provider has a trust relation ship with the Broker and needs the check result to trust the genuineness of the access token from the Data Consumer. If the check result is negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the check result.
3.	The Data Provider authorizes the Data Consumer to receive the requested data.	The Data Provider checks the authorization of the Data Consumer for the requested data. Optionally, the Data Provider can choose to rely on an external Authorization Registry for the validation of authorizations. In this case we assume that the Data Provider manages the registration and validation of the authorizations internally. The Data Provider authorizes the Data Consumer based on the positive result of the internal authorization check. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the successful authorization.

4.	The Data Provider sends a response including the requested data to the Data Consumer.	The Data Provider sends a response to the Data Consumer including the requested data. The requested data is encrypted within the response and can be only read by the Data Consumer. Even though the response is sent through the communication channel of iSHARE, the requested data is only accessible to the parties authorized by the Data Provider. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to the Data Consumer.
	Final state	The Data Consumer receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, the Data Consumer checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

7. Share data involving a Broker and an Authorization Registry (H2M)

Description

This use case is again about the authentication of a user through an Identity Provider involving a Broker that handles the authentication of the user and the communication between the Data Provider and the Identity Provider and the authorization of the user by managing the communication between the Data Provider and the Authorization Registry. Again, the user (Data Consumer) will be seen as a person who directly interacts with the iSHARE network and the Data Provider is a machine answering to data requests from the user (Data Consumer). The use case is initiated by the user (Data Consumer) who logs into the system of the Identity Provider with user credentials. Upon the successful login, the Identity Provider issues authentication data to the Data Consumer.

As a next step, the user (Data Consumer) sends a request for specific data to the Data Provider and includes the authentication data from the Identity Provider. The Data Provider needs to check the authentication data at the Identity Provider. Since there is no logical connection between the Data Provider and the Identity Provider, the Data Provider uses a Broker who can route the request to the matching Identity Provider.

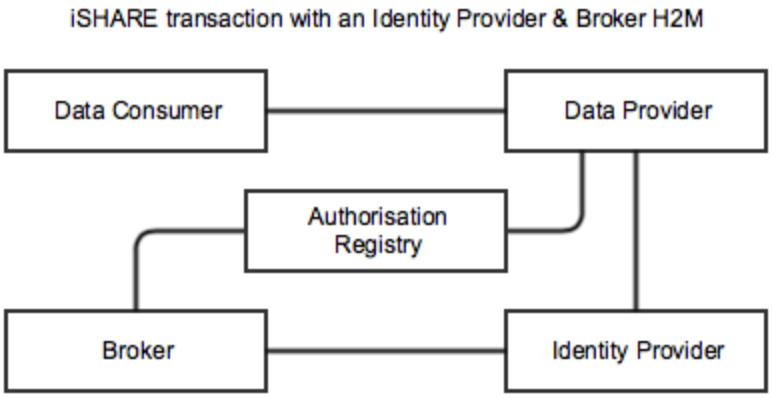
If the authentication data from the user (Data Consumer) is valid, the Identity Provider returns this information to the Broker who in turn will forward the successful authentication result to the Data Provider. Upon the receipt of the successful authentication data validation from the Identity Provider via the Broker, the Data Provider requests a check of the authorization rights of the Data Consumer from the Broker. The Broker forwards the authorization check request to the Authorization Registry that checks the authorization rights of the Data Consumer. Once the user (Data Consumer) has valid authorization rights to access the requested data, the Authorization Registry returns the successful authorization check result to the Broker who then in turn forwards it to the Data Provider. With the valid authentication and authorization checks performed by the Identity Provider and the Authorization Registry, respectively, and communicated via the Broker, the Data Provider sends the requested data to the user (Data Consumer) signed by the certificate of the Data Provider, as identity proof to the Data Consumer.

Practical examples

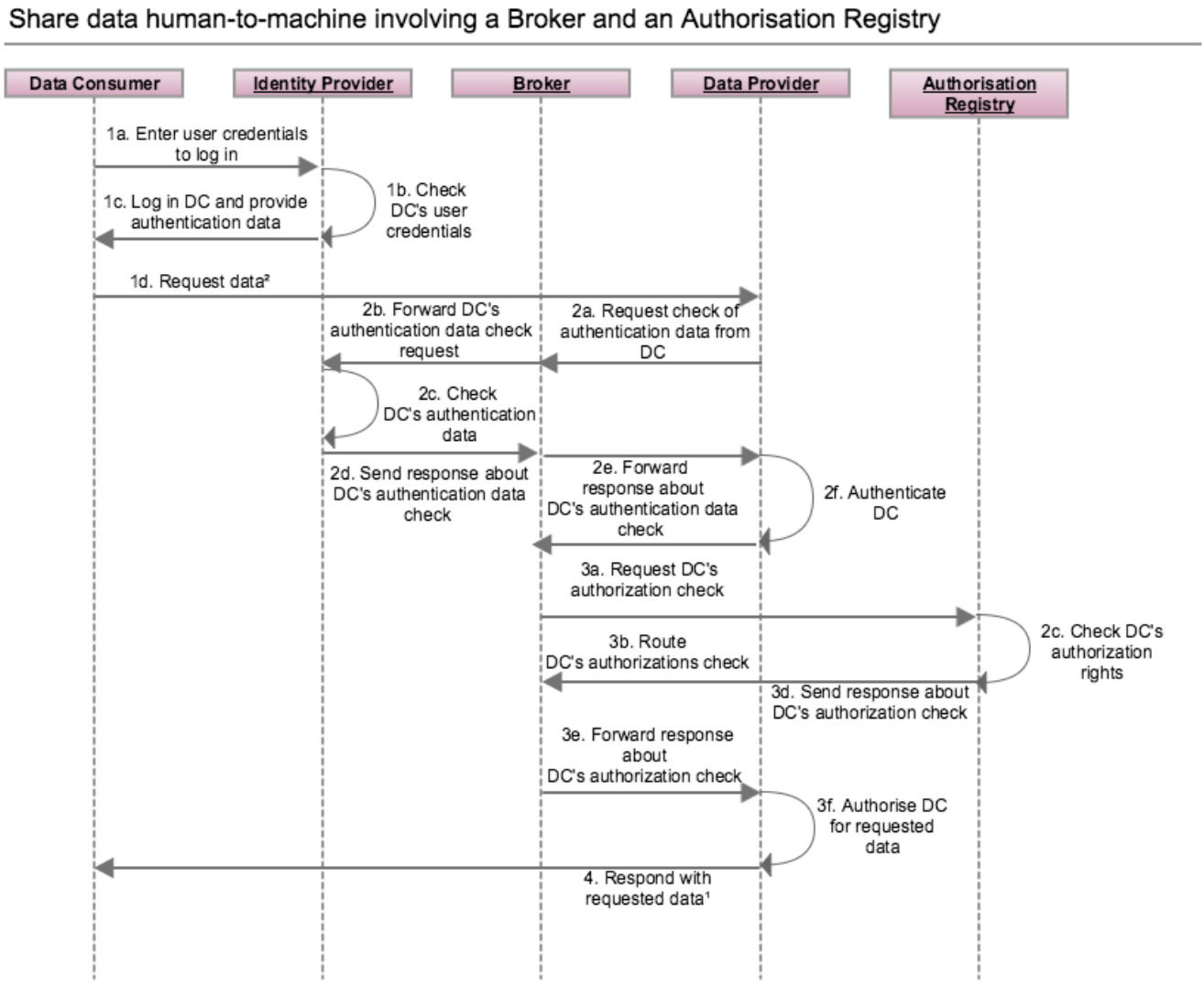
All Functional working group-members are invited to add practical examples of this use case in the comment section.

Roles

1. Data Consumer
2. Data Provider
3. Identity Provider
4. Broker
5. Authorisation Registry



Sequence Diagram



Footnote:

¹: Message is signed with digital certificate issued by the Certificate authority as a means of identity proof.

[2]: Message includes an authentication data issued by the Identity Provider as means of identity proof.

Step no.	Action	Description
	Initial state	The Data Consumer is represented by a person operating a personal computer or mobile device. The Data Provider is represented by a machine that is programmed to automatically evaluate requests and give responses. The Identity Provider previously on-boarded the person representing the Data Consumer as user and issued log-in credentials. The Certificate Authority previously issued certificates to the Broker, the Identity Provider, the Authorization Registry, the Data Provider and the Data Consumer. The Data Provider relies on an external Authorization Registry for the validation of authorizations. The Data Provider accepts authentication data issued by the Identity provider. The Data Provider relies on a Broker for the routing to the correct Identity Providers and Authentication Registry and for the further handling of authentication data and authorizations. The Data Provider and the Data Consumer made a bilateral agreement with regards to access rights and shared data.
1a.	The Data Consumer enters the user credentials into the system of the Identity Provider.	The Data Consumer types in the user credentials into the system of the Identity Provider. The system can be a browsed website or a mobile application.
1b.	The Identity Provider checks the user credentials of the Data Consumer.	Before the Data Consumer gets access to the system of the Identity Provider, the Identity provider first has to check and validate the user credentials provided by the Data Consumer as part of the log-in process.
1c.	The Identity Provider logs in the Data Consumer and issues authentication data to the Data Consumer.	The Data Consumer is logged in to the system of the Identity Provider and receives authentication data that can be included in data requests as identity proof for Data Providers.
1d.	The Data Consumer sends a request for data to the Data Provider	The Data Consumer sends a data request to the Data Provider and includes the authentication data issued by the Identity Provider. The authentication data included in the request is a means of identity proof for the Data Consumer to the Data Provider.
2a.	The Data Provider sends a request to the Broker to check the authentication data of the Data Consumer.	The Data Provider requests a check of the authentication data of the Data Consumer from the Broker.
2b.	The Broker forwards the check request for the authentication data of the Data Consumer to the Identity Provider.	The Broker first authenticates the Data Provider by checking the certificate (included in the authentication check request) at the Certificate authority. The Broker then routes the check request for the authentication data of the Data Consumer to the matching Identity Provider.
2c.	The Identity Provider checks the authentication data of the Data Consumer.	The Identity Provider receives the request from the Broker to check the authentication data of the Data Consumer. The Identity Provider first authenticates the Broker by checking the certificate (included in the authentication check request) at the Certificate authority. The Identity Provider validates the authentication data of the Data Consumer.
2d.	The Identity Provider sends a check response about the authentication data of the Data Consumer to the Broker.	The Identity Provider sends a response to the Broker including the check result of the authentication data of the Data Consumer.

2e.	The Broker forwards the check response about the authentication data of the Data Consumer to the Data Provider.	The Broker first authenticates the Identity Provider by checking the certificate (included in the authentication check response) at the Certificate authority. The Broker then routes the response of the check result of the authentication data of the Data Consumer from the Identity Provider to the matching Data Provider.
2f.	The Data Provider authenticates the Data Consumer	The Data Provider first authenticates the Broker by checking the certificate (included in the authentication check response) at the Certificate authority The Data Provider authenticates the Data Consumer based on the response of the Broker that the authentication data from the Data Consumer is valid. The Data Provider has a trust relation ship with the Broker and needs the check result to trust the genuineness of the authentication data from the Data Consumer. If the check result is negative, the Data Provider can decide to cancel the data request from the Data Consumer. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the check result.
3a.	The Data Provider sends a request to the Broker to check the authorizations of the Data Consumer for the requested data.	The Data Provider requests a check of the authorizations of the Data Consumer for the requested data from the Broker.
3b.	The Broker forwards the check request for the authorizations of the Data Consumer to the Authorization Registry.	The Broker first authenticates the Data Provider by checking the certificate (included in the authorization check request) at the Certificate authority. The Broker routes the check request for the authorizations of the Data Consumer to the matching Authorization Registry.
3c.	The Authorization Registry checks the authorizations of the Data Consumer.	The Authorization Registry receives the request from the Broker to check the authorizations of the Data Consumer for the requested data. The Authorization registry authenticates the Broker by checking the certificate (included in the authorization check request) at the Certificate authority. The Authorization Registry checks the authorizations of the Data Consumer for the requested data.
3d.	The Authorization Registry sends a check response about the authorizations of the Data Consumer to the Broker.	The Authorization Registry sends a response to the Broker including the check result of the authorizations of the Data Consumer.
3e.	The Broker forwards the check response about the authorizations of the Data Consumer to the Data Provider.	The Broker authenticates the Authorization Registry by checking the certificate (included in the authorization check response) at the Certificate authority. The Broker routes the response of the check result of the authorizations of the Data Consumer from the Authorization Registry to the matching Data Provider.
3f.	The Data Provider authorizes the Data Consumer to receive the requested data.	The Data Provider authenticates the Broker by checking the certificate (included in the authorization check response) at the Certificate authority. The Data Provider authorizes the Data Consumer based on the positive result of the Authorization Registry. Optionally, the Data Provider can send a notification to the Data Consumer to inform about the successful authorization.
4.	The Data Provider sends a response including the requested data to the Data Consumer.	The Data Provider sends a response to the Data Consumer including the requested data. The requested data is encrypted within the response and can be only read by the Data Consumer. Even though the response is sent through the communication channel of iSHARE, the requested data is only accessible to the parties authorized by the Data Provider. The response from the Data Provider is signed with the certificate issued by the Certificate Authority. The digital signature included in the request is a means of identity proof for the Data Provider to the Data Consumer.
	Final state	The Data Consumer receives the response included with the requested data from the Data Provider. To verify the authenticity and have complete trust that the response originated from the Data Provider, the Data Consumer checks the validity of the certificate with which the response is signed at the Certificate Authority.

Note: For every use case, the interface specifications between the interacting roles and the technical standards & specifications according to which the use case is functioning, will be both defined and determined during the course of the functional and technical iSHARE workshops.

Secondary use cases

The secondary use cases have an administrative background. They support and form the basis for the primary use cases where the acts of sharing data are defined. Secondary use cases include the issuance and management of certificates and credentials, registration and management of authorisations, the execution of audits and so on.

The secondary use cases will be further discussed and developed in the course of the iSHARE functional working groups.

Detailing key features

This section dives deeper into the key features of the iSHARE scheme. What we detail per key feature is the following:

- Key feature: Provide trust framework for PKI certificates
  - PKI trusted list
  - iSHARE's own PKI
- Key feature: Provide flexibility in authorization
  - Granular authorization
  - Multiple authorization registration points
- Key feature: Allow for management of consent
- Key feature: Support multiple interaction models

A section on Federated identity is also added here - as it is related to the above key features.

PKI trusted list

One of iSHARE's key features is to provide a trust framework for PKI certificates including a list of certificate roots (also called PKI roots), or Certificate Authority in other words, that meet the iSHARE requirements and can be trusted by all iSHARE participates. We call it the PKI trusted list that will be developed and provided by the iSHARE scheme.

The European Union (EU) implemented the eIDAS regulation providing a list of criteria for Certificate Authorities which can be re-used within iSHARE.

However, the eIDAS regulation does only hold for EU member states and not for countries outside of the EU. As iSHARE might not be restricted to only European countries in the future, Certificate Authorities outside of the EU do not have to comply with the eiDAS regulation. In that case, we cannot fully rely on the criteria provided by eIDAS and would want to deviate from that in order to be able to work with Certificate Authorities from outside the EU.

Also, it might turn out in the course of the workshops in phase 2 that not all criteria listed in the eIDAS regulation are necessary and applicable to the iSHARE requirements. A reduced version of the eIDAS regulation could be sufficient for the iSHARE objectives. Equally, the iSHARE scheme could encounter that stronger criteria are needed which are not listed in eIDAS.

During the co-creation process we could also come across the need for specific certificates, i.e. to prove the authenticity of Authorization Registries, that are currently not provided by Certificate Authorities. In that case, the need for iSHARE's own PKI could arise.

Here a list of questions we need to ask ourselves in this regard during the co-creation process in phase II:

- Based on which criteria do we evaluate Certificate Authorities and put them on our PKI trusted list?
  - 'Level of Assurance' (LoA)?
  - Interoperability?
  - Issuing process of certificates?
  - Supervision and control of Certificate Authorities?
- To which extend are we going to re-use the criteria listed in the eIDAS regulation?Will the criteria be a 'light' or 'heavy' version of the eiDAS regulation?
- Are we going to include criteria that are not listed in eIDAS yet?
- Is there a need to create specific certificates and establish our own Certificate Authority?

iSHARE's own PKI

It is assumed that existing PKIs are sufficient to meet all iSHARE requirements. If, during the course of phase 2, this assumption turns out to be false, an additional iSHARE specific PKI can be created.

We foresee the following scenario's for the role of iSHARE (non-exhaustive):

- iSHARE could be part of an existing PKI scheme and make use of certificates issued by existing certificate authorities
- iSHARE could be part of an existing PKI scheme and take the role of a certificate authority issuing its own certificates
- iSHARE could implement its own PKI scheme and make use of certificates issued by existing certificate authorities
- iSHARE could implement its own PKI scheme and take the role of a certificate authority issuing its own certificates

Granular authorization

One of the iSHARE key features is the flexibility in authorization with regards to the authorization scope, granularity and source. In this section we will expand on the granularity for authorizations.

By granular authorization we mean the level of details an authorizing process requires to limit and separate privileges (= the right to access a resource).

A single authorization may enable a number of privileges the same way as a privilege may require multiple authorizations. An authorizing authority should be capable of handling both scenario's.

Granularity is not based on either authorization requests or privileges, but on functions. Those functions are processed in computer algorithms that express the rules defined in authorization policies. XACML for instance is a standard that defines a declarative, fine-grained, attribute-based access control policy language that can be used to write computer algorithms.

Fine grain authorization

Fine grain authorization defines very specific functions that are applicable to specific tasks. Each authorization request is broken up into tasks and each task is then assigned to a function.

Role-based access control is an example for "fine grain": access to a resource depends on user's role (not only on user), and user can have multiple roles (having access to multiple resources).

Attribute-based access control is an example for "finer grain" authorization: access to a resource depends on attributes that the user has to bring along to proof that they meet the authorization requirements (the policies).

Coarse grain authorization

It is simpler and different from fine grain authorization as there are no lower detail tasks within the functions.

Access control lists (ACL's) are an example for "coarse grain" authorization: once the user is authenticated, the user is allowed access to the requested resource depending on whether that user's ID is on a whitelist (or blacklist, in case user is blocked).

Example for coarse, fine, finer grain authorization

- Coarse: User A, User C, User F & User L can access container A.
- Fine: Truck companies have access to container A.
- Finer: The users that can proof to be a trucker from company B, working for the Data Provider in week X, can access container A.

Multiple authorization registration points

One of the iSHARE key features is the flexibility in authorization with regards to the authorization scope, granularity and source. In this section we will expand on the authorization sources and the possibility of having multiple registration points. Data Providers have different options when it comes to the management of their authorization information.

Authorization registration point resides at Data Provider

The authorization information can reside very close to the data, namely at the Data Provider. They register, validate and execute their own authorization policies.

Authorization registration point resides at separate source

The authorization information can be handled by an external third party separated from the Data Provider.

Authorization registration point resides at Data Consumer that delegates rights

Data Providers can give permission to Data Consumers to delegate their right to access a specific resource to another Data Consumer. In that case authorization information of the Data Provider can partly reside at a Data Consumer that delegates its own right to another Data consumer.

Federated identity

A federated identity is a 'summarised' identity that is spread out and recognised across multiple systems. A person's electronic identity and attributes are linked and stored across multiple, distinct identity management systems.

The use of federated identities could reduce costs by eliminating the need for proprietary identity solutions. By proprietary solutions we mean products and services provided by one vendor. The lack of competition of other vendors can make the acquisition and maintenance of the solution costly. Secondly, it may not be fully interoperable with other solutions in the field. Cloud service providers for instance are known for having proprietary identity management systems.

Single Sign On (SSO) is a federated identity solution.

Open standards & specs for federated identities

For the implementation of federated identity solutions and the realisation of interoperability between parties, the use of open industry standards and openly published specifications is a must.

Examples for technical specifications & standards of federated identity solutions are SAML, OAuth, OpenID, Security Tokens (Simple Web Tokens, JSON Web Tokens, and SAML assertions), Web Service Specifications, Microsoft Azure Cloud Services, and Windows Identity Foundation. If you want to read more about them, we refer to the section Technical.

Examples for digital federated identity platforms that allow their users to log onto other third party mobile & web applications are Google Account, Twitter, LinkedIn, PayPal, Foursquare, MySpace, AOL, Amazon.

Single Sign On (SSO)

Single Sign On (SSO) is a federated identity solution.

It is however important to note that not all federated identity solutions include SSO. The difference between SSO and other federated identity solutions is that SSO has the requirement to authenticate the user once and remain in the authenticated state across multiple systems. The users fill in their credentials once for one particular website to prove their identity and can access multiple websites automatically without the need to re-enter their credentials until the sessions times out (password is remembered for a certain period of time). Ordinary federated identity system do hold the requirement to be recognised across multiple systems as well, but not necessarily after authenticating once at one website to remain authenticated across many websites without being asked to enter credentials a second time.

It may also be interesting to know that Single Sign Off exists as well where a signing out action in one environment terminates the access to all previously signed-in environments.

Functional requirements per role

This section will describe the functional requirements (e.g. accountability) per role in the iSHARE network - it will be detailed by the Functional working group.

User interface requirements

Accessibility

The iSHARE network is fundamentally designed to work for all people, organisations and companies who are related to the logistics sector and would like to share data.

Language

To encourage the international possibilities of the iSHARE network, interfaces will be in English or in several (other) languages.

Note that this section will be further detailed by the Functional working group.

Identifiers

A unique identifier (UID) including the related processes is required, in order to be able to uniquely identify all users in the entire iSHARE network.

Identity attributes

Before users are issued digital certificates and other credentials (i.e. authentication and authorization means) to request (as Data Consumer) or share data (as Data Provider), they first need to pass through an on-boarding or registration process. Hereunder a list of personal identifiers that uniquely identify persons:

- Unique identifier (i.e. unique serial, random numbers linked to a person's identity)
- Biometrics (i.e. fingerprints, iris scan, voice recognition etc.)
- Name
- Organisation
- Job title
- Role within organisation

Note that personal data such as passport numbers or social security numbers are not allowed to be registered according to the privacy regulations set by the European Union and the Dutch government.

At it stands now, it is proposed that iSHARE is going to reuse existing identity solutions in the Dutch market such as eHerkenning and iDIN, and once expanding to other countries, international identity solutions. In this light, iSHARE is going to accept existing on-boarding solutions and processes that are already put in place and to which iSHARE users adhere. However, the possibility of developing a new iSHARE identity solution (opposed to re-using existing identity solution such as i.e. eHerkenning) is not excluded and has to be decided in the iSHARE set of agreements.

The on-boarding process and identity attributes should be thoroughly discussed in the course of the iSHARE functional workshop, because based on the outcome of the on-boarding process, it is decided if a user is allowed to make use of iSHARE and is issued the required credentials for further use,.

Authorization attributes

Attributes can serve as additional information in the authorization validation process as they play a role in the validation of authorization policies (examples are authorization policies are i.e. "All DC's holding the role of manager get access to document X" or "All DC's at a distance from X meters from location Y get access to document Z").

Before Data Providers can give access to their data, they might want to receive additional information from Data Consumers proving that they meet the authorization policies. To trust the authenticity of the additionally provided information from the Data Consumer, there might be the need for a proving mechanism.

The following questions have to be addressed in the course of the functional work groups:

1. Which attributes do we foresee to play a role in the authorization policies defined by Data Providers?
2. How can Data Consumers prove the authenticity of their delivered attributes?
3. How to proof the authenticity of
    - location?
    - role?
    - containers?

Technical

This section covers the relevant Technical topics of the iSHARE scheme:

- Interface specifications: the eHerkenning interface specifications that (can) serve as input for the interface specifications for iSHARE
- Security: five important key aspects of information security are listed. They (can) form the basis and have tobe covered by the technical requirements of iSHARE
- Relevant standards: a number of existing technical standards are listed that are successfully implemented in well-established digital services related to authentication, access management and the secure exchange of information

These topics (and possibly others that arise during Phase 2) are detailed by the Technical working group.

Interface specifications

The interfaces between the participating roles in iSHARE listed in Roles & Responsibilities will be defined in the course of the 'Functional' co-creation workshop. For a first impression of how interfaces can be described for the traffic between roles in a network, we added a link hereunder to the interface specifications of eHerkenning.

The following link leads to the interface specifications between all participating roles in eHerkenning: https://afsprakenstelsel.etoegang.nl/display/as/Interface+specifications

Security

In this section we describe the following five key concepts of information security whose purpose it to prevent the unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information:

- Confidentiality
- Integrity
- Authenticity
- Availability
- Non-repudiation

Confidentiality

In the context of information security, confidentiality refers to the protection of information from disclosure to unauthorised parties.

The message the recipient gets can be proven not to have been read by anyone else and that the information is kept between the sender and recipient. Confidentiality can be achieved by i.e. the use of cryptography and the encryption of information, as well as through the enforcement of file permissions and access control lists to restrict access to sensitive information.

Integrity

In the context of information security, integrity refers to the protection of information from being modified by unauthorized parties.

The message the recipient receives from the sender can be proven not to have been changed during the transmission. Integrity can be achieved

by i.e. hash functions (hashing the received data and comparing it with the hash of the original message).

Authenticity

Authenticity (from Greek: *authentikos*, "real, genuine") in the context of information security refers to the truthfulness of messages and if they have been sent by an authentic sender. There are ways for a recipient to prove if the received message has been sent by a "genuine, true" positively-identified sender.

Authenticity can be achieved by i.e. by digitally signing the message with the private key from the sender. The recipient can verify the digital signature with the matching public key.

The public and private key pairs are issued by Certificate Authorities. Those are entities within a Public Key Infrastructure (PKI) facilitating the trust framework.

Availability

Availability in the context of information security refers to the ability of authorized parties to access their resources whenever they need to.

Availability can be achieved by i.e. back-ups to limit the damage caused by incidents (such as broken hard drives, natural disasters etc.).

Non-repudiation

Non-repudiation means "onweerlegbaarheid" in Dutch and refers in the context of information security to the guarantee that a message is actually sent by the sender and received by the recipient. The broadcast and receipt (the authenticity) of the message cannot be denied by neither of the involved parties (sender and recipient).

Non-repudiation can be achieved by digital signatures in combination with message tracking.

Relevant standards

The section "Relevant standards" covers a number of technical standards we consider to be relevant for the realisation of the iSHARE set of agreements. The table hereunder matches the technical standards to their main purposes (i.e. authentication, authorization, cryptography, data exchange, data formatting).

Technical standard (in alphabe-tical order)	Authentication	Authorization	Cryptography	Data exchange	Data formatting	What is it about?
JSON						*Formatting/structuring data in object units*
OAuth	()**					*Standard for authorization (delegated access control, password handling)*
OpenID						*Authentication layer built on top of OAuth 2.0 protocol*
SAML	()*	()*				*XML-based data format for exchange of authentication and authorization data*
SOAP					()*	*Network protocol for the exchange of structured information*
TLS						*Cryptographic protocol for secure communication of computer networks*
UMA						*OAuth-based access management protocol standard*
X.509						*Cryptographic standard for PKI's (digital certificates & keys)*
XACML						*Standard for authorization policies (language, architecture, processing model)*
XML						*Formatting/structuring text documents to be both human- and machine-readable*
XML Signature					()*	*Standard defining an XML syntax for digital signatures to sign XML documents*

()*: It is associated with the above-mentioned topic in the table, but not in the first place.

()**: The use of OAuth as an authentication method may be referred to as pseudo-authentication where the access token is used as proof of identity.

HTTP

On this page a brief description of HTTP is provided. For the most recent version of the specification click on this link.

Description

HTTP is short for 'Hypertext Transfer Protocol'.

HTTPS stands for 'Hypertext Transfer Protocol Secure' (or HTTP over TLS or HTTP over SSL). It is a protocol for secure communication over a computer network and is widely used on the Internet.

Difference between HTTP and HTTPS

The difference between HTTP and HTTPS is that HTTPS consists of communication over the Hypertext Transfer Protocol that is encrypted by TLS or its forerunner SSL.

The main reason for the use of HTTPS is the authentication of the visited website and the protection of the privacy and integrity of the shared data.

JSON

On this page a brief description of JSON is provided. For the most recent version of the specification click on this link.

Description

JSON is short for 'JavaScript Object Notation' and is an open standard data format that does not depend on a specific programming language. This compact data format makes use of human-readable (easy to read) text to exchange data objects (structured data) between applications and for data storage

JSON is most commonly used for asynchronous communication between browsers and servers.

OAuth

On this page a brief description of OAuth is provided. For the most recent version of the specification click on this link.

Description

OAuth is an open standard for authorization which is used by i.e. Google, Facebook, Microsoft, Twitter etc. to let their users share information about their accounts with other applications or websites. OAuth is designed to work with HTTP.

Through OAuth users can authorize third party applications or websites to access their account information on other "master" systems without the need of sharing with them their credentials to login onto the platform. OAuth provides a "secure delegated access" to resources (email accounts, pictures accounts, etc.) on behalf of the resource owner

It specifies a method for resource owners to authorize third parties access to their resources without sharing their credentials (username, password). Authorization servers (of the platform) issue access tokens to third party clients (applications or websites) with the approval of the resource owner (= end user). The third party client needs the access token to get access to the resources that are stored on the resource server (of the master system)

OAuth in relation to other standards & specifications

OAuth is not the same as OATH (Initiative for Open Authentication) which is a reference architecture for authentication and not a standard for authorization.

OAuth is linked to OpenID Connect since OIDC is the authentication layer built upon OAuth 2.0.

OAuth is not the same as XACML which is an open standard for authorization policies but can be use within XACML for ownership consent and access delegation.

OAuth 2.0

OAuth 2.0 provides specific authorization flows for web applications, desktop applications, **mobile phones**, and living room devices.

OAuth 2.0 is not backwards compatible with OAuth 1.0.

Because OAuth 2.0 is more of a framework than a defined protocol, one OAuth 2.0 implementation is less likely to be naturally interoperable with another OAuth 2.0 implementation.

OAuth 2.0 does not support signature, encryption, channel binding, or client verification. It relies completely on TLS for some degree of confidentiality and server authentication.

OAuth's phishing vulnerability

The most shocking OAuth security breach is the phishing vulnerability: every application/website using OAuth is visually (not technically) asking the end users to fill in their credentials of the master system (where the resources are stored).

Hacker's can visually emulate this process of third party clients and let end users believe that they are filling in their credentials on a genuine website. In doing so, hackers can succeed in stealing credentials. Two-factor authentication (two types of evidence/credentials) does not add extra security as phishing website can steal those extra types of credentials as well.

OpenID Connect

On this page a brief description of OpenID Connekt (which we would like to stress is the most recent version of OpenID and an authentication layer on top of OAuth) is provided. For the most recent version of the specification click on this link.

Description

Open ID Connect (OIDC) is the authentication layer that is built on top of OAuth 2.0 protocol which is an authorization framework. The OIDC authentication layer allows clients to verify the ID and obtain basic profile information of their end-users

The authentication is performed by the authorization server (managing the access rights and conditions) in an interoperable and REST-like manner.

OpenID Connect's building blocks

OIDC specifies a RESTful HTTP API using JSON as data format.

REST (Representational state transfer) or RESTful web services provide a method to achieve interoperability between computer systems and the internet.

APIs (Application Programming interfaces) enable Machine to Machine (M2M) communication where one machine calls upon the software functionality of another machine. They facilitate connectivity between applications. It is a software architectural approach that revolves around the view on digital interfaces that APIs provide self-service, one-to-many, reusable interfaces.

With OIDC a broad range of clients (web-based, mobile, JavaScript) can request and receive data about authentication sessions end-user profiles.

The specification is extensible (meaning it takes future growth into consideration) and supports optional features for encryption, ID data, discovery of OpenID providers and session management

OpenID Connect 1.0

Open ID Connect 1.0 is most recent version of OpenID.

OpenID Connect performs many of the same tasks as OpenID 2.0, but in an API-friendly way and usable by native and mobile applications.

OpenID Connect defines optional mechanisms for robust signing and encryption.

Whereas integration of OAuth 1.0a with OpenID 2.0 required an extension, in OpenID Connect, OAuth 2.0 capabilities are integrated with the protocol itself.

SAML

On this page a brief description of SAML is provided. For the most recent version of the specification click on this link.

Description

SAML is short for "Security Assertion Markup Language" and is an open standard and XML-based data format to exchange authentication and authorization data between identity providers and service providers

SAML specifies the assertions (= claims) in XML passed from the user to identity provider and to the service provider.

After the user requests a service from the service provider, the service provider obtains an ID assertion from the ID provider which the service provider can use to make an access control decision ("Is user authorized to use the requested service?"). Before the ID provider shares the ID assertion with the service provider, the ID provider may ask for extra information from the user (i.e. user name, password, fingerprint) for authentication reasons.

In SAML, one single ID provider may provide SAML assertions to many service providers. Likewise, one single service providers may rely on assertions from multiple ID providers

One of SAML's most important requirement is that of Single Sign On (SSO): after users log in once for a service (web or local environment) for which they have authorization, they can access the same service repeatedly/multiple times without log-in credentials being asked and validated again.

Important note: The most recent version SAML 2.0 was built with the assumption of the client being a web browser from desktops/laptops. Unfortunately because of this presumption it doesn't adapt well into the mobile application ecosystem

SAML's basic standards

SAML is built on the following existing standards:

- XML (eXtensible Markup Language)
- XSD (XML Schema Definition)
- XML signature standard for authentication and message integrity
- XML encryption standard to encrypt identifiers, attributes and assertions. XML encryption is reported to have security concerns
- HTTPS (Hypertext Transfer Protocol Secure) as communications protocol
- SOAP (Simple Object Access Protocol): a network protocol for the exchange of structured information

The SAML specifications recommend and even mandate (for some cases) specific security standards and protocols such as TLS 1.0 (for transport-level security) and XML Signature and XML Encryption (for message-level security)

SAML's building blogs

SAML includes assertions, protocols, bindings and protocols.

- Assertions: the syntax and semantics of the assertions are described in "SAML Core", together with the protocol needed to request and transmit assertions
- Protocols: "SAML protocol" focusses on what is transmitted, not how (as this is determined by the choice of binding)
- Bindings: "SAML binding" describes how how SAML requests and responses map onto to other standard messaging or communication protocols. An example of an (synchronous) binding is the SAML SOAP binding
- Profiles: "SAML profile" is a specific form (profile) of a defined use case with a given combination of assertions, protocols and bindings

SAML 2.0

SAML 2.0 replaces SAML 1.1: In SAML 1.1 Web Browser SSO Profiles are initiated by the ID Provider. In SAML 2.0, however, the flow begins at the service provider who issues an explicit authentication request to the ID provider (significant new feature).

It makes use of security tokens containing assertions to pass information about a user.

It enables web-based authentication and authorization scenarios including cross-domain SSO, which helps reduce the administrative overhead of distributing multiple authentication tokens to the user

When SAML 2.0 was built, it was built with the assumption of the client being a web browser from desktops/laptops. Unfortunately because of this presumption it doesn't adapt well into the mobile application ecosystem

SOAP

On this page a brief description of SOAP is provided. For the most recent version of the specification click on this link.

Description

SOAP stands for 'Simple Object Access Protocol' and is a network protocol for the exchange of structured information. The SOAP message format follows the "XML Information Set" (XML InfoSet) which is a specification describing the data model for an XML document as a set of information items.

SOAP relies on application layer protocols for message negotiation and transmission such as HTTP or "Simple Mail Transfer Protocol (SMTP)".

TLS

On this page a brief description of TLS is provided. For the most recent version of the specification click on this link.

Description

Transport Layer Security (TLS) is a cryptographic protocol that describes communication security for computer networks. The first version of TLS 1.0 is built upon and is an upgrade of SSL 3.0.

Differences and similarities between TLS and SSL

Both TLS and SSL provide means for data encryption and authentication between applications, machines and servers when data is sent through insecure network.

The differences between TLS and its forerunner "Secure Sockets Layer" (SSL) are the addressed vulnerabilities. TLS for instance works with

- a wider variety of hash functions.
- more secure and stronger cipher suites, such as the Advanced Encryption Standard (AES) cipher suits which are integrated into TLS version 1.1.
- browser security warnings. TLS has more alert descriptions than SSL.

TLS versions

TLS 1.0: upgrade of version SSL 3.0. The differences between TLS 1.0 and SSL 3.0 are not big, but significant enough to exclude interoperability between TLS 1.0 and SSL 3.0. Version TLS 1.0 does include a means by which a TLS implementation can downgrade the connection to SSL 3.0.

TLS 1.1: Added protection against cipher-block chaining (CBC) attacks. (CBC = each block of plaintext is XORed with the previous cipher text block before being encrypted), added support for Internet Assigned Numbers Authority (IANA) registration of parameters

TLS 1.2: improved hash functions (MD5-SHA-1), improvement in the client's and server's ability to specify which hash and signature algorithms they accept, expansion of support for authenticated encryption ciphers, added TLS Extensions definition and Advanced Encryption Standard cipher suites

TLS 1.3: removing support for some hash functions (MD5 and SHA-224), requiring digital signatures even when a previous configuration is used, integrating use of session hash

UMA

On this page a brief description of UMA is provided. For the most recent version of the specification click on this link.

Description

UMA is short for User-managed Access and is an OAuth-based access management protocol standard.

Its purpose is to “enable a resource owner to control the authorization of data sharing and other protected-resource access made between online services on the owner’s behalf or with the owner’s authorization by an autonomous requesting party”.

UMA in relation to other standards & specifications

UMA does not depend or have to use the OpenID protocols (most recent version is OpenID Connect) to identify users or (optionally) collect identity claims from a requesting party (for access policy checks).

In the same fashion, UMA does not depend or have to use XACML as policy language (to write access policies and rules) and validate authorization requests based on the policies and rules.

UMA has no restrictions regarding the policy format, as the Authorization Server is in charge and in control of the policy evaluation.

The UMA and XACML flows for requesting access have common features.

X.509

On this page a brief description of X.509 is provided. For the most recent version of the specification click on this link.

Description

X.509 is a cryptographic standard for public key infrastructures (PKI’s) that specifies the management of digital certificates and public-key encryption and keys of the Transport Layer Security (TLS) protocol that is used to secure web and email communication.

Apart from that, it also specifies the formats for public key certificates, certificate revocation lists (CRL’s), attribute certificates, and a certification path validation algorithm.

It assumes a strict hierarchical system of certificate authorities for issuing the certificates. Unlike web of trust models (i.e. encryption method "Pretty Good Privacy (PGP)") where anyone (not just special certificate authorities) may sign and thus verify the validity of others' key certificates.

Structure of X.509 certificates

The structure of X.509 digital certificates is expressed in a formal language: Abstract Syntax Notation One (ASN.1) which is a standard and notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking

The content of a digital certificate is structured and divided into fields. The fields of a X.509 digital certificate are listed hereunder:

- Certificate
- Version Number
- Serial Number: Used to uniquely identify the certificate
- Signature Algorithm ID: The algorithm used to create the signature ID.
- Issuer Name: Name of the entity that verified the information and issued the certificate
- Validity period
    - Not Before
    - Not After
- Subject name: Name of the person, or entity identified
- Subject Public Key Info
- Public Key Algorithm
- Subject Public Key
    - Issuer Unique Identifier (optional)
    - Subject Unique Identifier (optional)
- Extensions (optional)
- Certificate Signature Algorithm: The algorithm used to create the certificate signature
- Certificate Signature: The actual certificate signature to verify that it came from the issuer

Each extension (additional field) has its own ID, expressed as object identifier, which is a set of values, together with either a critical or non-critical indication. If the critical value cannot be recognised or processed, the certificate is rejected. Non-critical values may be ignored if not recognised, but must be processed if recognised.

Types of extensions

- Information about a specific usage of a certificate
- Certificate filename extensions

XACML

On this page a brief description of XACML is provided. For the most recent version of the specification click on this link.

Description

XACML (eXtensible Access Control Markup Language) is an XML-based specification that is designed to control access to applications. One of the main advantages of this specification is that applications and systems with their own and different authorization structure can be integrated into one authorization scheme. Authorizations and the rules surrounding it can be managed centrally regardless of authorization mechanism of the applications themselves. This phenomenon is called externalization. XACML is derived from SAML and provides the underlying specification for ABAC (Attribute-Based Access Control). XACML is also suitable to be used in combination with RBAC (Role-Based Access Control).

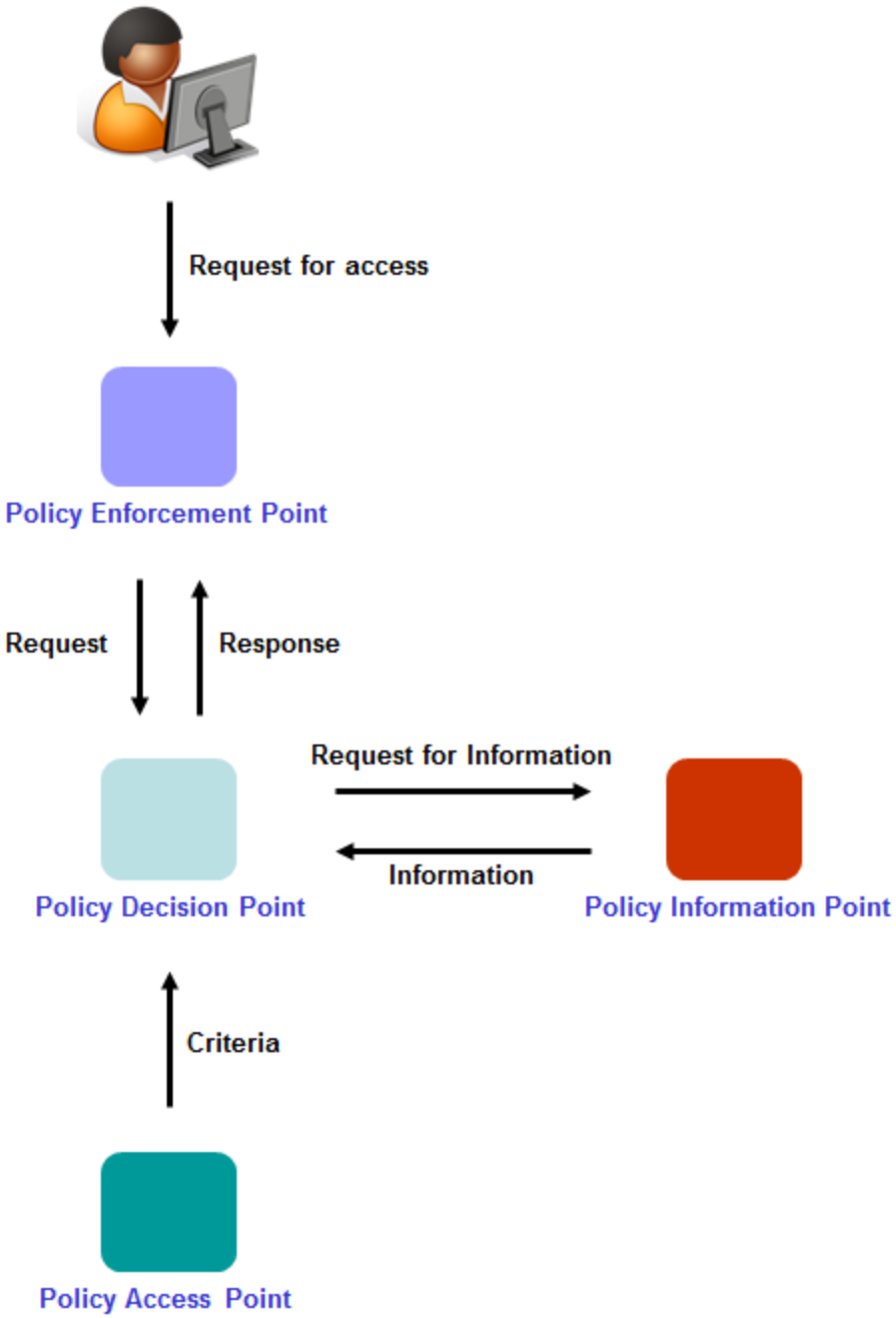
Moreover, with the help of XACML authorizations can be arranged and managed in detail. This is called fine-grained authorization. XACML supports the use of security labels, rules with arbitrary attributes, rules with a certain duration and dynamic rules.

In XACML two main functions can be distinguished. One function defines the criteria with which authorizations are assigned, such as 'only an experienced user from department X is allowd to modify documents'. The other function compares the criteria with the rules or policies to determine whether a person is allowed to perform the operation on the object or not.

The architecture of XACML is fairly complex. This is partly due to the fact that it is difficult to fit the various components of XACML in the application landscape. These components should be positioned in such a way that the owner of the data can somehow control the authorizations to his or her data, but at the same time the components should be positioned in such a way that the performance is not negatively influenced. This is extra important when independent parties need to cooperate with each other and want to jointly organize the access to their applications. Finally, applications need to be compatible with XACML.

Roles and interactions in XACML

The following figure shows the involved roles Policy Enforcement Point (PEP), Policy Decision Point (PDP), Policy Access Point (PAP) and Policy Information Point (PIP) in XACML and how they are interacting in order to process the user's request for access.



XML

On this page a brief description of XML is provided. For the most recent version of the specification click on this link.

Description

XML is short for "eXtensible Markup Language" to encode text documents in a format that is both human- and machine-readable.

XML Signature

On this page a brief description of XML Signature is provided. For the most recent version of the specification click on this link.

Description

XML signature is a standard for authentication and message integrity that defines an XML syntax for digital signatures to sign primarily XML documents.

It is used within i.e. SOAP & SAML.