



iSHARE

Version 1.0

Exported on 06/23/2017

Table of Contents

| | | |
|----------|--|------------|
| 1 | Table of Contents | 2 |
| 2 | Introduction | 5 |
| 2.1 | Goals and scope of the iSHARE scheme | 6 |
| 2.2 | Key features | 7 |
| 2.3 | Guiding principles | 9 |
| 2.4 | Assumptions | 13 |
| 2.5 | Roles & Responsibilities | 13 |
| 2.6 | Conventions & Versioning | 21 |
| 3 | Functional | 23 |
| 3.1 | Interaction models | 23 |
| 3.2 | Primary use cases | 24 |
| 3.3 | Secondary use cases | 50 |
| 3.4 | Functional requirements per role | 52 |
| 3.5 | User interface requirements | 52 |
| 4 | Technical | 54 |
| 4.1 | Interface specifications | 54 |
| 4.2 | 'Language' of Delegation and Authorisation | 71 |
| 4.3 | Relevant standards | 102 |
| 5 | Legal | 114 |
| 5.1 | Accession Agreement for Adhering Parties | 115 |
| 5.2 | Accession Agreement for Certified Parties | 116 |
| 5.3 | Terms of Use | 117 |
| 5.4 | Legal Framework | 123 |
| 6 | Operational | 126 |

| | |
|--|------------|
| 6.1 Governing body..... | 126 |
| 6.2 Audits..... | 126 |
| 6.3 Incident Management..... | 127 |
| 6.4 Change Management..... | 127 |
| 6.5 Service Level Agreements..... | 127 |
| 7 Glossary & Legal Notices..... | 129 |
| 7.1 Glossary | 129 |
| 7.2 Legal notices | 139 |

iSHARE 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 focuses on topics of authentication, authorisation 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.

Introduction

This document provides a full overview of the iSHARE scheme.

iSHARE 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 functional scope of the iSHARE scheme focuses on topics of authentication, authorisation and identification. In January 2018, the iSHARE scheme will be ready to open up to the market.

This chapter further describes the history and context of the iSHARE scheme, how the iSHARE scheme is established through co-creation with participating organisations, and what the purpose of this document is. The remainder of this section is dedicated to the [goals and scope of the iSHARE scheme](#), the [key features, guiding principles and assumptions](#) and a description of [roles and responsibilities](#) present within the scheme. For an overview of used terms and their explanation, please consult the [glossary](#).

History and planning

The project to establish the iSHARE scheme was initiated by the Neutral Logistics Information Platform (NLIP), as part of the government programme "[Topsector Logistiek](#)", through a tender project in 2016. NLIP requested 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 won the tender with their plan to set-up a scheme of multilateral agreements instead of, for instance, a technology-centric approach relying on a software platform. Since June 2016, the iSHARE project team facilitated the realisation of a 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, in which organisations were openly invited to participate in the initiative and which resulted in the so called "startdocument v0.1". Startdocument v0.1 provided 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 participating organisations worked collaboratively towards iSHARE scheme v1.0 which contains the first full set of agreements for improved data exchanging conditions. Participating organisations worked in four working groups to produce the first full version of the iSHARE scheme: the Legal, Functional and Technical working groups (the Operational working group was postponed to at the earliest phase 3). Participating organisations realised Proofs of Concept to verify the correct functional and technical workings of the iSHARE scheme;
- 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.

Establishment of the iSHARE scheme through co-creation

The iSHARE scheme is established through collaboration between its participating organisations. By going through a co-creation process, the collective expertise of all participants leads 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 the whole of the iSHARE scheme is constantly scrutinised by its participants and constantly grows towards maturity. What the iSHARE scheme eventually entails or does not 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 main topics within the scheme agreements: **Legal**, **Operational**, **Functional** and **Technical** (LOFT) agreements (presented in FTLO order in the scheme, for readability). The assumption is that for a fully functional scheme, at least these topics need to be discussed and organised;
- The relevant working groups for these four topics 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;
- Working groups have regular meetings facilitated by a chairman and a secretary who registers made agreements.

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

Purpose of this document

The purpose of this document is to provide a complete overview of the current state of the iSHARE scheme. The iSHARE scheme and this document are a "growing document" to which additions and changes are regularly made.

Notes accompanying current version of this document (Version 1.0)

iSHARE scheme version 1.0 contains the first full version of the Functional and Technical specifications, as well as the first versions of required Legal documents. Due to unforeseen budget limitations during phase II, the current version does not include the following aspects:

- Operational guidelines, as the Operational working group did not start during phase 2;
- Limited Legal descriptions and specifications relating to the legal characteristics of the Scheme Owner (due to dependency on the Operational Working group);
- Description and specification of the reference implementation (including certification and testing capabilities).

Goals and scope of the iSHARE scheme

The iSHARE scheme is a collaborative effort to improve the exchange of data between organisations involved with the Dutch logistics sector. The iSHARE scheme will result in a set of agreements based on which improved data exchange can be achieved.

The ambition of the iSHARE project is to lower barriers for 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 data can flow in a controlled and smart way, it 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 crucial in any data exchange context:

1. [Identification](#);
2. [Authentication](#);
3. [Authorisation](#).

iSHARE focuses on these three aspects as they are considered indispensable in any communication between parties, also in the context of exchanging logistical data. Within the iSHARE scheme, agreements are made on the above 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 on what terms.

These three aims can only be reached when a variety of perspectives are 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 invest 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](#).

*Note: iSHARE's scope does not include the specification of possible business models for sharing data and/or payments related to data exchange.

**Note: The iSHARE scheme can in some way be compared with the institute of the passport: the iSHARE scheme will be useable by anyone who owns a digital identity compatible within the iSHARE scheme. This will greatly simplify authentication and authorisation processes, also between different organisations (however: even though organisations can have valid certificates, it does not rule out possible malign intentions).

Key features

Based on the research resulting from Phase 1, the iSHARE scheme should at least support the following key features:

- [Provide flexibility in authorisation](#)
- [Allow for management of consent](#)
- [Support multiple interaction models](#)
- [Provide a PKI trust framework](#)
- [Facilitate the use of federated identity\(s\)](#)

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

Provide flexibility in authorisation

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

- **Flexible authorisation scope**

iSHARE aims to provide a way to add a layer of authorisation to any resource or any selection or combination of resources. The authorisation scope refers to the objects or resources of a specific party, to which authorisations 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 or services). Either way, the scope is always governed by a formal agreement and implemented by technical means. In the current version of the iSHARE scheme, the flexibility of authorisations is captured in the [language for delegation and authorisation](#).

- **Granular authorisations**

iSHARE aims to provide a granular way to use authorisations for resources. The authorisation granularity refers to the characteristics of both the requested resources and the rules (policies, conditions) that apply. Authorisations to resources 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 authorisations 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. In the current version of the iSHARE scheme, the granularity of authorisations is captured in the [language for delegation and authorisation](#). For more information on [granularity of authorisations](#), please consult the [glossary](#).

- **Flexible authorisation source**

iSHARE aims to provide flexibility to where authorisation rules are stored and can be retrieved. The authorisation source refers to the location of the rules (policies, conditions) and the attributes (e.g. subject attributes, object attributes) that govern the authorisations. These can be located near the data, at a dedicated source, or a combination thereof. In the current version of the iSHARE scheme, the flexibility in authorisation source is described as "Policy Information Point" or PIP under the [Primary Use cases](#).

How the iSHARE scheme develops the flexibility in authorisation will be dependent on factors like data ownership, formal agreements, communication and security.

Allow for management of consent

For appropriate recognition of authorisations 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 aims to support multiple interaction models. Within the current version of the iSHARE scheme, the "Human to machine (H2M)" and "Machine to Machine (M2M)" interaction models are foreseen. Both these models can be characterised as request-and-response models. For more information on the current use of these interaction models, please refer to the functional descriptions of the [interaction models](#).

Depending on utility and future growth, other interaction models like "Peer to Peer (P2P)" and "Publish and Subscribe" might be added.

Provide a PKI trust framework

The iSHARE scheme relies on public key encryption for several core processes, amongst which the following:

- Proof of origin of data;
- Proof of authenticity of identities;
- Protection of data against unauthorised access or disclosure.

A [Public Key Infrastructure](#) (PKI) is required, in order to:

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

iSHARE aims to provide a list of certificate roots (also called PKI roots), or [Certificate Authorities](#), that meet the iSHARE requirements. These Certificate Authorities can be (and must be) trusted by all iSHARE participants for the registration and issuance of digital certificates. iSHARE will at least trust the certificate authorities/service providers listed by the EU under EIDAS regulation (for a list of trusted service providers, [click here](#). More information on EIDAS can be found [here](#)).

Facilitate the use of federated identity(s)

iSHARE aims to facilitate (but not impose) the use of one or more federated identity(s). A federated identity is an identity that is spread out and recognised across multiple, independent systems.

Within iSHARE, the use of federated identities would reduce costs by eliminating the need for proprietary, or newly issued identity solutions. In order for an identity to become part of iSHARE's federation, the identity provider must be certified under the iSHARE scheme.

Guiding principles

To achieve the goals of the iSHARE scheme, it is paramount to stay close to a set of guiding principles. As time progresses new principles can be defined, existing principles can be adapted or dropped if deemed necessary. The guiding principles were defined using the format as suggested* by [TOGAF 8.1.1 architectural principles \(external link\)](#).

The following principles define the iSHARE scheme and must be kept in mind at all times during further development (see details of guiding principles below):

| Principle # | Principle name |
|--------------------|---|
| 1 | Generic building block to enable data exchange |
| 2 | Limited scope: Identification, authentication & authorisation |
| 3 | Leverage existing (international) building blocks |
| 4 | Agnostic towards nature and content of data |
| 5 | Benefits outweigh investment for all types of participants |
| 6 | International orientation |

Guiding principles details:

| Principle 1 | Generic building block to enable data exchange |
|---------------------|---|
| Statement | iSHARE is a generic identification, authentication and authorisation scheme to be used as enabler for data exchange in logistics |
| Rationale | In every exchange of data, identification, authentication and authorisation are fundamental factors. iSHARE aims to simplify processes of identification, authentication and authorisation as a generic solution to facilitate data exchange in the logistics sector. |
| Implications | <ul style="list-style-type: none"> the iSHARE scheme will allow for extension or adaptability so it can be used in situation/ sector specific cases the iSHARE scheme will not cater to a specific sector or market, it is applicable in an N amount of cases the iSHARE scheme will not be a point solution |

| Principle 2 | Limited scope: Identification, authentication & authorisation |
|--------------------|---|
| Statement | The iSHARE scheme's scope is limited to topics of identification, authentication and authorisation in the context of data exchange |
| Rationale | iSHARE aims to improve the circumstances for data exchange throughout the logistics sector and provides focus on the topic of identification, authentication and authorisation. Identification, authentication and authorisation are a fundamental part of any data exchange, but are not solved in a scalable or standardised way at the moment. |

| | |
|---------------------|--|
| Implications | <ul style="list-style-type: none"> Without this principle, there is a risk of "scope creep": related topics could take away the focus off the intended topics |
|---------------------|--|

| | |
|---------------------|--|
| Principle 3 | Leverage existing (international) building blocks |
| Statement | Where possible, iSHARE should be realised using existing and proven standards, technology or initiatives |
| Rationale | By reusing building blocks already available and in use, the impact on organisations to participate in iSHARE and the time to realise the iSHARE scheme are lowered. Standards, technology and initiatives preferably have a broad (international) usage base and are backed by a professional organisation charged with maintenance of the standards, technology or initiatives. |
| Implications | <ul style="list-style-type: none"> the iSHARE scheme will build on or use existing (international) standards, technology or initiatives where possible the iSHARE scheme will aim to use open standards, technology or initiatives the iSHARE scheme may use proprietary standards, technology or initiatives if existing and/or proven standards, technology or initiatives do not provide what is needed, alternative solutions will be sought |

| | |
|---------------------|---|
| Principle 4 | Agnostic towards nature and content of data |
| Statement | The iSHARE scheme does not concern itself with the contents or nature of data |
| Rationale | Given the generic nature of the iSHARE scheme and the aim to be applicable throughout the logistics sector, iSHARE needs to function with any type of possible data and/or any relevant data exchange interaction model. To this end, the contents of data are only considered where it concerns the facilities needed within iSHARE to adequately exchange various types of data (e.g. requirements to security, encryption, etc.). It is up to the participating organisations to ensure that iSHARE adequately fulfills requirements to the process of identification, authentication and authorisation in the context of data exchange. |
| Implications | <ul style="list-style-type: none"> the iSHARE scheme will not specify the (allowed) content of data exchanges done within an iSHARE context the iSHARE scheme does not specify content specific data standards the iSHARE scheme should not have limitations connected to types of data or standards used |

| | |
|---------------------|---|
| Principle 5 | Benefits outweigh investment for all types of participants |
| Statement | The iSHARE scheme needs to be attractive to use and implement for all types of participants/roles. |
| Rationale | The iSHARE scheme knows different roles with different responsibilities. When a potential participant considers taking a (or multiple) role(s) in the iSHARE scheme, the iSHARE scheme should aim to have the lowest possible threshold to participate for the potential participant. Depending on what the character of the potential participant is (e.g smaller size or larger size organisations) and which role the participant wants to take, this could mean that the impact of implementation needs to be small or that the implementation is kept relatively simple. |
| Implications | <ul style="list-style-type: none"> the iSHARE scheme aims to keep thresholds to participate in the iSHARE scheme (e.g. in terms of implementation impact or onboarding/certification effort) as low as possible for all possible roles the iSHARE scheme strives for the lowest possible impact for participants when changes occur in the future. Changes to used standards will take place; within the iSHARE scheme and its specifications thought needs to be given to how change is dealt with in an efficient way. |

| | |
|---------------------|---|
| Principle 6 | International orientation |
| Statement | The iSHARE scheme needs to look over geographic boundaries to foster international involvement and cooperation |
| Rationale | The logistics sector is per definition an international sector. The iSHARE scheme needs to facilitate, to the extent that it is practical and possible, international involvement. |
| Implications | <ul style="list-style-type: none"> the iSHARE scheme needs its participants to provide knowledge and experience on how the iSHARE scheme can stay (and become) attractive in the international context |

*Format used for defining guiding principles, based on TOGAF standard:

| | |
|-----------------------|---|
| Principle name | Should both represent the essence of the rule as well as be easy to remember. Specific technology platforms should not be mentioned in the name or statement of a principle. Avoid ambiguous words in the Name and in the Statement such as: "support", "open", "consider", and for lack of good measure the word "avoid", itself, be careful with "manage(ment)", and look for unnecessary adjectives and adverbs (fluff). |
| Statement | Should succinctly and unambiguously communicate the fundamental rule. For the most part, the principles statements for managing information are similar from one organisation to the next. It is vital that the principles statement be unambiguous. |

| | |
|---------------------|---|
| Rationale | Should highlight the business benefits of adhering to the principle, using business terminology. Point to the similarity of information and technology principles to the principles governing business operations. Also describe the relationship to other principles, and the intentions regarding a balanced interpretation. Describe situations where one principle would be given precedence or carry more weight than another for making a decision. |
| Implications | Should highlight the requirements, both for the business and IT, for carrying out the principle - in terms of resources, costs, and activities/tasks. It will often be apparent that current systems, standards, or practices would be incongruent with the principle upon adoption. The impact to the business and consequences of adopting a principle should be clearly stated. The reader should readily discern the answer to: "How does this affect me?" It is important not to oversimplify, trivialise, or judge the merit of the impact. Some of the implications will be identified as potential impacts only, and may be speculative rather than fully analysed. |

Assumptions

The iSHARE scheme was developed with the following assumptions in mind:

1. **Conditions for the exchange of data - or calls upon services - are assumed to be established**
The iSHARE scheme needs to rely upon the responsibility of participants to know what rights they have to what data and/or services. iSHARE is meant as an instrument to exchange data or call upon services in a uniform, controlled and straightforward way; it is not meant as a means to resolve questions of data ownership. In practice this means that for instance a Service Provider bears responsibility to sufficiently establish if a service consumer is authorised to receive certain data or call upon certain services.
2. **Data formats and semantics are assumed to be in place**
In order to be able to exchange data, a mutual understanding of the meaning of data and the way data is structured is required. Within the iSHARE scheme it is assumed that this mutual understanding exists and thus the exchange of data between involved parties is possible (also see [guiding principle 4](#): "Agnostic towards nature and content of data")
3. **Data and service classification has taken place**
It is assumed that within the iSHARE scheme, participants have sufficiently identified and classified their data and services. Data owners are responsible for the classification of their data and services, the iSHARE scheme does not prescribe its participants how to classify their resources (See "[Data Classification](#)" in the [glossary](#) for an explanation of Data Classification)

Roles & Responsibilities

This section describes the iSHARE basic framework, the functional roles within the iSHARE basic framework, and the general responsibilities of the functional roles. A more detailed explanation of each role's functional behaviour, and interaction between roles is described in the section containing the [Functional](#) descriptions.

The section is presented as follows:

- [Basic framework of service provision](#)

- Adherence, certification and compatibility
 - Entitled Party
 - Service Consuming Entity
 - Service Consumer
 - Human Service Consumer
 - Service Provider
 - Authorisation Registry
 - Identity Provider
 - Identity Broker
 - Service Broker (obsolete)
- Scheme Owner

Basic framework of service provision

Note that the term "data exchange" was deemed too narrow for the scope of iSHARE, therefore a wider term was introduced: "service provision".

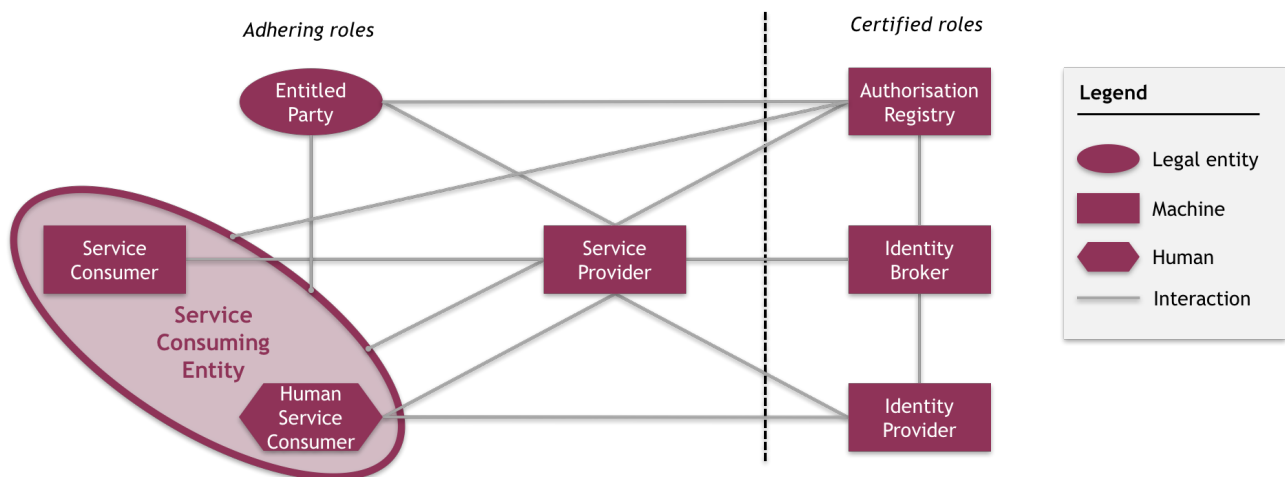
Introduction

This section presents an overview of all functional roles present within the iSHARE basic framework, as well as an overview of basic situations that might occur. First the basic framework is explained in full, after which two generic situations are explained. For an in depth description of the named functional roles, please refer to the section on [adherence, certification and compatibility](#).

iSHARE basic framework

iSHARE aims to provide a generic building block widely applicable in the logistics sector. This requires a framework which can be applied to the wide variety of cases possible in practice. This section explains this framework and its roles, step-by-step.

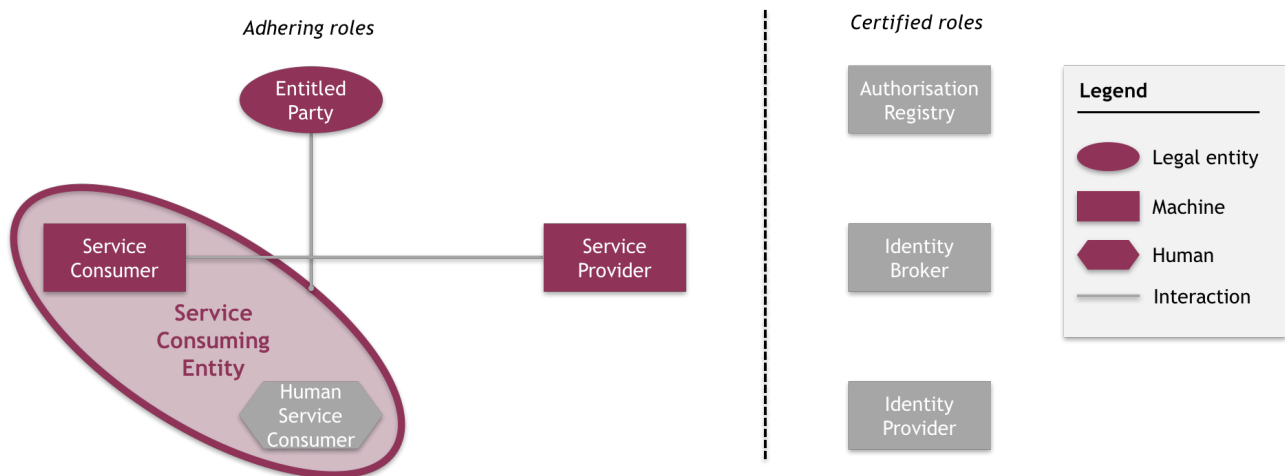
The iSHARE basic framework consists of eight functional roles that, depending on the situation, interact with each other based on the iSHARE scheme rules (see image below). Every role has a certain function in the overall scheme and bears certain responsibilities, as described later. In principle, the basic framework can be applied throughout the logistics sector. Do note that an organisation can fulfill several roles within the scheme, depending on the context and what service is consumed/provided.



The depicted difference between adhering- and certified roles is described [here](#).

In the least complex of situations, a Service Consumer (machine) or a Human Service Consumer interacts with a Service Provider, as depicted in the example below.

Example: Service Consumer interacts with Service Provider



The **Entitled Party** is the legal entity that requires the right(s) to service(s) provided by a **Service Provider**. In this case, the Entitled Party arranges (in a legal agreement with a Service Provider) that it has the right to consume its services.

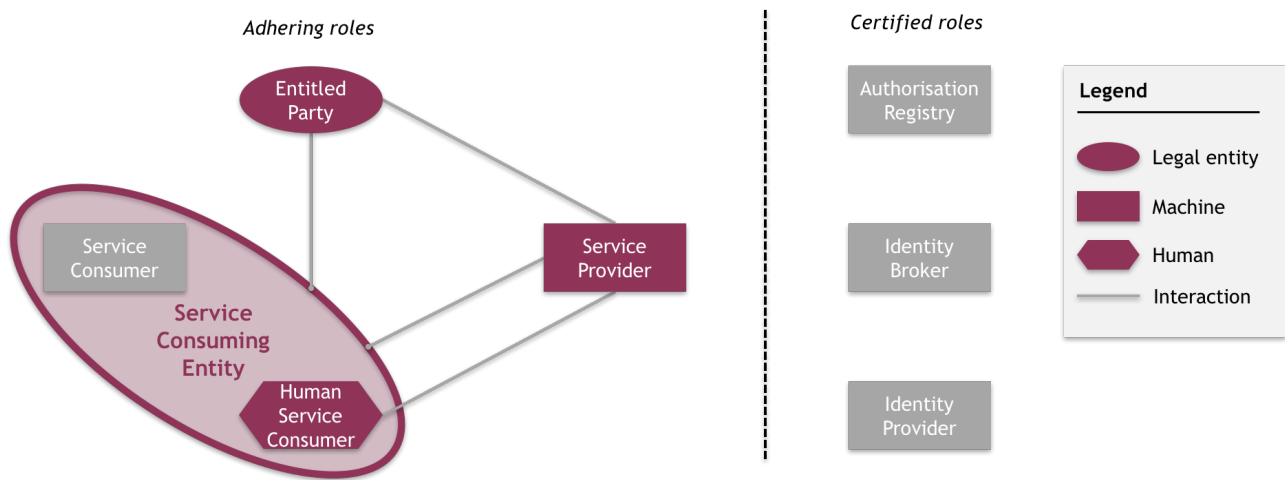
The **Service Consumer** represents a machine that requests, receives, and uses certain services, from a Service Provider. The Service Consumer is managed by the so called **Service Consuming Entity**, the legal entity that is responsible for the machine that consumes a service. A Service Consuming Entity is delegated by the Entitled Party to consume services on the Entitled Party's behalf. This means that the Service Consuming Entity can authorise its Service Consuming machine to consume services based on the delegated rights. In the least complex of situations, however, no such delegation takes place: in such situations the Entitled Party is also the Service Consuming Entity

and manages Service Consuming machines itself: in other words, the Entitled Party is also the Service Consuming Entity. This is the case in the example above.

The Service Provider represents a machine that provides certain services, such as data, to (a) (Human) Service Consumer(s). In this case, the services of the Service Provider are consumed by the Service Consuming machine of the Service Consuming Entity that is also the Entitled Party.

The above example is already very similar to primary [use case 1](#) that is detailed under [Functional](#). In the following example, a Human Service Consumer is involved instead of a Service Consumer.

Example: Human Service Consumer interacts with Service Provider



A **Human Service Consumer** represents a human (person) who requests, receives, and uses certain services, from a Service Provider. The Human Service Consumer works for the Service Consuming Entity.

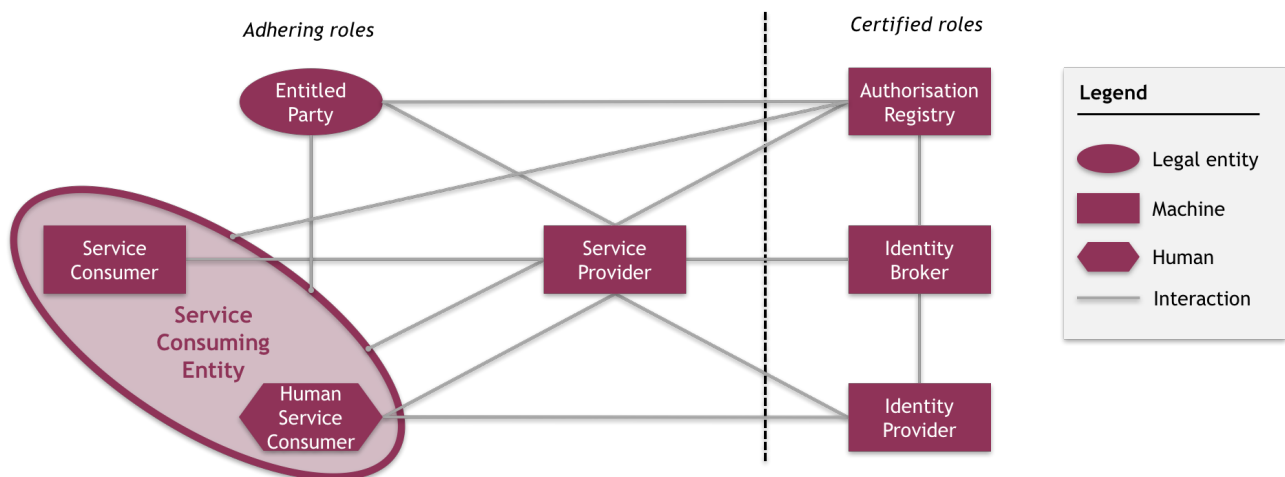
In this case, the Entitled Party is also the Service Consuming Entity (so no delegation is required). The Service Consuming Entity registers which Human Service Consumers are authorised to consume services at the Service Provider. On the basis of this authorisation, the Service Provider will provide the requested service(s) to the Human Service Consumer upon request.

This example is very similar to primary [use case 2](#) as also detailed under [Functional](#).

Note that the roles Service Consuming Entity and Service Provider are not fixed to particular entities. In other words, a Service Provider may be a Service Consuming Entity in another context of service provision. Likewise, depending on the context, the concepts of data ownership, responsibility and accountability can take different forms.

Adherence, certification and compatibility

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.



A party fulfilling a role in the iSHARE basic framework MUST be iSHARE adhering or iSHARE certified – as required for that role. Note that a party can fulfil more than one role.

The iSHARE roles depicted on the left side of the framework are roles for which parties MUST adhere to the iSHARE scheme. An **iSHARE adhering party** adheres to the [iSHARE terms of use](#). An iSHARE adhering party MUST sign an adherence agreement with the [Scheme Owner](#).

The iSHARE roles depicted on the right side of the framework are roles for which parties MUST be certified within the iSHARE scheme. Roles for which certification is required facilitate certain functions for the iSHARE scheme that every party within iSHARE must be able to rely upon. An **iSHARE certified party** MUST apply to the [Scheme Owner](#) for certification and, after providing sufficient proof, MUST sign a certification agreement with the [Scheme Owner](#). While the exact bases for certification need to be determined, the eHerkenning responsibilities and requirements per role serve as a starting point. In this way the [eHerkenning admission process](#) is completely reused. eHerkenning certified parties would be asked to fulfil only some extra responsibilities and requirements to also become iSHARE certified.

Next to iSHARE adherence and certification, it is considered to be beneficial to the iSHARE scheme to add the possibility of products and/or processes being assessed as **iSHARE compatible**. iSHARE compatible products and/or processes comply to the iSHARE agreements and standards, from a functional and technical perspective. A conformity test will be developed at a later stage to affirm iSHARE compatibility.

Entitled Party

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The **Entitled Party** is the legal entity that has one or more rights to something, e.g. to data at a Service Provider that it has a legal agreement with. The Entitled Party is either the same entity as the [Service Consuming Entity](#), or delegates its rights to another Service Consuming Entity. In the latter case, this other Service Consuming Entity can consume services on the Entitled Party's behalf.

The Entitled Party is a role for which iSHARE [adherence](#) is REQUIRED.

Service Consuming Entity

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The **Service Consuming Entity** is the legal entity that consumes the Service Provider's service on the basis of the **Entitled Party's** rights to that service. It can do so because the Service Consuming Entity is either the same legal entity as the Entitled Party (i.e. it already has these rights), or because the Entitled Party has delegated rights to the Service Consuming Entity.

The Service Consuming Entity does not interact with the Service Provider; it authorises (and uses) a **Service Consumer** or **Human Service Consumer** to do so.

The Service Consuming Entity is a role for which iSHARE **adherence** is REQUIRED.

Service Consumer

The **Service Consumer** is a role that represents a machine that requests, receives, and uses certain services, such as data, from a Service Provider on behalf of and authorised by the **Service Consuming Entity**.

The Service Consumer is not a separate role, but it belongs to the adhering party Service Consuming Entity.

Human Service Consumer

The **Human Service Consumer** is a role that represents a human (person) who requests, receives, and uses certain services, such as data, from a Service Provider on behalf of and authorised by the **Service Consuming Entity**.

The Human Service Consumer is not a separate role, but belongs to the adhering party Service Consuming Entity.

Service Provider

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The **Service Provider** is a role that provides certain services, such as data, to a Service Consuming Entity. In case the service pertains to data provisioning, the Service Provider is either the Data Owner, or has explicit consent of the Data Owner to provide the services.

The Service Provider is **responsible** for the availability of services, and **accountable** for these services if it also the **Data Owner**.

The Service Provider is a role for which iSHARE **adherence** is REQUIRED.

Authorisation Registry

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The **Authorisation Registry**:

- Manages records of delegations and authorisations of Entitled Parties and/or Service Consuming Entities;
- Checks on the basis of the registered permission(s) whether a (Human) Service Consumer is authorised to take delivery of the requested service, and;
- Confirms the established powers towards the Service Provider.

Within the iSHARE scheme, the term Authorisation Registry always refers to an external Authorisation Registry (not part of the Service Provider or Entitled Party).

The Authorisation Registry is a role for which iSHARE [certification](#) is REQUIRED. This certification builds on the eHerkenning certification for '[Machtigingenregister](#)'.

Identity Provider

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The **Identity Provider**:

- Provides identifiers for Human Service Consumers;
- Issues credentials to Human Service Consumers;
- Asserts to the system that such an identifier presented by a user is known to the Identity Provider, and;
- Possibly provides other information (which are frequently referred to as attributes) about the user that is known to the Identity Provider.

In the iSHARE environment an Identity Provider could support various methods of authentication, such as:

- Password authentication;
- Hardware-based authentication (smartcard, token);
- Biometric authentication;
- Attribute-based authentication.

The Identity Provider is a role for which iSHARE [certification](#) is REQUIRED. This certification builds on the eHerkenning certifications for both '[Middelenuitgever](#)' and '[Authenticatiedienst](#)'.

Identity Broker

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

If multiple distinct [Service Providers](#) exist where each data set is protected under a distinct trust domain, multiple [Identity Providers](#) may be needed. Moreover, the iSHARE scheme may require different levels of certainty for specific data and may wish to designate specific Identity Providers for specific services.

In order to support multiple Identity Providers (with possible multiple rules) and Service Providers, an **Identity Broker** is required. An Identity Broker allows [Human Service Consumers](#) to select the Identity Provider they prefer to authenticate themselves at. It prevents the need for a direct relationship between all Service Providers and all Identity Providers.

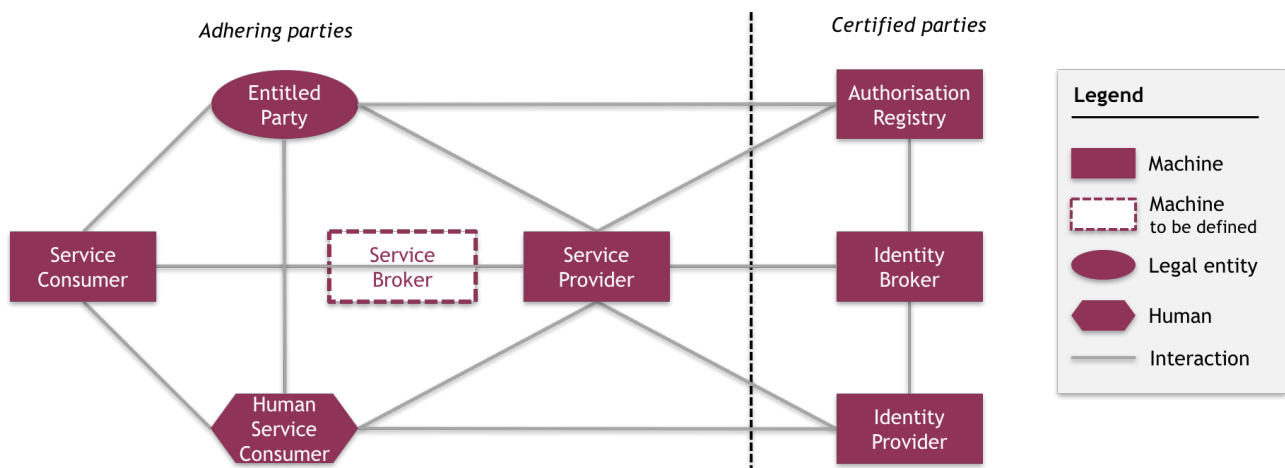
The Identity Broker is a role for which iSHARE [certification](#) is REQUIRED. This certification builds on the eHerkenning certification for '[Herkenningmakelaar](#)'.

Service Broker (obsolete)

The Service Broker-role was deemed unnecessary for the proper functioning of the iSHARE scheme by the Functional working group. This page explains what the role of Service Broker would entail and why it was deemed unnecessary. Note that the original basic framework included the Service Broker-role - with the disclaimer that it was yet to be defined.

A **Service Broker** is an abstract role that represents a machine that brokers interaction between a Service Consumer and (a) Service Provider(s). After a Service Brokers receives a request for certain services from a Service Consumer, the Service Broker requests (the input for) these services from a relevant Service Provider. As a result, the Service Broker connects Service Consumers to relevant Service Providers.

Service Broker deemed unnecessary: Service Provider role provides sufficient functional freedom for Service Brokering



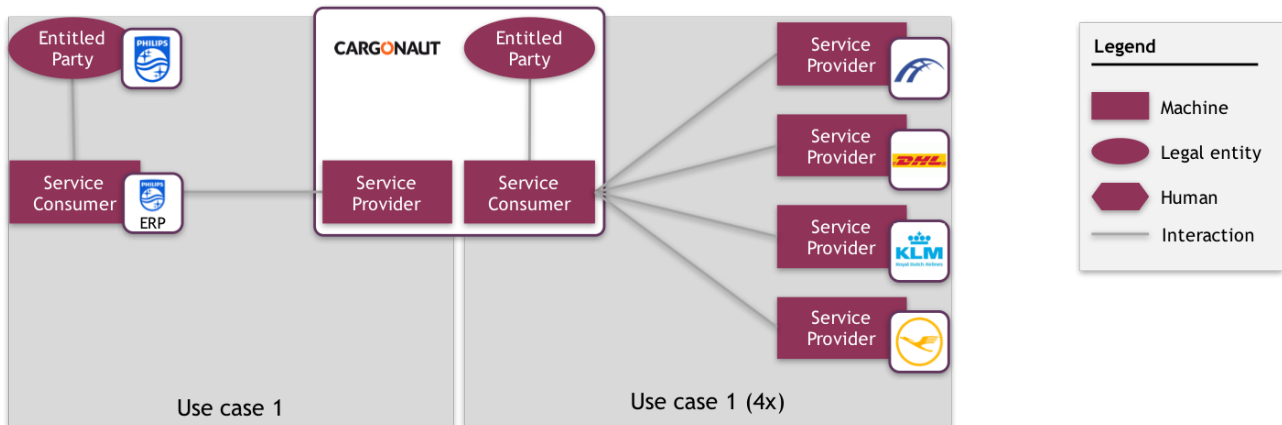
The extra Service Broker-role was deemed unnecessary as its functionality and its use cases are covered within the role of Service Provider. A Service Broker is simply a party that acts as a Service Provider in one use case and as a Service Consumer in a subsequent use case(s).

Example case of service brokering within the scope of a Service Provider

Consider the following example. At any given moment, Philips wants to see where its shipments are in the world and what their status is. Philips is a customer of Cargonaut and uses the Cargonaut platform to gain insight into where its shipments are. There are several companies that transport shipments for Philips, in this case AirBridgeCargo, DHL, KLM and Lufthansa. The Cargonaut platform interfaces with the transporting companies and retrieves information relevant to Philips shipments. Cargonaut then reports back to Philips on what the status of their various shipments is.

Cargonaut brokers services for Philips, but its responsibilities are no different than those of a Service Provider (in its interaction with Philips), and a Service Consumer (in its interaction with the transportation companies): the interaction between Philips and Cargonaut is characterised as a Service Consumer (Philips) communicating with a

Service Provider (Cargonaut) as in [use case 1](#), while the interaction between Cargonaut and the transportation companies is characterised as a Service Consumer (Cargonaut) communicating with several Service Providers (AirBridgeCargo, DHL, KLM and Lufthansa), i.e. 4 instances of use case 1 - as depicted below:



Adding the abstract Service Broker-role to the basic framework, while both its functionality and responsibilities are already covered within existing scheme roles, was deemed unnecessary.

Scheme Owner

The **Scheme Owner** represents the body that governs the iSHARE scheme and its participants. It is to be specified by the Operational working group, which will also drafts the body's exact mandate in coordination with the Legal working group.

The Scheme Owner is not represented in the [basic framework of service provision](#) because it is not a [certified or adhering](#) role itself, and because it has no interaction with other parties in the [primary use cases](#). As part of the [secondary use cases](#), however, parties will need to register themselves as certified or adhering at the Scheme Owner. They will also need to consult the Scheme Owner to check whether their counterparty is adherent or certified, and whether a counterparty's certificate is valid.

Conventions & Versioning

This section includes notational conventions and notes on versioning. The section is presented as follows:

- [Notational conventions](#)
- [Versioning](#)

Notational conventions

Within the iSHARE scheme documentation, the following notational conventions apply:

- 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>).

Please note: Other conventions can be added in due time.

Versioning

Unique version numbers will be assigned to unique states of the iSHARE scheme. For a full overview of previous versions of the iSHARE scheme documentation, please consult the version history [on Confluence](#).

Functional

This section covers the functionality that is in scope of the the iSHARE scheme.

It starts by explaining the two [interaction models](#) that are at the basis of iSHARE's primary use cases: Machine to Machine (M2M) and Human to Machine (H2M). This is followed by the [three primary use cases](#), which form the core of the Functional section:

1. Machine to Machine service provision;
2. Human to Machine service provision with authorisation and identity info held at the Service Provider;
3. Human to Machine service provision with identity info held at the Identity Provider.

These primary use cases have several derived use cases, most of which are explained in detail.

The [secondary use cases](#) that follow the primary use cases include processes related to registration, and processes that recur in primary use cases. The section is then concluded by functional requirements - those [per role in the scheme](#) and those to the iSHARE [user interface](#) in H2M use cases.

Interaction models

At this moment in time, two interaction models are at the basis of iSHARE's primary use cases: Machine to Machine (M2M) and Human to Machine (H2M). This chapter explains both models.

Machine to Machine

Sometimes called Server to Server, **Machine to Machine** interaction is the automated exchange of data and the performance of actions between electronic devices without requiring the assistance of humans. In some M2M applications, electronic devices exchange their data with a central control unit or app(lication), which processes the data for humans.

To exchange (send and receive) data (in the form of electronic signals), a communication network or channel is required such as a telecommunication network, the internet (Wifi, 3/4G), radio-frequency identification (RFID) or Bluetooth.

Human to Machine (H2M)

Human to machine interaction is data transmission between a human (user) and an electronic 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. This interface can include software (i.e. what is visible to the human on the computer monitor) and hardware (i.e. the mouse, keyboard and other devices).

Depending on the interaction model and which roles hold information, three [primary use cases](#) have been defined.

Note that all of these use cases are based on a **request-response** interaction model - in which the (Human) Service Consumer requests a certain service, and a Service Provider responds. In line with iSHARE's [key features](#), the **publish-subscribe** interaction model - in which the (Human) Consumer subscribes to a service that is

(repeatedly) published by the Service Provider - remains in scope. No high-priority practical use cases of this interaction model have been identified yet, however.

Primary use cases

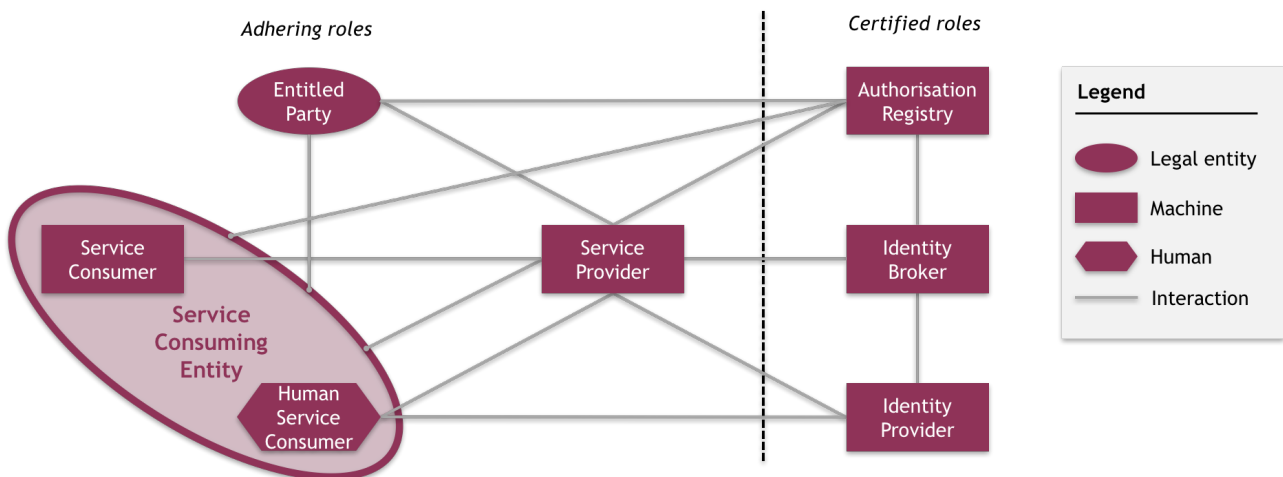
iSHARE knows three primary use cases that form the functional core of the scheme. This most important part of the Functional section explains the following:

- The iSHARE basic framework: its goal, roles, relations and types of information;
- The three primary use cases: Machine to Machine, Human to Machine with authorisation info and identity info held at the Service Provider, Human to Machine with identity info held at an Identity Provider;
- The possible variations to the three primary use cases, depending on where identity information, authorisation information or delegation information is held.

iSHARE basic framework

iSHARE aims to provide a generic building block widely applicable in the logistics sector. This requires a framework which can be applied to the wide variety of cases possible in practice.

The iSHARE basic framework consists of eight functional roles that, depending on the situation, interact with each other based on the iSHARE scheme rules (see image below). Every role has a certain function in the overall scheme and bears certain responsibilities (see [Roles & Responsibilities](#)). In principle, the basic framework can be applied throughout the logistics sector. Do note that an organisation can fulfill several roles within the scheme, depending on the context and what service is consumed/provided.



On the left side of the framework, there are so called adhering roles. These roles consume or provide services and adhere to the iSHARE terms of use. On the right side there are certified roles. These roles facilitate certain functions for the scheme relating to identification, authentication, authorisation and delegation. Parties fulfilling certified roles need to register and certify themselves with the - to be established - iSHARE [Scheme Owner](#). A third type of roles are compatible roles - as further explained [here](#).

In the least complex situations, a (Human) Service Consumer of a Service Consuming Entity interacts with a Service Provider. In the more complex situations, a (Human) Service Consumer interacts with a Service Provider that can only provide its service when it retrieves additional information from certified parties. This might be the case when authorisations for data are stored with a certified Authorisation Registry, or when a Human Service Consumer needs to be identified by an Identity Provider. Depending on the practical context, different roles of the basic framework are called upon.

Within the iSHARE scheme, four types of information are recognised that are needed to facilitate identification, authentication and authorisation:

- **Entitlement info:** information indicating what Entitled Parties are entitled to what (parts of) services;
- **Delegation info:** information indicating which (parts of) an Entitled Party's rights (as registered at the Service Provider or the Authorisation Registry) are delegated to a Service Consuming Entity;
- **Authorisation info:** information indicating which (Human) Service Consumers are authorised to act on a Service Consuming Entity's behalf;
- **Identity info:** information about a Human Service Consumer's identity (only applicable in H2M use cases).

Depending on the specific situation, the required types of information, and which party can provide the required information, specific use cases can be derived. Within iSHARE, three primary use cases are recognised and 21 use cases derived from the primary use cases.

Three primary use cases

Two interaction models are recognised within iSHARE: Machine to Machine (M2M) and Human to Machine (H2M). Depending on the interaction model and which roles hold the information summed up in the above, three primary use cases have been defined:

1. Machine to Machine service provision;
Primary use case 1 caters to all Machine to Machine cases
2. Human to Machine service provision with authorisation and identity info held at the Service Provider;
Primary use case 2 caters to all Human to Machine cases where the Service Provider resides over both identity information and authorisation information and does not need to consult other information points
3. Human to Machine service provision with identity info held at the Identity Provider.
Primary use case 3 caters to all Human to Machine cases where identity information is held at the Identity Provider

The primary use cases all know a variety of derived use cases. Derived use cases are variations of the primary use cases in which information required by the Service Provider is held by and retrieved from different parties. We call the party holding delegation- and/or authorisation information a **Policy Information Point (PIP)**. This PIP, as in XACML, acts as the source of the information. There are different use case variations for different PIPs for delegation- and/or authorisation information, as presented in the use case tables below. Note that entitlement info is always held by the Service Provider which is (consequently) not depicted in the tables below.

The Service Provider requests (from the PIP(s)) and evaluates the information required to decide whether or not to grant a (Human) Service Consumer access to a service. After making its decision based on the received information,

it grants this access (or not) to the (Human) Service Consumer. The Service Provider therefore acts as **Policy Enforcement Point (PEP)** and **Policy Decision Point (PDP)** in all use cases.

Primary use case 1 (and derived use cases)*: M2M service provision

| | Delegation info PIP | | | |
|---------------------|----------------------|------------------|----------------|-------------------|
| | <i>No delegation</i> | Service Provider | Entitled Party | Authorisation Reg |
| Derived use cases** | 1 | 1a | 1b | 1c |

*Use case 1 and its variations can also be initiated by a Human Service Consumer through an app. In such case, the Service Consumer acts as a proxy between the Human Service Consumer and the Service Provider as described [here](#).

**Primary use case 1 assumes that authorisation information is always present in a valid token used by the Service Consumer. Therefore primary use case 1 has no derived use cases where authorisation information is retrieved from other parties.

Note that interaction sequences are not described in the table above. In derived use cases 1b and 1c, several interaction sequences are possible depending on who requests delegation info from the PIP. If the Entitled Party is the delegation info PIP:

1. The Service Provider can request delegation info after a service request from the Service Consumer;
2. The Service Consumer can request delegation info and include it in its service request to the Service Provider;
3. The Entitled Party can push delegation info to the Service Consumer, so it can include it in its service request to the Service Provider.

If the Authorisation Registry is the delegation info PIP:

1. The Service Provider can request delegation info after a service request from the Service Consumer;
2. The Service Consumer can request delegation info and include it in its service request to the Service Provider.

Use case 1 only has one interaction pattern as there is no delegation info PIP. Derived use case 1a also has one interaction pattern as the Service Provider is the Delegation info PIP and therefore already has the delegation info it needs.

Primary use case 2 (and derived use cases): H2M service provision with authorisation info and identity info held at the SP

| | Delegation info PIP | | | |
|--|----------------------|------------------|----------------|-------------------|
| | <i>No delegation</i> | Service Provider | Entitled Party | Authorisation Reg |
| | | | | |

| | | | | | |
|----------------------|------------------|-------------------|----|----|----|
| Auth info PIP | Service Provider | 2 | 2a | 2b | 2c |
|----------------------|------------------|-------------------|----|----|----|

Primary use case 3 (and derived use cases): H2M service provision with identity info held at the IDP

| | | Delegation info PIP | | | |
|----------------------|--------------------|----------------------------|------------------|----------------|----------------------|
| | | <i>No delegation</i> | Service Provider | Entitled Party | Authorisation Reg |
| Auth info PIP | Service Provider | 3 | 3a | 3b | 3c |
| | Entitled Party | 3.1 | 3a.1 | 3b.1 | 3c.1 |
| | Authorisation Reg | 3.2 | 3a.2 | 3b.2 | 3c.2 |
| | Identity Provider* | 3.3 | 3a.3 | 3b.3 | 3c.3 |

*The Identity Provider cannot hold explicit authorisation info, but it can hold info about a Human Service Consumer's identity that implies authorisation - i.e. 'working for truck company X'

Note again that interaction sequences are not described in the tables above. Unlike a Service Consumer, a Human Service Consumer cannot include delegation (or authorisation) info in its service request to the Service Provider. In use cases 2 and 3 (and derived use cases), therefore, the Service Provider will always request delegation- and/or authorisation info from the respective PIP(s) after a service request from the Human Service Consumer.

Several interaction sequences are still theoretically possible depending on who requests a login from the Identity Provider. During the Functional working groups, however, it appeared that in practice, a Human Service Consumer will never request login from an Identity Provider before requesting a service from the Service Provider. Until proven otherwise, therefore, the only interaction sequence in scope for use cases 2 and 3 (and derived use cases) is the one in which the Service Provider (also) requests login from the Identity Provider after a service request from the Human Service Consumer.

In use case 3 (and derived use cases), an [Identity Broker](#) can be introduced to broker the relation between the Service Provider and the Identity Provider(s) and/or the Service Provider and the Authorisation Registry(s). This is optional and useful in situations with several Identity Providers and/or Authorisation Registries. [Use case 3](#) is detailed both without an Identity Broker and with one, while derived use cases [3.2](#) and [3c.2](#) both include an Identity Broker.

For both use case 2 and 3 (and derived use cases), an interface is required. Requirements to this interface are summarised [here](#).

Please note that all use cases that contain a hyperlink (in their respective tables) are detailed on their own Confluence page - as follows:

- Roles;
- Depiction;
- Description;
- Sequence diagram.

1. M2M service provision

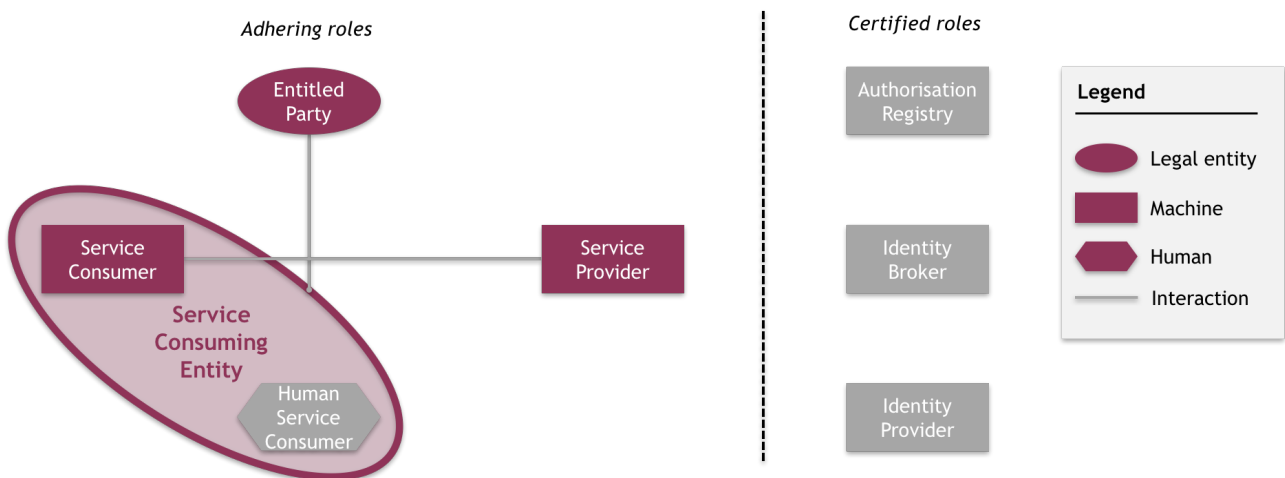
In use case 1, a service is provided by the Service Provider to the Service Consumer.

Roles

| | Delegation info PIP | | | |
|--------------------|----------------------|------------------|----------------|-------------------|
| | <i>No delegation</i> | Service Provider | Entitled Party | Authorisation Reg |
| Use case variation | 1 | 1a | 1b | 1c |

As no delegation takes place, the Entitled Party is also the Service Consuming Entity.

Depiction



Description

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
- The Service Consumer is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Service Consumer.

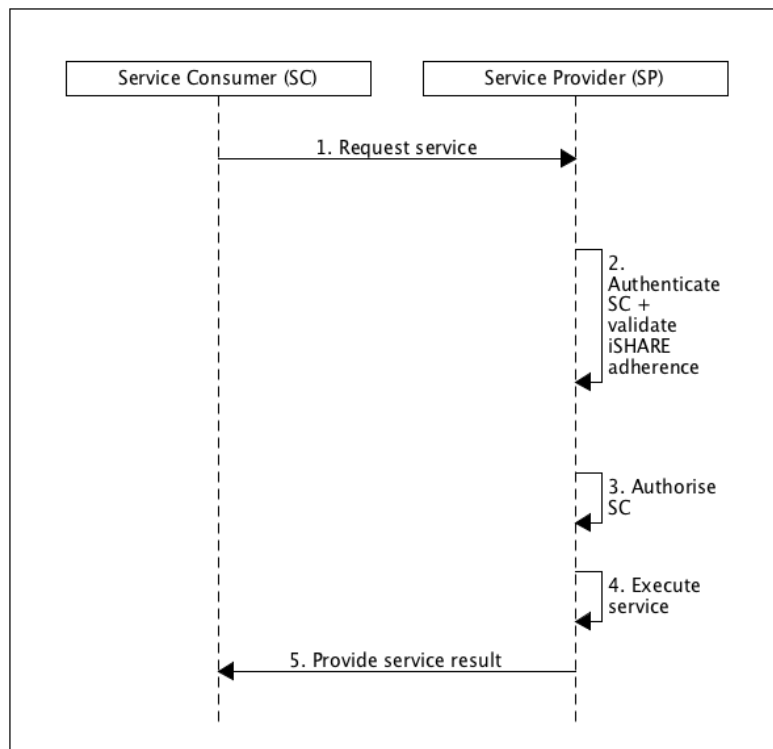
- In this use case the Entitled Party is also the Service Consuming Entity.

*The Service Provider can outsource this function to a third party

The use case consists of the following steps:

1. The Service Consumer requests a service from the Service Provider;
2. The Service Provider authenticates the Service Consumer and validates the iSHARE adherence of the Service Consuming Entity;
3. The Service Provider authorises the Service Consumer of the Service Consuming Entity based on the entitlement information registered with the Service Provider;
4. The Service Provider executes the requested service;
5. The Service Provider provides the service result to the Service Consumer.

Sequence diagram



1b. M2M service provision with the EP as the delegation info PIP

In use case 1b, a service is provided by the Service Provider to the Service Consumer. The Service Consuming Entity has been delegated by the Entitled Party.

Roles

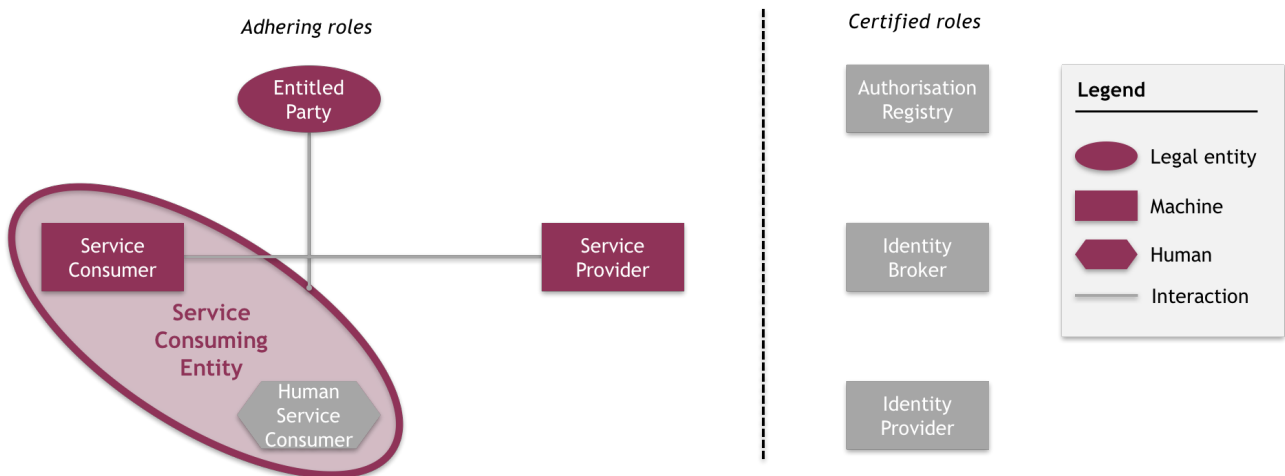
| | Delegation info PIP | | | |
|--------------------|---------------------|------------------|----------------|-------------------|
| | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Use case variation | 1 | 1a | 1b | 1c |

Note that interaction sequences are not described in the table above. In derived use case 1b, three interaction sequences are possible depending on who requests delegation info from the PIP:

1. The Service Provider can request delegation info after a service request from the Service Consumer;
2. The Service Consumer can request delegation info and include it in its service request to the Service Provider;
3. The Entitled Party can push delegation info to the Service Consumer, so it can include it in its service request to the Service Provider.

Interaction sequence 3 is detailed below.

Depiction



Description

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
- The Service Consumer is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Service Consumer;

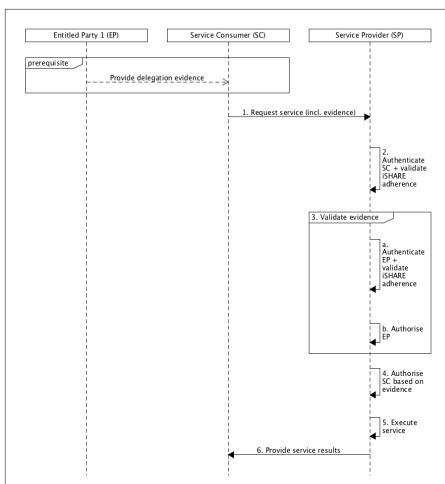
- The Entitled Party delegates (part of) its rights (as registered at the Service Provider) to the Service Consuming Entity. He provides the Service Consumer of the Service Consuming Entity with evidence of this delegation.

*The Service Provider can outsource this function to a third party

The use case consists of the following steps:

1. The Service Consumer requests a service from the Service Provider. With this requests he includes the evidence obtained from the Entitled Party;
2. The Service Provider authenticates the Service Consumer and validates the iSHARE adherence of the Service Consuming Entity;
3. The Service Provider validates the received delegation evidence through the following steps:
 - a. The Service Provider authenticates the Entitled Party and validates its iSHARE adherence based on the delegation evidence;
 - b. The Service Provider authorises the Entitled Party based on the entitlement information registered with the Service Provider.
4. The Service Provider authorises the Service Consumer of the Service Consuming Entity based on the validity of the delegation evidence;
5. The Service Provider executes the requested service;
6. The Service Provider provides the service result to the Service Consumer.

Sequence diagram



1c. M2M service provision with the AR as the delegation info PIP

In use case 1c, a service is provided by the Service Provider to the Service Consumer. The Service Consuming Entity has been delegated by the Entitled Party, and delegation evidence is registered at an Authorisation Registry.

Roles

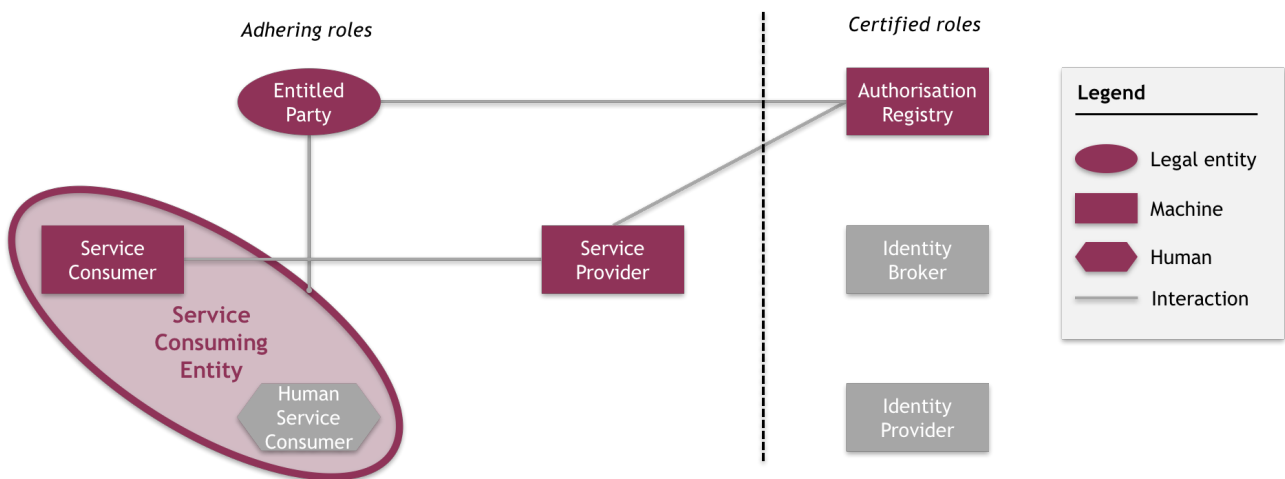
| | Delegation info PIP | | | |
|--------------------|---------------------|------------------|----------------|-------------------|
| | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Use case variation | 1 | 1a | 1b | 1c |

Note that interaction sequences are not described in the table above. In derived use case 1c, two interaction sequences are possible depending on who requests delegation info from the PIP:

1. The Service Provider can request delegation info after a service request from the Service Consumer;
2. The Service Consumer can request delegation info and include it in its service request to the Service Provider.

Interaction sequence 1 is detailed below.

Depiction



Description

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
- The Service Consumer is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Service Consumer;
- The Entitled Party delegates (part of) its rights (as registered at the Service Provider) to the Service Consumer Entity. He registers this delegation in an Authorisation Registry;
- The Service Provider knows which Authorisation Registry to request the delegation evidence from;
- The Service Provider is able to authenticate the Authorisation Registry;

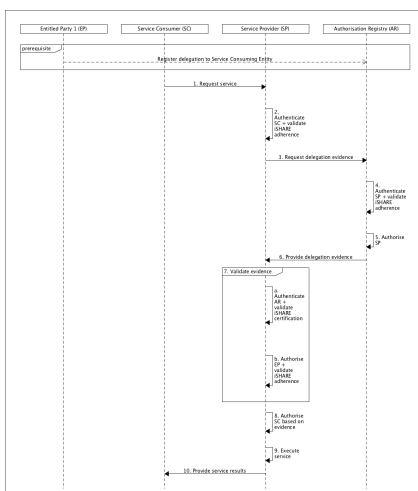
- The Authorisation Registry is able to authenticate the Service Provider;
- It is clear, through scheme agreements, under what conditions an Authorisation Registry can provide delegation information to a Service Provider.

*The Service Provider can outsource this function to a third party

The use case consists of the following steps:

1. The Service Consumer requests a service from the Service Provider;
2. The Service Provider authenticates the Service Consumer and validates the iSHARE adherence of the Service Consuming Entity;
3. The Service Provider requests delegation evidence from the Authorisation Registry;
4. The Authorisation Registry authenticates the Service Provider and validates its iSHARE adherence;
5. The Authorisation Registry authorises the Service Provider based on the scheme agreements for providing delegation information;
6. The Authorisation Registry provides the delegation evidence;
7. The Service Provider validates the received delegation evidence through the following steps:
 - a. The Service Provider authenticates the Entitled Party and validates its iSHARE adherence based on the delegation evidence;
 - b. The Service Provider authorises the Entitled Party based on the entitlement information registered with the Service Provider.
8. The Service Provider authorises the Service Consumer of the Service Consuming Entity based on the validity of the delegation evidence;
9. The Service Provider executes the requested service;
10. The Service Provider provides the service result to the Service Consumer.

Sequence diagram



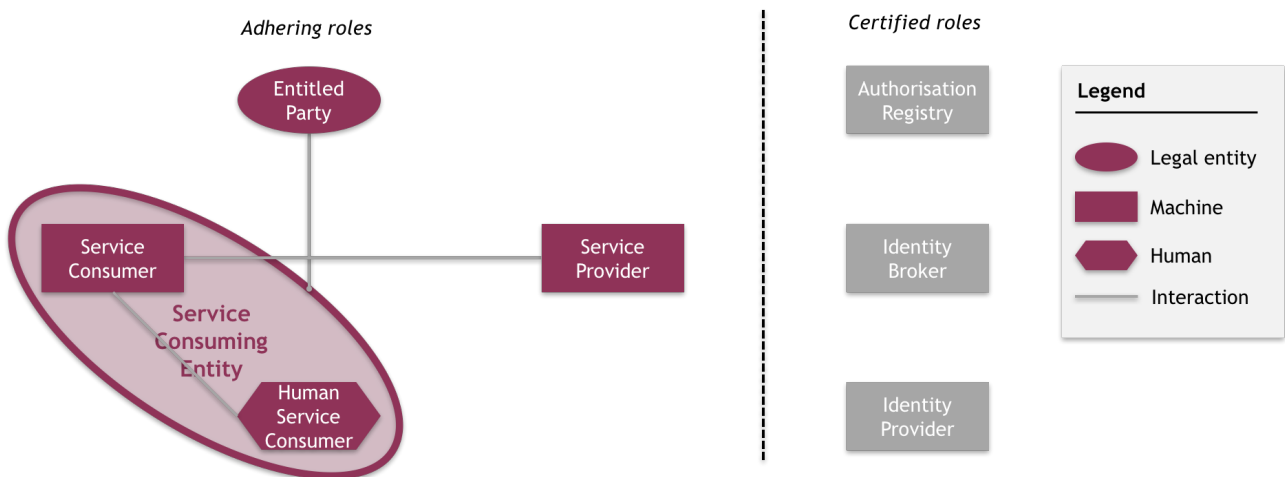
M2M service provision including an app

Use case 1 and its variations can be initiated by a Human Service Consumer through an app. In such case, the Service Consumer acts as a proxy between the Human Service Consumer and the Service Provider.

Roles

| | Delegation info PIP | | | |
|--------------------|---------------------|------------------|----------------|-------------------|
| | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Use case variation | 1 | 1a | 1b | 1c |

Depiction



Description

As to use case 1, it is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
- The Service Consumer is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Service Consumer.
- In this use case the Entitled Party is also the Service Consuming Entity.

*The Service Provider can outsource this function to a third party

The use case consists of the following steps:

- The Human Service Consumer uses an app to request a service at the Service Consumer - the Human Service Consumer's identity is included in the request;

- The request is mapped to a service request;
- 1. The Service Consumer requests a service from the Service Provider;
- 2. The Service Provider authenticates the Service Consumer and validates the iSHARE adherence of the Service Consuming Entity;
- 3. The Service Provider authorises the Service Consumer of the Service Consuming Entity based on the entitlement information registered with the Service Provider;
- 4. The Service Provider executes the requested service;
- 5. The Service Provider provides the service result to the Service Consumer;
- The Human Service Consumer accesses the result through app.

Sequence diagram

To follow.

2. H2M service provision with identity info at the SP

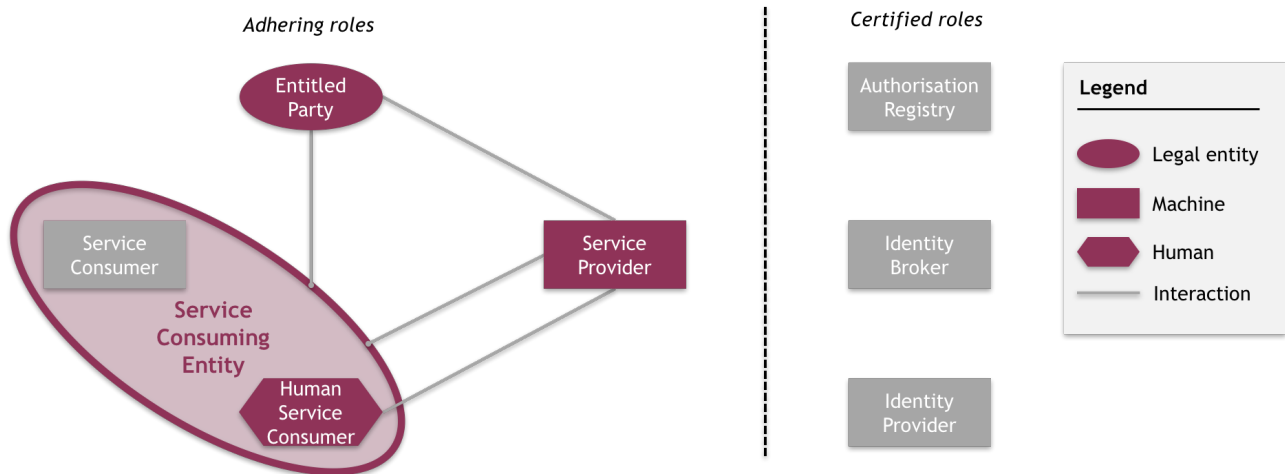
In use case 2, a service is provided by the Service Provider to the Human Service Consumer. Identity info is held at the Service Provider.

Roles

| | | Delegation info PIP | | | |
|----------------------|------------------|----------------------|------------------|----------------|-------------------|
| | | <i>No delegation</i> | Service Provider | Entitled Party | Authorisation Reg |
| Auth info PIP | Service Provider | 2 | 2a | 2b | 2c |

As no delegation takes place, the Entitled Party is also the Service Consuming Entity.

Depiction



Description

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
 - The Service Consuming Entity has and manages its own authorisation information indicating which Human Service Consumers are authorised to act on its behalf**;
 - The Service Consuming Entity registers the authorisation information at the Service Provider;
 - The Human Service Consumer is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Human Service Consumer;
 - The Human Service Consumer has been issued identity credentials by the Service Provider.
- In this use case the Entitled Party is also the Service Consuming Entity.

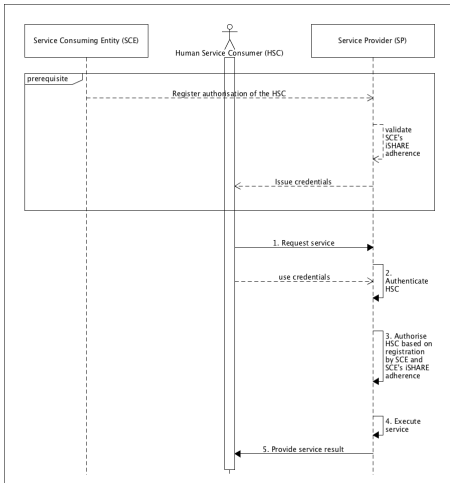
*The Service Provider can outsource this function to a third party

**The Service Consuming Entity can outsource this function to a third party

The use case consists of the following steps:

1. The Human Service Consumer requests a service from the Service Provider;
2. The Service Provider authenticates the Human Service Consumer, and validates the iSHARE adherence of the Service Consuming Entity;
3. The Service Provider authorises the Human Service Consumer of the Service Consuming Entity based on the entitlement information registered with the Service Provider;
4. The Service Provider executes the requested service;
5. The Service Provider provides the service result to the Human Service Consumer.

Sequence diagram



3. H2M service provision with identity info at the IP

In use case 3, a service is provided by the Service Provider to the Human Service Consumer. Identity info is held at the Identity Provider.

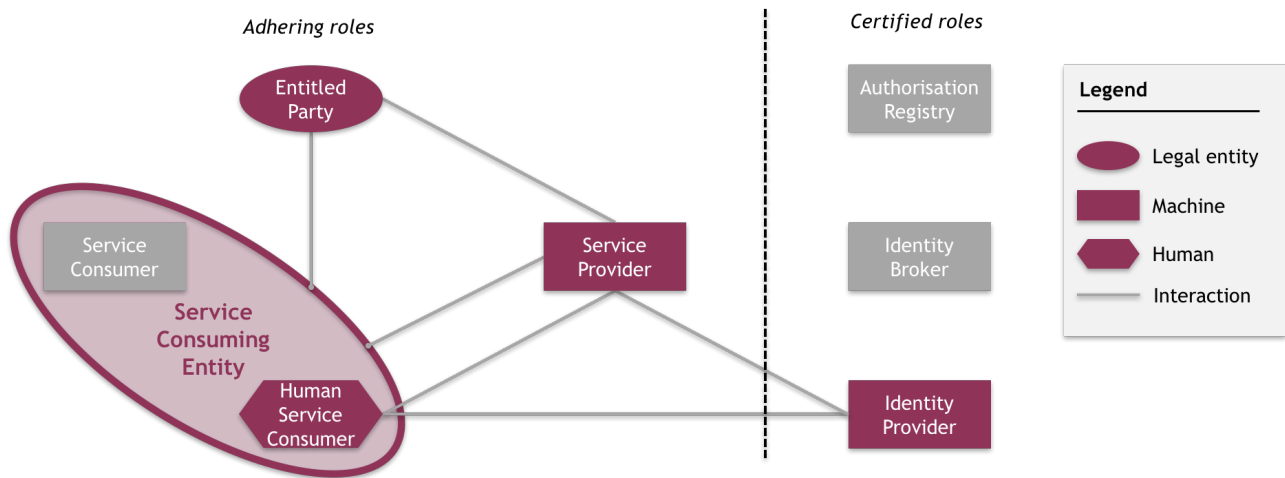
Roles

| | | Delegation info PIP | | | |
|---------------|--------------------|---------------------|------------------|----------------|-------------------|
| | | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Auth info PIP | Service Provider | 3 | 3a | 3b | 3c |
| | Entitled Party | 3.1 | 3a.1 | 3b.1 | 3c.1 |
| | Authorisation Reg | 3.2 | 3a.2 | 3b.2 | 3c.2 |
| | Identity Provider* | 3.3 | 3a.3 | 3b.3 | 3c.3 |

*The Identity Provider cannot hold explicit authorisation info, but it can hold info about a Human Service Consumer's identity that implies authorisation - i.e. 'working for truck company X'

As no delegation takes place, the Entitled Party is also the Service Consuming Entity.

Depiction



Description

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
 - The Service Consuming Entity has and manages its own authorisation information indicating which Human Service Consumers are authorised to act on its behalf**;
 - The Service Consuming Entity registers the authorisation information at the Service Provider;
 - The Human Service Consumer is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Human Service Consumer;
 - The Identity Provider is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Identity Provider;
 - The Human Service Consumer has been issued identity credentials by the Identity Provider.
- In this use case the Entitled Party is also the Service Consuming Entity.

*The Service Provider can outsource this function to a third party

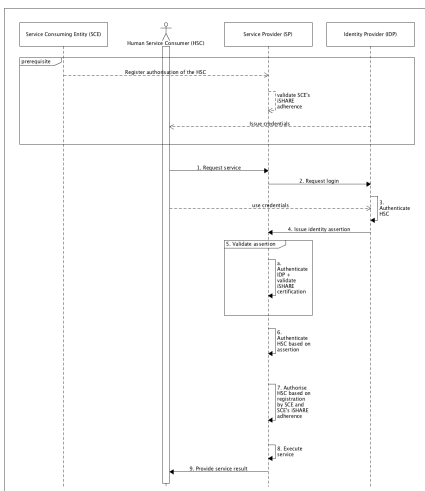
**The Service Consuming Entity can outsource this function to a third party

The use case consists of the following steps:

1. The Human Service Consumer requests a service from the Service Provider;
2. The Service Provider requests a login from the Identity Provider;
3. The Identity Provider authenticates the Human Service Consumer;
4. The Identity Provider issues an identity assertion to the Service Provider;
5. The Service Provider validates the identity assertion through the following steps:
 - a. The Service Provider authenticates the Identity Provider and validates its iSHARE certification.

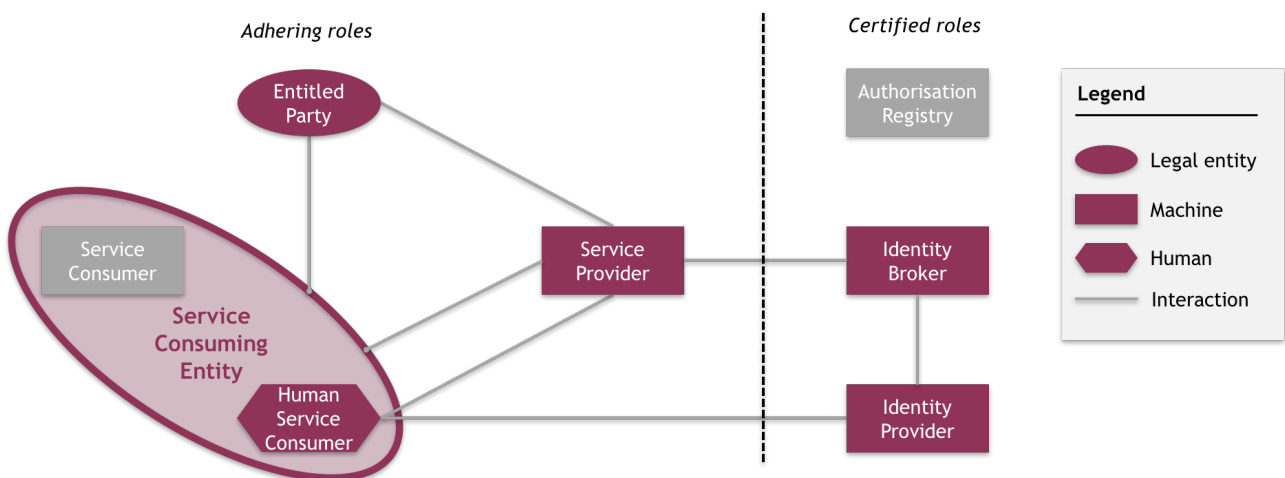
6. The Service Provider authenticates the Human Service Consumer based on the validity of the identity assertion, and validates the iSHARE adherence of the Service Consuming Entity;
7. The Service Provider authorises the Human Service Consumer of the Service Consuming Entity based on the entitlement information registered with the Service Provider;
8. The Service Provider executes the requested service;
9. The Service Provider provides the service result to the Human Service Consumer.

Sequence diagram



Note that an **Identity Broker** can be introduced to broker the relation between the Service Provider and the Identity Provider(s) and/or the Service Provider and the Authorisation Registry(s). This is optional and useful in situations with several Identity Providers and/or Authorisation Registries. This use case would look as follows with a Service Broker:

Depiction with Identity Broker



Description with Identity Broker

It is prerequisite of this use case that:

- The Service Provider has and manages its own authorisation information indicating what Entitled Parties are entitled to what (parts of) services*;
 - The Service Consuming Entity has and manages its own authorisation information indicating which Human Service Consumers are authorised to act on its behalf**;
 - The Service Consuming Entity registers the authorisation information at the Service Provider;
 - The Human Service Consumer is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Human Service Consumer;
 - The Identity Provider is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Identity Provider;
 - The Identity Broker is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Identity Broker;
 - The Human Service Consumer has been issued identity credentials by the Identity Provider.
- In this use case the Entitled Party is also the Service Consuming Entity.

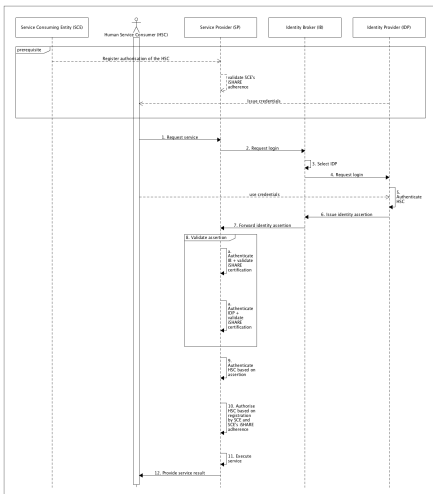
*The Service Provider can outsource this function to a third party

**The Entitled Party can outsource this function to a third party

The use case consists of the following steps:

1. The Human Service Consumer requests a service from the Service Provider;
2. The Service Provider requests a login from the Identity Broker;
3. The Identity Broker asks the Human Service Consumer to select his Identity Provider;
4. The Identity Broker requests a login from the Identity Provider;
5. The Identity Provider authenticates the Human Service Consumer;
6. The Identity Provider issues an identity assertion for the Service Provider to the Identity Broker;
7. The Identity Broker forwards the identity assertion to the Service Provider;
8. The Service Provider validates the identity assertion through the following steps:
 - a. The Service Provider authenticates the Identity Broker and validates its iSHARE certification;
 - b. The Service Provider authenticates the Identity Provider and validates its iSHARE certification.
9. The Service Provider authenticates the Human Service Consumer based on the validity of the identity assertion, and validates the iSHARE adherence of the Service Consuming Entity;
10. The Service Provider authorises the Human Service Consumer of the Service Consuming Entity based on the authorisation information registered with the Service Provider;
11. The Service Provider executes the requested service;
12. The Service Provider provides the service result to the Human Service Consumer.

Sequence diagram with Identity Broker



3.2. H2M service provision with identity info at the IP and the AR as the authorisation info PIP

In use case 3c.2, a service is provided by the Service Provider to the Human Service Consumer. Identity info is held at the Identity Provider. The Service Consuming Entity has been delegated by the Entitled Party, and delegation evidence is registered at an Authorisation Registry. Authorisation info is held at another Authorisation Registry.

Roles

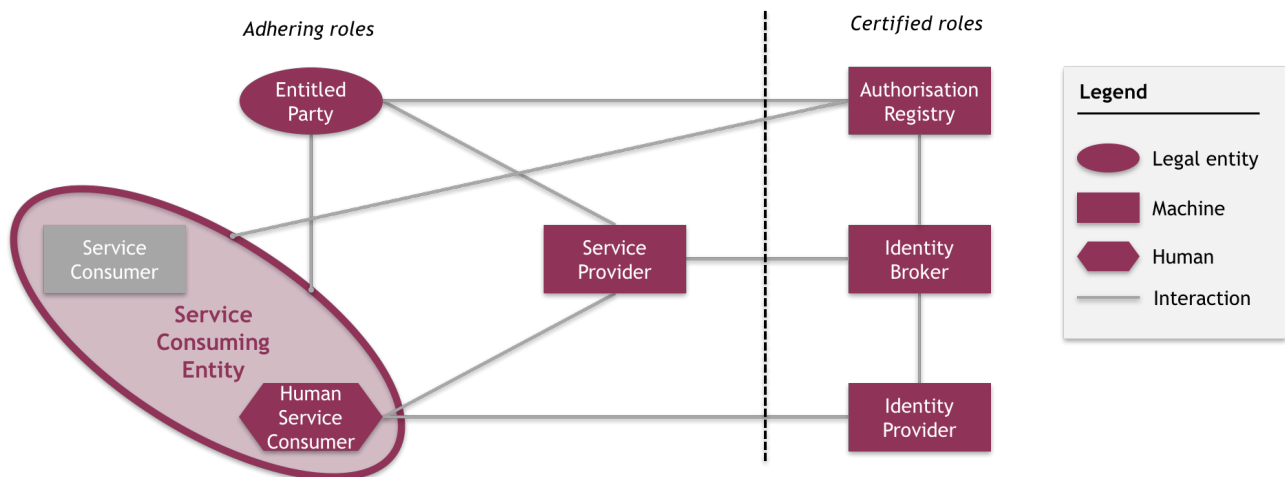
| | | Delegation info PIP | | | |
|---------------|--------------------|---------------------|------------------|----------------|-------------------|
| | | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Auth info PIP | Service Provider | 3 | 3a | 3b | 3c |
| | Entitled Party | 3.1 | 3a.1 | 3b.1 | 3c.1 |
| | Authorisation Reg | 3.2 | 3a.2 | 3b.2 | 3c.2 |
| | Identity Provider* | 3.3 | 3a.3 | 3b.3 | 3c.3 |

*The Identity Provider cannot hold explicit authorisation info, but it can hold info about a Human Service Consumer's identity that implies authorisation - i.e. 'working for truck company X'

As no delegation takes place, the Entitled Party is also the Service Consuming Entity.

Note that an **Identity Broker** can be introduced to broker the relation between the Service Provider and the Identity Provider(s) and/or the Service Provider and the Authorisation Registry(s). This is optional and useful in situations with several Identity Providers and/or Authorisation Registries.

Depiction



Description

It is prerequisite of this use case that:

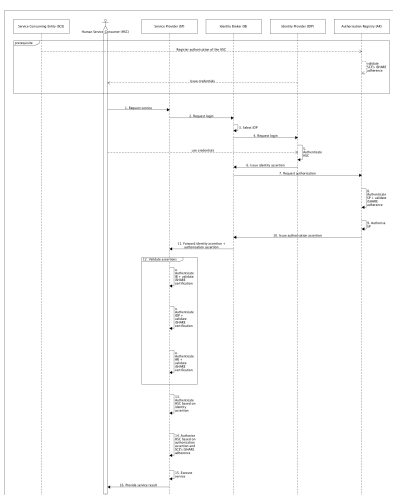
- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
 - The Service Consuming Entity has and manages its own authorisation information indicating which Human Service Consumers are authorised to act on its behalf**;
 - The Service Consuming Entity registers the authorisation information at the Authorisation Registry;
 - The Human Service Consumer is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Human Service Consumer;
 - The Authorisation Registry is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Authorisation Registry;
 - The Identity Provider is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Identity Provider;
 - The Identity Broker is able to authenticate the Service Provider;
 - The Service Provider is able to authenticate the Identity Broker;
 - The Identity Broker knows which Authorisation Registry to request the authorisation evidence from;
 - The Human Service Consumer has been issued identity credentials by the Identity Provider.
- In this use case the Entitled Party is also the Service Consuming Entity.

*The Service Provider can outsource this function to a third party

**The Service Consuming Entity can outsource this function to a third party

The use case consists of the following steps:

1. The Human Service Consumer requests a service from the Service Provider;
2. The Service Provider requests a login from the Identity Broker;
3. The Identity Broker asks the Human Service Consumer to select his Identity Provider;
4. The Identity Broker requests a login from the Identity Provider;
5. The Identity Provider authenticates the Human Service Consumer;
6. The Identity Provider issues an identity assertion for the Service Provider to the Identity Broker;
7. The Identity Broker requests authorisation evidence from the Authorisation Registry;
8. The Authorisation Registry authenticates the Service Provider and validates its iSHARE adherence;
9. The Authorisation Registry authorises the Service Provider;
10. The Authorisation Registry issues an authorisation assertion for the Service Provider to the Identity Broker;
11. The Identity Broker forwards the identity assertion and the authorisation assertion to the Service Provider;
12. The Service Provider validates the identity assertion through the following steps:
 - a. The Service Provider authenticates the Identity Broker and validates its iSHARE certification;
 - b. The Service Provider authenticates the Identity Provider and validates its iSHARE certification;
 - c. The Service Provider authenticates the Authorisation Registry and validates its iSHARE certification.
13. The Service Provider authenticates the Human Service Consumer based on the validity of the identity assertion, and validates the iSHARE adherence of the Service Consuming Entity;
14. The Service Provider authorises the Human Service Consumer of the Service Consuming Entity based on the validity of the authorisation assertion;
15. The Service Provider executes the requested service;
16. The Service Provider provides the service result to the Human Service Consumer.

Sequence diagram

3c.2. H2M service provision with identity info at the IP, an AR as the authorisation info PIP, and another AR as the delegation info PIP

In use case 3c.2, a service is provided by the Service Provider to the Human Service Consumer, who has been delegated by the Entitled Party. Delegation evidence is now registered at a Authorisation Registry.

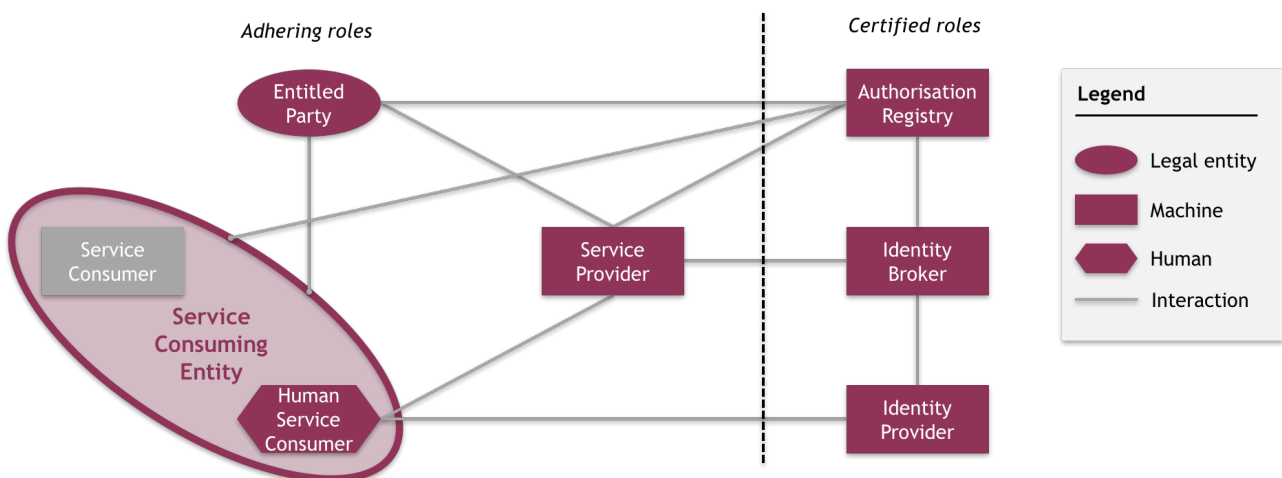
Roles

| | | Delegation info PIP | | | |
|---------------|--------------------|---------------------|------------------|----------------|-------------------|
| | | No delegation | Service Provider | Entitled Party | Authorisation Reg |
| Auth info PIP | Service Provider | 3 | 3a | 3b | 3c |
| | Entitled Party | 3.1 | 3a.1 | 3b.1 | 3c.1 |
| | Authorisation Reg | 3.2 | 3a.2 | 3b.2 | 3c.2 |
| | Identity Provider* | 3.3 | 3a.3 | 3b.3 | 3c.3 |

*The Identity Provider cannot hold explicit authorisation info, but it can hold info about a Human Service Consumer's identity that implies authorisation - i.e. 'working for truck company X'

Note that an **Identity Broker** can be introduced to broker the relation between the Service Provider and the Identity Provider(s) and/or the Service Provider and the Authorisation Registry(s). This is optional and useful in situations with several Identity Providers and/or Authorisation Registries.

Depiction



Description

In this derived use case, the Entitled Party delegates its rights to the Service Consuming Entity. Note that because the Entitled Party utilises another Authorisation Registry to register its delegation info than the Service Consuming Entity to register its authorisation info, two Authorisation Registries appear.

It is prerequisite of this use case that:

- The Service Provider has and manages its own entitlement information indicating what Entitled Parties are entitled to what (parts of) services*;
- The Entitled Party delegates (part of) its rights (as registered at the Service Provider) to the Service Consuming Entity. He registers this delegation in Authorisation Registry 2;
- The Service Consuming Entity has and manages its own authorisation information indicating which Human Service Consumers are authorised to act on its behalf**;
- The Service Consuming Entity registers the authorisation information at Authorisation Registry 1;
- The Human Service Consumer is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Human Service Consumer;
- Both Authorisation Registries are able to authenticate the Service Provider;
- The Service Provider is able to authenticate both Authorisation Registries;
- The Service Provider knows which Authorisation Registry to request the delegation/authorisation info from;
- It is clear, through scheme agreements, under what conditions an Authorisation Registry can provide delegation/authorisation information to a other parties;
- The Identity Provider is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Identity Provider;
- The Identity Broker is able to authenticate the Service Provider;
- The Service Provider is able to authenticate the Identity Broker;
- The Identity Broker knows which Authorisation Registry to request the authorisation evidence from;
- The Human Service Consumer has been issued identity credentials by the Identity Provider

* The Service Provider can outsource this function to a third party

** The Entitled Party can outsource this function to a third party

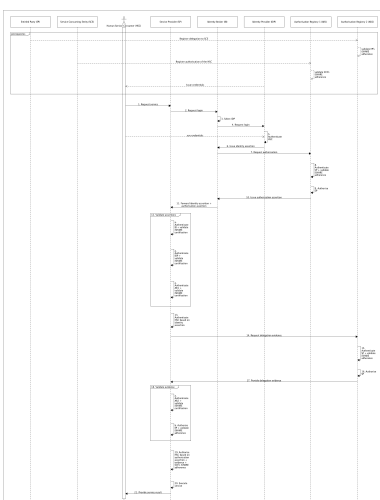
The use case consists of the following steps:

1. The Human Service Consumer requests a service from the Service Provider;
2. The Service Provider requests a login from the Identity Broker;
3. The Identity Broker asks the Human Service Consumer to select his Identity Provider;
4. The Identity Broker requests a login from the Identity Provider;
5. The Identity Provider authenticates the Human Service Consumer;
6. The Identity Provider issues an identity assertion for the Service Provider to the Identity Broker;
7. The Identity Broker requests authorisation evidence from Authorisation Registry 1;
8. Authorisation Registry 1 authenticates the Service Provider and validates its iSHARE adherence;
9. Authorisation Registry 1 authorises the Service Provider;
10. Authorisation Registry 1 issues an authorisation assertion for the Service Provider to the Identity Broker;

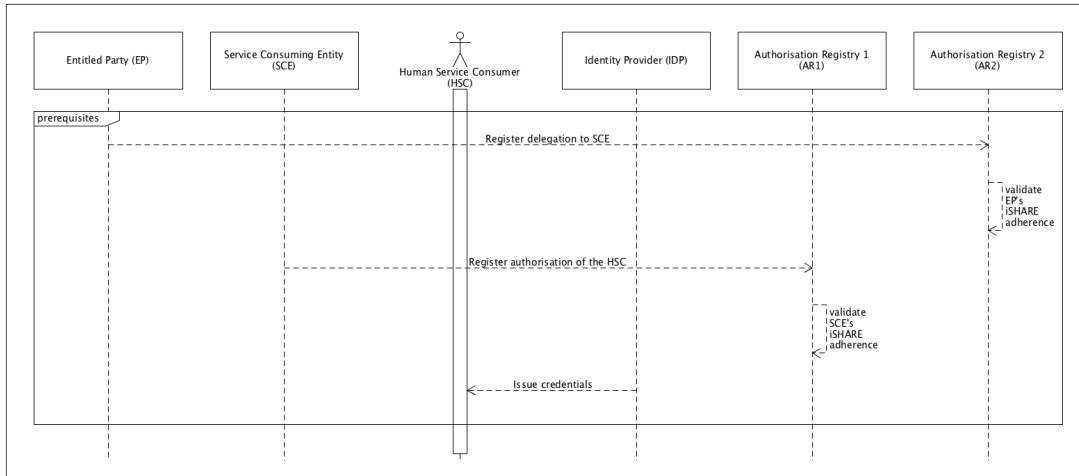
11. The Identity Broker forwards the identity assertion and the authorisation assertion to the Service Provider;
12. The Service Provider validates the identity assertion through the following steps:
 - a. The Service Provider authenticates the Identity Broker and validates its iSHARE certification;
 - b. The Service Provider authenticates the Identity Provider and validates its iSHARE certification;
 - c. The Service Provider authenticates Authorisation Registry 1 and validates its iSHARE certification.
13. The Service Provider authenticates the Human Service Consumer based on the validity of the identity assertion, and validates the iSHARE adherence of the Service Consuming Entity;
14. The Service Provider requests delegation evidence from Authorisation Registry 2;
15. Authorisation Registry 2 authenticates the Service Provider and validates its iSHARE adherence;
16. Authorisation Registry 2 authorises the Service Provider based on the scheme agreements for providing authorisation information;
17. Authorisation Registry 2 provides the delegation evidence;
18. The Service Provider validates the received delegation evidence through the following steps:
 - a. The Service Provider authenticates Authorisation Registry 2 and validates its iSHARE certification;
 - b. The Service Provider authorises Entitled Party 1 based on the entitlement information registered with the Service Provider, and validates its iSHARE adherence.
19. The Service Provider authorises the Human Service Consumer of the Service Consuming Entity based on the validity of the delegation evidence;
20. The Service Provider executes the requested service;
21. The Service Provider provides the service result to the Human Service Consumer.

Sequence diagrams

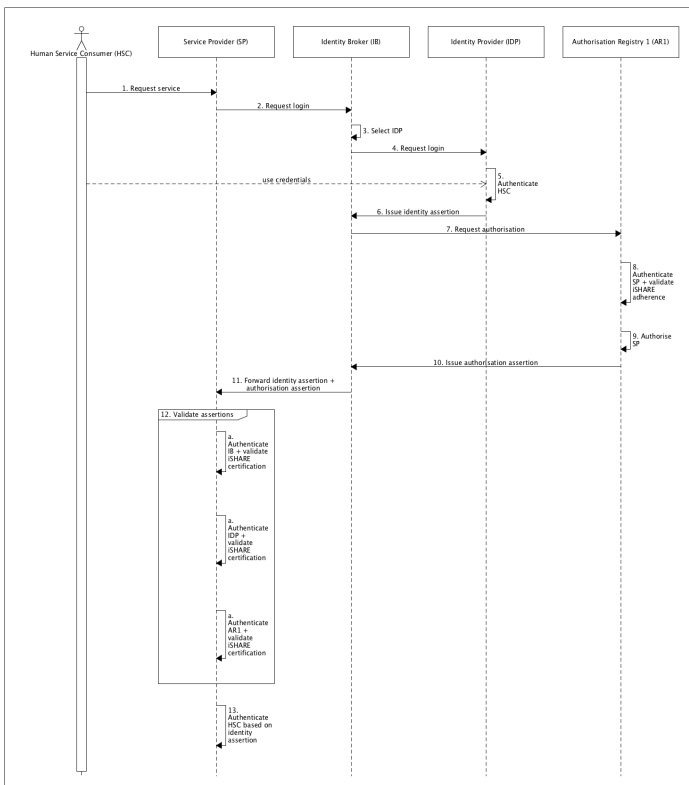
Total



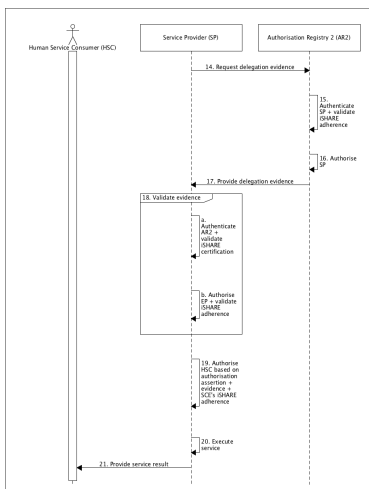
Prerequisites



Authentication and Authorisation



Delegation



Licenses

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

For a Service Consumer, the result of any of iSHARE's [primary use cases](#) is consuming a service (for example: receiving data). Entitled Parties can specify what Service Consumers may and may not do with the result of a consumed service. For this purpose, **licenses** are introduced. A license contains statements about how a Service Consumer is supposed to use the result of a service it has consumed from a Service Provider. The license covering a service can consist of the following elements:

- LicensePurpose
 - A code indicating for what purpose the Service Consumer is allowed to use the received data
 - Optional, defaults to 0000, indicating reshaping with iSHARE adhering parties*
- LicenseSubLicense
 - An integer indicating the number of times the Service Consumer can share the result with third parties
 - Optional, defaults to 9999
- LicenseEndDate
 - Within what timeframe the Service Consumer can keep and use the result of the service it has consumed
 - Optional, defaults to 9999-12-31


Establishing a license

Entitled Parties and/or Service Providers are responsible for establishing the license covering a service or services, either based on an agreement with a Service Consumer or unilaterally declared. Entitled Parties must register licenses at a PIP.

Entitled Parties can also choose *not* to establish a license; in this case a default applies - as described above.

Putting the license to use

Whenever a service is requested by a Service Consumer from a Service Provider, the request SHOULD include the requested LicensePurpose, as described in the technical [interface specifications](#). The response from the Service Provider SHOULD include the LicensePurpose, LicenseSubLicense & LicenseEndDate for the license covering this specific service.

 It is highly discouraged to set a LicenseEndDate since this puts high requirements on the implementation of the Service Consumer.

If *no* license is established by the Entitled Party and/or mentioned in the response from the Service Provider, the default license applies as described above. The details of licenses, and their legal value in case of a dispute, will be detailed in the Legal part of this scheme.

*During co-creation, it was suggested that the default should also include that the result of the service should not be used to damage the Entitled Party. This is a general requirement of service consumption, however, and will therefore be part of the Terms & Conditions for iSHARE adhering parties.

Purpose code list

| Purpose code | Description |
|--------------|---|
| 0000 | No limitations |
| 0001 | Re-sharing with Adhering Parties only |
| 0002 | Internal use only |
| 0003 | Non-commercial use only: licensee may not use the data to generate revenue |
| 0004 | Licensee may enrich received data with own data before re-sharing |
| 0005 | Licensee may enrich received data with data of others before re-sharing |
| 0006 | Licensee may enrich received data with own data before re-sharing on a non-commercial basis |
| 0007 | Licensee may enrich received data with data of others before re-sharing on a non-commercial basis |
| 9999 | As determined between Parties |

Identifiers

Within iSHARE companies will mainly be identified by their Economic Operators Registration and Identification (EORI) number. The EORI number is used as an identifier for companies throughout the European Union. The format of the EORI number consists of a country code followed by a unique code which is established within an EU member state. For example, in the Netherlands the EORI consists of: NL, followed by an RSIN number (Rechtspersonen en Samenwerkingsverbanden IdentificatieNummer). If the NL-RSIN contains less than 9 digits, the EORI is prefixed with 0's. For more information on the EORI number, please consult the European [EORI website](#). For more information on the EORI for Dutch parties, please consult the website of the [Dutch tax authorities](#).

If no EORI is available, it is alternatively allowed for Dutch entities to use the unique Chamber of Commerce number as alternative identifier.

Certified parties also include their "role" when identifying themselves, using the following identifiers:

| Role identifier |
|------------------------|
| IDENTITY_PROVIDER |
| IDENTITY_BROKER |
| AUTHORISATION_REGISTRY |

For technical guidance on role identifiers, please consult the [Scheme Owner's API specification](#).

Secondary use cases

iSHARE's [three primary use cases](#) are supported by seven secondary use cases. These include:

- Processes related to registration
- Processes that recur in primary use cases

Processes related to registration

These four secondary use cases need to be completed before any, or specific, primary use cases can be initiated.

Any party needs to:

- 1a. Register adherence/certification at [Scheme Owner](#)
and later needs to be able to:
- 1b. Modify adherence/certification at Scheme Owner

Before initiating Human to Machine use cases, the **Service Consuming Entity** needs to:

2a. Create Service Consuming Entity and/or [Human Service Consumer](#) identity at [Identity Provider](#)

Prerequisites:

- An agreement needs to be in place between Service Consuming Entity and Identity Provider
- An agreement needs to be in place between Service Provider and Identity Provider

later, a Service Consuming Entity needs to be able to:

2b. Modify Service Consuming Entity and/or Human Service Consumer identity at Identity Provider

When delegating rights, the **Entitled Party** needs to:

3a. Register delegation at [Service Provider](#), Entitled Party, or [Authorisation Registry](#)

Prerequisite:

- For registration at Service Provider or Authorisation Registry, an agreement needs to be in place between Entitled Party and Service Provider or Authorisation Registry

later, an Entitled Party needs to be able to:

3b. Modify delegation at Service Provider, Entitled Party, or Authorisation Registry

When authorising something or -one, the **Service Consuming Entity** needs to:

4a. Register authorisation at Service Provider, Entitled Party, or Authorisation Registry

Prerequisite:

- For registration at Service Provider or Authorisation Registry, an agreement needs to be in place between Service Consuming Entity and Service Provider or Authorisation Registry

later, a Service Consuming Entity needs to be able to:

4b. Modify authorisation at Service Provider, Entitled Party, or Authorisation Registry

Processes that recur in primary use cases

These three secondary use cases form the wiring of all primary use cases. Without them, primary use cases cannot be completed successfully.

In any primary use case, **any party** needs to:

5a. Check whether its counterparty is iSHARE adherent/certified (with the Scheme Owner)

5b. Check whether its counterparty's certificate is valid (with the Scheme Owner)

In any primary use case, the **Service Provider** *also* needs to:

6. Determine an authorisation decision based on entitlement-, delegation-, and/or authorisation info in its own contract administration and/or from external PIPs

When delegation- or authorisation info is requested by a Service Provider, an **Authorisation Registry** or **Entitled Party** also needs to:

7. Determine authorisation decision based on Service Consumer assertion included in Service Provider's request

Please note that the secondary use cases will not be detailed more than the above. No depictions or sequence diagrams are to be developed (contrary to for the primary use cases). This (deliberately) leaves freedom in implementation.

Functional requirements per role

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The functional requirements per role will be summarised and made explicit in the next iteration of the iSHARE scheme. They can already be found (implicitly) throughout the [Roles & Responsibilities](#) and [primary use cases](#).

User interface requirements

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

For all [Human to Machine interactions](#), as in [primary use case 2](#) and [3](#), an interface is required. This interface MUST comply with the following guidelines:

- The name of the party that provides a broker service or identity provisioning service MUST be clearly visible.
- During the process of authentication, information not directly relating to the identity provision process or supporting the identity provision process MAY NOT be present. Links to websites irrelevant to the identity provisioning process or advertisements MAY NOT be present.
- Parties facilitating the identity provision process MAY use their own corporate styling and logos

- The iSHARE brand MUST be shown during the identity provision process. Showing the iSHARE brand MUST be in line with iSHARE communication guidelines (Communication guidelines have not yet been determined).
- Users that are being identified through the use of a browser MUST be able to verify the URL and used SSL certificate during all steps of identity provisioning process.

Please note that extra guidance will need to be added for the context of apps: how can users verify that they are not being tricked?

Technical

This section covers the technical details of the iSHARE scheme.

The section starts out with a chapter containing the [basic API specifications](#), including an [example implementation](#) based on use case 1c. The chapter also includes [role specific API requirements](#) and the [APIs that are exposed by the Scheme Owner](#).

The chapter on "[Language of delegation and authorisation](#)" explains how the process of delegation and authorisation are implemented within the iSHARE scheme.

The section then ends with an overview of relevant [technical standards](#).

Interface specifications

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

Note that a complete overview of the current draft of the iSHARE API specifications (work in progress) can be found [here](#) on SwaggerHub.

The iSHARE API makes use of [standard HTTP methods](#) inspired by the RESTful architectural style. Standard CRUD operations are mapped to standard HTTP methods in the following table:

| HTTP Verb | CRUD |
|-----------|----------------|
| POST | Create |
| GET | Read |
| PUT | Update/Replace |
| PATCH | Update/Modify |
| DELETE | Delete |

Use of OAuth 2.0 in iSHARE in general

OAuth 2.0 is used in iSHARE M2M use cases directly and as part of OpenID Connect 1.0 in iSHARE H2M use case.

For both uses of OAuth 2.0 the following requirements apply:

- Clients MUST NOT be pre registered. A look-up in the iSHARE adherence registry is sufficient. It is up to the server ¹
- Clients MUST authenticate through the `private_key_jwt` method as specified in OpenID Connect 1.0 [Chapter 9](#). ^{1 2}

- The `private_key_jwt` MUST always contain the `iat` claim
- The `iss` and `sub` claims MUST contain the valid iSHARE identifier of the client ¹
- The `aud` claim MUST contain only the valid iSHARE identifier of the server. (Including multiple audiences creates a risk of impersonation and is therefore not allowed)
- The `private_key_jwt` MUST be signed using a certificate containing the client's valid iSHARE identifier
- The `private_key_jwt` MUST be set to expire in 30 seconds
- A server SHALL NOT accept a `private_key_jwt` more than once. However within it's time to live a Service Provider MAY forward a `private_key_jwt` from a Service Consumer to another server (Entitled Party or Authorisation Registry) to obtain additional evidence on behalf of the Service Consumer
- A server SHALL only accept a forwarded `private_key_jwt` if the `aud` claim of the forwarded `private_key_jwt` matches the `iss` claim of the `private_key_jwt` from the client
- For interoperability reasons clients MUST only make HTTP GET calls to all OAuth /token and iSHARE /delegation endpoints. (Services are free to implement other HTTP verbs)
- Servers SHALL only issue access tokens with "bearer" token type
- Servers SHALL NOT issue refresh tokens
- Access tokens SHOULD expire within 3600 seconds by default. Depending on scope, servers MAY choose to limit the expiration period

Additional rationale

¹ In OAuth 2.0 clients are generally pre-registered. Since in iSHARE we want to interact with clients that have been previously unknown this doesn't suit us. We go for a generic client identification and authentication scheme, based on iSHARE whitelisted PKI roots.

² Since OAuth 2.0 doesn't specify a PKI base authentication scheme, but OpenID Connect 1.0 does, iSHARE chooses to use the later in all use cases. This is preferred above defining a new proprietary scheme.

Use of OAuth 2.0 in iSHARE M2M

For use of OAuth 2.0 in iSHARE M2M use cases the following additional requirements apply:

- Only the Client Credentials Grant SHALL be used

Use of OpenID Connect 1.0 in iSHARE H2M

For use of OpenID Connect 1.0 in iSHARE H2M use cases the following additional requirements apply:

- Only the Authorization Code Grant SHALL be used
- Since clients aren't pre-registered, the `redirect_uri` parameter MUST be present in the authorization request.
- OpenID Connect clients MUST use the `state` parameter
- iSHARE only allows the following scope values:
 - `openid` - REQUIRED scope value
 - `name` - OPTIONAL scope value requests access to: `name`, `family_name`, `given_name`, `middle_name` and `gender`
 - `contact_details` - OPTIONAL scope value requests access to: `email`, `email_verified`, `phone_number` and `phone_number_verified`
 - `company_id` - OPTIONAL scope value requests access to: `company_id`
 - `company_info` - OPTIONAL scope value requests access to: `company_name`, `company_type`, `company_address` and `company_url`
- A client SHALL only request claims from the `/userinfo` endpoint based on the access token. Scope values or claims request parameters SHALL NOT be used.
- For interoperability reasons clients MUST only make HTTP GET calls to the `/userinfo` endpoint.

Caching

For every API exposed under iSHARE caching MUST Be made explicit to the API consumer.

If a response is not cacheable it MUST contain the following headers:

| Adherence information |
|---|
| Cache-Control: no-store Pragma: no-cache |

If a response is cacheable it MUST contain the following headers:

| Adherence information |
|---------------------------------|
| Cache-Control: max-age=31536000 |

Note: max-age MAY vary

Security

Organisations participating in the iSHARE scheme need to consider aspects of security. Depending on the character of services and data of an organisation, appropriate security measures need to be taken. Please refer to the following glossary topics for more guidance on security:

- [Confidentiality](#)
- [Integrity](#)
- [Authenticity](#)
- [Availability](#)
- [Non-repudiation](#)

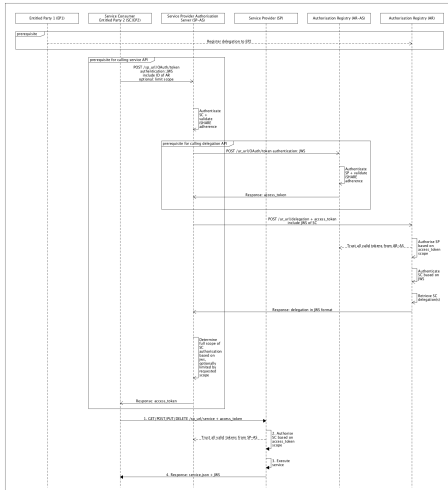
Response codes

Within the iSHARE scheme, the HTTP standard concerning response codes is followed as established by the IETF. Please refer to the [IETF website](#) for further specification. Within iSHARE the HTTP response codes 401, 403, 406, 409 and 412 are most relevant.

| HTTP Verb | CRUD | Entire Collection (e.g. /customers) | Specific Item (e.g. /customers/{id}) |
|-----------|------------------|---|--|
| POST | Create | 201 (Created), 'Location' header with link to /customers/{id} containing new ID. | 404 (Not Found), 409 (Conflict) if resource already exists.. |
| GET | Read | 200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists. | 200 (OK), single customer. 404 (Not Found), if ID not found or invalid. |
| PUT | Update / Replace | 404 (Not Found), unless you want to update/replace every resource in the entire collection. | 200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid. |
| PATCH | Update / Modify | 404 (Not Found), unless you want to modify the collection itself. | 200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid. |
| DELETE | Delete | 404 (Not Found), unless you want to delete the whole collection—not often desirable. | 200 (OK). 404 (Not Found), if ID not found or invalid. |

API example use case 1c

Step by step detailed technical overview of use case 1c



Pre-requisite for calling service APIs at Service Provider

Access Token request from Service Consumer

```
GET /oauth2.0/token HTTP/1.1
Host: example.service-provider.com

grant_type=client_credentials
&scope=iSHARE
&client_id=NL000000001
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
&client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjNkZGRkYXNkYXNkYXNkZGQ0NTY3ODkwIiwiaWF0Ij03ir88bMY_WyzYu-cwnaIr20gLLWZIQ3W7dq4--
JqMwnlVb3xunr6YHm4ivGftvdVbpS2sPqoLxNHCsYgb2L2X0NJKurhpgZ_00B5FwPHJ1nqvX_fwymwNejPZPgqFLvUN-U
&authorisation_registry=NL123456789
```

Pre-requisite for calling /delegation API at Authorisation Registry

Access Token request from Service Provider

```
GET /oauth2.0/token HTTP/1.1
Host: example.authorisation-registry.com

grant_type=client_credentials
```

```
&scope=iSHARE
&client_id=NL000000002
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer
&client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhc2RzYXN1YiI6IjEyM2RkZGRhc2Rhc2Rhc2RkZDQ1Njc4OTAiLCJyZWllIjoibm9ob2B2UjZG1pbiI6dHJ1ZX0.ay5Ghz_X6It4h8KnNUiar03hTPWJ_ahqfaTzZ_NwNGJecC0GXlJefm0NyCOUq9jlyZel8_mmrfbtDZDZixov8QEInoC7Eihsq07o9xih0vhCRTbnx_G98UV8X2STGiN0Ppz3TDWKEH-R1dAFL6E5KFLG-Ybi7ZqzplHbey-ZcEw
```

Access Token response from Authorisation Registry

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "AGxpJB7hl9too8AULLpncK1Kih5beXbjnbe0DHp2EN48U09BDpvtgScF05aIXwH9T",
  "token_type": "bearer",
  "expires_in": 3600
}
```

Delegation Evidence request from Service Provider

```
GET /ishare1.0/delegation HTTP/1.1
Authorization: Bearer AGxpJB7hl9too8AULLpncK1Kih5beXbjnbe0DHp2EN48U09BDpvtgScF05aIXwH9T
Host: example.authorisation-registry.com

service_consumer_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMkZGRkYXNkYXNkYXNkZGQ0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ij0iYWRtaW4iOnRydWV9.w-0FT6yHL2cnXHicWvKKNLhd1nTft8jHSFLL_Fit03ir88bMY_WyzYu-cwnaIr20gLWZIQ3W7dq4--
JqMwnlVb3xunR6YHm4ivGftvdVbpS2sPqoLxNHCsYgb2L2X0NJKurhpgZ_00B5FwPHJ1nqvX_fwymwNejPZPgqFLvUN-U
```

Delegation Evidence response from Authorisation Registry

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "delegation_token":
  "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ij0iYWRtaW4iOnRydWV9.
  EkN-
  DOsnsuRjR06BxXemmJdM3HbxrbRzXglbN2S4s0kopaU4IsDxTI8j019W_A4K8ZPJijnLis4EZsHeY559a4DF0d50_0qHGGuERTqYZyuhF3
  9yxJPAjUESwxk2J5k_4zM30-vtd1Ghyo4IbqKKSy6J9mTniYJPenn5-HIirE"
}
```

Access Token response from Service Provider

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "be0DHp2EN48U09BDpvtgScF05aIXwH9TAGxpJB7hl9too8AULLpncK1Kih5beXbjn",
```

```

    "token_type": "bearer",
    "expires_in": 3600
  }

```

Service request from Service Consumer

```

GET /service HTTP/1.1
Authorization: Bearer be0Dhp2EN48U09BDpvtgScF05aIXwH9TAGxpJB7hL9tooI8AULLpncK1Kih5beXbjn
Host: example.service-provider.com
LicensePurpose: RESHARE_ISHARE

```

Service response from Service Provider

```

HTTP/1.1 200 OK
Content-Type: application/json
LicensePurpose: RESHARE_ISHARE
LicenseSubLicense: 10
LicenseEndDate: 9999-12-31

{
  ... service specific content ...
}

```

Scheme Owner APIs

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The iSHARE Scheme Owner exposes the following APIs.

/scheme_owner/ishare1.0/parties/{party_id}

Returns a signed JWT containing the all registered information of the requested party.

Note: this is the registered information at the time, not describing certification at the time

Party information

```

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "date_time": "2017-06-07T06:17:23Z",
  "party_id": "EU.EORI.123456789",
  "adherence": {

```

```

    "status": "ACTIVE",
    "start_date": "2017-06-07T06:17:23Z"
  },
  "certifications": [
    "certification": {
      "role": "iSHARE.v10.IDENTITY_PROVIDER",
      "start_date": "2017-06-07T06:17:23Z",
      "end_date": "2017-12-31T23:59:59Z"
    },
    "certification": {
      "role": "iSHARE.v11.IDENTITY_PROVIDER",
      "start_date": "2017-06-07T06:17:23Z"
    },
    "certification": {
      "role": "iSHARE.v10.IDENTITY_BROKER",
      "start_date": "2017-06-07T06:17:23Z"
    },
    "certification": {
      "role": "iSHARE.v11.AUTHORISATION_REGISTRY",
      "start_date": "2017-06-07T06:17:23Z"
    }
  ]
}

```

Values for "adherence":status" can be:

- ACTIVE
- NOT_ACTIVE
- SUSPENDED

End_date for adherence and certification is optional.

Note: if a date time is provided in the request, the result becomes final and therefor MUST be cashable.

HTTP Header for cacheable party information

```

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: max-age=31536000

```

...

/scheme_owner/ishare1.0/certified_parties

Returns a signed JWT containing the certifications of all certified parties

Certification information

HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

```

{
  "date_time": "2016-12-31T12:45:23Z",
  "certified_parties": [
    {
      "party_id": "EU.EORI.123456789",
      "adheres": "ACTIVE",
      "certifications": [
        "certification": {
          "role": "iSHARE.v10.IDENTITY_PROVIDER",
          "start_date": "2017-06-07T06:17:23Z",
          "end_date": "2017-12-31T23:59:59Z"
        },
        "certification": {
          "role": "iSHARE.v11.IDENTITY_PROVIDER",
          "start_date": "2017-06-07T06:17:23Z"
        },
        "certification": {
          "role": "iSHARE.v10.IDENTITY_BROKER",
          "start_date": "2017-06-07T06:17:23Z"
        },
        "certification": {
          "role": "iSHARE.v11.AUTHORISATION_REGISTRY"
          "start_date": "2017-06-07T06:17:23Z"
        }
      ]
    }
  ]
}
{
  "party_id": "EU.EORI.234567890",
  "certifications": [
    "certification": {
      "role": "iSHARE.v11.IDENTITY_PROVIDER",
      "start_date": "2017-06-07T06:17:23Z"
    }
  ]
}
]
}

```

Certifications are specified in the following format:

iSHARE.<version>.<role>

Note: if a date time is provided in the request, the result becomes final and therefore MUST be cashable.

/scheme_owner/ishare1.0/trusted_list

tbd

/scheme_owner/ishare1.0/certificate_validation

Input is a x.509 certificate in PEM format

Returns a signed JWT containing the validity of a certificate

Adherence information

```

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "date_time": "2016-12-31T12:45:23Z",
  "certificate": "EU.EORI.123456789",
  "validity": "TRUE"
}

```

Note: if a date time is provided in the request, the result becomes final and therefor MUST be cachable.

Values for "validity" can be:

- TRUE
- FALSE
- UNKNOWN

Role specific API requirements

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

The below is a summary of the requirements per service to be requested from a party with a certain role, as found in the iSHARE API specifications [on SwaggerHub](#).

The roles from which a service can be requested covered here are [Service Consumer](#), [Service Provider](#), [Entitled Party](#), [Authorisation Registry](#), [Identity Provider](#), and [Scheme Owner](#). The services covered are GET, POST, PUT, PATCH, and DELETE as in [standard HTTP methods](#) inspired by the RESTful architectural style. Standard CRUD operations are mapped to standard HTTP methods in the table on [this page](#).

Service Consumer

GET /service_consumer/webhook_url

Service Consumer defined URL that is registered to receive certain event types from the Service Provider

- The service request MUST contain the parameter `event_id`

Service Provider

GET /service_provider/oauth2.0/token

Used to obtain an OAuth access token from a Service Provider

- The format of `access_token` MUST be defined by the Service Provider
- The format of `access_token` SHOULD be opaque to the Service Consumer
- The service request MUST contain the parameters `grant_type`; `client_id`; `client_assertion_type`, and; `client_assertion`
- The service request MAY contain the parameters `scope`; `delegation`; `authorisation_registry`, and; `entitled_party_registry`
- The `grant_type` parameter MUST contain "client_credentials"
The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Consumer
- The `client_assertion_type` parameter MUST contain "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- The `client_assertion` parameter MUST contain a JWT token conform iSHARE specifications, signed by the client

GET /service_provider/openid_connect1.0/return

OpenID Connect end-point received by a Human Service Consumer after authentication

- This OpenID Connect end-point MAY have any name the Service Provider chooses
- The service request MUST contain the parameters `code`, and; `state`
- The `state` parameter MUST contain the state as provided by the Service Provider in the service request to the Identity Provider or Identity Broker

POST /service_provider/webhooks

Used to subscribe to certain events defined by the Service Provider by registering a webhook url

- The service request MUST contain the parameters Authorisation, and; Request Body
- The Authorisation parameter MUST contain "Bearer" + access token value

GET /service_provider/webhooks

Used to subscribe to certain events defined by the Service Provider by registering a webhook url

- The service request MUST contain the parameter Authorisation
- The Authorisation parameter MUST contain "Bearer" + access token value

GET /service_provider/webhooks/{webhook_id}

Used to obtain info on a certain webhook url

- The service request MUST contain the parameters Authorisation, and; webhook_id
- The Authorisation parameter MUST contain "Bearer" + access token value

DELETE /service_provider/webhooks/{webhook_id}

Used to delete a certain webhook url

- The service request MUST contain the parameters Authorisation, and; webhook_id
- The Authorisation parameter MUST contain "Bearer" + access token value

GET /service_provider/events/{event_id}

Used to obtain info on a certain event

- The service request MUST contain the parameters Authorisation, and; event_id
- The Authorisation parameter MUST contain "Bearer" + access token value

GET /service_provider/service

Example service of a Service Provider

- An iSHARE-adherent Service Provider MUST apply iSHARE conformant OAuth to every iSHARE enabled service
- An iSHARE enabled service MAY have any name the Service Provider chooses
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `service_consumer_assertion`; `LicensePurpose`; `Do-Not-Sign`, and; `Service-Headers`
- The `Authorisation` parameter MUST contain "Bearer" + access token value

POST /service_provider/service

Example service of a Service Provider

- An iSHARE-adherent Service Provider MUST apply iSHARE conformant OAuth to every iSHARE enabled service
- An iSHARE enabled service MAY have any name the Service Provider chooses
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `service_consumer_assertion`; `LicensePurpose`; `Do-Not-Sign`, and; `Service-Headers`
- The `Authorisation` parameter MUST contain "Bearer" + access token value

PUT /service_provider/service

Example service of a Service Provider

- An iSHARE-adherent Service Provider MUST apply iSHARE conformant OAuth to every iSHARE enabled service
- An iSHARE enabled service MAY have any name the Service Provider chooses
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `service_consumer_assertion`; `LicensePurpose`; `Do-Not-Sign`, and; `Service-Headers`
- The `Authorisation` parameters MUST contain "Bearer" + access token value

PATCH /service_provider/service

Example service of a Service Provider

- An iSHARE-adherent Service Provider MUST apply iSHARE conformant OAuth to every iSHARE enabled service
- An iSHARE enabled service MAY have any name the Service Provider chooses
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `service_consumer_assertion`; `LicensePurpose`; `Do-Not-Sign`, and; `Service-Headers`
- The Authorisation parameters MUST contain "Bearer" + access token value

DELETE /service_provider/service

Example service of a Service Provider

- An iSHARE-adherent Service Provider MUST apply iSHARE conformant OAuth to every iSHARE enabled service
- An iSHARE enabled service MAY have any name the Service Provider chooses
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `service_consumer_assertion`; `LicensePurpose`; `Do-Not-Sign`, and; `Service-Headers`
- The Authorisation parameters MUST contain "Bearer" + access token value

Entitled Party

GET /entitled_party/oauth2.0/token

Used to obtain an OAuth access token from an Entitled Party

- The service request MUST contain the parameters `grant_type`; `client_id`; `client_assertion_type`, and; `client_assertion`
- The service request MAY contain the parameter `scope`
- The `grant_type` parameter MUST contain "client_credentials"
- The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Consumer
- The `client_assertion_type` parameter MUST contain "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- The `client_assertion` parameter MUST contain a JWT token conform iSHARE specifications, signed by the client

GET /entitled_party/ishare1.0/delegation

Used to obtain delegation evidence from an Entitled Party

- A Service Provider MUST validate that the Entitled Party only provides information about his own delegations
- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameters `scope`, and; `service_consumer_assertion`
- The `Authorisation` parameter MUST contain "Bearer " + access token value

Authorisation Registry

GET /authorisation_registry/oauth2.0/token

Used to obtain an OAuth access token from an Authorisation Registry

- The service request MUST contain the parameters `grant_type`; `client_id`; `client_assertion_type`, and; `client_assertion`
- The service request MAY contain the parameter `scope`
- The `grant_type` parameter MUST contain "client_credentials"
- The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Consumer
- The `client_assertion_type` parameter MUST contain "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- The `client_assertion` parameter MUST contain a JWT token conform iSHARE specifications, signed by the client

GET /authorisation_registry/ishare1.0/delegation

Used to obtain delegation evidence from an Authorisation Registry

- The service request MUST contain the parameters `Authorisation`
- The service request MAY contain the parameters `scope`, and; `service_consumer_assertion`
- The `Authorisation` parameter MUST contain "Bearer " + access token value

Identity Provider

GET /identity_provider/openid_connect1.0/authorize

OpenID Connect end-point for redirecting a Human Service Consumer for authentication by the Identity Provider

- The service request MUST contain the parameters `response_type`; `client_id`; `redirect_uri`; `scope`, and; `state`
- For the `response_type` parameter, using the Authorization Code Flow with value 'code' is REQUIRED
- The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Provider
- The `scope` parameter MUST contain the 'openid' scope value and MAY contain 'name'; 'contact_details'; 'company_id', and; 'company_info' scope value(s)

GET /identity_provider/openid_connect1.0/token

OpenID Connect end-point for obtaining the OAuth access token and OpenID Connect id token

- The service request MUST contain the parameters `grant_type`; `code`; `redirect_uri`; `client_id`; `client_assertion_type`, and; `client_assertion`
- The `grant_type` parameter MUST contain "authorization code"
- The `code` parameter MUST contain value of authorisation code received from the Identity Provider
- The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Provider
- The `client_assertion_type` parameter MUST contain "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- The {{client_assertion parameter}} MUST contain a JWT token conform iSHARE specifications, signed by the client

GET /identity_provider/openid_connect1.0/userinfo

OpenID Connect end-point for obtaining attributes of a Human Service Consumer conform scope defined in access token

- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameter `Do-Not-Sign`
- The `Authorisation` parameter MUST contain "Bearer " + access token value

Scheme owner

GET /scheme_owner/auth2.0/token

Used to obtain an OAuth access token from the iSHARE Scheme Owner

- The service request MUST contain the parameters `grant_type`; `client_id`; `client_assertion_type`, and; `client_assertion`
- The service request MAY contain the parameter `scope`
- The `grant_type` parameter MUST contain “client_credentials”
- The `client_id` parameter MUST contain a valid iSHARE identifier of the Service Provider
- The `client_assertion_type` parameter MUST contain “urn:ietf:params:oauth:client-assertion-type:jwt-bearer”
- The `client_assertion` parameter MUST contain a JWT token conform iSHARE specifications, signed by the client

GET /scheme_owner/ishare1.0/parties/{party_id}

Used to obtain adherence information on an iSHARE participant from the iSHARE Scheme owner

- The service request MUST contain the parameters `Authorisation`, and; `party_id`
- The service request MAY contain the parameter `date_time`
- The `Authorisation` parameter MUST contain "Bearer" + access token value
- The `date_time` parameter MUST be cacheable

GET /scheme_owner/ishare1.0/parties/certified_parties

Used to obtain certification information on all iSHARE participants from the iSHARE Scheme owner

- The service request MUST contain the parameter `Authorisation`
- The service request MAY contain the parameter `date_time`
- The `Authorisation` parameter MUST contain "Bearer" + access token value
- The `date_time` parameter MUST be cacheable

GET /scheme_owner/trusted_list

Used to obtain the iSHARE trusted list of certificate/seal roots from the iSHARE Scheme owner

- The service request MUST contain the parameter `Authorisation`
- The `Authorisation` parameter MUST contain "Bearer" + access token value

GET /scheme_owner/ishare1.0/certificate_validation

Used to assess whether a PKI certificate is valid and trusted under iSHARE

- The service request SHOULD not be used more than x per y for each certificate
- The service request MUST contain the parameters Authorisation, and; certificate
- The Authorisation parameter MUST contain "Bearer" + access token value

'Language' of Delegation and Authorisation

As suggested in the fourth Technical working group meeting, this is a first attempt to specify an iSHARE language for delegation and authorisation. We need this delegation and authorisation language to specify statements from the Entitled Party or Authorisation Registry. Possibly also for the restrictions of tokens. Preferably we use an existing standard (like a JSON port XACML 3.0).

Below you will find a list of authorisation and delegation cases that need to be supported both in natural language and JSON format, followed by a suggestion for "types of access" one can have. Lastly, a suggestion is done on how to represent delegation evidence in JWT/JWS format.

General cases that must at least be supported

| Delegation and authorisation cases in natural language | JSON |
|--|---|
| I, Party X, grant Party Y read access to the ETA of Container #12345 | <pre> { "NotBefore":"2017-02-24T15:04:29.329Z", "NotOnOrAfter":"2017-04-24T15:04:29.329Z", "MaxDelegationDepth":"2", "DelegationActive":"TRUE", "Delegator": { "type": "NL.KVK", "value": "Party X 12345678" }, "type":"EU.EORI", "value":"Party Y NL123456789", "DelegatedResource": { "name":"OBJECTS.CONTAINER", "value":"12345" }, "DelegatedAction": { "action":"READ" }, "attributes": { "value":"ETA" } } </pre> |

I, Party X, grant Party Y read access to the ETA of all my Containers

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{ "type": "NL.KVK", "value": "Party X 12345678" },
  "type":"EU.EORI",
  "value":"Party Y NL123456789"
  "DelegatedResource":
  {
    "name":"OBJECTS.CONTAINER",
    "value":"*"
  },
  "DelegatedAction":
  {
    "action":"READ"
  },
  "attributes":
  {
    "value":"ETA"
  }
}
```

I, Party X, grant Party Y read access to all information of Container #12345

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{ "type": "NL.KVK", "value": "Party X 12345678" },
  "type":"EU.EORI",
  "value":"Party Y NL123456789"
  "DelegatedResource":
  {
    "name":"OBJECTS.CONTAINER",
    "value":"12345"
  },
  "DelegatedAction":
  {
    "action":"READ"
  },
  "attributes":
  {
    "value":"*"
  }
}
```

I, Party X, grant Party Y read access to all my information

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{ "type": "NL.KVK", "value": "Party X 12345678" },
  "type":"EU.EORI",
  "value":"Party Y NL123456789"
  "DelegatedResource":
  {
    "name":"*",
    "value":"*"
  },
  "DelegatedAction":
  {
    "action":"READ"
  },
  "attributes":
  {
    "value":"*"
  }
}
```

I, Shipper A, grant RWS read access to my 'hazardous goods information' if my ship is within 5 miles of critical infrastructure

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{ "type": "NL.KVK", "value": "Shipper A" },
  "type":"EU.EORI",
  "value":"RWS"
  "DelegatedResource":
  {
    "name":"*",
    "value":"*"
  },
  "DelegatedAction":
  {
    "action":"READ"
  },
  "attributes":
  {
    "NAME":"some value",
    "SPEED":"some value",
    "DIRECTION":"some value",
    "CONTAINS_DANGEROUS_GOODS":"some value"
  },
  "Condition":
  {
    "name":"RANGE_IN_KM",
    "value":"5"
  }
}
```

I, RWS, grant read access to the police to all objects that have a 'calamity' flag raised

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{ "type": "NL.KVK", "value": "RWS" },
  "type":"Investigation service",
  "value":"PLW001828"
  "DelegatedResource":
  {
    "name":"*",
    "value":"*"
  },
  "DelegatedAction":
  {
    "action":"READ-"
  },
  "attributes":
  {
    "value":"*"
  },
  "Condition":
  {
    "flag":"CALAMITY",
    "boolean":"TRUE"
  }
}
```

Real life example cases:

| Delegation and authorisation cases in natural language | JSON |
|--|--|
| <p>I, Planner from Carrier X, grant Substitute Carrier Y the right to delegate to third parties (such as other carriers and drivers)</p> | <pre> { "NotBefore":"2017-02-24T15:04:29.329Z", "NotOnOrAfter":"2017-04-24T15:04:29.329Z", "MaxDelegationDepth":"2", "DelegationActive":"TRUE", "Delegate":{ "type": "NL.KVK", "value": "Carrier X" }, "type":"EU.EORI", "value":"Substitute Carrier Y", "DelegatedResource": { "name":"*", "value":"*" }, "DelegatedAction": { "action":"*" }, "attributes": { "value":"*" } } </pre> |

I, Planner from Carrier X, grant Substitute Carrier Y the right to delegate to third parties (such as other carriers and drivers), except the right to delegate

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "Carrier X"
  },
  "type": "EU.EORI",
  "value": "Substitute Carrier Y",
  "DelegatedResource":
  {
    "name": "*",
    "value": "*"
  },
  "DelegatedAction":
  {
    "action": "*"
  },
  "attributes":
  {
    "value": "*"
  }
}
```

I, Carrier X, grant ILT (Inspectie Leefomgeving en Transport) the right to view freight orders

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "Carrier X"
  },
  "type": "EU.EORI",
  "value": "ILT",
  "DelegatedResource":
  {
    "name": "OBJECTS.FREIGHTORDER",
    "value": ""
  },
  "DelegatedAction":
  {
    "action": "READ"
  },
  "attributes":
  {
    "value": ""
  }
}
```


I, Carrier X, grant Party Y the right to view RTIs (Reusable Transport Item) on the CMR (Convention Relative au Contract de Transport International de Marchandises par la Route)

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "Carrier X"
  },
  "type": "EU.EORI",
  "value": "Party X",
  "DelegatedResource":
  {
    "name": "OBJECTS.CMR",
    "value": ""
  },
  "DelegatedAction":
  {
    "action": "READ"
  },
  "attributes":
  {
    "value": "RTI"
  }
}
```

I, Carrier X, grant IMS the right to update ETA on the CMR (Convention Relative au Contract de Transport International de Marchandises par la Route)

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "Carrier X"
  },
  "type": "EU.EORI",
  "value": "IMS",
  "DelegatedResource":
  {
    "name": "OBJECTS.CMR",
    "value": ""
  },
  "DelegatedAction":
  {
    "action": "UPDATE"
  },
  "attributes":
  {
    "value": "ETA"
  }
}
```

| Delegation and authorisation cases in natural language | JSON |
|--|--|
| <p>I, Ship X, grant Party Y the right to view the AIS (Automatic Identification System) during the journey from Basil to Rotterdam</p> | <pre>{ "notBefore":"2017-02-24T15:04:29.329Z", "notOnOrAfter":"2017-04-24T15:04:29.329Z", "maxDelegationDepth":"1", "delegationActive":"TRUE", "delegator": { "identifiers": [{ "type": "EU.EORI", "value": "Jan de Rijk NL123456789" }, { "type": "NL.KVK", "value": "Jan de Rijk NL123456111" }] }, "delegatee": { "identifiers": [{ "type": "NL.KVK", "value": "Portbase NL123456788" }] }, "delegatedResource": { "serviceProvider": ["TransFollow"], "name": "OBJECTS.FREIGHT_DOCUMENT", "value": "ALL_CREATED_BY_THIS_RULE" }, "delegatedActions":</pre> |

```
{
  "actions": ["CREATE","READ","UPDATE"]
},
"conditions":
[
  {
    "name": "departure",
    "value": "Basel"
  },
  {
    "name": "direction",
    "value": "Rotterdam"
  }
],
"attributes":
{
  "value": "AIS"
}
}
```

I, Ship X, grant RWS the right to view the AIS (Automatic Identification System) and cargo

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{
    "type":"NL.KVK",
    "value":"Ship X"
  },
  "type":"EU.EORI",
  "value":"RWS"
  "DelegatedResource":
  {
    "name":"OBJECTS.SHIP",
    "value":"*",
    "name":"OBJECTS.CARGO",
    "value":"*"
  },
  "DelegatedAction":
  {
    "action":"READ"
  }
  "attributes":
  {
    "value":"AIS",
    "value":"CARGO"
  }
}
```

I, Ship X, grant RWS the right to view the AIS (Automatic Identification System) and cargo

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{
    "type": "NL.KVK",
    "value": "Ship X"
  },
  "type":"EU.EORI",
  "value":"RWS"
  "DelegatedResource":
  {
    "name":"OBJECTS.SHIP",
    "value":"AIS"
  },
  "DelegatedAction":
  {
    "action":"READ-"
  }
}
```

I, Expeditor X, grant Airfreight Company Y the right to update weight and dimension of shipment item

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate":{
    "type":"NL.KVK",
    "value":"Expeditor X"
  },
  "type":"EU.EORI",
  "value":"Airfreight Company Y"
  "DelegatedResource":
  {
    "name":"OBJECTS.CARGO",
    "value":"dimension",
    "value":"weight"
  },
  "DelegatedAction":
  {
    "action":"UPDATE"
  }
}
```

| | |
|--|---|
| <p>I, Ship X, grant Expeditor Y the right to view (and subsequently publish) the ETA, if I carry his cargo</p> | <pre> { "NotBefore":"2017-02-24T15:04:29.329Z", "NotOnOrAfter":"2017-04-24T15:04:29.329Z", "MaxDelegationDepth":"2", "DelegationActive":"TRUE", "Delegate":{ "type": "NL.KVK", "value": "Ship X" }, "type":"EU.EORI", "value":"Expeditor Y" "DelegatedResource": { "name":"OBJECTS.CARGO", "value":"" }, "DelegatedAction": { "action":"READ" } "attributes": { "value":"ETA", }, "Condition": { "carrier":"SHIP X", "classification":"public" } } </pre> |
| | |

| Delegation and authorisation cases in natural language | JSON |
|--|--|
| I, APM Terminals, grant Customs the right to clear documents for every container | <pre> { "NotBefore":"2017-02-24T15:04:29.329Z", "NotOnOrAfter":"2017-04-24T15:04:29.329Z", "MaxDelegationDepth":"2", "DelegationActive":"TRUE", "Delegate":{ "type": "KVK", "value": "APM Terminals" }, "type":"Investigation Service", "value":"Customs" "DelegatedResource": { "name":"OBJECTS.CONTAINER", "value":"" }, "DelegatedAction": { "action":"READ" } "attributes": { "value":"" }, "Condition": { "clearance":"TRUE" } } </pre> |

I, Party X, grant Party Y to pick up containers to APM Terminals

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "Party X 12345678"
  },
  "type": "EU.EORI",
  "value": "Party X"
  "DelegatedResource":
  {
    "name": "OBJECTS.CONTAINER",
    "value": "*"
  },
  "DelegatedAction":
  {
    "action": "READ"
  }
  "attributes":
  {
    "value": "*"
  },
  "Condition":
  {
    "destination": "APM Terminals"
  }
}
```

I, APM Terminals, grant Party X the right to provide clearance

```
{
  "NotBefore":"2017-02-24T15:04:29.329Z",
  "NotOnOrAfter":"2017-04-24T15:04:29.329Z",
  "MaxDelegationDepth":"2",
  "DelegationActive":"TRUE",
  "Delegate": {
    "type": "NL.KVK",
    "value": "APM Terminals"
  },
  "type": "EU.EORI",
  "value": "Party X"
  "DelegatedResource":
  {
    "name": "OBJECTS.CARGO",
    "value": "*"
  },
  "DelegatedAction":
  {
    "action": "READ"
  }
  "attributes":
  {
    "value": "*",
  },
  "Condition":
  {
    "clearance": "TRUE"
  }
}
```

| | |
|--|---|
| I, Secure Logistics, grant Portbase the right to read company information with EAN | <pre> { "NotBefore":"2017-02-24T15:04:29.329Z", "NotOnOrAfter":"2017-04-24T15:04:29.329Z", "MaxDelegationDepth":"2", "DelegationActive":"TRUE", "Delegate": { "type": "NL.KVK", "value": "Secure Logistics" }, "type": "EU.EORI", "value": "Portbase" "DelegatedResource": { "name": "OBJECTS.ORGANISATION", "value": "*" }, "DelegatedAction": { "action": "READ" } "attributes": { "value": "EAN", } } </pre> |
|--|---|

Types of access*

Within iSHARE the following operations are defined:

| Operation | Description |
|-----------|--|
| Create | Allows a Service Consumer to create new data at the Service Provider |

| | |
|-----------------|--|
| Read | Allows a Service Consumer to view data from the Service Provider |
| Read- | Allows a Service Consumer to view anonymised data from the Service Provider. In order to use these rights a Service Provider MUST have available this kind of data. It is not an iSHARE requirement to have available this kind of data, nor does iSHARE what is required to make data anonymised |
| Update | Allows a Service Consumer to modify data at the Service Provider |
| Delete | Allows a Service Consumer to remove data from the Service Provider |
| DelegatedAction | Indicates that an Entitled Party passes on its rights to another party. An Entitled Party SHOULD specify how many times rights can be delegated. However, this can never exceed two times. If an Entitled Party does not specify this, every party MUST assume rights can be delegated only once |

As a result any combination of rights can be expressed.

| | Create | Read | Read- | Update | Delete | DelegatedAction |
|---------|--------|------|-------|--------|--------|-----------------|
| RIGHT_1 | | X | | | | |
| RIGHT_2 | | X | | X | | |
| RIGHT_3 | | | X | | | |
| RIGHT_N | X | X | | X | X | N |

*Please note that the rights established here will be updated according to the latest insights on [licenses](#) soon

Possible representation of delegation evidence using JWT/JWS format

The most logical presentation of delegation evidence seems to be a signed JWT/JWS, format
<header>.<payload>.<signature>

JWT Header

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

JWT Payload

```
{
  "NotBefore": "2017-02-24T15:04:29.329Z",
}
```

```

    "NotOnOrAfter": "2017-04-24T15:04:29.329Z",
    "MaxDelegationDepth": "2",
    "DelegationActive": "yes",
    "Delegate": {
      "type": "EU.EORI",
      "value": "NL123456789"
    },
    "type": "NL.KVK",
    "value": "12345678",
    "DelegatedAction":
  [
    {
      "DelegatedResource":
    [
      {
        "name": "OBJECTS.CONTAINER",
        "value": "12345"
      },
      {
        "name": "OBJECTS.CONTAINER",
        "value": "67890"
      }
    ],
    "attributes":
  {
    "NAME": "some value",
    "SPEED": "some value",
    "DIRECTION": "some value",
    "CONTAINS_DANGEROUS_GOODS": "some value"
  },
    "Condition":
  [
    {
      "name": "RANGE_IN_KM",
      "value": "5"
    }
  ],
    "DelegatedAction":
  {
    "action": "READ",
    "DELEGATE": "1"
  }
  ],
  {
    "DelegatedResource":
  [
    {
      "name": "OBJECTS.CONTAINER",
      "value": "*"
    }
  ],
    "attributes": "*",
    "Action": "READ-",
  }
}

```

}

Delegation rules

In this page the rules are described, to which the processes of delegation should adhere. The rules will be implemented as policies in the policy information point(s) (PIP). The PIP provides the attribute values to the policy decision point (PDP) needed to make the decisions about delegations and authorisations.

Delegation can be explained as the act of empowering to act for another or to represent other(s). In the iSHARE scheme delegation always pertains to authorisation. Thus, one party can delegate another party to have access to services (data) on his or her behalf. However, the party who delegates always remains accountable for whichever actions are performed by the party to whom authorisations are delegated. In other words, accountability can never be delegated.

Delegation chain

A party to whom authorisations are delegated is allowed to delegate the same authorisations (or a subset thereof) to yet another party. This can occur with a total maximum of *two (2) times* (expressed by MaxDelegationDepth), excluding the originating party. The delegation information (token) must always contain the identity information of all previous delegating parties, including the originating party. It is the responsibility of the delegating party to know and trust the party to whom authorisations are delegated.

If any party revokes its delegation, all parties down the delegation chain will lose their authorisations that are acquired from the same delegation.

Delegation conditions

The operations (rights, actions) that are defined within the iSHARE scheme (see the table further below) may be subject to certain conditions. For example, a delegated party may read certain data and subsequently publish that data; or a delegated party may read certain data and subsequently share that data with other parties within the iSHARE scheme. Those conditions should adhere to the internal policies of the service provider. In turn, these policies are based on the data classification definitions of the service provider. The conditions only apply to the operations Read and Read-, as all other operations (i.e. Create, Update, Delete and DelegatedAction) can be enforced technically. Any conditions should be expressed in the JSON delegation policies of the PIP.

iSHARE conditions on the operations Read and Read-:

| Condition | Description |
|------------------|--|
| Public | Data is intended to be shared with other parties outside the iSHARE scheme |
| For internal use | Data is intended to be shared with parties within the iSHARE scheme only |
| Confidential | Data is intended to be shared with explicitly designated parties within of outside the iSHARE scheme |

| Condition | Description |
|-----------|---|
| Secret | Data is intended to be shared with explicitly designated parties within the iSHARE scheme only Security and privacy are the primary criteria here, efficiency and cost are secondary |

iSHARE operations:

| Operation | Description |
|-----------------|---|
| Create | Allows a Service Consumer to create new data at the Service Provider |
| Read | Allows a Service Consumer to view data from the Service Provider |
| Read- | Allows a Service Consumer to view anonymised data from the Service Provider. In order to use these rights a Service Provider MUST have available this kind of data. It is not an iSHARE requirement to have available this kind of data, nor does iSHARE what is required to make data anonymised |
| Update | Allows a Service Consumer to modify data at the Service Provider |
| Delete | Allows a Service Consumer to remove data from the Service Provider |
| DelegatedAction | Indicates that an Entitled Party passes on its rights to another party. An Entitled Party SHOULD specify how many times rights can be delegated. However, this can never exceed two times. If an Entitled Party does not specify this, every party MUST assume rights can be delegated only once |

Specification

The [XACML v3.0 Administration and Delegation Profile Version 1.0](#) will be used as the specification to define and implement delegation of authority in the iSHARE scheme. However, XACML v3.0 has been defined in traditional XML format, which is not lightweight enough for most use cases in the iSHARE scheme. Therefore, a JSON profile will be used to implement the delegation policies instead.

Notice that the XACML v3.0 specification has defined a [JSON profile for XACML requests and responses](#) only, not for the XACML policies. For this reason, the iSHARE scheme needs to provide for a JSON profile itself for implementing delegation of authority.

The [RBAC \(Role-Based Access Control\) specification](#) will not be implemented in the iSHARE scheme, since this is a local responsibility of the participating parties. However, the iSHARE scheme will provide for an attribute (DelegatedRole; see the table below) that can be used to support roles.

Terminology

The XACML v3.0 specification has defined the following terms as related to delegation of authority.

| Definition | Explanation |
|------------------------|--|
| Access policy | A policy that governs access |
| Access request | A request to determine whether access to a resource should be granted |
| Administrative policy | A policy that authorizes a delegate to issue policies about constrained situations |
| Administrative request | A request to determine whether a policy was issued by an authorized source |
| Backward Chaining | Finding a chain of administrative and access policies beginning with an access policy, such that each policy is authorized by the next one |
| Delegator | Someone authorized by an administrative policy to issue policies |
| Delegatee | Someone to whom something is delegated |
| Forward chaining | Finding a chain of administrative and access policies beginning at a trusted policy, such that each policy authorizes the next one |
| Issuer | A set of attributes describing the source of a policy |
| Reduction | The process by which the authority of a policy associated with an issuer is verified or discarded |
| Situation | A set of properties delineated by the Attributes elements of an access request context |
| Trusted policy | A policy without a PolicyIssuer element |

JSON attributes

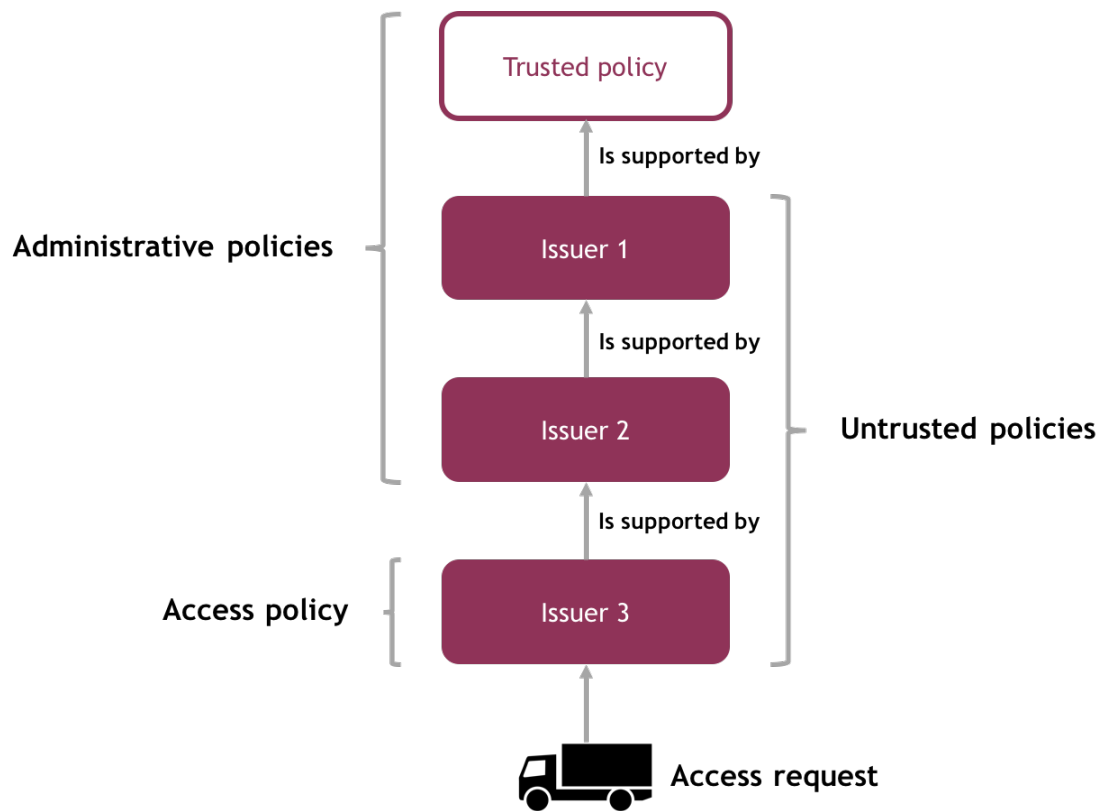
The following attributes that are related to the subject of delegation of authority will be used in the JSON profiles of the iSHARE scheme.

| Attribute | Explanation |
|---------------|---|
| Action | The action or operation that the subject is allowed to, i.e. Create, Read, Read- (view anonymised data), Update, Delete |
| Condition | The condition or conditions, under which the action(s) or operation(s) are allowed |
| Delegator | The subject authorised by an administrative policy to issue policies |
| Delegatee | The subject to whom something is delegated |
| DelegatedRole | The role that is authorised by an administrative policy to perform a delegated action or operation |

| | |
|--------------------|---|
| DelegatedAction | The delegated action or operation that the subject is allowed to, i.e. Create, Read, Read- (view anonymised data), Update, Delete |
| DelegatedResource | The object to which the delegated action or operation is allowed, i.e. Create, Read, Read- (view anonymised data), Update, Delete |
| DelegationActive | The boolean that determines whether the delegation is active or not |
| MaxDelegationDepth | The integer indicating the maximum depth of delegation that is authorised by the policy, excluding the initial node |
| NotBefore | The condition specifying the date and/or time, before which the delegation and delegated action(s) are not valid |
| NotOnOrAfter | The condition specifying the date and/or time, on or after which the delegation and delegated action(s) are not valid |
| Policy | The set of rules that is evaluated by the PDP, each time a subject performs an action or operation |
| PolicyIssuer | The source of the policy. A missing PolicyIssuer attribute means that the policy is trusted |
| Resource | The object to which the action or operation is allowed, i.e. Create, Read, Read- (view anonymised data), Update, Delete |

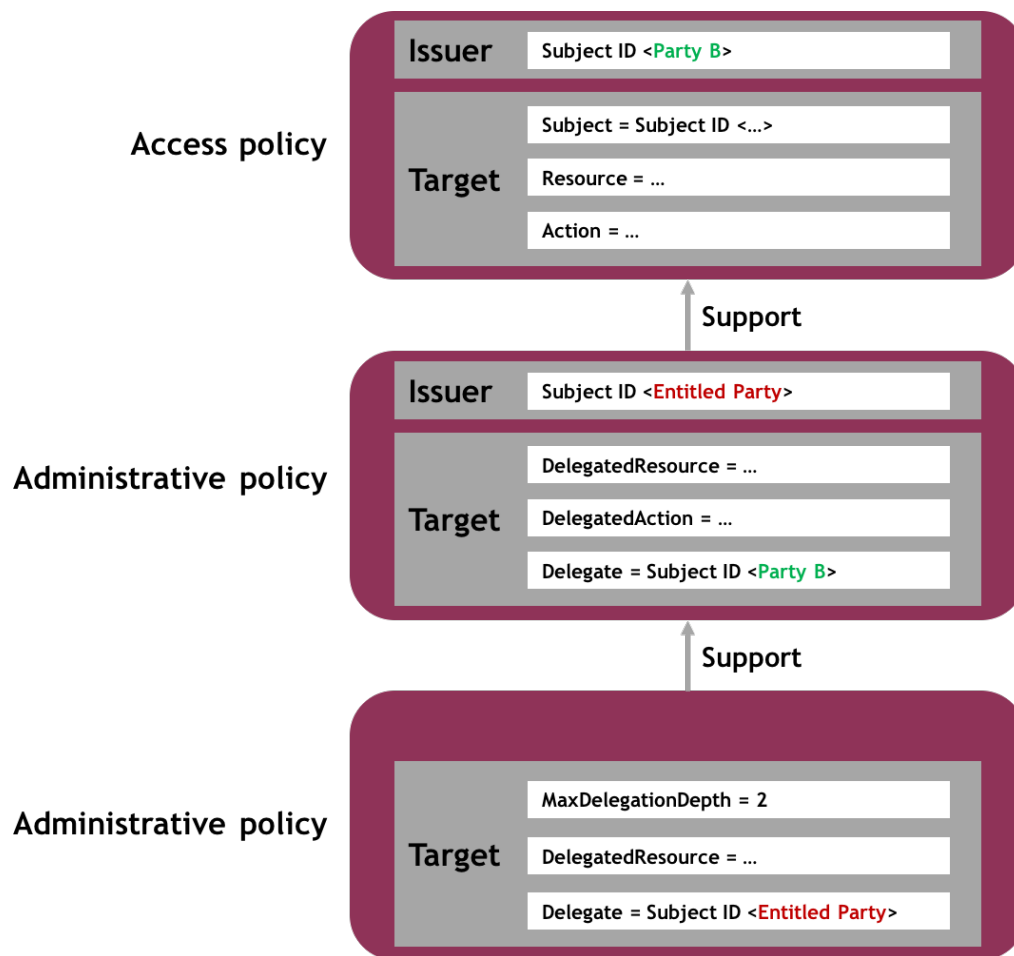
Delegation policy chain

The following figure depicts the chain of delegation policies as it will be implemented in the iSHARE scheme.



Delegation policy architecture

The following figure depicts the architecture with regard to delegation policies as it will be implemented in the iSHARE scheme.



The JSON attributes applied to primary use case 1b

The following table shows which JSON attributes are applied to [primary use case 1b M2M service provision based on delegation](#).

| Attribute | Explanation |
|-------------------|--|
| Action | The Service Consumer performs an action or operation at the Service Provider (e.g. read data) |
| Condition | Not applicable in this use case |
| Delegate | The Entitled Party and the delegated party that issue policies |
| DelegatedAction | The delegated Service Consumer performs an action or operation at the Service Provider (e.g. read data) |
| DelegatedResource | The service at the Service Provider, to which the delegated action or operation of the Service Consumer is allowed (e.g. data) |

| | |
|--------------------|---|
| DelegationActive | TRUE |
| MaxDelegationDepth | Not applicable in this use case |
| NotBefore | Not applicable in this use case |
| NotOnOrAfter | Not applicable in this use case |
| Policy | The set of rules at the PIP of the Service Provider that is evaluated by the PDP of the same Service Provider, each time the (delegated) Service Consumer performs an action or operation |
| PolicyIssuer | The source of the policy, i.c. both the Entitled Party and the delegated party (who has also become an Entitled Party through delegation) |
| Resource | The service at the Service Provider, to which the action or operation of the Service Consumer is allowed (e.g. data) |

Token and delegation lifetime

The iSHARE scheme requires a window of time, during which signed tokens and delegation evidence are considered valid. Each token has a timestamp attribute as well as a time-to-live attribute indicating the allowable lifetime of the token (in milliseconds) after the token timestamp. Tokens that contain authorization and/or delegation information should always have an expiration time, so that the time is limited a potential attacker can abuse a token that is intercepted by attacks (such as a man-in-the-middle attack, a session hijacking attack or a replay attack), so that the risk of impersonation or unauthorised access will be reduced.

Notice that the validity of delegation evidence may not only be determined by duration, but also by the number of times it is allowed to be used.

The Service Provider may also want to determine the lifetime during which it is allowed to access its services (data). This lifetime may overrule the central lifetime. The following provides some guidelines and a structure that can be used to determine the lifetime of the tokens.

The more sensitive the information that is accessed at the Service Provider with the token, the shorter the lifetime of the token should be. Also, the more intrusive the access rights a token provides, the shorter the lifetime of the token should be. The value of the lifetime of a token must be in milliseconds, whereas the value of the timestamp of a token must be in Unix time, i.e. the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT).

Refresh token

Tokens can be refreshed, but only for Human Service Consumers. Refresh tokens carry the information necessary to get a new token. A prerequisite to get a refresh token is that successful authentication of the Human Service Consumer is required.

The following use cases apply to refresh tokens:

- A token has expired
- A Human Service Consumer accesses a new resource for the first time

Attributes

The following attributes must be used to express the timestamp and lifetime of tokens respectively:

- "Timestamp" : "some value"
- "TokenLiveTime" : "some value"

Error messages

Once the token has expired, the following error message will be displayed:

```
error="invalid_token",
error_description="The token has expired"
```

Relevant standards

This part of the iSHARE scheme is considered normative and is therefore compliant with RFC 2119.

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, authorisation, cryptography, data exchange, data formatting).

| Technical standard (in alphabetical order) | Authentication | Authorisation | Cryptography | Data exchange | Data formatting | Description |
|---|----------------|---------------|--------------|---------------|-----------------|--|
| JSON | | | | | ✓ | <i>Formatting/structuring data in object units</i> |
| OAuth | (✓)** | ✓ | | | | <i>Standard for authorisation (delegated access control, password handling) Version 2.0 MUST be used</i> |

| | | | | | | |
|-----------------------|------|------|---|---|------|--|
| OpenID Connect | ✓ | | | | | <i>Authentication layer built on top of OAuth 2.0 protocol</i> |
| SAML | (✓)* | (✓)* | | | ✓ | <i>XML-based data format for exchange of authentication and authorisation data Version 2.0 MUST be used</i> |
| SOAP | | | | ✓ | (✓)* | <i>Network protocol for the exchange of structured information</i> |
| TLS | | | ✓ | | | <i>Cryptographic protocol for secure communication of computer networks Version 1.2 MUST be used SSL (either version) MUST NOT be used</i> |
| UMA | | ✓ | | | | <i>OAuth-based access management protocol standard</i> |
| X.509 | | | ✓ | | | <i>Cryptographic standard for PKI's (digital certificates & keys) Version 3.0 MUST be used</i> |
| XACML | | ✓ | | | | <i>Standard for authorisation policies (language, architecture, processing model) Either version 2.0 or 3.0 MUST be used</i> |
| XML | | | | | ✓ | <i>Formatting/structuring text documents to be both human- and machine-readable</i> |
| XML Signature | | | ✓ | | (✓)* | <i>Standard defining an XML syntax for digital signatures to sign XML documents</i> |

(✓)*: 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 exchanged 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 authorisation which is used by i.e. Google, Facebook, Microsoft, Twitter etc. to let their users exchange information about their accounts with other applications or websites. OAuth is designed to work with HTTP.

Through OAuth users can authorise third party applications or websites to access their account information on other "master" systems without the need of exchanging 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 authorise third parties access to their resources without exchanging their credentials (username, password). Authorisation 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 authorisation.

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 authorisation policies but can be use within XACML for ownership consent and access delegation.

OAuth 2.0

OAuth 2.0 provides specific authorisation 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 Connect (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 authorisation 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 authorisation 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 an adapted version of OpenID, combined with OAuth 2.0.

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 the 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 authorisation 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 authorised 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 authorisation, 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 blocks

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 authorisation 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 authorisation of data exchange and other protected-resource access made between online services on the owner’s behalf or with the owner’s authorisation 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 authorisation requests based on the policies and rules.

UMA has no restrictions regarding the policy format, as the Authorisation 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 authorisation structure can be integrated into one authorisation scheme. Authorisations and the rules surrounding it can be managed centrally regardless of authorisation mechanism of the applications themselves. This phenomenon is called externalisation. 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).

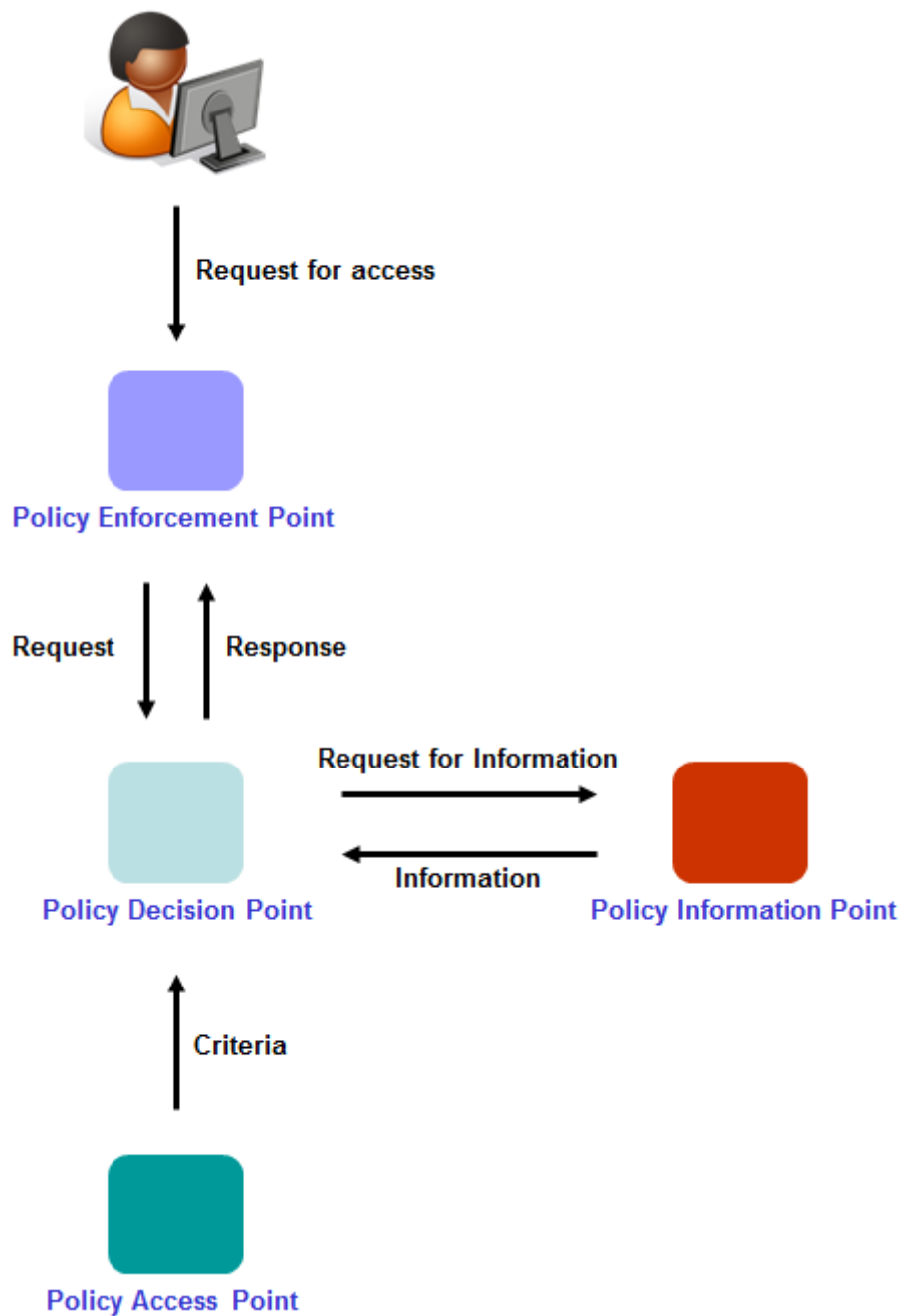
Moreover, with the help of XACML authorisations can be arranged and managed in detail. This is called fine-grained authorisation. XACML supports the use of security labels, rules with arbitrary attributes, rules with a certain duration and dynamic rules.

In XACML two main functions can be distinguished. One function defines the criteria with which authorisations are assigned, such as 'only an experienced user from department X is allowed 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 authorisations 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 organise 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](#).

Legal

This section covers the legal specifications of the iSHARE scheme. The legal specifications consist of three documents:

1. Accession Agreement (separate versions for Adhering Parties and Certified Parties);

The Accession Agreement is a bilateral agreement between each separate participant and the [Scheme Owner](#) (or [Governing Body](#)). The Operational working group will determine who the Scheme Owner will be at a later stage. By signing the Accession Agreement, an Adhering or Certified Party becomes a participant of the iSHARE Scheme, and declares and agrees with the conditions set forth in the Accession Agreement and the Terms of Use. A participant also declares that it will abide to all relevant laws and regulations that apply to its business, amongst which the laws and regulations described in the Legal Framework.

2. Terms of Use;

The Terms of Use are an appendix to and integral part of the Accession Agreement. The Terms of Use further define the rights and obligations of the various roles within the iSHARE scheme. The Terms of Use provide a uniform set of rules for both the participants and the scheme owner, thereby fostering a level playing field between all parties involved.

The Terms of Use are drafted in such a way that data can be exchanged by participants even if they have no other contractual arrangement in place. In that case, the default requirements as set forth in the Terms of Use govern their legal relationship. This includes the (license) conditions that apply to the exchange of data. But the Terms of Use leave room for participants to derogate from or further detail the provisions of the Terms of Use on a bilateral basis. However, there will be certain requirements that participants should comply with at any time, and from which they will not be able to deviate. These are the requirements that deal with the proper functioning of the iSHARE Scheme, such as each party's responsibility to safeguard the security of its IT-systems (articles 3.5 and 4.1).

Furthermore, the Terms of Use include a number of annexes, amongst which the pre-defined conditions of exchange, the Legal Framework and the iSHARE Scheme standards and specifications.

3. Legal Framework.

The Legal Framework deals with the legal context of the iSHARE Scheme. It describes the laws and regulations that are of particular importance for participants when exchanging data within the iSHARE Scheme: the eIDAS regulation, the General Data Protection Regulation, competition law and the Dutch civil code. As stipulated in the Accession Agreement and the Terms of Use, all participants are expected to comply with these and all other applicable national and international pieces of legislation.

Accession Agreement for Adhering Parties

ACCESSION AGREEMENT FOR PARTICIPATION

ADHERING PARTIES - iSHARE SCHEME

The Scheme Owner and the [COMPANY NAME] (hereafter: ‘the Adhering Party’) enter into an agreement which specifies the terms and conditions under which:

- the Adhering Party shall participate in the exchange of Data under the rules and specifications of the iSHARE Scheme;
- the Scheme Owner shall [SPECIFY ROLE SCHEME OWNER IN RESPECT OF ADHERING PARTIES].

The Adhering Party hereby declares to comply with the following rules for participation in the iSHARE Scheme:

- The Adhering Party must comply to the iSHARE specific requirements <TBD by other working group> as defined in the (annexes to the) Terms of Use.
- The Adhering Party can apply for participation in the iSHARE Council of Participants and the Change Advisory Board as defined in the statutes of the Scheme Owner <not yet set up or established / depending on further development and role of the iSHARE Scheme>.
- The Adhering Party agrees with and accepts the Terms of Use as specified in Appendix 1.
- Participation in the iSHARE Scheme is subject to a [TBD: ANNUAL/MONTHLY] participation fee. Participation fees are non-refundable fees and are stated in Appendix 2. The Scheme Owner may adjust participation fee rates once a year with effect from January the 1st with two (2) months’ prior written notice to the Adhering Party.
- The Scheme Owner’s invoices are due upon receipt and must be fully paid within 30 days after the invoice date.

Duration

The Accession Agreement is entered into for an initial period of twelve (12) months. During the initial period, the Adhering Party may only terminate the Accession Agreement as set forth in the Terms of Use. After the initial period, the Accession Agreement shall be tacitly extended for an indefinite period of time and may be terminated subject to the notice period as stated in the Terms of Use.

The Adhering Party declares compliance to all rules set forth in this Accession Agreement, including the referenced appendices.

| | | |
|--|-----------------------|---------------------|
| | Adhering Party | Scheme Owner |
|--|-----------------------|---------------------|

| | | |
|-----------|--|--|
| Name | | |
| Company | | |
| Place | | |
| Date | | |
| Signature | | |

APPENDIX 1: TERMS OF USE iSHARE SCHEME

APPENDIX 2: PARTICIPATION FEES

Accession Agreement for Certified Parties

ACCESSION AGREEMENT FOR PARTICIPATION

CERTIFIED PARTIES - iSHARE SCHEME

The Scheme Owner and the [COMPANY NAME] (hereafter: ‘the Certified Party’) enter into an agreement which specifies the terms and conditions under which:

- the Certified Party shall [SPECIFY SERVICES] under the rules and specifications of the iSHARE Scheme;
- the Scheme Owner shall [supervise that the Certified Party shall act in a reliable and professional manner, in compliance with applicable law and all relevant technical specifications, to safeguard consistency across the whole iSHARE Scheme].

The Certified Party hereby declares to comply with the following rules for participation in the iSHARE Scheme:

- The Certified Party must comply to the iSHARE specific requirements and specifications as defined in the (annexes to the) Terms of Use.
- The Certified Party can apply for participation in the iSHARE Council of Participants and the Change Advisory Board as defined in the statutes of the Scheme Owner <not yet set up or established / depending on further development and role of the iSHARE Scheme>.
- The Certified Party agrees with and accepts the Terms of Use as specified in Appendix 1.
- Participation in the iSHARE Scheme is subject to a [TBD: ANNUAL/MONTHLY] participation fee. Participation fees are non-refundable fees and are stated in Appendix 2. The Scheme Owner may adjust participation fee rates once a year with effect from January the 1st with two (2) months’ prior written notice to the Certified Party.

- The Scheme Owner’s invoices are due upon receipt and must be fully paid within 30 days after the invoice date.

Duration

The Accession Agreement is entered into for an initial period of twelve (12) months. During the initial period, the Certified Party may only terminate the Accession Agreement as set forth in the Terms of Use. After the initial period, the Accession Agreement shall be tacitly extended for an indefinite period of time and may be terminated subject to the notice period as stated in the Terms of Use.

The Certified Party declares compliance to all rules set forth in this Accession Agreement, including the referenced appendices.

| | Certified Party | Scheme Owner |
|-----------|------------------------|---------------------|
| Name | | |
| Company | | |
| Place | | |
| Date | | |
| Signature | | |

APPENDIX 1: TERMS OF USE iSHARE SCHEME

APPENDIX 2: PARTICIPATION FEES

Terms of Use

TERMS OF USE

iSHARE SCHEME

ARTICLE 1. APPLICABILITY

- 1.1. These Terms of Use apply to each Party participating in the iSHARE Scheme.
- 1.2. In addition to the laws and regulations described in the Legal Framework, these Terms of Use will apply to each Party participating in the iSHARE Scheme and govern the rights and obligations of each Party as well as the relationships between the Parties.
- 1.3. In the event of a conflict between the Parties' private agreement(s) and these Terms of Use, the private agreement(s) will prevail, with the exception of the matters covered by the following Articles [mandatory articles to be determined in consultation with other working groups].

ARTICLE 2. DEFINITIONS

The terms used in these Terms of Use, both in the singular and plural, shall be understood to mean the following:

- 2.1. **Accession Agreement:** the agreement that governs the admission of Adhering Parties and Certified Parties to the iSHARE Scheme. In the event of a conflict with the Terms of Use, the provisions in the Accession Agreement will prevail.
- 2.2. **Adhering Party:** an Entitled Party, a Service Consuming Entity or a Service Provider.
- 2.3. **Annex(es):** the annex(es) that are inextricably linked with the Terms of Use. In the event of a conflict with the Terms of Use, the provisions in the Terms of Use will prevail.
- 2.4. **Authorisation Registry:** a party that holds authorisation information, information on licences and information on proxies that Service Providers can use to determine the rights of the Service Consuming Entity in relation to a specific Dataset.
- 2.5. **Certified Party:** an Authorisation Registry, an Identity Broker or an Identity Provider that has been certified by the Scheme Owner.
- 2.6. **Conditions of Exchange:** the licence conditions that are inextricably linked with the exchanged Data as established by the Entitled Party.
- 2.7. **Data or Dataset:** the data exchanged in the context of the iSHARE Scheme.
- 2.8. **Entitled Party:** a Party that grants a (sub-)license to a Service Consuming Entity in relation to a specific Dataset.
- 2.9. **Human Service Consumer:** a natural person who acts on behalf of and under the responsibility of the Service Consuming Entity.
- 2.10. **Identity Broker:** a party whose services a Service Provider can use to connect to one of more Identity Providers.
- 2.11. **Identity Provider:** a party that holds the digital identity information on a Human Service Consumer which that Human Service Consumer can use to identify himself/herself towards a Service Provider.
- 2.12. **iSHARE Scheme:** the set of specifications which govern the relationships between the Parties in the iSHARE Scheme, including, without limitation, the exchange mechanism and the actual exchange of Data.
- 2.13. **Legal Framework:** the non-exhaustive overview of relevant and applicable laws and regulations in respect of the iSHARE Scheme. The Legal Framework is described in Annex II to these Terms of Use.

- 2.14. **Scheme Owner:** the entity <not yet set up or established / depending on further development of the scheme> responsible for management and continued development of the iSHARE Scheme[, as well as for controlling and monitoring the Parties' compliance with the iSHARE Scheme].
- 2.15. **Party:** an entity that participates in the iSHARE Scheme.
- 2.16. **Service Consuming Entity:** a Party who requests the Service Provider to provide a service relating to the exchange of Data.
- 2.17. **Service Provider:** a Party who provides a service relating to the Data to be exchanged with a Service Consuming Entity.
- 2.18. **Terms of Use:** this document, including the Annexes.
- 2.19. **Website:** ishare-project.org

ARTICLE 3. RIGHTS AND OBLIGATIONS OF ADHERING PARTIES

- 3.1. The Adhering Party who is sending the Data is responsible for linking the Conditions of Exchange to the Data to be exchanged. Each Dataset can be provided with an attribute. This is a code to which the Conditions of Exchange of the Adhering Party who is exchanging the Data are linked. It is up to the Adhering Parties who are exchanging the Data to agree on any commercial arrangements with regard to that exchange.
- 3.2. The Service Provider is responsible for determining the assurance level of identification of the Human Service Consumer within the iSHARE Scheme.
- 3.3. The rights of the Service Consuming Entity related to the exchange of a specific Dataset is determined by the Conditions of Exchange. The various licence conditions are linked to the Dataset by means of a data exchange code. The data exchange codes and their meaning are described in Annex I to these Terms of Use. If a Dataset does not contain a data exchange code, the default Conditions of Exchange as indicated in Annex I apply. The Service Provider and the Service Consuming Entity agree to comply with the Conditions of Exchange.
- 3.4. Service Consuming Entities will supervise and are responsible for their Human Service Consumers. Service Consuming Entities will not permit any practice that could lead to improper handling by their Human Service Consumers, including, without limitation, the unauthorised use of authentication tokens linked to individuals and/ or the organisation, or the use of authentication tokens for any purpose other than the purpose for which they were issued. Service Consuming Entities will make their Human Service Consumers aware of these Terms of Use.
- 3.5. An Adhering Party is responsible for the security and monitoring of the network connections and systems that it uses in the context of the iSHARE Scheme. An Adhering Party will take appropriate technical and organisational measures in order to safeguard the security as described in Annex III.
- 3.6. In case an Adhering Party notices or suspects irregularities in the Data it receives, that Party shall immediately notify the Service Consuming Entity(ies) and/or the Service Provider concerned. Where applicable, the Service Provider shall immediately notify the Entitled Party.
- 3.7. The Scheme Owner grants the Adhering Party a limited, non-exclusive and non-transferable license to use - during the term of the Accession Agreement - the trademarks and trade names "iSHARE" and "iSHARE Adhering Party" and any other trademarks or trade names related to the iSHARE Scheme, as determined by Scheme Owner from time to time hereafter. The trademarks and trade names may only be used in connection with iSHARE Scheme related activities. In the event the Scheme Owner decides to modify or discontinue the use of one or more of the trademarks and trade names or to use one or more additional or substitute trademarks or trade names, the Adhering Party agrees to immediately and fully comply with the instructions of the Scheme Owner in that respect.

ARTICLE 4. RIGHTS AND OBLIGATIONS OF CERTIFIED PARTIES

4.1. The Certified Party is responsible for the security and monitoring of the network connections and systems that it uses in the context of the iSHARE Scheme. All Certified Parties will take appropriate technical and organisational measures in order to safeguard the security, including those measures and use of standards that are specified in the iSHARE Scheme <include reference to document still to be drafted by another working group>.

4.2. In addition to its own statutory obligations, the Certified Party shall notify the Scheme Owner of a (potential) network failure or (suspicion of) a security breach within [XX] hours of becoming aware of said failure and/or breach and shall promptly take adequate remedial measures. The Certified Party shall warrant that the information it provides is complete and accurate.

4.3. The duty to notify as referred to in the previous paragraph includes in any event details regarding:

- the (suspected) cause of the network failure and/or security breach;
- the (currently known and/or anticipated) consequences thereof;
- the (proposed) solution;
- the contact details in connection with follow-up action;
- what measures have already been implemented.

4.4. The Scheme Owner grants the Certified Party a limited, non-exclusive and non-transferable license to use - during the term of the Accession Agreement - the trademarks and trade names “iSHARE” and “iSHARE Certified Party” and any other trademarks or trade names related to the iSHARE Scheme, as determined by the Scheme Owner from time to time hereafter. The trademarks and trade names may only be used in connection with iSHARE Scheme related activities. In the event the Scheme Owner decides to modify or discontinue the use of one or more of the trademarks and trade names or to use one or more additional or substitute trademarks or trade names, the Adhering Party agrees to immediately and fully comply with the instructions of the Scheme Owner in that respect.

ARTICLE 5. RIGHTS AND OBLIGATIONS OF THE SCHEME OWNER

5.1. The Scheme Owner is not allowed to access exchanged Data. [The Scheme Owner will facilitate the iSHARE Scheme and will only have an administrative role with regard to these Terms of Use and other legal documents associated with the iSHARE Scheme.] <depending on the role of the Scheme Owner>

5.2. The Scheme Owner will maintain and publish a publicly accessible registry of Parties and their respective roles within the iSHARE Scheme.

5.3. The Scheme Owner is entitled to suspend a Party, or terminate its participation and registration in the iSHARE Scheme, if that Party breaches these Terms of Use and/or applicable laws and regulations in respect of the iSHARE Scheme.

5.4. The Scheme Owner determines which Parties can be admitted to the iSHARE Scheme and on what conditions. The standards and (technical) specifications under which Certified Parties will be accredited are specified in Annex III to these Terms of Use. <input from the functional working group is required here>

5.5. [The Scheme Owner will endeavour to make a decision within four (4) weeks regarding the possible admission of the Certified Party.]<depending on the admission procedure which will be described in Annex IV>

5.6. The Certified Party shall conduct an annual audit through an independent certified auditor to verify compliance with the conditions, standards and (technical) specifications under which the Certified Party is accredited. In addition to the annual audit, the Scheme Owner in its sole discretion, may determine that more frequent audits are required when there are specific grounds for suspecting a possible breach of these conditions,

standards or (technical) specifications. Unless otherwise agreed with the Scheme Owner, the Certified Party will conclude each audit within a period of thirty (30) days. The findings resulting from any audit will be evaluated in mutual consultation by the Scheme Owner and the Certified Party. The costs of all audits will be borne by the Certified Party.

ARTICLE 6. CONFIDENTIALITY AND PRIVACY

6.1. The Party to whom information (including the Data) is provided shall only use that information for the purpose for which it has been provided. Neither Party shall provide the information to any third party other than those to whom he may provide information within the framework of the iSHARE Scheme, or as otherwise agreed between the Parties, unless it is obliged to do so in pursuance of a statutory duty or required by court order. Furthermore, the Parties shall accept the duty to observe strict secrecy when the information is marked as confidential or when the receiving Party knows or should reasonably suspect that the information was intended to be confidential.

6.2. The Parties shall protect the information against unauthorised access using a level of protection that is reasonable given the nature of the information.

6.3. The Parties only process personal data if and to the extent necessary for the performance of its rights and obligations within the framework of the iSHARE Scheme. The processing of personal data shall be in accordance with applicable privacy and data protection law.

ARTICLE 7. LIABILITY

7.1. The liability of the Parties shall be in accordance with and determined by the general rules of Dutch law.

7.2. To the extent permitted by law, the Scheme Owner expressly disclaims any and all liability for damages of any kind incurred by any Party. However, the Scheme Owner's liability is not limited regarding damages that are the result of deliberate recklessness or wilful misconduct by the Scheme Owner and/or its management.

ARTICLE 8. SETTLEMENT OF DISPUTES

8.1. In the event of disputes between the Parties arising from and/or in connection with the performance of operations within the framework of the iSHARE Scheme, including disputes regarding compensation for damages, the Parties should first endeavour to resolve the disputes by mutual agreement.

8.2. If the dispute cannot be resolved through constructive dialogue between the Parties, the Parties may submit the dispute for resolution to the Complaints and Disputes Committee <rules not available / discuss iSHARE role or role of external dispute resolution body>. Furthermore, the Parties may always submit disputes to the competent civil courts or any other dispute resolution body.

ARTICLE 9. AMENDING THE TERMS OF USE

9.1. The Scheme Owner is entitled to amend or supplement these Terms of Use and its Annexes in accordance with the rules and procedures as described in Annex V.

9.2. Amendments will apply subject to a term of 30 days following publication of the amendment on the Website or after announcement by electronic communication. Minor changes can be implemented at any time.

9.3. Notwithstanding article 10, if an Adhering Party does not accept an amendment to the Terms of Use, that Party's participation in the iSHARE Scheme can be terminated on the date on which the amended Terms of Use take effect.

ARTICLE 10. DURATION

10.1. These Terms of Use shall remain in force as long as a Party remains registered with the Scheme Owner or for the duration described in the Conditions of Exchange, whichever is longer.

10.2. A Party can cancel his registration by terminating the Accession Agreement. Termination is subject to a one month's notice period for Adhering Parties, and a six months' notice period for Certified Parties. Promptly after giving notice of termination of the Accession Agreement, a Certified Party shall communicate the termination of its participation to all Parties affected.

ARTICLE 11. FINAL PROVISIONS

11.1. These Terms of Use are governed by Dutch law and the Parties agree to submit to the courts of [TBD].

11.2. The Parties are not authorised to transfer their rights and obligations under the iSHARE Scheme to any third party, except with written permission from the Scheme Owner.

11.3. The Parties have a continuous obligation to keep their registration with the iSHARE Scheme up-to-date and to notify the Scheme Owner of any material changes in the corporate structure and/or ownership of its business.

11.4. If any provision of these Terms of Use (or part of any provision) is found by any court or other authority of competent jurisdiction to be invalid, illegal or unenforceable, that provision or part-provision shall, to the extent required, be deemed not to form part of these Terms of Use, and the validity and enforceability of the other provisions of these Terms of Use shall not be affected. In such an event, the Scheme Owner shall include a suitable replacement provision.

ANNEXES

Annex I: Conditions of Exchange

Annex II: [Legal framework](#)

Annex III: Standards and (technical) specifications of the iSHARE Scheme

Annex IV: Admission procedure

Annex V: Change procedure

Legal Framework

As stipulated in the [Accession Agreement](#) and the [Terms of Use](#), all iSHARE participants are expected to comply with all laws and regulations that apply to their business. The following rules and regulations are of particular relevance when exchanging data within the iSHARE scheme:

1. Competition law
2. Privacy and data protection law
3. eIDAS Regulation
4. Dutch Civil Code

1.1 Competition law

1.1.1 Agreements

First and foremost, it should be noted that it is not the intention of the iSHARE scheme to limit competition in any shape or form. In all cases, an important principle of the iSHARE scheme is to create a level playing field and foster efficiency gains. Nonetheless, it is important to carefully draft the agreements (i.e. the Accession Agreement and Terms of Use) and always assess whether they could restrict competition, and whether a restriction could be justified by - for example - efficiencies. Admittedly, it is mainly up to the participants sharing data to comply with competition law, but the iSHARE scheme itself is not designed in a way to directly or indirectly have an adverse effect on competition.

Depending on whether an agreement or other behaviour has an effect in the entire EU or not, EU competition law or national competition law (and enforcement) applies. Competition law prohibits agreements that restrict competition, unless there is a justification for them.

There are different types of agreements with different rules. The rules for agreements between companies at the same level of the production chain are generally stricter than those for companies at different levels of the production chain. The iSHARE scheme facilitates both horizontal and vertical exchanges of information.

What is problematic under competition law, is the exchange of information that is sensitive to competition, such as price lists, data on turnover, etc. Restrictive effects may, for instance, be found in cases where exchanges of information enables companies to be better aware of each other's market strategies. Agreements that have as their purpose or effect the restriction of competition (such as price fixing, market sharing) are very likely to be prohibited.

On the other hand, a justification may be found for exchanging information. Exchanging information can lead to efficiency gains. For the determination of efficiency gains, there are three further conditions to be taken into account:

1. the efficiency must at least be partially passed on to the consumers which are affected by the restriction (e.g. quicker delivery of products or reduction of search costs);
2. the agreement must not restrict competition more than is necessary for the attainment of the efficiency gains (proportionality requirement);
3. the restriction of competition must not result in the total elimination of competition.

As a result, competition law leaves room for such agreements. The iSHARE scheme could lead to efficiencies (e.g. in terms of costs or by removing barriers).

1.1.2 Dominant position

Competition law also deals with the abuse of a dominant position. Companies can also have a dominant position collectively. Whether there is a dominant position, is assessed on the basis of market shares, amongst other factors. When there is a (collective) dominant position, it is important to assess whether, for example, parties not participating in iSHARE are excluded from the market via abuse of dominance. A dominant position is not in itself anti-competitive. Only when that position is exploited to eliminate competition, it is considered an abuse. Examples of practices that can (but do not necessarily have to) lead to abuse of dominance are exclusive dealing agreements, a refusal to supply, and certain pricing practices.

Currently, the iSHARE scheme is intended to be an open framework, accessible for just any party – admitted to the iSHARE scheme or not - seeking to use its functionalities. For the parties wishing to participate in iSHARE, the requirements that will be formulated in order to become a participant must also not restrict competition to the extent that they, for instance, could be perceived as a refusal to supply. In other words, the requirements must not be so exacting that they exclude specific parties, thereby enabling the participants to corner the market. The result of the foregoing may be that the economic competitiveness will be jeopardized.

1.2 Privacy and data protection law

On the 25th of May 2018, our Dutch privacy law (*‘Wet bescherming persoonsgegevens’*) is set to be overhauled by a European privacy regulation, the ‘General Data Protection Regulation’ (GDPR). This regulation will ensure that the same privacy rules apply throughout the entire EU and will entail substantial changes for businesses and industry.

Two of those changes are the requirements of ‘privacy by design’ and ‘privacy by default’. Broadly speaking, this means that privacy must be taken into account throughout the entire process in which products and services are developed. This can be achieved by using techniques such as pseudonymisation and by processing as few personal data as possible, i.e. by processing only the necessary personal data. This requirement of necessity also applies to the accessibility of data (i.e. who has access to which data) and the period for which data are retained. The default settings of a product or service must also be as privacy-friendly as possible. Products and services will therefore have to be developed and designed in such a way as to ensure that they are ‘privacy proof’.

Personal data must be protected adequately, via technical and organizational measures. For example: passwords, encryption, secure (SSL/TLS) network connections and pseudonymisation of data. Technical norms such as the ISO 27001 are not mandatory, but in practice they are the best way to make sure a service provider uses adequate protection. Service providers who are able to provide a statement from an independent auditor offer even more security. The most well-known statements are the SAS70, ISAE 3402 and the SSAE No. 16.

Although the majority of data shared via the iSHARE scheme may not be personal data, there could be personal data involved. For example, data relating to employees or clients of participating parties. If personal data is shared via the iSHARE scheme, the participating parties will need to have a legal basis to do so. A legal basis can be, for example, consent of the data subjects, or an agreement to which the data subject is a party.

When data is exchanged between two data controllers, both need a legal basis for this. A data sharing agreement then also needs to be concluded. When a data processor processes personal data on behalf of the controller, they are obliged to enter into a data processing agreement. The GDPR explains what such an agreement should contain. The iSHARE scheme should put the participating parties in control of the types and amount of data they like to share and in this respect should also easily facilitate the conclusion of data processing or data sharing agreements.

In certain cases, the GDPR requires that the privacy effects of a project are assessed in advance (a Privacy Impact Assessment). This is the case when the processing of personal data constitutes a high risk for the data subjects. For

certain companies, for example, companies which monitor individuals or systematically process sensitive data, it will become mandatory to have a Privacy Officer.

1.3 eIDAS Regulation

The eIDAS Regulation – formally the Regulation on electronic identification and trust services for electronic transactions in the internal market – was adopted on 23 July 2014. It aims to provide a predictable regulatory environment to enable secure and seamless electronic interactions between businesses, citizens and public authorities throughout the entire EU. It ensures that people and businesses can use their own eIDs to access public services in other EU countries and enhances cross-border interoperability of electronic trust services.

The first section of the eIDAS Regulation relates to the government-recognized eIDs and establishes a legal framework that will allow all EU countries to recognize each other's eIDs. The second section of eIDAS deals with the various electronic signatures (i.e. simple, advanced and qualified). It clarifies existing rules, but also introduces a new legal framework for electronic signatures, seals and timestamps. The new legal framework is not mandatory but introduces certain requirements that can be followed in order to grant greater legal certainty and to improve the reliability of these services.

Furthermore, the eIDAS Regulation draws a distinction between the parties providing the electronic signatures: qualified and non-qualified trust service providers. The eIDAS Regulation sets forth certain requirements that the qualified trust service providers must adhere to. For instance, the qualified trust service providers need to inform the supervisory body of any change in the provision of its services and must maintain sufficient financial resources or obtain appropriate liability insurance. Furthermore, each EU country is required to 'establish, maintain and publish trusted lists, including information related to the qualified trust service providers for which it is responsible, together with information related to the qualified trust services provided by them'.

For the purpose of the iSHARE scheme, it needs to be considered which eIDs are to be used (either offering 'low', 'substantial' or 'high' assurance levels) and which trust service providers are to be engaged (qualified or non-qualified) and the roles these trust service providers have within the iSHARE scheme. The selection of eIDs and trust service providers are also relevant for the international orientation of the iSHARE scheme and to foster the cross-border interoperability of electronic trust services.

1.4 Dutch Civil Code

In setting up the iSHARE scheme, the relevant provisions of the Dutch Civil Code should also be taken into account. This primarily relates to the Accession Agreement and the Terms of Use, which need to be drafted in accordance with Dutch contract law. With the expansion of the iSHARE scheme, other national laws may become relevant as well. Any specific (national and international) rules for the transport and logistics sector, such as rules for agreements on the carriage of goods, fall outside the scope of this legal framework. These types of sector specific rules are not relevant for operating and using the iSHARE scheme, although participants may need to adhere to them when contracting services through the scheme.

Operational

This section will cover the operational topics of the iSHARE scheme.

As described in the [introduction](#), the Operational working group did not start during phase 2. Therefore, this section is only a framework including descriptions of what needs to be agreed on - to be filled in by the Operational working group soon. The framework set up consists of the following chapters:

- [Service Level Agreements](#)
- [Audits](#)
- [Incident Management](#)
- [Change Management](#)
- [Governing Body](#)

Governing body

The iSHARE scheme is an initiative with a long-term ambition to improve the circumstances for data exchange in 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: the [Scheme Owner](#). Which form this Scheme Owner 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.

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 service provisions 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 working group.

Incident Management

The Incident Management process is a process to settle different types of incidents within the iSHARE scheme - 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 scheme. Incidents could include:

- Disruptions: events that lead to (parts of) the iSHARE service(s) being partially or entirely unavailable;
- Information security incidents
- Fraud or the presumption of fraud by, for example, an employee or a hacker.

The Operational Working group will in due time define a policy for incident management. The working group shall identify possible incidents and set-up procedures to handle these incidents.

Change Management

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

- Scheme documentation;
- Scheme implementations.

It will be detailed by the Operational working group.

Service Level Agreements

This section describes the service level agreements that apply to participants of the iSHARE scheme. A **service level agreement** (SLA) is a contract that defines the level of service expected between parties in the iSHARE scheme.

SLAs are output-based in that their purpose is specifically to define what the party requesting a service will receive.

Service level agreements include the following topics:

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

Up-time

Up-time is a measure of the time a machine has been working and available. Uptime is the opposite of downtime.

The times which are issued by participants and the [Scheme Owner](#) guarantee the availability the iSHARE scheme.

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 the time it takes to process a request and return a response.

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

The norm for processing of messages for participants could be:

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.

Monitoring

The Operational Working group will establish to what extent monitoring of activity within the iSHARE scheme is necessary, useful and/or possible.

Logging

The Operational working group will establish to what extent logging will be required by which parties and which specific reasons.

Archiving

The Operational working group will establish to what extent archiving will be necessary (in which situations, by which parties, for what reasons).

Reporting

The Operational working group shall establish to what extent Reporting will be required (which parties, what information, to what end).

Glossary & Legal Notices

This section includes the iSHARE glossary and legal notices. The section is presented as follows:

- [Glossary](#)
- [Legal notices](#)

Glossary

[ABAC](#)

[Access](#)

[Accountability](#)

[Authentication](#)

[Authenticity](#)

[Authorisation](#)

[Authorisation Registry](#)

[Availability](#)

[Broadcast](#)

[Broker](#)

[Certificate Authority](#)

[CIA Triad](#)

[Co-creation](#)

[Confidentiality](#)

[Credentials](#)

[Service Consumer](#)

[Service Provider](#)

[Data Owner](#)

[Data classification](#)

[Data retention](#)

[Delegation](#)

[EAN](#)

[Encryption](#)

[EORI](#)

[Exchange \(of Data\)](#)

[Granularity \(of authorisations\)](#)

[Hashing](#)

Identification
Identity Broker
Identity Provider
Integrity
IPsec
Levels of Assurance
Multicast
Non-repudiation
PDP
PEP
Public Key Infrastructure (PKI)
PKI root
RBAC
Responsibility vs Accountability
Scheme
Service provision
Signing
Single Sign On (SSO)
Session
SSL/TLS
Token
Trust framework
Use case
Validation

ABAC

ABAC (Attribute-Based Access Control) or is assigning authorisations based on attributes (contextual pieces of information that are relevant to an access decision, such as device type, RBAC role, time, location, or CRUD level). The attributes can be associated with all entities that are involved with certain actions, such as the subject, the object, the action itself and the context (e.g. time, location). The attributes are compared with policies to decide which actions are allowed in which context.

Accountability

Accountability can be described as being liable or answerable for the completion of a certain task. A person who is accountable oversees and manages the stakeholder(s) who are responsible for performing the work effort. In order to be effective, accountability SHOULD be with a sole person or role.

Access

A way of getting near, at, or to something or someone. In the context of information technology access mostly refers to activities related to information systems and to activities (creating, reading, updating, deleting) to digital data.

Authentication

The process of determining or validating whether someone or something is, in fact, who or what it is declared to be. There are several means of authenticating the identity of an entity, which can be used alone or in combination:

- Something the entity knows – examples include a password, PIN, passphrase, or answer to a secret question.
- Something the entity possesses – examples include electronic keycard, smartcard, token, and smartphone.
- Something the entity is (biometrics) – examples include recognition by fingerprint, retina, iris, and face.
- Something the entity does (behavioral dynamics) – examples include recognition by voice pattern, swipe characteristics, handwriting characteristics, and typing rhythm.
- Something about the context of the entity – examples include IP address, device type, geolocation, and time of day.

Authenticity

Authenticity in the context of information security refers to the truthfulness of information and if this has been sent or created by an authentic sender. Authenticity can be achieved 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 key is issued by a [Certificate Authority](#).

Authorisation

Authorisation is the process of giving someone or something permission to do or have, for example getting access to services, data or other functionalities. Authorisation is enabled by authentication. Policies and attributes determine what types of activities are permitted by the entity.

The owner of the environment (Service Provider) can decide to perform the [authorisation management and validation process](#) internally or to rely on an [Authorisation Registry](#) for that. The Service Provider decides which authorisation attributes have to be presented and which policies to adhere to by the entity before getting access to the service.

Availability

Availability in the context of information security refers to the ability of authorised parties to access their resources whenever they need to. It can be achieved by a number of controls, e.g. backup procedures, failover mechanisms, and disaster recovery procedures.

Broadcast

An act of casting or scattering in all directions, e.g. a message or a radio signal.

Broker

Person or entity that performs actions, arrangements or negotiations between parties, to provide for interoperability and to avoid $n(n-1)$ connections between parties.

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 assertions made with the private key that corresponds to the certified public key.

A **Registration Authority** verifies the identity of entities requesting digital certificates to be issued by the CA and validates the correctness of the registration.

A **Validation Authority** verifies the validity of digital certificates on behalf of the CA.

CIA Triad

Model with the three key principles [confidentiality](#), [integrity](#) and [availability](#), that is designed to guide policies for information security.

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 but the legitimate sender and recipient. Confidentiality can be achieved by the use of cryptography, as well as access control.

Credentials

Attestation or evidence of identity, authority, status, authorisations, rights, or entitlement. Can be in digital form (e.g. username combined with a password) or in written form (e.g. a name combined with a signature).

Data Owner

The Data Owner is the (legal) person who is accountable for the confidentiality, integrity, availability and accurate reporting of data.

The Data Owner can be the Service Provider. In this case, he is not only accountable for the availability of service, but also responsible. Read more on the relation between responsibility and accountability [here](#).

Data Classification

The classification of data in categories is an important pre-requisite for proper authorisation. Data can be classified in categories defining their type, location, sensitivity and protection level. Authorisation depends on the access rights of the (Human) Service Consumer that are checked as part of the service requesting process. Clustering the data in categories does not only simplify the authorisation process, it also provides a clear overview to the Service Provider over their data and lowers the risk of exchanging sensitive data with unauthorised (Human) Service Consumers. A risk analysis is part of the data classification process.

Data retention

Refers to the storage and archiving of data (records) for compliance, historical or business reasons.

Delegation

The act of empowering to act for another or to represent other(s). A delegated party acts on behalf of an Entitled Party and is either allowed to assign authorisations or to delegate yet another party, depending on the relevant policy.

EAN

(European Article Number; also called International Article Number) Used worldwide for marking products that are sold at retail point of sale.

Encryption

Encryption is the process of converting data from plaintext to ciphertext. Plaintext (also called cleartext) represents data in its original (readable) format, whereas ciphertext (also called cryptogram) represents data in encrypted (unreadable) format.

Decryption is the process of converting data from ciphertext to plaintext.

The algorithm represents the mathematical or non-mathematical function used in the encryption and decryption process.

A cryptographic key represents the input that controls the operation of the cryptographic algorithm. With symmetric encryption the same key is used for encryption and decryption, whereas with asymmetric encryption two different, but mathematically related keys are used for either encryption or decryption, a so-called public key and a private key.

A crypto system represents the entire cryptographic environment, including hardware, software, keys, algorithms and procedures.

EORI

(Economic Operator Registration and Identification) Unique identification number that companies are required to use when exchanging data with customs in all EU member states.

Exchange (of Data)

An act of giving one thing and receiving another in return. A transaction is a type of exchange.

Granularity (of authorisations)

One of the iSHARE key features is [flexibility in authorisation](#) with regards to authorisation scope, granularity and source. In this section we will expand on the granularity for authorisations.

By granular authorisation we mean the level of detail that an authorisation process requires to limit and separate privileges (e.g. the right to access a resource).

A single authorisation may enable a number of privileges the same way as a privilege may require multiple authorisations. An authorising authority should be capable of handling both scenarios.

Granularity is not based on either authorisation requests or privileges, but on functions. Those functions are processed in computer algorithms that express the rules defined in authorisation 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-grained authorisation

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

Role-based access control is an example for "fine-grained": 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-grained" authorisation: access to a resource depends on attributes that the user has to bring along to prove that they meet the authorisation requirements (the policies).

Coarse-grained authorisation

Course-grained authorisation is simpler and different from fine-grained authorisation as there are no lower detail tasks within the functions.

Access control lists (ACL's) are an example for "coarse-grained" authorisation: 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).

Examples of coarse-, fine-, finer-grained authorisation

- 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 Service Provider in week X, can access container A.

Hashing

Hashing is a one-way mathematical function used to verify the integrity of data. Putting it differently, to ensure that data (message, file or software) has not been modified.

A thorough hash function has the following characteristics:

- The hash value (output) should not be predictable
- The hash value should be collision resistant. It should not be computationally feasible to find another input value that generates the same hash value
- The hash value should be impossible to invert. It should not be possible to derive the input value from the hash value, and
- The hash value should be deterministic. A given input should always generate the same hash value.

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 service request of the iSHARE exchange which is [authentication](#).

Integrity

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

IPsec

Protocol suite that provides for both encryption and authentication of IP packets in network communication. Since IPsec works at the internet layer of the TCP/IP model (network layer in the OSI model), applications do not need to be aware of it. Hence, IPsec is able to protect all traffic in an IP network, regardless of the application(s) used.

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.

Handreiking over betrouwbaarheidsniveaus

The Dutch government published the following '[handreiking](#)' about Levels of Assurance for authentication.

Multicast

An act of casting or scattering to a defined group of receivers, e.g. an electronic message.

Non-repudiation

Non-repudiation (Dutch 'onweerlegbaarheid') refers in the context of information security to the fact that the sending (or broadcast) and receipt of the message cannot be denied by neither of the involved parties (sender and recipient).

Non-repudiation is closely related to [authenticity](#) and can be achieved by digital signatures in combination with message tracking.

PDP

(Policy decision point) Entity that evaluates access requests that are received from the policy enforcement point (PEP). Subsequently an answer is sent back to the PEP.

PEP

(Policy enforcement point) Entity that determines whether an action is permitted or not. It takes any access requests and forwards these to the policy decision point (PDP).

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.

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.

RBAC

(Role-Based Access Control) Assigning authorisations through business roles. An RBAC role represents a set of tasks or activities translated into authorisations, reflecting one or more of the following:

- Organisational structure
- Business processes
- Policies (rules)

RBAC authorisations can either give access to the front door of the information system or can be translated to access rights within the information system (often through application roles or groups).

Responsibility vs Accountability

There is a clear distinction between responsibility and accountability.

Responsibility can be described as tasked with getting the job done. A person who is responsible performs the actual work effort to meet a stated objective.

Accountability can be described as being liable or answerable for the completion of a certain task. A person who is accountable oversees and manages the stakeholder(s) who are responsible for performing the work effort. In order to be effective, accountability SHOULD be with a sole person or role.

Responsibility may be delegated, but accountability cannot.

Scheme

In the context of iSHARE a scheme can be defined as a collaborative effort of organisations to achieve a common goal. In [Goals and scope of the iSHARE scheme](#) the purpose of the iSHARE scheme is described.

An analogy is the card scheme, such as Visa, MasterCard, American Express etc.

Service provision

An act of providing or supplying something for consumption or use. One of the most common forms of service provision is the [exchange of data](#).

Session

Interactive information exchange between two or more computers (or other communicating devices), or between a human and a computer (or another communicating device).

Signing

Signing is the process of encrypting data (message, document, transaction) with the private key of the sender. It enables a receiver to confirm the [authenticity](#) of the data. Signing also provides for [non-repudiation](#), so that it is ensured that a sender cannot deny having sent a message.

In most cases, a hash of the data is encrypted. Thus, both the [integrity](#) and the [authenticity](#) of the data can be verified. Confirmation takes place by the receiver using the public key of the sender. The public key is contained in the digital certificate that is sent by the sender along with the signed data. The association of the key pair with the sender MUST be assured by a [Certificate Authority](#).

Single Sign On (SSO)

Single Sign On (SSO) is often implemented as cross-domain SSO, which is a federated identity solution.

It is 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 systems 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 one or all previously signed-in environments.

SSL/TLS

SSL/TLS (Secure Sockets Layer/Transport Layer Security) are a set of protocols that provide for secure communication in computer networks. SSL/TLS make use of cryptography and are widely used by a variety of applications such as web browsing, email and voice-over-IP.

Token

Something that serves as a verifiable representation of some fact, e.g. an identity or entitlement.

Trust framework

Structure that aims to provide confidence in public internet environments. A trust framework is specified according to rules drawn up by a party or community that is inherently trusted, such as a government or a combination of profit and nonprofit parties. Service providers who wish to participate in the trust framework must comply with those rules to achieve a certain level of trust or level of assurance.

The rules include functional, technical, operational and legal rules.

Validation

Action of proving the validity or accuracy of something; declaring that something is legally or officially acceptable.

Legal notices

No part of these specifications may be reproduced in any form by print, photo print, microfilm or any other means or stored in an electronic retrieval system, without the prior written consent of the iSHARE project organisation, which must never be presumed.

Note: the Operational working group will, in coordination with other working groups and the iSHARE project organisation, decide under what terms these pages will be governed and a final position on intellectual property rights will be established. New legal notices might be added to this page in due time.