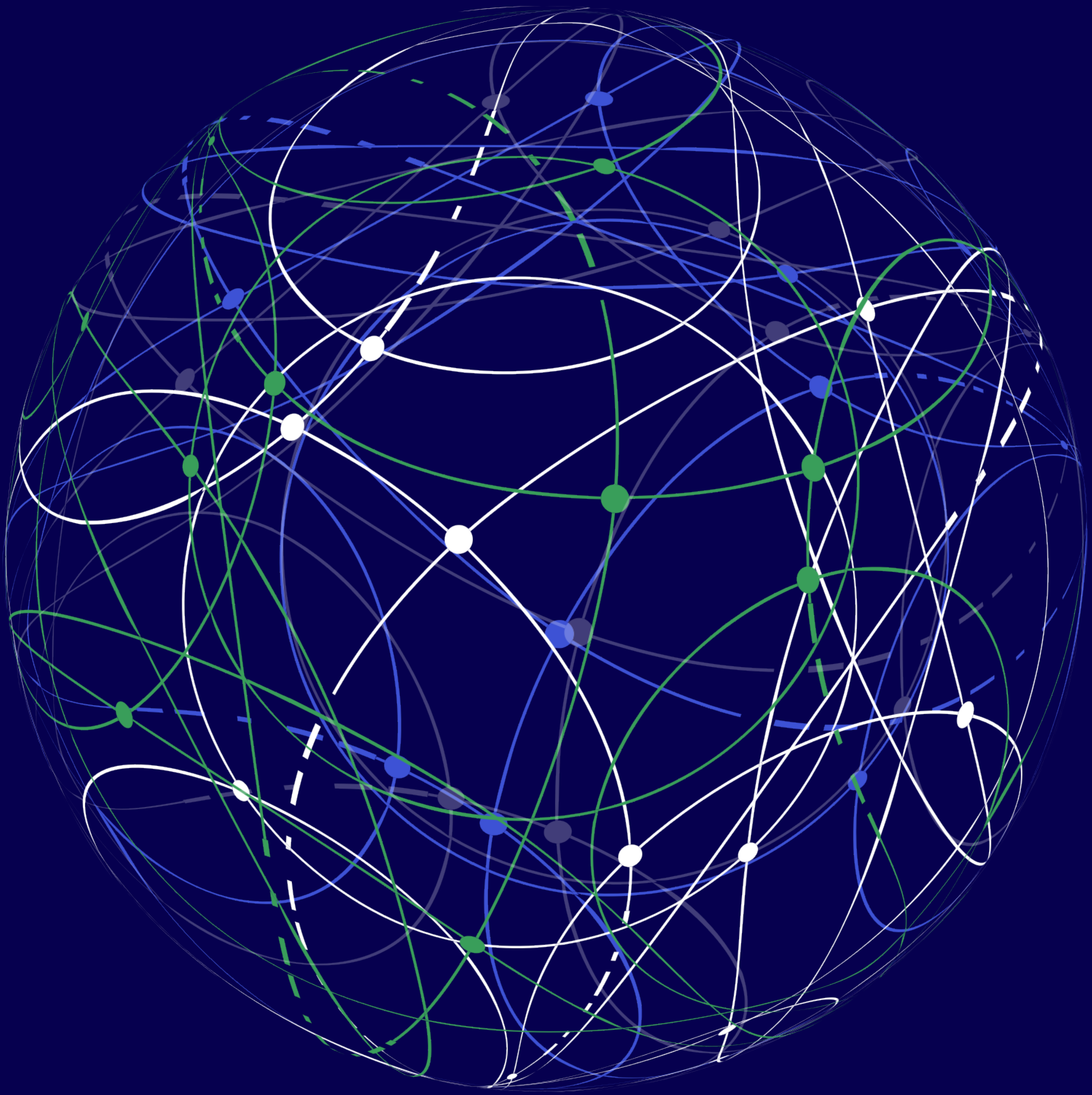
DATA SHARING COALITION

Data Sharing Canvas

A stepping stone towards cross-domain data sharing at scale



Version 1.0, April 2021

Authors and Contributors

This document is authored by the Data Sharing Coalition Expert Group and Data Sharing Coalition project team:

Victor den Bak	Visma Connect
Arjen de Bake	HDN
Rick van Beek	EDSN
Willem ter Berg	Dexes
Elsbeth Bodde	Visma Connect
Tjerk van Dalen	Data Sharing Coalition
Freek Dijkstra	SURF
Edwin Edelenbos	Netbeheer Nederland
Tom van Engers	University of Amsterdam
Leon Gommans	University of Amsterdam
Casper van der Harst	MedMij
Eefje van der Harst	Data Sharing Coalition
Johan Hobelman	MedMij
Gerard van der Hoeven	iSHARE
Gerard Huis in 't Veld	SBR Nexus
Vincent Jansen	INNOPAY
Erik Kentie	SURF
Ruud Koreman	EDSN
Wouter Los	University of Amsterdam
Bert Meerman	GO FAIR
Joep Meindertsma	Dexes
Robin Oostrum	SIVI
Rajiv Rajani	iSHARE
Bauke Rietveld	Data Sharing Coalition
Michiel Schok	SURF
Jan Schrama	HDN
Hayo Schreijer	Dexes
Pieter Schuurmans	Data Sharing Coalition
Lisa Spellman	SAE ITC
Sebastian Steinbuss	IDSA
Marnix Vermaas	Visma Connect / iSHARE
Jurriaan Wesselink	Data Sharing Coalition
Jolien van Zetten	NEN

Table of Contents

Introduction

This section provides context on the purpose of the *Data Sharing Coalition* and this document, as well as information on how to interpret this document. Furthermore, the guiding principles established in the *Data Sharing Coalition* which steer, and influence content and discussions are presented.

01.
Context

1.1. The Data Sharing Coalition

The *Data Sharing Coalition* is an open and growing, international initiative in which a large variety of organisations collaborate to unlock the value of cross-*Domain Data Sharing*. The *Data Sharing Coalition* aims to drive cross-*Domain Data Sharing* under control of the entitled party, by enabling *Interoperability* between *Domains*.

Several *Data Sharing Initiatives* exist (as of 2020), and these are often focused on a specific sector or *Domain*. Examples include *Initiatives* such as, HDN for the mortgage *Domain*, MedMij for the healthcare *Domain*, or SURF for the higher education and research *Domain*. These facilitate *Data Sharing* for their *Participants*. Additionally, generic Initiatives such as GO FAIR, AMdEX, iSHARE, NEN, and the International Data Spaces Association provide overarching principles, standards or functionalities which can be used in new and existing *Data Sharing Initiatives*. The *Data Sharing Coalition* aims to build on these existing *Data Sharing Initiatives* to strengthen them in unlocking the value of *Data Sharing* in and across their domain.

The coalition started in January 2020 with support of the Dutch Ministry of Economic Affairs and Climate Policy. The expected lifespan of the project phase of the coalition is until 2025. By 2025, the *Data Sharing Coalition* is expected to have transferred its results and activities to an entity that operates and governs a *Trust Framework* which facilitates cross-*Domain Data Sharing*. The first phase of the *Data Sharing Coalition* is a study into the *Harmonisation* potential to enable cross-*Domain Data Sharing*. For more information on the *Data Sharing Coalition*, visit: **www.datasharingcoalition.eu**

1.2. The Data Sharing Canvas

The *Data Sharing Canvas*, this document, provides input for the future *Trust Framework* for cross-*Domain Data Sharing* and is the main deliverable of the first phase of the *Data Sharing Coalition*.

The goal of the *Data Sharing Canvas* is to serve as a first steppingstone for the further research into and development of *Data Sharing* agreements between *Domains*. Due to the document's goal, the *Data Sharing Canvas* aims to give an indication of topics and their implication but does not aim to be exhaustive or to complete the detailing of these topics. As a result, the statements and findings presented in this document will provide guidance for future work of the *Data Sharing Coalition*, but do not yet represent any binding agreements or requirements.

The *Data Sharing Canvas* captures the results of a collaborative exploration of what type of agreements are required to achieve *Interoperability* across *Domains*. This includes determining the topics that require agreements to achieve interoperability, the extent to which agreements are necessary for these topics and the gathering of best practices regarding these future agreements.

The *Data Sharing Canvas* is a product of the *Data Sharing Coalition* Expert Group. Together, through extensive discussions, collaborative research, and knowledge sharing, facilitated by the *Data Sharing Coalition* project team, they have produced the *Data Sharing Canvas*. The Expert Group identified and discussed topics relevant for cross-*Domain Data Sharing* and combined this with insights from the *Data Sharing Coalition* use cases and an analysis of existing *Data Sharing Initiatives*, (see **Figure 1**). An overview of the sources of input which have been processed in the *Data Sharing Canvas* is provided below:

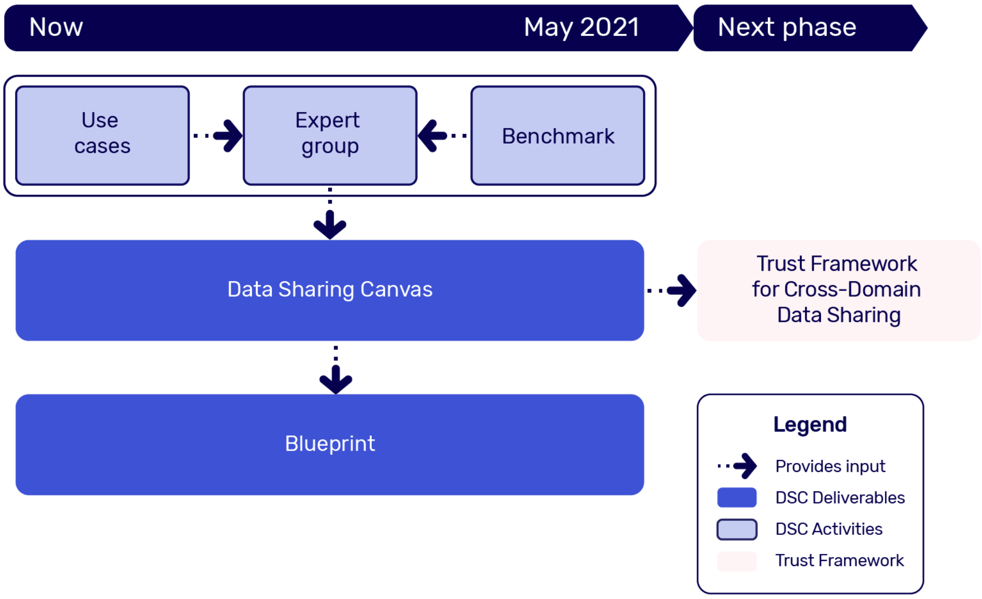
- **Expert input:** Experts delegated by *Data Sharing Coalition Participants* provide input on a wide range of identified topics which are relevant for *Data Sharing*. On all the topics discussed, they provide insights based on their specific experience and expertise. See document **Authors and Contributors**, for an overview of the experts who contributed to this document.

- **Use cases:** The *Data Sharing Coalition* supports the realisation of cross-sectoral use cases of *Data Sharing*[1]. In these use cases, the aim is to realise *Interoperability* across *Domains* for *Data Sharing* in a specific context. Although *Interoperability* requirements might be use case specific, the learnings from these use cases are generalised to be included in the *Data Sharing Canvas*.

[1] – https://datasharingcoalition.eu/use-cases/

- **Analysis of existing *Data Sharing Initiatives*:** The *Data Sharing Coalition* project team analyses how *Data Sharing Initiatives* that are participating in the *Data Sharing Coalition* are designed in relation to certain topics (e.g. requirements on identity proofing, standards used for *Metadata*, etc.). This provides insights into the setup of different *Data Sharing Initiatives* and therefore what is required for *Interoperability* between these *Data Sharing Initiatives* and *Domains* in general.

Figure 1: Relationship of the data sharing canvas with other documents



1.3. Related Documents

This *Data Sharing Canvas* is related to several other documents within the *Data Sharing Coalition*. Figure 1 shows these relationships, and Table 1 gives an overview of the other documents and their status. The *Data Sharing Canvas* provides input for two future documents, the *Data Sharing Coalition Blueprint*, and the *Trust Framework* for cross-*Domain Data Sharing*.

Table 1: Overview of documents related to the data sharing canvas

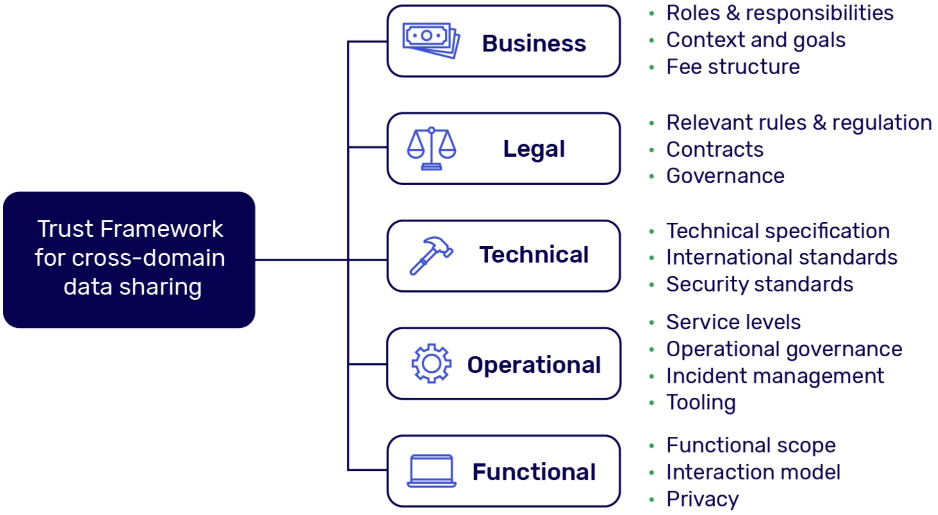
Document	Description	Status
Blueprint	The Blueprint provides an actionable approach for development of a *Data Sharing Domain* through an overview of relevant topics, including insights from the *Data Sharing Canvas*. It will inform, inspire, and accelerate *Domains* in *Data Sharing* and support them in setting up *Data Sharing* activities	To be completed by Q2 2021 as part of current phase of the *Data Sharing Coalition*
Trust Framework for cross-*Domain Data Sharing*	The set of agreements that facilitate interoperable, cross-*Domain Data Sharing* under the *Data Sharing Coalition Governance*. (see next paragraph for details)	Development to start in the next phase of the *Data Sharing Coalition*

1.4. The Future Trust Framework for Cross-Domain Data Sharing

In order to enable *Interoperability* and establish the *Trust* between the actors required to enable seamless *Data Sharing* across *Domains*, the *Data Sharing Coalition* will develop multilateral agreements on a wide range of relevant topics (e.g. digital identities, legal context, *Metadata*, etc.). These agreements will be captured in the future overarching *Trust Framework* for cross-*Domain Data Sharing*. This *Trust Framework* allows *Domains* that implement and adhere to these multilateral agreements to *Trust* each other and become *Interoperable*. This then enables *Domains* to facilitate their participants in sharing *Data* with minimal efforts with actors from other *Domains* that have also agreed to adhere to these multilateral agreements, with a minimum of additional agreements between these actors.

The *Trust Framework* will specify agreements and requirements across five disciplines: Business, Legal, Operational, Functional and Technical (BLOFT), see **Box 1** for an overview of the BLOFT model and included topics. An indicative overview of the contents and structure of the future *Trust Framework* for cross-*Domain Data Sharing* can be found in **Figure 2**.

Figure 2: Preliminary content and structure of the future Trust Framework for cross-domain data sharing

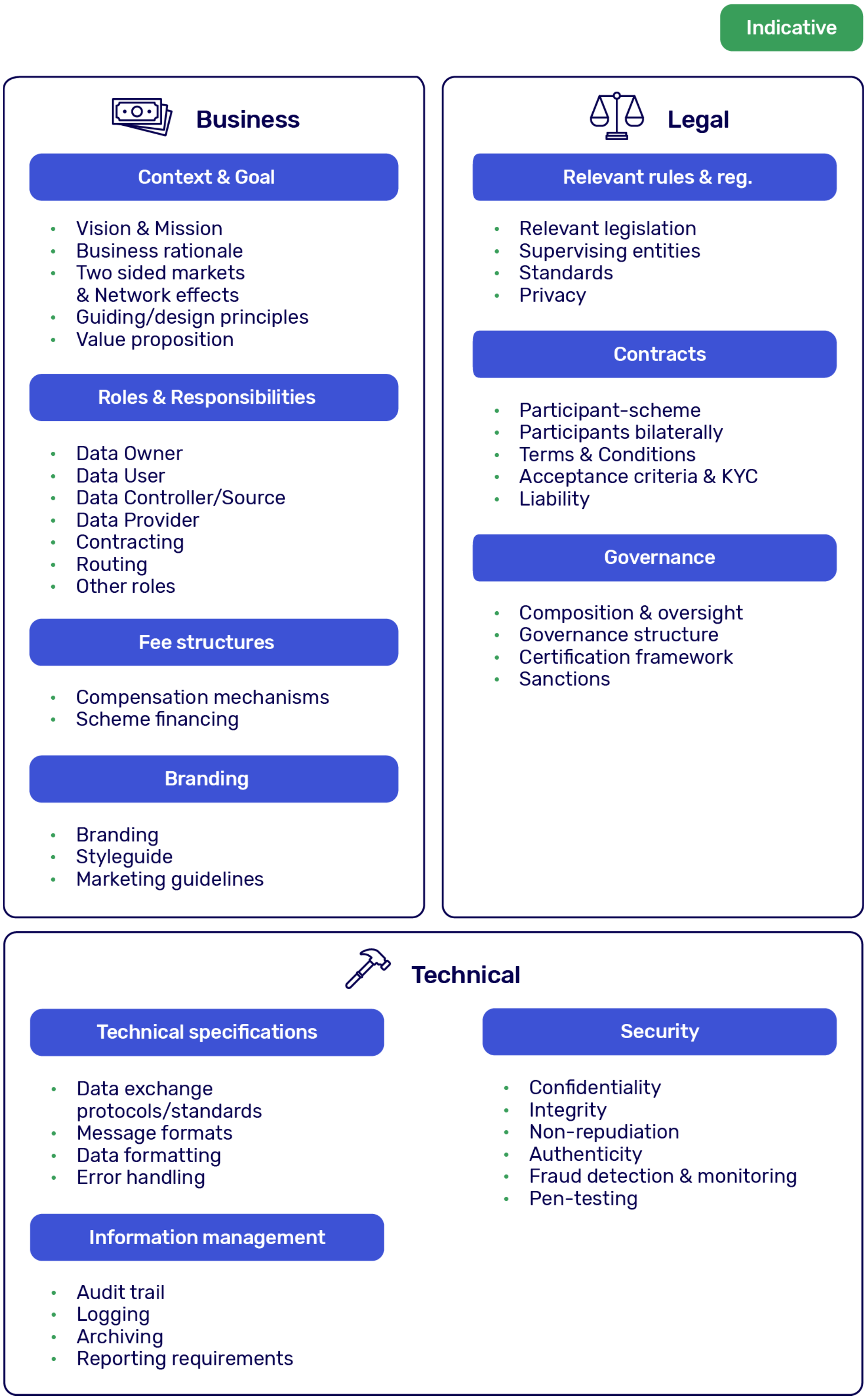


Box 1

Complete BLOFT framework

The BLOFT model has been developed based on experience in the creation of *Trust Frameworks* in the past. It contains an extensive list of topics that together form a starting point to create a blueprint for a *Trust Framework*. See **Figure 3** for a high-level overview of the topics included within the model.

Figure 3: Overview of topics in the BLOFT model



Indicative

Business

Context & Goal

- Vision & Mission
- Business rationale
- Two sided markets & Network effects
- Guiding/design principles
- Value proposition

Roles & Responsibilities

- Data Owner
- Data User
- Data Controller/Source
- Data Provider
- Contracting
- Routing
- Other roles

Fee structures

- Compensation mechanisms
- Scheme financing

Branding

- Branding
- Styleguide
- Marketing guidelines

Legal

Relevant rules & reg.

- Relevant legislation
- Supervising entities
- Standards
- Privacy

Contracts

- Participant-scheme
- Participants bilaterally
- Terms & Conditions
- Acceptance criteria & KYC
- Liability

Governance

- Composition & oversight
- Governance structure
- Certification framework
- Sanctions

Technical

Technical specifications

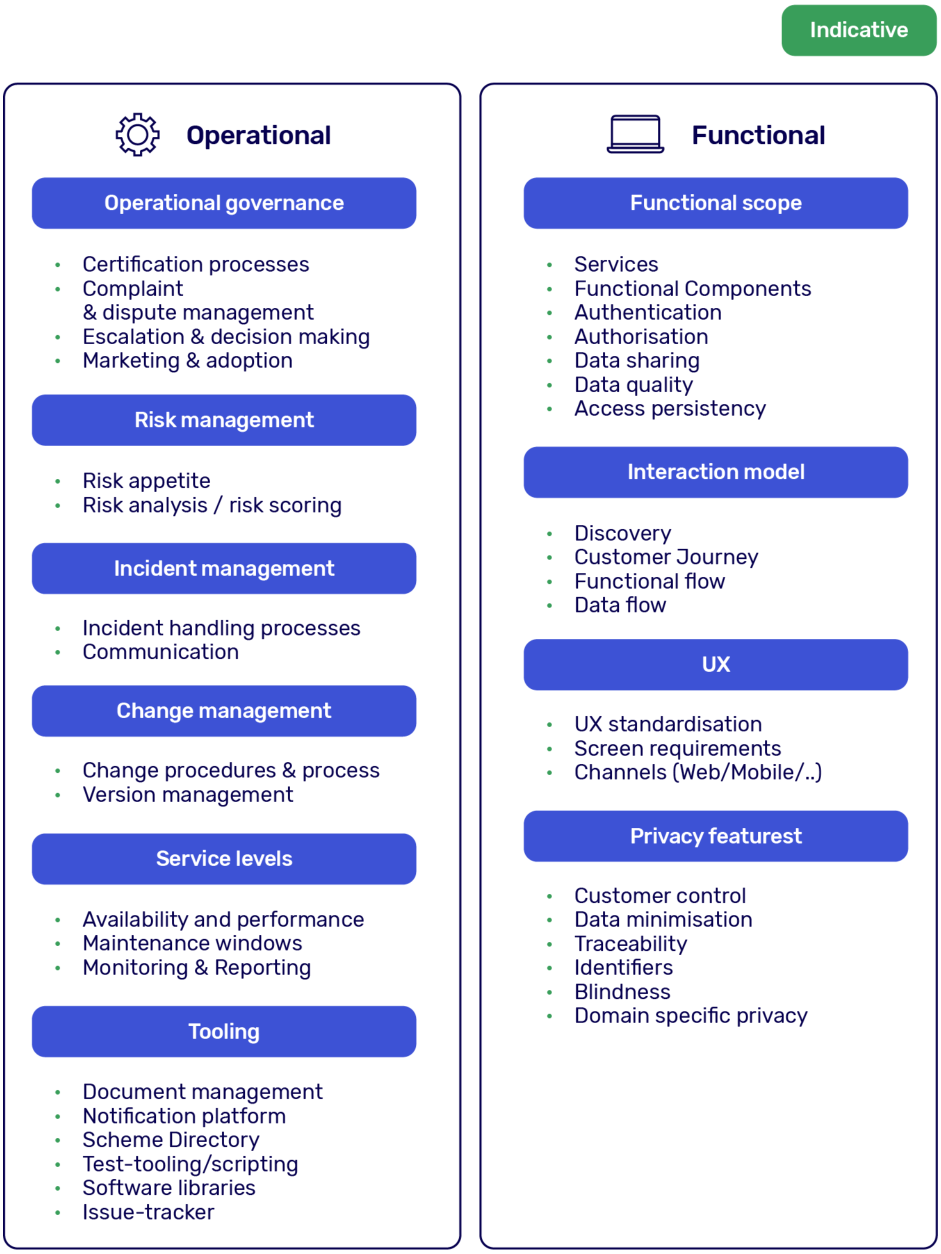
- Data exchange protocols/standards
- Message formats
- Data formatting
- Error handling

Information management

- Audit trail
- Logging
- Archiving
- Reporting requirements

Security

- Confidentiality
- Integrity
- Non-repudiation
- Authenticity
- Fraud detection & monitoring
- Pen-testing



Indicative

Operational

Operational governance

- Certification processes
- Complaint & dispute management
- Escalation & decision making
- Marketing & adoption

Risk management

- Risk appetite
- Risk analysis / risk scoring

Incident management

- Incident handling processes
- Communication

Change management

- Change procedures & process
- Version management

Service levels

- Availability and performance
- Maintenance windows
- Monitoring & Reporting

Tooling

- Document management
- Notification platform
- Scheme Directory
- Test-tooling/scripting
- Software libraries
- Issue-tracker

Functional

Functional scope

- Services
- Functional Components
- Authentication
- Authorisation
- Data sharing
- Data quality
- Access persistency

Interaction model

- Discovery
- Customer Journey
- Functional flow
- Data flow

UX

- UX standardisation
- Screen requirements
- Channels (Web/Mobile/..)

Privacy featurest

- Customer control
- Data minimisation
- Traceability
- Identifiers
- Blindness
- Domain specific privacy

At first glance, this model gives a comprehensive overview. In practice, the separation of topics is not as clear as indicated as there is overlap between topics and topics can be discussed from different perspectives. Therefore, this extensive BLOFT model is used as a starting point to ensure all topics will be discussed during the co-creation of the *Trust Framework*.

02.
Reading guide

2.1. Intended audience

People and organisations that are a stakeholder in the development of the future *Trust Framework* are the main audience of this document.
As a standalone document, the *Data Sharing Canvas* provides relevant insights for:

- Members of and people interested in the *Data Sharing Coalition* in general,
- People that are active in *Data Sharing Domains* that want to learn how to achieve interoperability with other *Data Sharing Domains*,
- People interested in (cross-*Domain*) *Data Sharing* in general.

2.2. Typography

The typography in this document follows the following rules:

- Regular text appears like this,
- *Defined terms from the glossary appear like this,*

Boxes: are used to give examples and extension on certain content

2.3. Glossary

Table 2: Glossary

Glossary item	Definition
Access Control Rules	*Policies* that are assessed and enforced prior to the establishment of a *Data Service Agreement*, which regulate how *Data Services* can be accessed
Authentication	The process where the validity of a claimed identity is verified
Authorisation	The permissions or rights of an actor (humans, machines, proxies, etc.) to perform an action
Data	A reinterpretable representation of information in a machine-readable format, suitable for communication, interpretation, or processing
Data Service	Any service offered by a *Data Service Provider* aimed at exchanging or processing *Data* (for example, this includes basic *Data Services* such as *Data* pull, *Data* push, bringing an algorithm to the *Data* as well as complex use cases based on combinations of these basic types)
Data Service Consumer	The actor that makes use of a *Data Service* offered by the *Data Service Provider*
Data Service Discovery	The mechanism through which a *Data Service Consumer* and *Data Service Provider* can find each other across *Domains*
Data Service Provider	The actor that offers a *Data Service* to the *Data Service Consumer*
Data Service Transaction	The event of executing a *Data Service* between *Data Service Provider* and *Data Service Consumer*. Depending on the type of *Data Service* the *Data Service Transaction* can be a single moment or take place for a length of time.
Data Service Transaction Agreement	The agreement (handshake) between a *Data Service Consumer* and *Data Service Provider* to enable trust and accept the terms on which the *Data Service Transaction* can take place

Glossary item	Definition
Data Sharing	The machine actionable exchange of structured *Data* through a *Data Service Transaction* between *Data Service Providers* and a *Data Service Consumers*
Data Sharing Coalition	A collaborative initiative that aims to enable organisations to easily share *Data* across *Domains*
Data Sharing Initiative	Organisation that enables *Data Sharing* in a certain *Domain* by providing a coherent set of specifications and requirements and by providing supervision
Data Standards	Provide the semantics, structure, and formatting of *Data*
Delegation	The provision of explicit rights (to perform an action) to a third party
Domain	Flexibly defined as any number organisations collaboratively working together to share *Data* to achieve a shared purpose
Dispute	When actors within the *Trust Framework* cannot settle disagreements between them according to specific service level agreements
Dispute Management	The process of managing *Disputes* when they have been reported to the *Trust Framework Authority*
Entitled Party	The entity which has rights over data. This may include the storage of the data as well as the access and usage of the data
Fair Principles	A set of principles to improve *Findability*, *Accessibility*, *Interoperability* and *Reusability* of *Data*. See Box 13 for more details.
Guiding Principle	A principle that gives direction in the decision-making process of establishing and maintaining the content of the *Data Sharing Canvas*
Governance	The management and maintenance of the *Trust Framework* agreements and network
Governing Body	The entity managing the *Governance* structure of the future *Trust Framework*
Harmonisation	Establishing agreements, standards, and requirements between actors to enable *Data Sharing* between them
Data Sharing Canvas	This document

Glossary item	Definition
Harmonisation Domain	Network of *Proxies*
Identification	The process of claiming an identity by a subject or the process of attributing/issuing an identity to a subject by an authority
Implied Regulation and Agreements	*Regulation and Agreements* that hold, but that is not explicitly stated in documentation such as agreement documentation and *Metadata*
Information Security	Preservation of the confidentiality, integrity, and availability of information though the implementation of technical or organisational information security measures
Initiative	Synonym for *Data Sharing Initiative*
Interoperability	The ability of systems of different actors to exchange *Data* in a meaningful way that is mutually understandable and satisfactory
Logging	The recording of actions with goal to create a reliable overview of events that have occurred
Metadata	*Describes* everything about *Data*, *Data Services*, and *Data Service Transactions* in *Data Sharing* that cannot be assumed to be known
Obligations and Advice	*Policies* that are assessed and enforced after the establishment of a *Data Service Agreement*, on what must be carried out after a *Data Service* is approved. *Advice* is similar to obligation with the difference that enforcement of the advice is not mandatory
Participant	(*Trust Framework*) *Participants* are parties which have joined the *Trust Framework* (potentially through *Domains*) and adhere to its agreements to facilitate *Data Services* with other *Participants*. Note that *Domains* also have *Participants* in their context. The use of this term will be clarified in the text
Policies	Define rules for access to and usage of *Data Services*, can be split into *Access Control Rules* and *Obligations* and *Advice*. *Terms and Conditions* are formalised into *Policies*
Proxy Model	Solution for multilateral Interoperability across *Domains* where different *Data Sharing Domains* implement *Proxies*. The *Data Sharing Coalition* will initially use this model for implementation of the *Trust Framework* for cross-*Domain Data Sharing*

Glossary item	Definition
Proxy	A module that translates between specifications and requirements from a *Data* sharing *Domain* and *Harmonised* specifications and requirements (and vice versa) to achieve Interoperability and trust across *Domains*
Scheme	Synonym for *Trust Framework*
Service Registry	Contains necessary *Data Service* information to perform *Data Service Discovery*. Can be considered similar to a telephone book
Terms and Conditions	Define the concepts as well as the duties and rights, the powers and liabilities that apply to the actors engaged in *Data Service Transactions*
Trust	A situation between actors where (perceived) risks are sufficiently reduced to enable *Data* sharing. The amount of risk deemed as acceptably low is determined by each actor themselves and therefore varies between actors
Trust Framework	Enables many-to-many transactions though business, legal, operational, functional, and technical agreements, tools, and processes which facilitate trusted transactions between *Participants*
Trust Framework Authority	The cross-*Domain Data Sharing* authority defines and manages the *Trust Framework*, monitors compliance, and settles disputes to facilitate cross-*Domain Data Sharing*
Trust Framework Governance	Needed to develop, manage, and maintain the *Trust Framework* agreements and network

03.
Guiding Principles

A number of principles will be used to guide the creation of the *Data Sharing Canvas* and future *Trust Framework* for cross-D*omain Data Sharing*. They provide a basis for decision-making; however, the *Guiding Principles* are no absolute truth or hard requirements but need to be considered in the context of each decision. In no particular order, the following five principles have been identified:

- Future proof,
- Trustworthy,
- Inclusive,
- As generic as possible, as specific as needed,
  Cost-efficient.

Future proof

An adaptive, extensible, and dynamic design caters for changes in technology, user behaviour, regulation, and a growing number of *Data Service Transactions*

Trustworthy

Trust between *Participants* is required on a transactional level to achieve wide-reaching adoption

Inclusive

The design should be generic, usable, and feasible for all types of *Participants* to provide a level playing field that enables all *Participants* to achieve collaborative advantages

As generic as possible, as specific as needed

By being as lightweight as possible the implementation costs for *Participants* are lowered to drive adoption. This minimisation of collaborative *Domain* requirements maximises the competitive *Domain*

Cost-efficient

Enabling cost savings at an ecosystem level, lowers the barrier of entry for potential *Participants*, which enables long term sustainable participation

3.1. Future proof

Statement

The Trust Framework for cross-*Domain Data Sharing* should be future proof and therefore extensible and non-static.

Rationale

A future proof design entails a Trust Framework which supports different implementations and is, to some extent, able to cater for changes in technology, user behaviour, regulation and for a growing number of Data Service Transactions. An adaptive, extensible, and non-static design enables scalability of the Trust Framework.

Objectives

1. Create a cooperative *Domain* that allows *Participants* to innovate their services.
2. Support scalable and fully *Interoperable Participant* implementation.

3.2. Trustworthy

Statement

The *Trust Framework* should be designed and maintained in a way that establishes *Trust* for all *Participants* and organisations, fitting the transaction context.

Rationale

Trust is required on all levels of the *Trust Framework* to achieve wide-reaching adoption. *Trust* is required across *Domains* and on a transactional level to facilitate cross-*Domain Data Service Transactions*.

Objectives

1. Enable *Trust* between actors from different *Domains*.
2. Ensure that *Data* remains sovereign and is used for authorised purposes only, as controlled by *Entitled Party*.
3. Define levels of trust dependent on a transaction context to perform a transaction.
4. Facilitate the use of required *Data* security and privacy mechanisms.
5. Be transparent towards *Participants* and related organisations.
6. Be transparent in process and *Dispute* resolution.
7. Install measures/sanctions against *Participants* and related organisations violating trust.

3.3. Inclusive

Statement

The *Trust Framework* for cross-*Domain Data Sharing* should be generic, usable, and feasible to all organisations or *Domains*, regardless of sector and *Data Sharing* context.

Rationale

Inclusivity is fundamental to enabling solution independent *Data Sharing* across *Domains* and organisations. It ensures diversity by providing a level playing field and comparable opportunities for incomparable organisations. Inclusivity leads to collaborative advantages across all *Domains*.

Objectives

1. Neutrality by ensuring a non-discriminatory approach and *Policies* towards all organisations, users, and contexts.
2. Cater for different levels of maturity of *Domains* and their *Participants*.
3. Create a level playing field for Participants.

3.4. As generic as possible, as specific as needed

Statement

The *Trust Framework* for cross-*Domain Data Sharing* rules should be as generic as possible and as specific as needed, taking into account different transaction contexts.

Rationale

This principle is needed to keep the T*rust Framework* as lightweight as possible to drive adoption. It ensures that *Participants* are not held back by restricting agreements in order to keep implementation costs low. Furthermore, it ensures a broad reach amongst sectors and types of organisations.

Objectives

1. Maximise the competitive *Domain* by minimising the collaborative *Domain* requirements.
2. Keep the *Trust Framework* as lightweight as possible.
3. Minimise risk of over-engineering.
4. Ensure solutions are generic to enable as many use cases as possible.

5. Cost-efficient

Statement
The *Trust Framework* for cross-*Domain Data Sharing* should be cost-efficient.

Rationale
Controlling costs is essential in a collaborative *Domain* as it enables a fast and effective development. It lowers the threshold for organisations to participate and enables long-term sustainable participation.

Objectives

1. Enable cost savings at an ecosystem level, financially or in terms of effort.
2. Use proven and open standards where possible.
3. Learn from (inter)national best practices.
4. Ensure a transparent cost and benefit structure.
5. Minimise cost of entrance and impact of implementation.
6. Consider impact for *Participants* when changes occur in the future.

Harmonisation Topics

An overview of several key concepts for cross-*Domain Data Sharing* are introduced in Chapter 4. The following chapters in this section detail topics which have been identified to be relevant for cross-*Domain Data Sharing* and present the insights that were gained together with the *Expert Group*. In development of the future *Trust Framework* for cross-*Domain Data Sharing*, these topics and insights will be used as a basis.

Note: the order in which these topics are presented does not indicate relative importance.

04.
Introducing Cross-Domain Data Sharing

This chapter presents the Coalition's views on the key concepts for cross-*Domain Data Sharing* and provides some initial insights on how they could be implemented to achieve Interoperable *Data Sharing* across *Domains*. For that purpose, it is deemed useful to have a preliminary idea of what the final interoperability model will look like so that topics and concepts can be discussed specifically within a practical context to avoid deeply theoretical discussions. The exact manifestation and functionality of this model will be detailed in the *Trust Framework*.

4.1. Data Sharing

Data Sharing is the act of exchanging *Data* between a *Data* provider and a *Data* consumer. In the context of the *Data Sharing Coalition*, this provider and consumer per definition reside in different *Domains*, and therefore we talk about cross-*Domain Data Sharing*. Cross-*Domain Data Sharing* is enabled through a *Data Service* between a *Data Service Provider* and a *Data Service Consumer*, from different *Domains*. The *Data Sharing Coalition's* primary focus is on the transactional exchange of structured *Data* as this is the most scalable type of *Data Sharing*. In general, *Data Services* exist in different forms. See Table 3 for a non-exhaustive overview of the basic *Data Service* types. All basic *Data Services* can be used to achieve *Data Sharing* and generate value for the actors involved.

Table 3: A non-exhaustive overview of data service types

Data Service	Description
Data Pull	The *Data Service Consume*r acquires *Data* from the *Data Service Provider* so that the consumer can make use of the *Data*
Data Push	The *Data Service Consumer* pushes their *Data* to a *Data Service Provider* so that the provider can make use of the *Data*
Algorithm Pull / *Data* visiting	The *Data Service Consumer* requests an algorithm from the *Data Service Provider* to be sent so that it can process *Data*. The *Data* remains at the source at all times
Algorithm Push / *Data* visiting	The *Data Service Consumer* pushes an algorithm to a *Data Service Provider* so that the algorithm can process the *Data*. The *Data* remains at the source at all times

The basic *Data Service* types described in Table 3 can be combined to realise more complex use cases. For example, a single use case can include multiple *Data* pull services to combine *Data* from several different sources. Note that *Data Sharing* through these *Data Services* can be considered as a transactional *Data Sharing* model. Therefore, the combined act of performing and consuming these *Data Services* can be called a *Data Service Transaction*. Another alternative *Data service type* is the *Data* publication model, where *Data* should be always available for access by a *Data Service Consumer*. This can be captured within this model as a *Data* pull transaction.

The *Data Service* type of *Data* visiting differs from *Data* pull and *Data* push in that *Data* never leaves the source and the *Entitled Party* controls the *Data* at all points in time. From a technical perspective, the required implementation to achieve DATA visiting is much more complex than *Data Push/Data Pull*. This is as algorithms must be transferred across *Domains* and these cannot be easily translated, compared to data. Further additional requirements are needed for allowing an algorithm to be run in another *Domain* compared to the transfer of *Data*. Therefore, *Data* visiting services are not the main focus of this document. For multilateral *Data* visiting services, the *Fair Principles* (see **Box 13**) give a framework of relevant concepts for developing the *Data Service*. This overview of principles should be considered to enable a scalable solution. **Table 4** presents some concrete examples of how *Data Sharing* is done/can be done in different *Domains* and explicitly describes who has the roles of *Data Service Consumer* and *Data Service Provider*.

Table 4: Data sharing examples

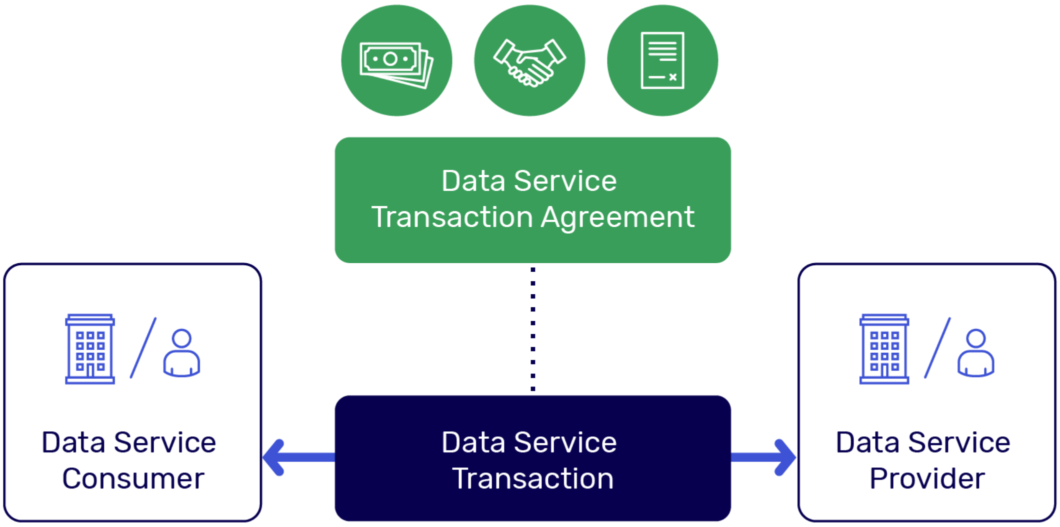
Use case		Data service type	Data service consumer	Data service provider
Green Loans	A house owner wants to share *Data* from his smart energy meter with his loan advisor and prospect loan provider so that he can obtain a loan for energy saving measures (e.g. solar panels). The loan advisor pulls the Data from smart meter	*Data* Pull	Interme-diary (loan advisor)	DSO (Distribution System Operator)
Virus Outbreak *Data* Network (VODAN)	A researcher in the health domain wants to analyse *Data* owned by other research institutions to discover patterns in the current COVID-19 pandemic and potential future epidemics. The researcher pushes the algorithm to the *Data* repository owned by a research institution	Algo-rithm Push	Researcher	Research institution
Smart Cleaning	Cleaning parties want to make use of building sensor *Data* from the sensor providers in a building, so that they can act on the *Data* with 'demand-based' cleaning services. The cleaning party's *Data* service processor (software provider) pulls the *Data* from the sensor provider	*Data* Pull	*Data* service processor (on behalf of cleaning parties)	Sensor provider

Use case		Data service type	Data service consumer	Data service provider
Tax administration	Accountants can push their client's income, VAT and profit tax *Data* towards the tax authority such that the tax authorities, in the role of *Data* service provider, can process tax returns automatically. The accountants push the *Data* to the tax authority	*Data* Push	*Account-ants*	Tax authority
Sharing shipment *Data* for improved risk management	A transport carrier in the logistics sector wants to enable the sharing of actual consignment *Data* using the e-CMR (digital waybill) with an insurer so that the claim handling process runs as smoothly as possible, and the insurer can assess risk more accurately. The Insurer pulls the *Data* from the e-CMR	*Data* Pull	Insurer	e-CMR provider

4.1.1. Data service transaction

As part of each *Data Service Transaction* between a *Data Service Consumer* and a *Data Service Provider*, an *Agreement* between the parties must be established, see Figure 4 (See Appendix II for the steps to reach a *Data Service Transaction Agreement*). This *Data Service Transaction Agreement* is specific to the transaction and its context and can be considered a handshake between the actors to confirm *Trust* and the mutual acceptance of the specific *Terms and Conditions* under which the *Data Service Transaction* takes place. In addition to the characteristics of the *Data Service* itself, many topics are relevant for the *Data Service Transaction Agreement* including, but not limited to: *Identification, Authentication & Authorisation, Terms and Conditions, Governance, and Information Security aspects*. See the Harmonisation Topics section, for further details about each topic.

Figure 4: Overview of a data service, including the data service transaction agreement



4.2. Interoperability and Harmonisation

Whenever organisations collaborate, they can make agreements with each other as they see fit to facilitate this collaboration. Within the context of the *Data Sharing Coalition*, a *Domain* is flexibly defined as any number of organisations collaboratively working together according to agreements to share *Data* to achieve a shared purpose. Examples include, but are not limited to:

- An initiative (e.g. a scheme or platform) which facilitates *Data* sharing between 100+ participant organisations,
- Organisations which share *Data* due to legal requirements, (e.g. sharing financial *Data* under PSD2),
- A small number of organisations which bilaterally share *Data* with each other based on proprietary standards.

The *Data Sharing Coalition* aims to also enable *Data Service Transactions* across *Domains* between actors that are part of different *Domains*, with a minimum number of additional agreements between these actors and despite the fact not all *Domains* adhere to the same agreements. This is enabled by a concept known as Interoperability; in the context of *Data Sharing*: "The ability of systems of different actors to exchange *Data* in a meaningful way that is mutually understandable". There are multiple approaches to achieve Interoperability.

In theory, *Interoperability* between *Domains* can be realised through full *Harmonisation* of *Domains*. This is the ideal solution to achieve multilateral Interoperability between *Domains* to enable *Data* Sharing across *Domains*. This means that existing *Data Sharing Domains* adjust their own requirements and implementations to follow a common, cross-*Domain* design. However, *Harmonisation* of *Domains* impacts all *Domain Participants* as they would need to adjust existing implementations, requiring significant investments. Given the impact (in effort and cost) full *Harmonisation* has on their *Participants*, immediate adoption of fully harmonised agreements by individual *Domains* will most likely be limited.

Another option to achieve bilateral cross-*Domain Data Sharing*, that does not require full *Harmonisation* of all *Domains*, is that individual *Domains* organise custom bilateral *Interoperability* for their use cases between only the actors involved. For this, they need bilateral agreements with organisations from another *Domain* and define and implement their own interoperable requirements. Such bilateral agreements will allow their single use case for cross-*Domain Data Sharing* but are dependent on individual participants implementing specific harmonised solutions and will therefore limit large scale cross-*Domain Data Sharing*.

Therefore, the *Data Sharing Coalition* initially aims for multilateral Interoperability between *Domains* through partial *Harmonisation* instead of full *Harmonisation*. Partial *Harmonisation* of a *Domain* can be realised through a new role: a *Proxy*. The role of a *Proxy* is to absorb the complexity of *Harmonisation* for *Domains* and *Participants* as much as possible by implementing all *Harmonisation Requirements*. This enables a *Data Service Provider* in one *Domain* to provide a *Data Service* to a *Data Service Consumer* in another *Domain*, while limiting impact for both the *Data Service Provider* and the *Data Service Consumer*.

4.3. The Proxy Model

A practical solution to enable multilateral Interoperability across *Domains* is for each *Domain* to implement a *Proxy*. *Proxies* are systems which are to be used by every *Domain* with the function of translating between *Domain* specific specifications and common, *Harmonised* inter-*Domain* specifications. Note, the *Proxy* model is the working hypothesis for a model to establish cross-domain interoperability with minimal impact on *Domains*. Its exact functionalities are not specifically defined yet and are subject to change.

The main functionality of the *Proxies* is to translate *Domain* specific transactions to their harmonised equivalents:

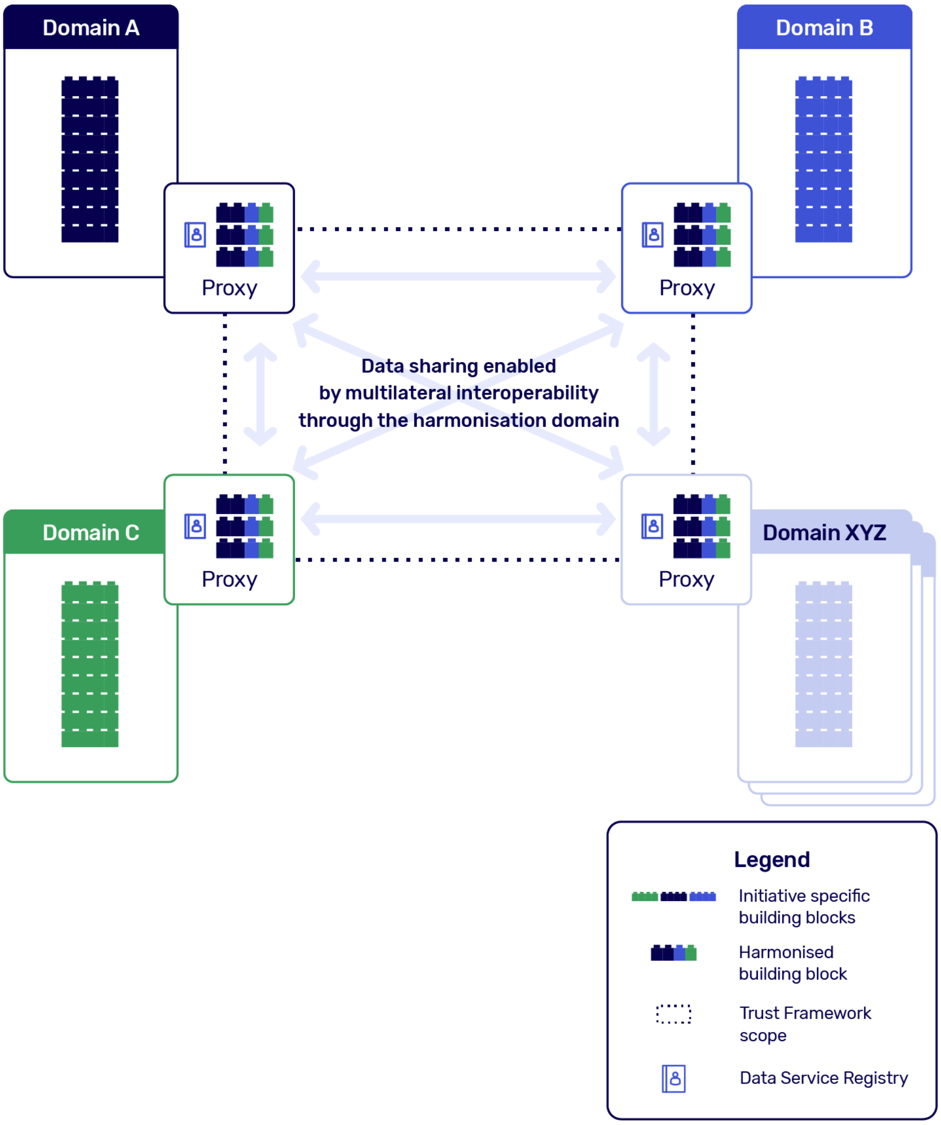
- *Proxies* will translate *Domain* specific language to a harmonised language in the *Harmonisation Domain* to enable multilateral end-to-end Interoperability,
- *Proxies* will facilitate *Trust* across *Domains* by conforming to the rules and agreements of the *Trust Framework*,
- *Proxies* will enable the discovery of *Data Services* across *Domains*.

The *Proxies* implemented by all *Domains* form a network, the inter-*Domain* space or *Harmonisation Domain*, which enables each *Domain* to share *Data* effortlessly with other *Domains*. The *Proxy* network will facilitate an *Interoperable* transaction capability and a understanding on concepts like *Trust* and security across *Domains*. The *Trust Framework* for cross-*Domain Data Sharing* will define the agreements on the setup of these *Proxies*.

Note that this many-to-many *Proxy* model solution does not exclude further bilateral agreements and technical implementations between *Domains* and/or actors in those *Domains*. However, as this is not aligned with the desire for a scalable solution, it shall not be the aim of the *Trust Framework*.

Individual *Domains* are responsible for the implementation, set-up and operation of a *Proxy* which adheres to the *Trust Framework* for cross-*Domain Data Sharing*. Figure 5 shows a visual representation of the *Proxy Model*.

Figure 5: Visual representation proxy model

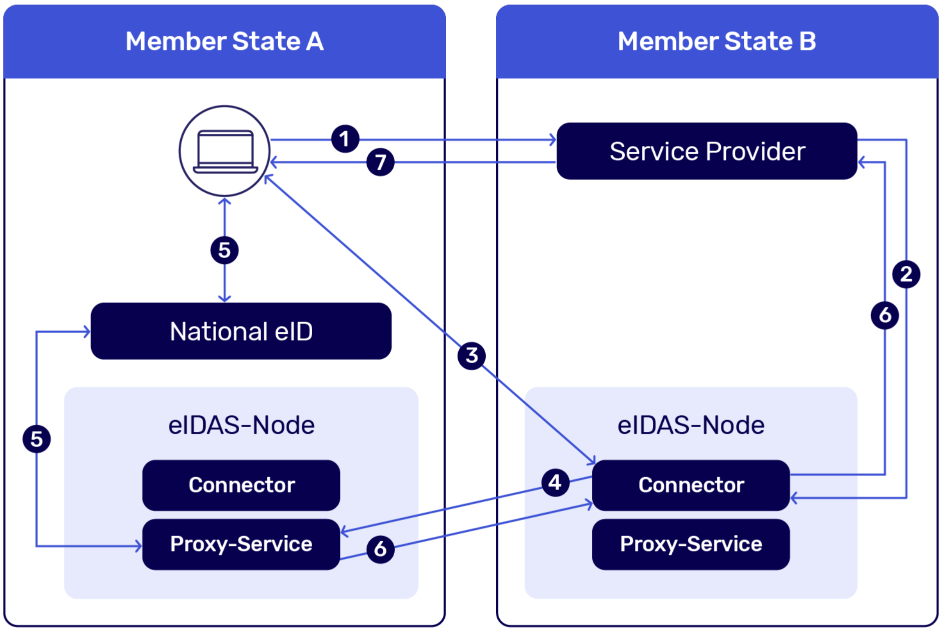

Domain A
Domain B
Proxy
Proxy
Data sharing enabled
by multilateral interoperability
through the harmonisation domain
Domain C
Proxy
Proxy
Domain XYZ
Legend
Initiative specific building blocks
Harmonised building block
Trust Framework scope
Data Service Registry

Similar uses of *Proxies* to enable cross-*Domain Interoperability* are already applied at scale in multiple contexts, see Box 2 for an example in the use of *Proxies* in eIDAS. However, a *Proxy Model* is no silver bullet. Whether *Data* will be shared across *Domains* will always depend on case specifics and decisions made by individual *Participants*.

Box 2:
Proxies in eIDAS

The eIDAS-nodes, formerly known as 'Pan European Proxy Server' (PEPS) are an implementation of *Proxies* used to enable *Interoperability* of digital identities across EU member states. Figure 6 shows how eIDAS Nodes are used between two member states[2].

Figure 6: Overview of the eIDAS authentication scheme depicting eIDAS nodes



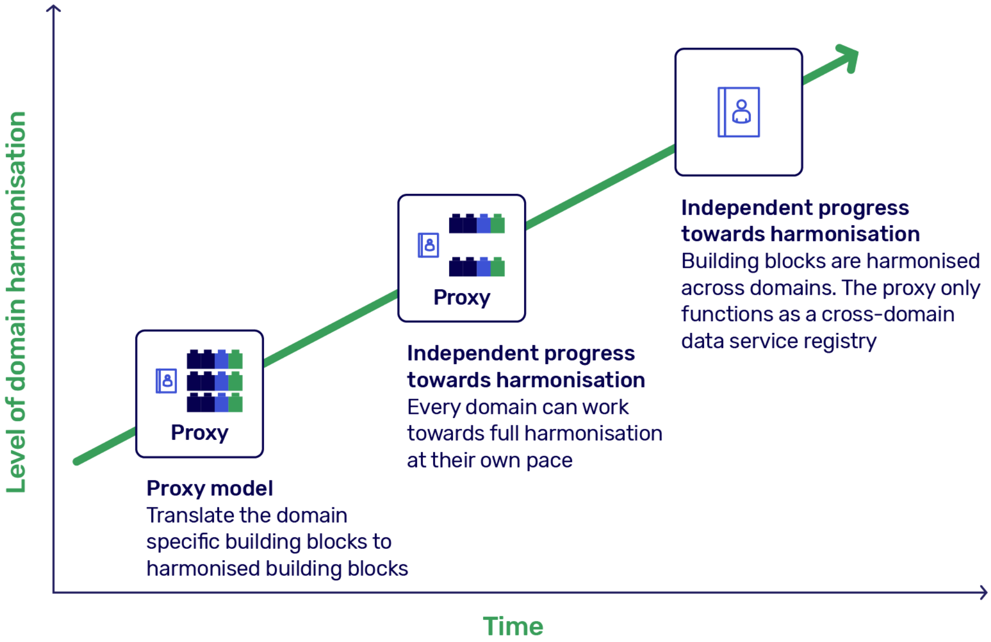
eIDAS is based on well-established standards, such as SAML, to achieve *Interoperability* and high security between EU member states. EU member states use different national eID solutions, that often involve nation specific implementations. The eIDAS Nodes translate the specific national solutions such that they can be understood across borders.

Harmonisation Topics

[2] – https://docs.wso2.com/display/IS570/ Electronic+Identification%2C+Authentication+and+Trust+Services+Regulation

The *Proxy* model further can serve as a foundation for future developments from *Domain Interoperability* towards full *Domain Harmonisation* through a phased approach. Individual *Domains* can work towards full *Harmonisation* at their own pace, following their own change management processes. The initial implementation of *Proxies* will become lighter, as the *Harmonised* components are transferred and embedded within the systems of participants of the Domain. Eventually, a Proxy may only need to carry out the function of cross-*Domain Data Service Registry* when all other elements are *Harmonised* within the *Domain*. See **Figure 7** for the possible development of *Proxies* in *Domains*.

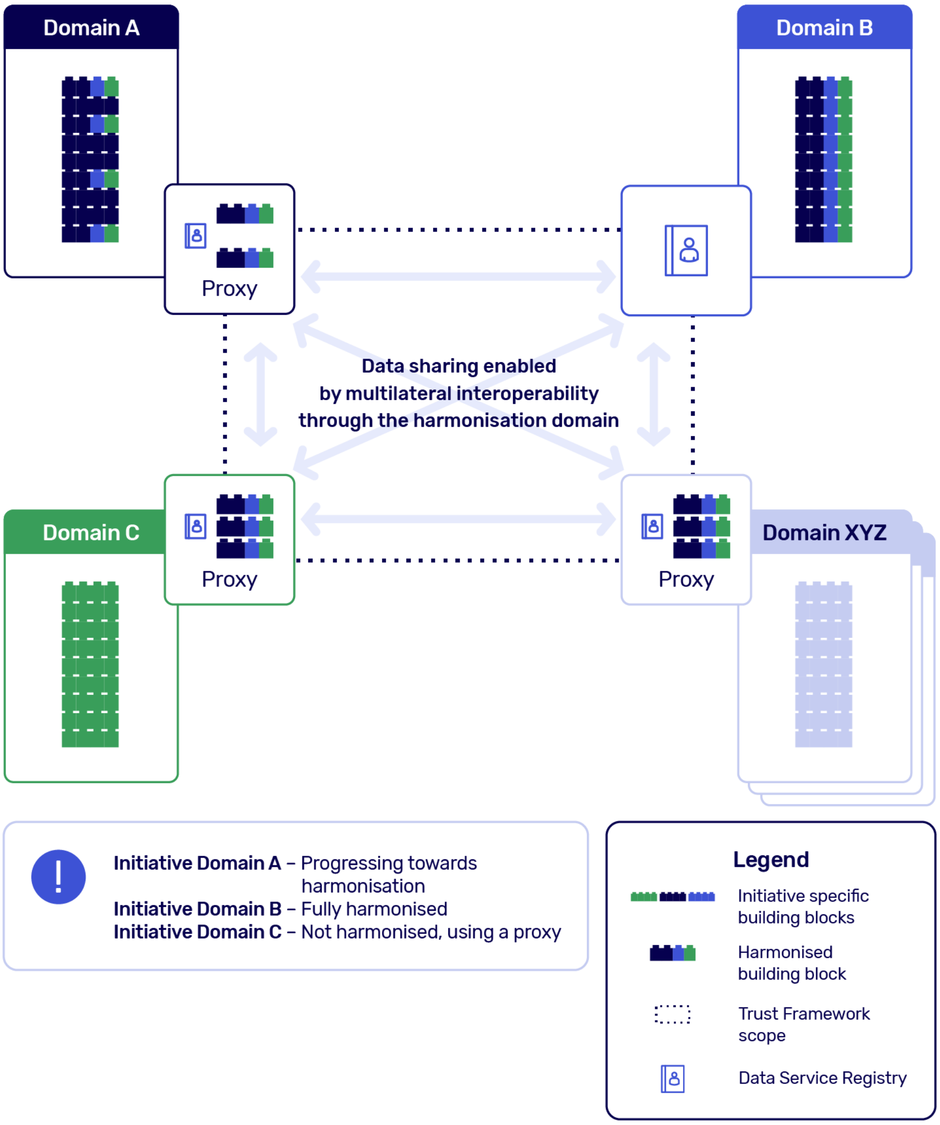
Figure 7: Development from the proxy model to full harmonisation


Level of domain harmonisation
Time
Proxy
Proxy model
Translate the domain specific building blocks to harmonised building blocks
Proxy
Independent progress towards harmonisation
Every domain can work towards full harmonisation at their own pace
Independent progress towards harmonisation
Building blocks are harmonised across domains. The proxy only functions as a cross-domain data service registry

It is very unlikely that *Domains* will progress towards full *Harmonisation* at the same pace, as *Domains* depend on the implementation pace of their *Participants*. However, the *Proxy* model enables *Domains* to remain fully interoperable at different levels of *Harmonisation*. The rules and agreements which hold for fully *Harmonised Domains* are the same as those for *Domains* with *Proxy Model* implementations. Therefore, *Data* can be shared across *Domains* irrespective of their phase of development.

Furthermore, for new developments of *Data Sharing Domains* or organisations aiming to develop their internal *Data Sharing*, the rules and agreements of a *Proxy* can be easily adopted to ease their internal development. This means these actors may be fully harmonised from the initial development. See **Figure 8** for a visual representation with *Domains* in different levels of progression towards full *Harmonisation*.

Figure 8: Data can be shared across Domains at different levels of progression toward full Harmonisation

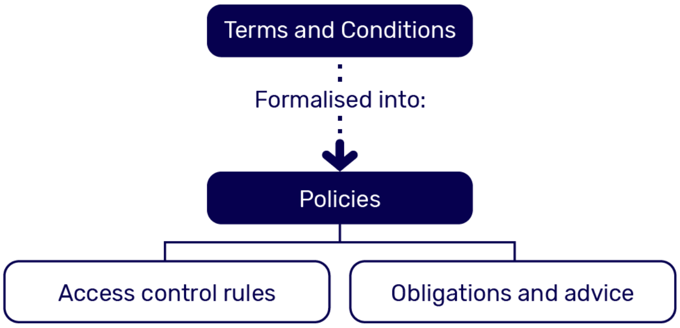

Domain A
Domain B
Proxy
Data sharing enabled
by multilateral interoperability
through the harmonisation domain
Domain C
Proxy
Proxy
Domain XYZ
!
Initiative Domain A – Progressing towards harmonisation
Initiative Domain B – Fully harmonised
Initiative Domain C – Not harmonised, using a proxy
Legend
Initiative specific building blocks
Harmonised building block
Trust Framework scope
Data Service Registry

05.
Data Service Terms and Conditions

5.1. Introduction

Data Service Terms and Conditions define the concepts, duties, rights, powers, and liabilities that apply to the actors on both sides of a *Data Service Transaction* that are captured in a *Data Service Transaction Agreement*. *Terms and Conditions* are formalised into *Policies*, which can be split into *Access Control Rules*, *Obligations* and *Advice* (see Figure 9). A *Data Service's Terms and Conditions* are set by the *Data Service Provider* directly and/or are (partially) a result of the rules of the *Data Sharing Domains* the *Data Service Provider* belongs and adheres to.

Figure 9: Terms and conditions are formalised in policies,
which can be split into access control rules and obligations and advice



5.2. Relevance

To enable Interoperability, the *Data Service Consumer* needs to understand the *Terms and Conditions* of a *Data Service* as specified and communicated by the *Data Service Provider*. Therefore, it is required that *Terms and Conditions* (formalised into *Policies*) can be machine interpreted across *Domains*. This way *Proxies* should be able to map individual *Policies* and the pieces of evidence that demonstrate adherence to these *Policies* to *Domain* specific *Policies* and evidence and vice versa. To achieve this, a shared understanding of and language for *Policies* and evidence is needed.

5.3. Description

To complicate things, within a single *Domain*, not everything that *Participants* should adhere to is made explicit and as a result not all *Data Service Terms and Conditions* are made explicit. They can also be ‘hidden’ in rule books, legislation, or certifications specific to the *Domain*, defined here as *Implied Regulation and Agreements*. In this case, both the *Data Service Provider* and *Data Service Consumer* operating within the same *Domain* are aware of these *Implied Regulation and Agreements*. However, *Participants* in other *Domains* are not likely to be aware of these *Domain* specific *Implied Regulation and Agreements*. Therefore, to enable cross-*Domain Data Service Transaction Agreements*, these *Implied Regulations and Agreements* should be made explicit. *Data Service Providers*, or *Proxies* on their behalf, may decide to make (parts of) the *Implied Regulation and Agreements* explicit and require explicit proof of adherence to those *Implied Regulation and Agreements*, similarly to other *Data Service Terms & Conditions*.

Whether or not proof of adherence to the *Terms and Conditions* is required as prerequisite for its service, is up to the *Data Service Provider* which requires a balanced consideration on factors such as risks, costs, and usability.

This chapter explains the need for a shared language and understanding on *Policies* in 5.3.1 and the split of *Policies* in 5.3.2.

5.3.1. Creation of a shared language and understanding

A shared language and understanding are needed to enable unambiguous communication on *Policies* and evidence that demonstrates the adherence to these *Policies*. It is not realistic to expect to create a shared language for all individual *Policies* given their variety across *Domains*. A solution might be to create *Policy* clusters and levels of adherence to *Policy* clusters (to express an assurance level). These *Policy* clusters might make it easier to define a shared language, as the clusters and levels might enable simple comparison across *Domains*.

Policy clusters are sets of *Policies*, in which *Policies* belong to the same cluster if they pursue the same objective. See Appendix III for a first set-up of *Policy* clusters. *Policy* cluster levels define whether a *Domain* meets specific criteria within a *Policy* cluster, based on underlying *Policies*. *Policy* cluster levels are formed differently for each cluster and can be defined along different axes (e.g. nominal, ordinal and interval) based on *Data Service Provider* requirements.

Policy clusters and *Policy* levels should be further explored and defined in the *Trust Framework* for cross-*Domain Data Sharing*.

In the eIDAS *Trust Framework*, the principle of creating a shared language for *Policies* via clusters and levels for clusters is applied at scale. This is further detailed in Box 3.

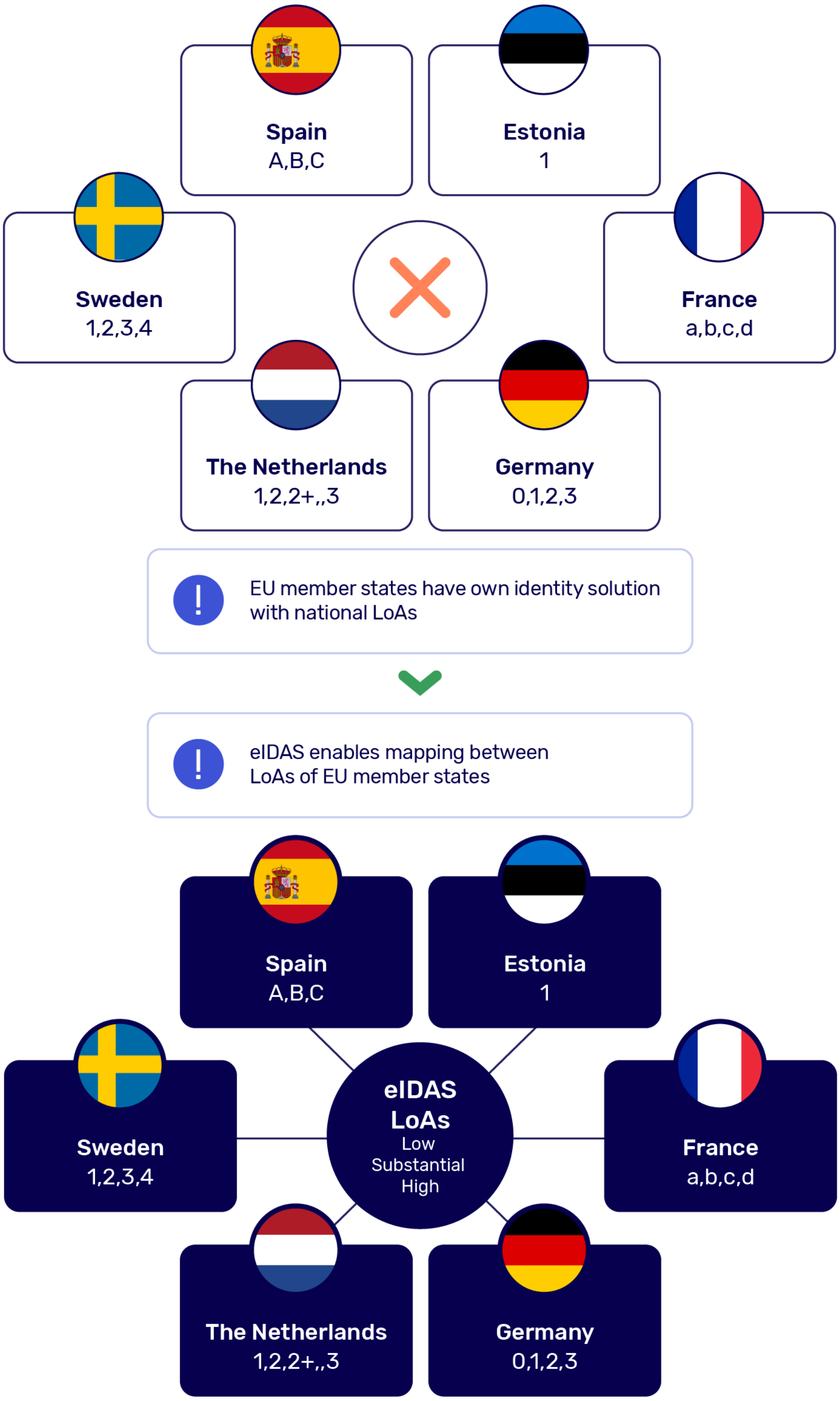
Box 3
eIDAS

In the last 15-20 years, most EU member states have developed their own national digital identity solutions for citizen *Authentication* based on member state specific requirements, resulting in member state specific Levels of Assurance (LoAs) for their digital identity.

In line with Europe's ambition to create one Digital Single Market, the European Union strived to enable people and businesses to use their own national electronic *Identification* schemes (eIDs) to access public services available online in other EU countries. To achieve this, the EU has created the common eIDAS3 framework. eIDAS (**e**lectronic **ID**entification, **A**uthentication and trust **S**ervices) is an EU regulation on electronic identification and trust services for electronic transactions in the European Single Market.

The variety of *Policies* and LoAs across countries initially made it impossible to create a shared language on individual Policies across EU member states. The eIDAS framework allows for mapping of national eID solutions and its member state specific LoAs to generic eIDAS LoAs, enabling *Interoperability*.

Figure 10: Creation of a mapping between levels of assurance in EU member states



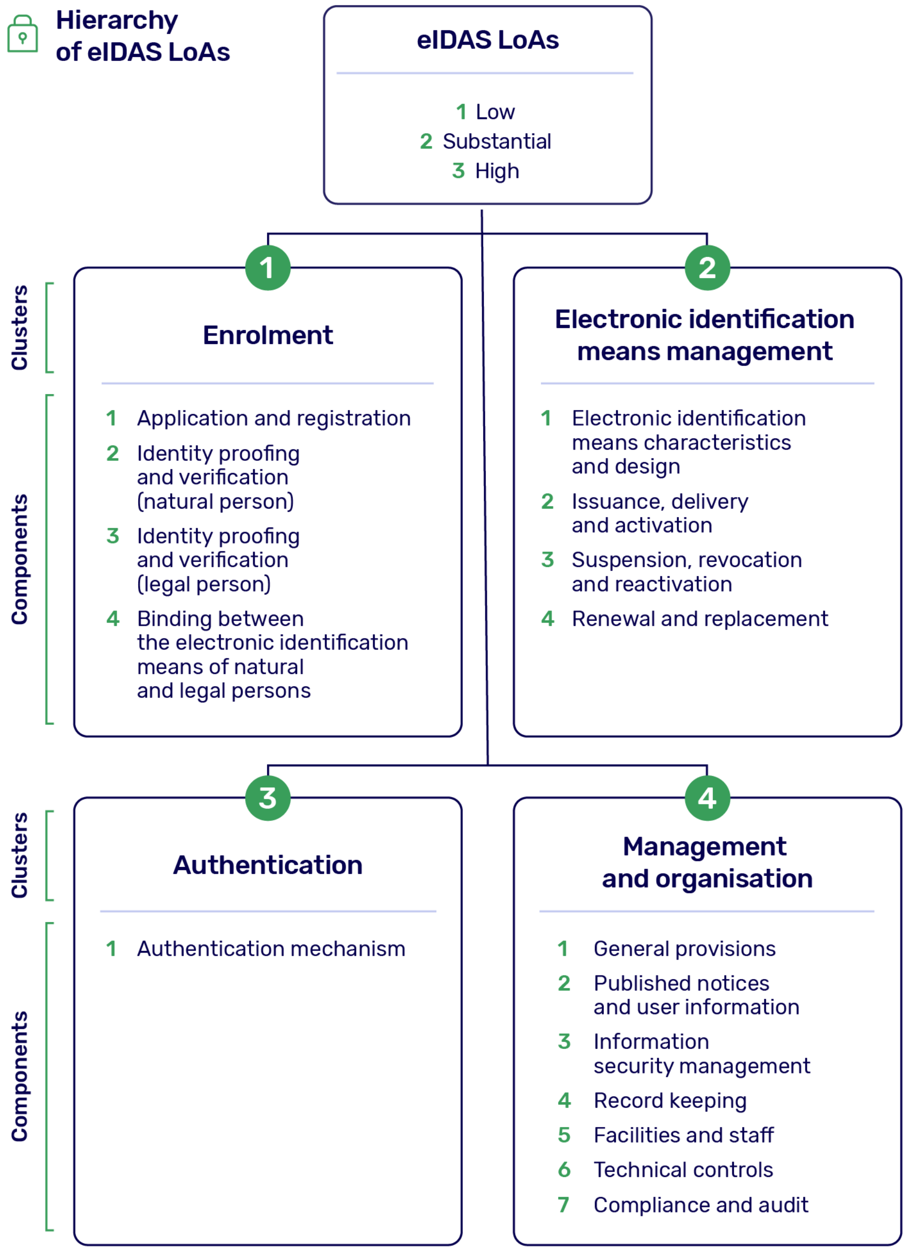
EU member states have own identity solution with national LoAs

eIDAS enables mapping between LoAs of EU member states

[3] – Commission implementing regulation (EU) 2015/1502, Office journal of the European Union

eIDAS *Policy* clusters consist of multiple components, with underlying *Policies*. The overall LoA of eIDs is based on the LoA of several clusters, where the lowest LoA of the individual clusters will determine the overall LoA. Each cluster contains several components, and the LoA of the cluster will be based on the lowest LoA of all the components. Per component, conditions are specified defining how a LoA can be reached.

Figure 11: Hierarchy of eIDAS LoAs



Hierarchy of eIDAS LoAs

eIDAS LoAs

1 Low
2 Substantial
3 High

Clusters

Components

1 Enrolment

1 Application and registration
2 Identity proofing and verification (natural person)
3 Identity proofing and verification (legal person)
4 Binding between the electronic identification means of natural and legal persons

2 Electronic identification means management

1 Electronic identification means characteristics and design
2 Issuance, delivery and activation
3 Suspension, revocation and reactivation
4 Renewal and replacement

Clusters

Components

3 Authentication

1 Authentication mechanism

4 Management and organisation

1 General provisions
2 Published notices and user information
3 Information security management
4 Record keeping
5 Facilities and staff
6 Technical controls
7 Compliance and audit

Overall LoA is based on LoA of clusters.
Lowest LoA of clusters defines overall LoA

LoA per cluster is based on LoA of components.
Lowest LoA of component defines LoA of a cluster

Conditions are specified per component leading to a certain LoA

LoA for 2.1 Electronic identification means characteristics and design

Assurance Level	Elements needed
Low	**1.** The electronic identification means utilises at least one authentication factor. **2.** The electronic identification means is designed so that the issuer takes reasonable steps to check that it is used only under the control or possession of the person to whom it belongs.
Substantial	**1.** The electronic identification means utilises at least two authentication factors from different categories. **2.** The electronic identification means is designed so that it can be assumed to be used only if under the control or possession of the person whom it belongs.
High	Level substantional, plus: **1.** The electronic identification means protects against duplication and tampering as well as against attackers with high attack potential. **2.** The electronic identification means is designed so that it can be reliably protected by the perso to whom it belongs against use by others.

5.3.2. Policies

Terms and Conditions are formalised into *Policies*, which can be split into *Access Control Rules* and *Obligations and Advice*, depending on whether the *Policies* are enforced before or after the *Data Service Agreement* is established.

Access control rules

Access Control Rules are *Policies* that are assessed and enforced prior to establishing the *Data Service Agreement* and validated at the moment of a *Data Service Transaction*. Some *Access Control Rules* are in place to assess the likelihood of adherence to *Implied Regulation and Agreements* (e.g. sector regulation and frameworks and general laws and regulation, through certifications and audit reports). Examples of *Access Control Rules*:

- Subject attributes (e.g. LoA of identity, role and age)
- Context/environment attributes (e.g. location and time)
- Proof of security certifications (e.g. ISO 27001)

Obligations and advice

Obligations and Advice are *Policies* that are assessed and enforced after the *Data Service Agreement* is established. They prescribe future requirements and optional guidance to the *Data Service Consumer*. It is up to the *Data Service Provider* (or the *Domain* rules to which the *Data Service Provider* adheres to) to determine whether a *Policy* is *Obligation* or *Advice*. *Policy* enforcement may vary (e.g. none, ad-hoc checks or by audit). Examples of *Obligations and Advice Policies*:

- Usage scope
- Storage requirements
- Time to live for datasets (deletion of *Data*)
- Pricing and other financial (reporting) requirements
- Operational reporting requirements

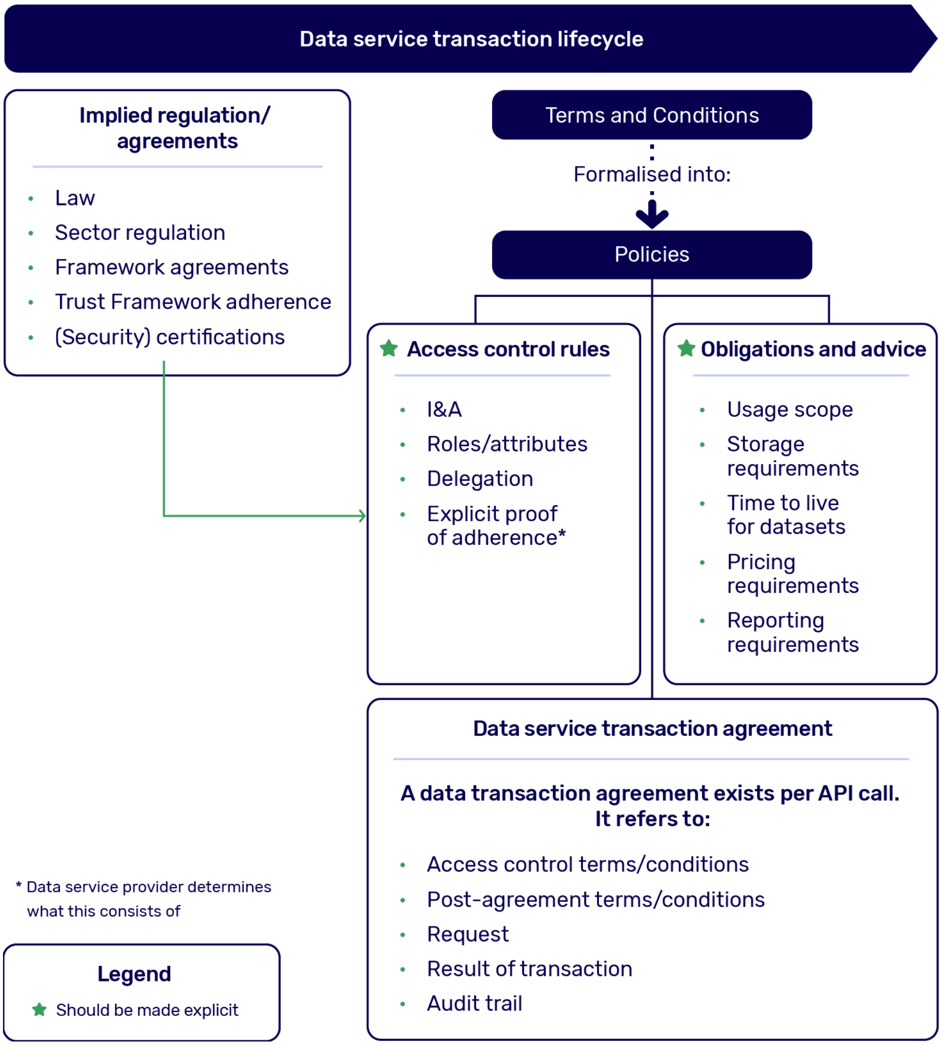
See **Appendix III Terms and Conditions**, for a complete overview of *Policies* split into *Access Control Rules and Obligations* and *Advice within Data Sharing Coalition* use cases.

One *Obligation* that requires attention is the restriction of the reselling of shared *Data*, which is a main worry of many organisations. If applicable to the transaction context, this obligation should be made explicit in the *Data Service* description before a *Data Service Transaction* (see **Chapter 9.3.1**). Depending on the *Data Service*, a burden of proof for the adherence to this *Obligation* could be mandated. The burden of proof could either be captured in required *Logging* (see **Chapter 10.3.3**) or in *Metadata* (see **Chapter 14**).

Figure 12 provides an overview of the relationship between a *Data Service Transaction Agreement*, the associated transaction (the API call) and the *Terms and Conditions* (formalised into *Policies*) within a *Data Service Transaction* lifecycle.

The term '*Data transaction* lifecycle' is introduced as a term to distinguish between the sequence in which *Policies* should be adhered to and the actual *Data Service Transaction*.

Figure 12: Data service transaction lifecycle with examples of data service transactions agreements and policies


Data service transaction lifecycle
Implied regulation/ agreements
Law
Sector regulation
Framework agreements
Trust Framework adherence
(Security) certifications
Terms and Conditions
Formalised into:
Policies
★ Access control rules
I&A
Roles/attributes
Delegation
Explicit proof of adherence*
★ Obligations and advice
Usage scope
Storage requirements
Time to live for datasets
Pricing requirements
Reporting requirements
Data service transaction agreement
A data transaction agreement exists per API call. It refers to:
Access control terms/conditions
Post-agreement terms/conditions
Request
Result of transaction
Audit trail
* Data service provider determines what this consists of
Legend
★ Should be made explicit

It is expected that only *Access Control Rules and Obligations* and *Advice Policies* will be specified in a *Data Service Transaction Agreement*, as these are relevant for the execution of a single API call.

In developing the future *Trust Framework* for cross-*Domain Data Sharing*, it should be explored to what detail *Implied Regulation and Agreements* should be made explicit.

06. Identification, Authentication and Authorisation

6.1. Introduction

For actors to reach a *Data Service Transaction Agreement*, they must be able to identify, authenticate and authorise other actors. It is required that actors are able to identify those they are interacting with and assess their assurance level (for *Identification and Authentication*) and know what permissions those other parties have (*Authorisation*). *Access Policies* define whether an entity should be permitted access to an object (target *Data*, database access, algorithm access, etc.). *Access Controls* are the mechanisms and methods used to enforce *Access Policies* using *Authorisation* mechanisms. Within *Domains*, various types of *Identification*, *Authentication* and *Authorisation* mechanisms are used and while this suffices for activities within a specific *Domain*, it is not trivial how these mechanisms and the resulting statements and evidence can find their way to another *Domain*. Furthermore, it is important that these mechanisms are implemented in such a way that they are scalable to enable potential high transactional volumes.

6.2. Relevance

When creating a *Harmonisation Domain*, *Proxies* in different *Domains* should be able to identify, authenticate and authorise one another to facilitate trusted, cross *Domain Data Sharing*. This will be part of the future creation of the *Trust Framework*.
To facilitate end-to-end cross-*Domain Interoperability*, *Identification*, *Authentication* and *Authorisation* from one *Domain* needs to be transportable to another *Domain* in a trustworthy manner. Language on *Identification*, *Authentication* and *Authorisation* should be created to enable this.

6.2.1. Identification

Actors must be able to establish the identity of actor(s) from other *Domain(s)* to determine the actor with whom a transaction is initiated. Currently, various Initiatives have different working implementations of *Identification* and *Authentication* mechanisms. Table 5 gives a non-exhaustive overview of the various Identification and Authentication solutions implemented by *Initiatives*.

Table 5: Overview of how identification and authentication are organised within initiatives

	SBR NEXUS	MedMij	SIVI	HDN	iSHARE
Type of initiative	**Domain specific**				**Generic**
Identifier	• *Natural person:* not applicable • *Legal person:* Chamber of Commerce number	• *Natural person:* BSN • *Legal person:* Organisation identification number (OIN)	• *Natural person:* Name, address, date of birth and client number* • *Legal person:* Chamber of Commerce number	• *Natural person:* not applicable • *Legal person:* Chamber of Commerce number	• *Natural person:* any as defined by the identity provider • *Legal person:* EORI number (Chamber of Commerce used for verification during onboarding)
Authentication methods	• *Natural person:* not applicable • *Legal person:* PKI Overheid certificate & eHerkenning	• *Natural person:* DigiD via "Toegangsverleningsservice" • *Legal person:* PKI Overheid certificate	• *Natural person:* e.g. IRMA, iDIN (maybe eHerkenning in future) • *Legal Person:* 2-Factor Authentication methods - following eHerkenning • *M2M:* ABZ certificaat*	• *Natural person:* not applicable • *Legal person:* HDN-specific certificate	• *Natural person:* depends on desired eIDAS LoA • *Legal person:* PKI Overheid certificate and eIDAS certificates. Possibility to add more certificate in the future
Requirements	• *Natural person:* not applicable • *Legal person:* eHerkenning niveau 2+	• *Natural person:* eIDAS High (DigiD sub or High) • *Legal person:* eIDAS High	• *Natural person:* Face-to-face • *Legal person:* eHerkenning • *Both:* (Trend towards) 2-Factor Authentication	• *Natural person:* not applicable • *Legal person:* copy ID, agreement with money-lender (moneylender has a "Wft-vergunning")	• *Natural person:* as specified by eIDAS for desired LoA • *Legal person:* as specified by eIDAS for desired LoA
Frameworks of identity assurance	• eHerkenning as a derivative of eIDAS	• eIDAS • DigiD	• eHerkenning as a derivative of eIDAS	• Not applicable	• eIDAS

Table 5 shows that the *Initiatives* use different identifiers. To enable cross-*Domain Data Sharing*, there must be a mutual understanding of identifiers between *Domains* such that *Data Service Transaction Agreements* can be made. If the *Domains* can understand each other's identities, a challenge remains in trusting the identities from another *Domain*. Therefore, a mechanism should be in place that allows the *Domains* to validate the authenticity of identities received from other *Domains* for different types of actors which could initiate a *Data Service Transaction*.

6.2.2. Authentication

Identities of actors must be authenticated to verify the validity of a claimed identity and protect against fraudulent use of identities. *Data Service Providers* can set requirements for the level of assurance of *Authentication* required from their *Data Service Consumers*. When those consumers reside in other *Domains*, the *Authentication* information (including LoA) must be communicated and mapped to the *Data Service Provider's* LoA definitions.

6.2.3. Authorisation

For *Data Service Providers* to be able to make proper *Authorisation* decisions regarding *Data Service Consumers* residing in another *Domain*, the information required for those decisions (attributes, roles, *Delegation* information and/or other information and decisions) must be communicated and mapped to the *Data Service Provider's* language and definitions. *Authorisation* should always originate from the *Entitled Party*.

6.3. Description

This chapter explains the need for a shared language and understanding on the topics of **Identification in 6.3.1**, **Authentication in 6.3.2**, and **Authorisation in 6.3.3**.

6.3.1. Identification

To come to a *Data Service Transaction Agreement* and share *Data*, sufficient *Trust* is required between all actors involved in the transaction. Knowledge and unambiguous understanding of the identity (and consequently the *Authorisations*) of all actors involved plays a crucial role in enabling this *Trust*. Furthermore, an understanding of identifiers for all objects and entities across *Domains* is required to be able to interpret the results of *Data Service Transactions*. Since different *Domains* make use of different types of identifiers, this unambiguous understanding is not trivial. There are several possibilities to facilitate an understanding of identifiers across domains, which are explored in this chapter:

- Using explicit identifiers,
- Mapping identifiers across *Domains*,
- Sharing transaction context.

Alternative solutions for dealing with identifiers across *Domains* includes the implementation of a single sign-on environment. However, this requires all *Domains* to make use of the same identity solution, which is not feasible for the *Trust Framework*. Another alternative is to use a decentralised identity solution, but these solutions are not yet sufficiently developed to consider at this point in time.

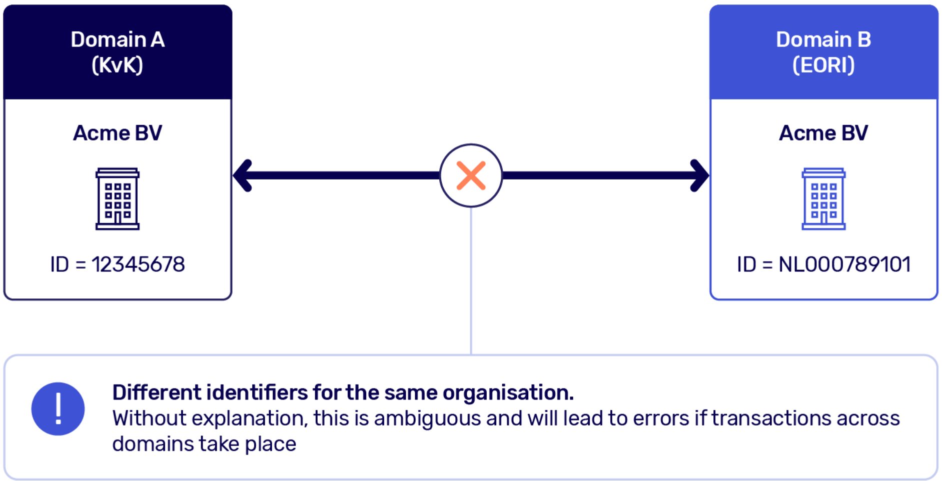
Using explicit identifiers

Ambiguity between identifiers across *Domains* can be solved by explicitly specifying the type of identifier used in all cross-*Domain* communication. Explicitly specifying the identifier used is possible through various mechanisms, such as the use of an attribute, *Metadata* description, or prefix (see **Box 4**). The exact method of specifying the identifier used should be detailed in the *Trust Framework*.

Box 4
Ambiguous identifiers

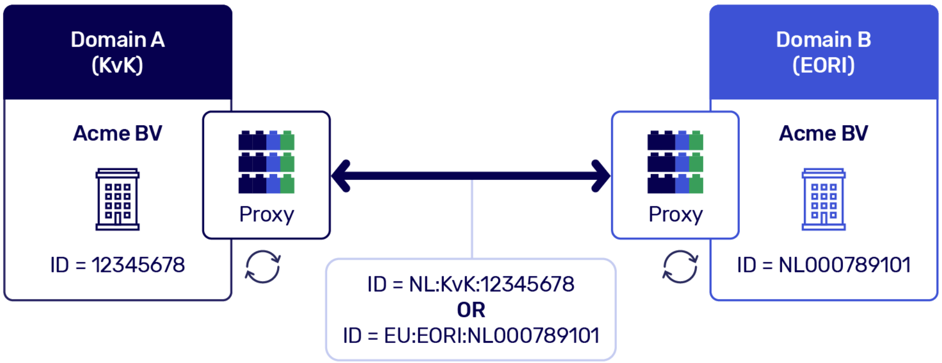
See Figure 13 for an example of a situation with ambiguous identifiers. Acme BV is a participant in both *Domain A* and *Domain B*. *Domain A* uses the KvK number (Chamber of Commerce number in the Netherlands) as identifier, *Domain B* uses the EORI number (Identification number for business in the European Union).

Figure 13: Ambiguity in identifiers should be resolved


Domain A
(KvK)
Acme BV
ID = 12345678
Domain B
(EORI)
Acme BV
ID = NL000789101
Different identifiers for the same organisation.
Without explanation, this is ambiguous and will lead to errors if transactions across domains take place

This ambiguity in used identifiers across *Domains* can be resolved by explicitly describing the identifier used. For example, this can be done using a prefix as shown in Figure 14.

Figure 14: Using prefixes for communication of IDs across domains solves ambiguity



Solving ambiguity in identifiers across *Domains* may facilitate an unambiguous mutual understanding of an identifier. However, it could be the case that the receiving *Domain* does not know how to interpret the explicit identifier. In this case, additional means are required to ensure an understanding of the identifier across *Domains*.

Mapping identifiers across domains

Identifiers communicated between *Domains* could be mapped to a known identifier within the receiving *Domain* to facilitate an understanding of the used identifiers. *Proxies* could play a role in performing the mapping of identifiers across *Domains*. For example, in Figure 14, the *Domain B Proxy* could receive the KvK number in a transaction from *Domain A*, and map this to an EORI number so it is understood in *Domain B* for further processing. If the mapping of identifiers in *Data* is not possible to facilitate an understanding of the used identifiers, the sharing of transaction context is required to ensure an understanding of the *Data* that is shared.

Sharing transaction context

Identifiers could be matched across *Domains* through the sharing of transaction context. This is a practical solution to match identifiers across *Domains* with sufficient assurance. Per *Data Service*, sufficient assurance should be achieved about the identity of the actor by providing context information while adhering to *Data* minimisation principles, and for personal *Data*, the General Data Protection Regulation (GDPR). The amount of context required to ensure sufficient assurance in the mapping of identities depends on the specific *Data Service* and therefore, should be determined on a case- by-case basis.

6.3.2. Authentication

Actors must be able to exchange identity information with each other and understand the level of assurance that is associated with the identity received. Depending on the type of actors involved, there are two different types of *Authentication*: Machine-to- machine *Authentication* and Human-to-machine *Authentication*. Furthermore, for any type of *Authentication* it may be necessary to transfer *Authentication* attributes across domains for specific use cases. These relevant topics are explored further in this chapter.

Assessing identity assurance

Actors must be able to understand the level of assurance (LoA) that is associated with an identity received from another *Domain* to determine whether the requested action can be performed. When communicating the LoA across *Domains*, the *Authentication* used to come to the LoA should be included in cross-*Domain* communication such that the receiving *Domain* has this information for their decision-making processes.

Furthermore, it is possible that an external authority can be used to verify and validate identity assurances.

For digital identity solutions, eIDAS has solved the *Interoperability* of Levels of Assurance (LoA) at an EU member state level, see **Box 3** for a detailed description. eIDAS allows EU member states with member state specific identity solutions with specific LoAs to be mapped to generic eIDAS LoAs to enable *Interoperability*.

The eIDAS framework with 3 LoAs (low, substantial, high) shall be used as a basis for interoperable LoAs in the *Trust Framework*. This is because the eIDAS framework is widely adopted already and has become the de facto standard for electronic *Identification* for eGovernment purposes in Europe.

Machine-to-machine authentication

An *Authentication* mechanism is required between machines (machine-to-machine, M2M) to autonomously authenticate each other's identity. This *Authentication* should take place for each transaction context and without a need for human interaction.

An example of machine-to-machine *Authentication* is in the usage of an IoT device service where the device must authenticate to the service servers. In the Trust Framework, machine-to-machine *Authentication* occurs when *Proxies* communicate with each other and must authenticate themselves.

To facilitate *Interoperability*, the Trust *Framework* should define a machine-to-machine Authentication method that all proxies can make use of. eIDAS Qualified Trust Services are anchored in EU law and widely used in Europe. Specifically, the Qualified Website *Authentication* Certificates (QWAC) and Qualified Seal are relevant to facilitate machine-to-machine *Authentication* methods. These eIDAS Qualified Trust Services could be used as a basis in the *Trust* Framework.

A Qualified Website Authentication Certificate is a digital certificate which ensures the authenticity and *Data* integrity of a connection and can be used to authenticate *Proxies* before a connection is made. A Qualified Seal is a signature which ensures the sender's non-repudiation and integrity of messages.

To ensure a correct usage of Qualified Trust Services, cybersecurity experts will be asked to provide insights and design principles so that these are implemented correctly for M2M *Authentication* within the *Trust Framework*.

Human-to-machine authentication

An *Authentication* mechanism (human-to-machine, H2M) is in place between natural acting persons and the *Domain* that they are a part of. However, when transacting across *Domains*, it may be necessary for natural acting persons to authenticate themselves in *Domains* other than the one they are located in. *Domains* should facilitate a customer journey to enable this. Natural acting persons in various *Domains* should therefore be able to be redirected to perform *Authentication* in other *Domains* within a single customer journey.

An example of human-to-machine *Authentication* is a log-in to an online service by using a Facebook account (via OAuth). In the *Trust Framework*, human-to-machine *Authentication* occurs when a natural acting person logs in to a service to perform an action. The person logs in a single time, requiring interaction, to set up a session during which they can perform the action, possibly consisting of multiple interactions, without having to authenticate themselves at every step.

Authentication is always performed within a specific *Domain* and therefore, there is no need to organise human-to-machine *Authentication* across *Domains*. However, it will occur that a natural acting person (human) must authenticate themselves in a *Domain* they are not present in, while initiating the transaction. To facilitate the transaction, the natural acting person needs to be redirected to the authorising *Domain* to authenticate. The Proxies should facilitate this redirect. To ensure a consistent user experience, User Experience (UX) Requirements should be defined for human-to-machine *Authentication*. The requirements for this redirect functionality by Proxies and the UX- requirements for *Identification* and *Authentication* (and *Authorisation*) should be included in the *Trust Framework*.

Forwarding authentication to another domain

For both H2M and M2M *Authentication*, it may be required to transfer *Authentication* attributes across *Domains*. For example, this may be needed to prove actor roles within another *Domain*. This topic should be discussed before development of the *Trust Framework*.

6.3.3. Authorisation

Once the identity of the *Data Service Consumer* has been determined with a sufficient level of assurance, the *Data Service Provider* must determine what actions they allow the *Data Service Consumer* to perform. In other words, what *Authorisation* the *Data Service Consumer* has. This follows the Fair Principle "Accessible", see Box 13 for more information. To determine the *Authorisation* an actor has, a sequence of actions in a specific flow should be carried out by specific roles. Furthermore, the *Authorisation* that an actor has can be delegated to a third party. These relevant topics are explored further in this chapter.

Roles in authorisation

For the *Data Service Provider* to determine the *Authorisation* of an actor, several different functional roles are established, each with their own responsibilities. Table 6 provides an overview of these roles and responsibilities and Box 5 provides an illustration of an *Authorisation* flow.

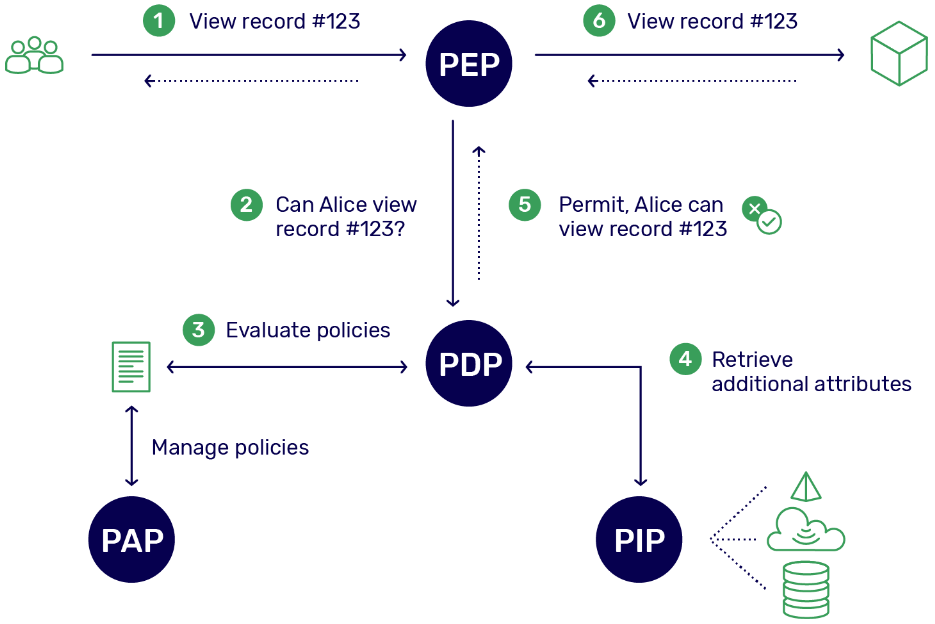
Table 6: Overview of authorisation roles and responsibilities

Roles	Responsibilities
Policy Administration Point (PAP)	The Policy Administration Point is where administrators, developers and business users can create and manage Authorisation Policies in order to be used by the PDP.
Policy Enforcement Point (PEP)	The Policy Enforcement Point is responsible for protecting the object by executing the access control decision. It intercepts API requests and forwards them on to the PDP.
Policy Decision Point (PDP)	The Policy Decision Point evaluates received Authorisation requests against Authorisation Policies using extra information if needed. All decisions reached are returned to the PEP.
Policy Information Point (PIP)	The Policy Information Point is any underlying information source of (meta) Data such as databases, user directories and Authentication details relevant for the Authorisation. If PEP provides insufficient Data to PDP, additional information can be retrieved via the PIP

Box 5
Authorisation roles

The following example *Authorisation* flow model can be applied to most *Authorisation* methods and provides a usable framework as basis for describing *Authorisation* concepts[4].

Figure 15: Example Authorisation flow as defined in the XACML standards


1 View record #123
PEP
6 View record #123
2 Can Alice view record #123?
5 Permit, Alice can view record #123
3 Evaluate policies
PDP
4 Retrieve additional attributes
Manage policies
PAP
PIP

1. A user sends a request which is intercepted by the Policy Enforcement Point (PEP).
2. The PEP converts the API request into an *Authorisation* request.
3. The PEP forwards the *Authorisation* request to the Policy Decision Point (PDP).
4. The PDP evaluates the *Authorisation* request against the loaded Policies. The Policies are managed by the Policy Administration Point (PAP). If needed, it also retrieves attribute values from underlying Policy Information Points (PIP).
5. The PDP reaches a decision (Permit / Deny / NotApplicable / Indeterminate) and returns it to the PEP.
6. The PEP enforces the decision and processes the request; in the case of a Permit, access is granted.

Note: This is a simplified model, and other *Authorisation* flows exist. These are further explored later in this chapter.

In practice, there is often not just a single implementation of several of the *Authorisation* roles. For example, there can be multiple Policy Decision Points which each take partial *Authorisation* decisions, these can be combined to collectively come to a final *Authorisation* decision. Furthermore, there are often multiple Policy Information Points,

[4] – https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

each providing different sets of information to the Policy Decision Points as needed. For cross-*Domain Authorisation* mechanisms, these roles (Policy Information Points and Policy Decision Points) can even be implemented in different *Domains*. Depending on the choice of possible distribution of the roles across *Domains*, *Interoperability* requirements are needed to facilitate the implementation of the roles.

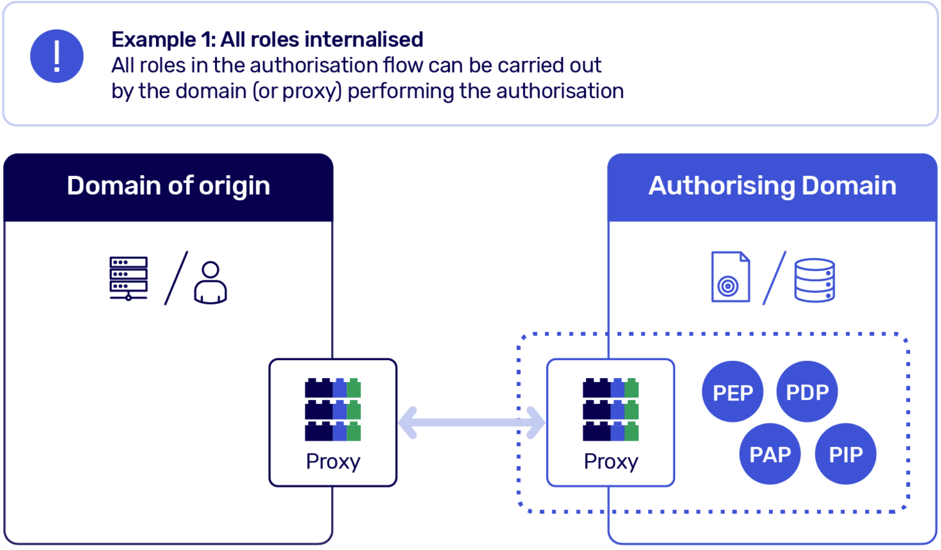
Requirements needed to facilitate the distribution of authorisation roles across domains

The roles required to facilitate *Authorisation* mechanisms could be distributed across different *Domains* to enable cross-*Domain* use cases. It is to be expected that the enforcement and administration of *Policies* will be located within the same *Domain*, which in turn makes it likely that the decision will also be made in the same *Domain*. In the context of *Authorisation*, it therefore makes sense to refer to *Domains* as administrative *Domains*, defined as the *Domain* where *Policies* are administrated and enforced.

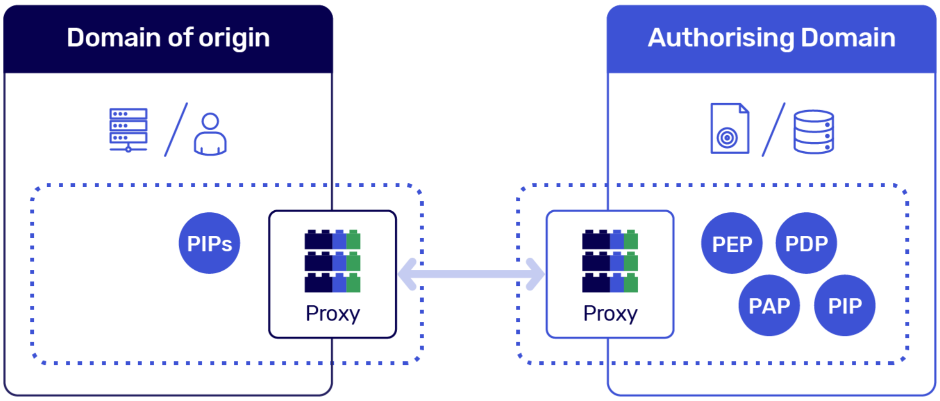
How an *Authorisation* decision is reached within a *Domain* can be the result of many (partial) decisions reached by different components within the *Domain*, However, the *Policy Decision Point* combines all partial decisions to a final decision. The details of how this is achieved is out of scope for the future *Trust Framework* for cross-*Domain Data Sharing* as it is the responsibility of a single *Domain*.

If use cases arise where it is necessary to out-source any of these *Authorisation* roles to other *Domains*, this will be further investigated to be included in the *Trust Framework* for cross-*Domain Data Sharing*. For now, this means the two most likely role distributions are as shown in Figure 16.

Figure 16: Most use cases can be captured in two different authorisation role distributions


!
Example 1: All roles internalised
All roles in the authorisation flow can be carried out
by the domain (or proxy) performing the authorisation
Domain of origin
Authorising Domain
Proxy
Proxy
PEP
PDP
PAP
PIP

Example 2: External PIP
In most cases, additional information is required on the subject/entity which stem from the domain of origin.
To achieve a PIP in the domain of origin for the authorising domain, interoperability requirements must be defined



When all the roles for *Authorisation* mechanisms can be realised within a *Domain* (Example 1 in Figure 16), there is no need for additional *Interoperability* requirements. However, in the case of Example 2 in Figure 16 where a role is in another *Domain*, or even outside of either *Domain*, Interoperability requirements are needed to enable this. Therefore, further investigation must be done into the following elements to be included in the *Trust Framework*:

- Language must be created to exchange *Authorisation Data* and attributes to transact,
- *Trust* is needed between *Domains* regarding the sharing of *Authorisation* attributes,
- Technical standards are needed to enable communication of attributes.

Flows for authorisation mechanisms

There are two most likely flows to determine the *Authorisation* an actor has needed to enable *Data Sharing*: the Pull and Push *Authorisation* sequence, as identified in RFC 29045. Both *Authorisation* sequences can be used for any type of *Data Service* model. Therefore, they can be considered independently from each other.

Pull authorisation sequence

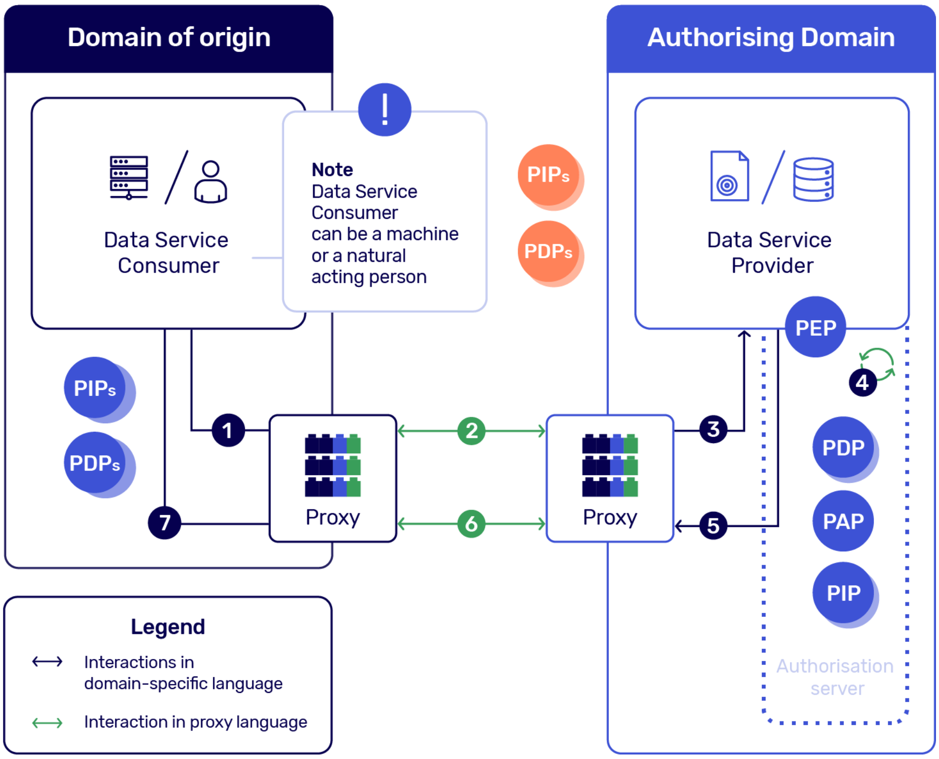
In a pull *Authorisation* sequence, the Policy Enforcement Point pulls the *Authorisation* decision from the Policy Decision Point in the authorising *Domain*. See Box 6 for more information on the pull *Authorisation* sequence.

[5] – https://tools.ietf.org/html/rfc2904

Box 6
Illustration of pull authorisation sequences in the proxy model

Figure 17 shows the *Proxy* interaction for a push *Authorisation* sequence.

Figure 17: Proxy interaction for a pull authorisation model



1. The *Data Service Consumer* sends a request for a *Data Service* to the *Domain* of Origin *Proxy* (including *Data Service Consumer* information for *Authorisation*)
2. The *Domain* of Origin *Proxy* translates the request and forwards it to the Authorising *Domain Proxy*
3. The Authorising *Domain Proxy* translates the request and forwards it to the Authorising *Domain*
4. Authorising *Domain* receives the request, processes it and the PDP takes the appropriate decision. The decision can be based on information and (sub) decisions received from outside of the Authorising *Domain*.
5. The *Data Service Provider* PEP provides access and *Data Service Provider* directly performs the action and sends back the result to the Authorising *Domain Proxy*
6. The Authorising *Domain Proxy* translates the results and forwards the result of the action to the *Domain* of Origin *Proxy*
7. The *Domain* of Origin *Proxy* translates the results and forwards the result of the action to the *Data Service Consumer*

Note: RFC 2904 additionally identifies the agent *Authorisation* sequence. From an *Interoperability* perspective, this can be considered the same as the pull sequence, as this only impacts how the decision is made in step 4.

An example of an *Authorisation* pull is when a Dutch citizen authorises a family member to perform their tax declaration using the NL mandate registry for citizens, DigiD Machtigen. The citizen must authorise the family member in advance at DigiD Machtigen, where this information is stored. The family member can then log in at the tax authority using their DigiD. The tax authority determines that they can perform the tax declaration based on an *Authorisation* pull from DigiD Machtigen.

Push authorisation sequence

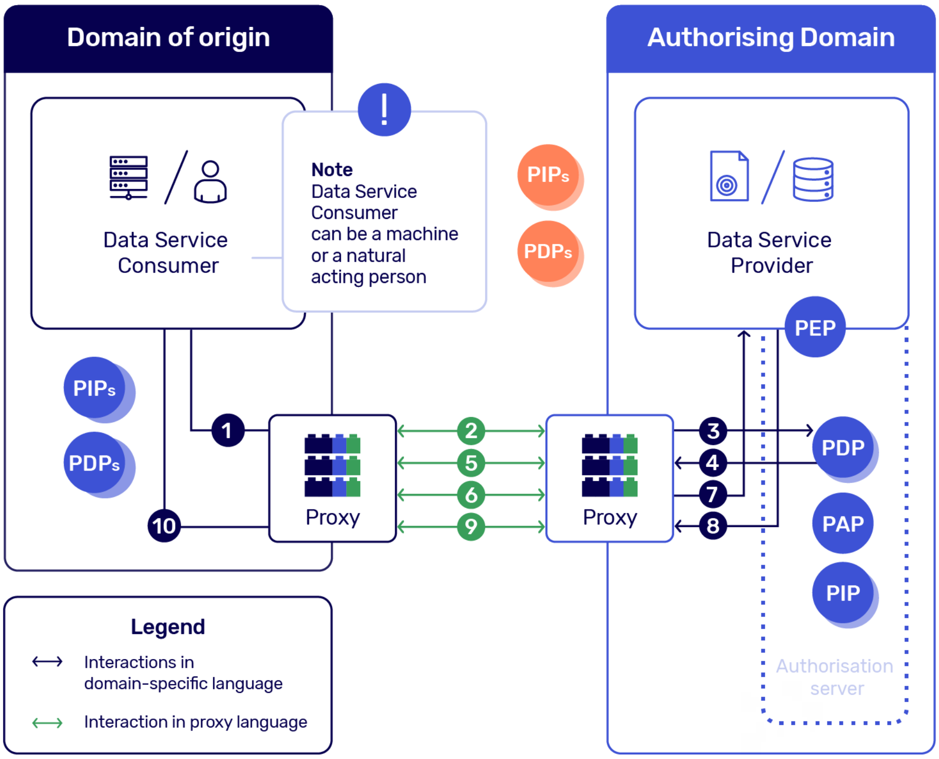
In a push *Authorisation* sequence, the Policy Enforcement Point gets pushed an *Authorisation* decision that the *Domain* of Origin has received from the Policy Decision Point. See Box 7 for more information on the push *Authorisation* sequence.

Box 7

Illustration of push authorisation sequences in the proxy model

Figure 18 shows the *Proxy* interaction for a push *Authorisation* sequence.

Figure 18: Proxy interaction for a push authorisation sequence



1. The *Data Service* Consumer sends an *Authorisation* request for a *Data Service* action to the *Domain* of Origin *Proxy* (including *Data Service Consumer* information for *Authorisation* and user redirect for consent, if necessary)
2. The *Domain* of Origin *Proxy* translates the *Authorisation* request and forwards it to the Authorising *Domain Proxy* (including information and redirect)
3. The Authorising *Domain Proxy* translates the *Authorisation* request and forwards it to the PDP in the Authorising *Domain* (including information and redirect)
4. PDP takes the appropriate decision and responds with the decision to the Authorising *Domain Proxy*. The decision can be based on information and (sub)decisions received from outside of the authorising *Domain*.
5. The Authorising *Domain Proxy* sends the decision to the *Domain* of Origin *Proxy*
6. The *Domain* of Origin *Proxy* sends a *Data Service* request (including decision) to the Authorising *Domain Proxy*
7. The Authorising *Domain Proxy* forwards the request to the *Data Services Provider* (including decision) where the PEP validates the decision and provides access
8. The *Data Service Provider* performs the action and sends the result to the Authorising *Domain Proxy*
9. The Authorising *Domain Proxy* translates the results and forwards the result to the *Domain* of Origin *Proxy*
10. The *Domain* of Origin *Proxy* translates the results and forwards the result of the action to the *Data Service Consumer*

An example of an *Authorisation* push is the OAuth 2.0 protocol in which users are redirected to provide consent for requests to access. This results in a long-term access token which can be used for the *Data Service Transactions*. The *Data Service* request includes the token and therefore, the Authorisation is pushed. These mechanisms are common to IoT setups and can be found in access control for home smart meters for electricity. The energy provider receives access to the home smart meter, based on a one-time consent of the user, on which the network operator (the owner of the metering infrastructure) issues an access token that can be used for all future requests for *Data*.

Delegated authority

Delegation is the provision of explicit *Authorisation* (to perform a specific action) to a third party. Several characteristics of the *Delegation* of authority include the action for which rights are given, who the authority is delegated to, and how long the rights remain. There are several different cases where *Delegation* of authority is required, such as:

- Companies cannot perform actions themselves and a service/employee must perform this on their behalf.
  - Natural persons, on behalf of companies, interact with other companies, such as non-standardised interactions using a web browser.
  - Machines, on behalf of companies, interact with other companies, such as PKI Overheid[6] (this is implicit *Delegation* of the machine, allowing machines to act for the company).
- Companies may delegate rights to other companies so that the other company can perform actions on their behalf in another *Domain*.
- Natural persons may give consent to another natural person to perform an action on their behalf, such as a colleague performing an action for you.

Therefore, *Delegation* of authority must be specified within the *Trust Framework*. The time frame at which the *Delegation* can be typically performed can be split into pre- configured and ad-hoc *Delegation*:

1. **Pre-configured delegation**
   - Pre-configured *Delegation* occurs well before the *Data Service* action takes place and is usually long lasting.
   - Examples of pre-configured *Delegation* can be seen in some iSHARE use cases, where *Delegation Policies* can be managed/stored in *Authorisation* registries which can be consulted at any time during *Data* requests to provide Authorisation. Another example is in the "Sharing e-CMR *Data* with insurers" use case, in which an insurer can be mandated by a shipper to retrieve *Data* from the e-CMR on their behalf.

2. **Ad-hoc delegation**
   - Ad-hoc *Delegation* occurs as the *Data Service* action is being performed and lasts for that single context.
   - An example of ad-hoc *Delegation* can be seen in the "Green Loans" use case in which mortgages can be provided based on energy usage *Data*. The mortgage intermediary can be granted access to the energy usage of a consumer to prepare a quotation for a mortgage.

[6] – http://www.pkioverheid.nl/

Communication required to validate pre-configured delegation

In pre-configured *Delegation*, the delegator gives consent for the delegatee in a single *Domain*. The delegatee can be given consent for generic rights, or rights to perform a specific action. The delegator does not know if the delegatee made use of the delegated rights and when or how they were used. Once the *Delegation* is performed, this must be stored within the *Domain* where this occurred and the delegatee is free to perform the action they were given consent for.

The process of pre-configured *Delegation* all takes place within a single *Domain* and therefore, there is no need for Interoperability requirements regarding the act of *Delegation*. Furthermore, if pre-configured *Delegation* takes place within the *Authorising Domain*, there is no need for additional Interoperability requirements as there is no need to communicate *Authorisation Data* across *Domains*.

If pre-configured *Delegation* takes place within the *Domain* of Origin, this must be communicated to the authorising *Domain* during a *Data Service Transaction*. The *Trust Framework* must facilitate a method to communicate this *Delegation* across *Domains*. Furthermore, a method for the *Authorising Domain* should be defined to validate the *Delegation* performed.

User experience requirements facilitate ad-hoc delegation

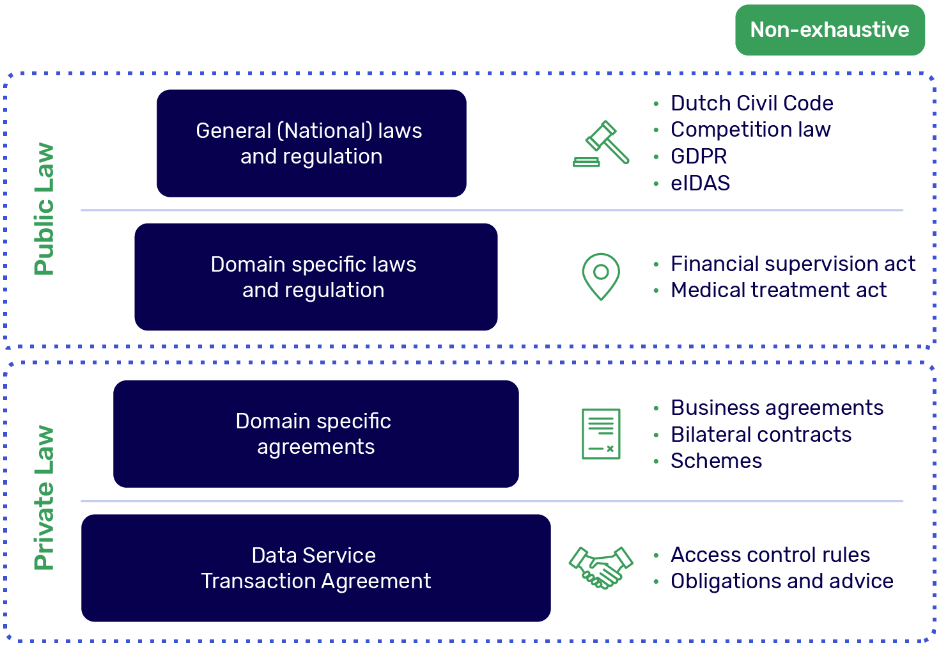
In ad-hoc *Delegation*, the delegatee is given specific rights to perform a *Data Service* action only during the transaction. The delegator knows that the delegatee made use of the delegated rights during only that transaction context. In this case, the *Authorisation* mechanism must take place within the *Authorising Domain*. To facilitate this, *Proxies* should include UX requirements for human-to-machine interaction to facilitate an actor delegating consent across *Domains*.

07.
Legal Context

7.1. Introduction

The first (and foundational) level of legal rules applicable to *Participants* of the ecosystem is existing general law, see Figure 19. This consists of the rules enacted as statutes by legislatures, adopted as regulations by government agencies, or determined by judicial decision. General law includes contract law, privacy law, antitrust law, and *Domain* specific laws such as the financial supervision act or medical treatment act, for example. These laws are public (i.e., written by governments), and applies to all *Data Sharing Domains* and transactions. Building on this foundation, private laws are defined that are voluntarily agreed upon by the *Participants* of that ecosystem. This includes *Domain* specific agreements such as schemes or contracts, and specifically the *Data Service Terms and Conditions* in a *Data Service Transaction Agreement* (see Chapter 5). The future overarching *Trust Framework* will become a scheme with legally binding agreements for all *Data Sharing Initiatives* and their *Participants* which choose to take part.

Figure 19: Hierarchy of rules, laws and regulations that must be considered for data sharing



7.2 Relevance

In general, agreements facilitate *Trust* between organisations as a prerequisite for most actions between them, including *Data Sharing*. When actors come to an agreement to be able to share *Data*, they form a *Domain*. These *Domain* specific agreements facilitate *Trust* by creating clarity about the legally binding rules under which *Data Sharing* takes place. As indicated in Figure 19, these *Domain* specific agreements are a further specification of what is allowed in addition to applicable rules, laws, and regulation. All *Data Sharing* transactions should be founded on a lawful basis. Depending on the *Data Service* and the actors involved, this could include a wide variety of possible bases. To enable cross-*Domain* agreements, a solution to facilitate cross-*Domain* agreements must be included in the *Trust Framework*.

7.3. Description

7.3.1. Legal status of the Trust Framework

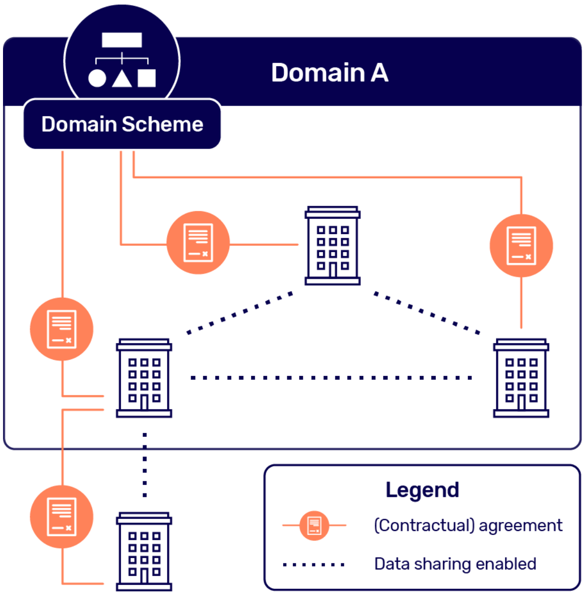
To achieve seamless transactions across *Domains* between actors that do not know each other, *Trust* is required between the involved actors. This can be facilitated through the *Trust Framework*. Actors that voluntary decide to join the *Trust Framework* are obligated by contract to follow the rules defined in the *Trust Framework*. The enrolment of actors into the *Trust Framework* as *Participants* is explored in Chapter 10.3.4 Enrolment. Because the future *Trust Framework* will be legally binding for all *Participants*, it allows *Participants* who do not know each other to *Trust* all other *Participants* adhere to the *Trust Framework* agreements. When sharing *Data*, the *Data Service Transaction Agreement* will explicitly refer to the *Trust Framework*, making it legally binding for the transaction. Thereby, the *Trust Framework* is meant to serve as the overarching rules for *Participants* entering an agreement for *Data Sharing*.

To ensure that sufficient multilateral *Trust* is provided, it is likely that *Participants* who wish to share data within the *Trust Framework* will require a certification to validate adherence to the *Trust Framework* and its agreements. The exact details and the legal implication of a required certification will be detailed when developing the *Trust Framework* for cross-*Domain Data Sharing*.

7.3.2. Contracts

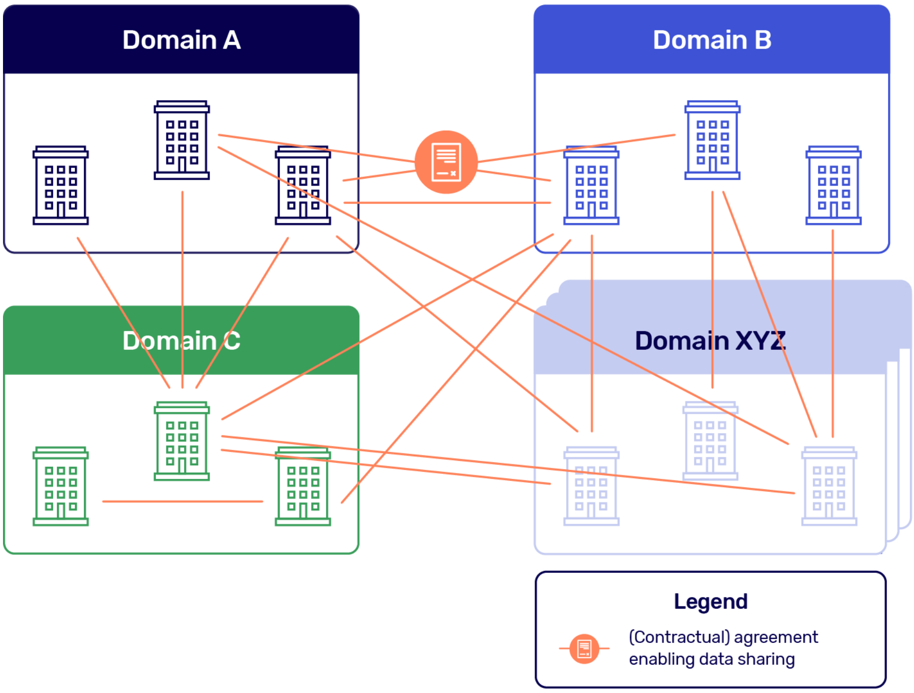
As a pre-requisite for sharing *Data*, any pair of organisations may have set up bilateral agreements and have implemented specific technology to enable *Data Sharing* between them. In case of a dispute between actors, the contracts provide the legal basis to which all parties involved in the *Data Service* should have adhered. These bilateral contracts need to be set up and maintained for all organisations to allow for *Data Sharing* between them. In a future where an increasing number of organisations is expected to share *Data*, the multitude of needed bilateral contracts is not efficient. Within some *Domains*, this has been resolved through the creation of a *Domain Scheme* to facilitate *Data Sharing* between organisations within the *Domain*, see Figure 20 *Domain Participants* have one contract with the *Domain Scheme* to enable *Data Sharing* with all other *Domain Participants*. This *Domain Scheme* is often managed collaboratively by actors in the *Domain*.

Figure 20: Some domains have implemented domain schemes to enable data sharing within the domain



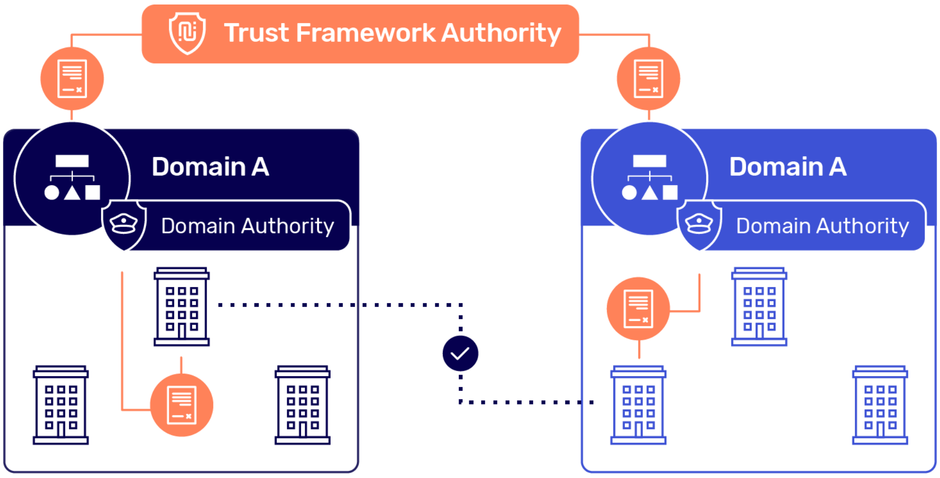
Domain Schemes facilitate multilateral *Trust* through contractual agreements to enable bilateral *Data Sharing* between *Domain Participants*. *Scheme* agreements lower barriers for *Data Sharing* by defining technical standards and legal agreements, including *Domain* specific laws and regulation. Beside these *Domain Scheme* agreements, organisations are free to make additional bilateral agreements with organisations outside of the *Domain* to enable cross-*Domain Data Sharing*. Where *Domain Schemes* have solved this need for bilateral agreements within a *Domain*, bilateral agreements remain relevant for cross-*Domain Data Sharing*, see Figure 21.

Figure 21: Closing bilateral contracts with every single organisation in cross-domain data sharing is not scalable



As a multitude of bilateral agreements between organisations from a multitude of *Domains* is not scalable, the *Trust Framework* should facilitate a scalable solution to legally bind all organisations across *Domains*. A solution to enable scalability is possible through multilateral agreements, which can be achieved via a chain of bilateral contracts as shown in Figure 22.

Figure 22: Enabling multilateral agreements via a chain of bilateral agreements



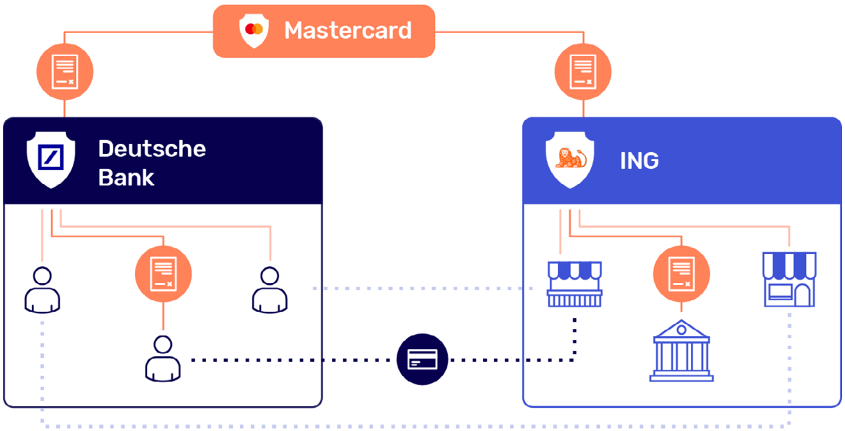
When each *Domain* scheme has a single bilateral contract with the overarching *Trust Framework Authority* and this bilateral contract enables a third-party effect, a chain of contracts is created which legally binds all organisations across all *Domains*. This is a scalable solution without laying the burden of multiple contracts on organisations. The exact contents of a contract required for *Data Sharing* will be detailed in the *Trust Framework* for cross-*Domain Data Sharing*. Once the required contracts are in place, organisations are free to share data. The legal basis of a transaction should be captured in the *Data Service Transaction Agreement*, see 14 Metadata.

An example of where this solution has a proven implementation can be seen in Box 8. As all organisations are connected across domains via the chain of multilateral contracts, there is no need for bilateral contracts between organisations in other *Domains*, however organisations are free to create bespoke agreements on top of the scheme agreements.

Box 8
A chain of bilateral contracts in the Mastercard ecosystem

Within the Mastercard ecosystem, a chain of bilateral contracts binds all actors to enable payments between actors, see Figure 23.

Figure 23: Example of a chain of contracts in the Mastercard ecosystem



- Deutsche Bank has a contract with Mastercard to enable them to issue Mastercard branded credit cards
- Deutsche bank issues Mastercard branded credit cards to their customers, who all have a contract with Deutsche Bank
- ING has a contract with Mastercard to enable them to facilitate accepting Mastercard payments at their merchants
- ING functions as an acquiring bank for their merchants, who all have a contract with ING
- Payments are facilitated between all Deutsche Bank customers and ING merchants

The *Trust Framework Authority* Is a role which is introduced to manage the contracts and ensure adherence to them. This includes the function of a monitoring body, which verifies that *Domain Schemes* adhere to the *Trust Framework* contract, and the function of an enforcement body which acts when contracts are violated. *Domain Authorities* are needed to aggregate the chain of contracts to connect all organisations in each *Domain*. Additionally, the *Domain Authority* functions as monitoring and enforcement body within the *Domain* (concerning the *Domain* specific agreements).

7.3.3. Lawful basis for sharing data

For all *Data Sharing* transactions, the *Data* that is shared can be personal or non-personal *Data*. For each of these types of data, different lawful bases exist that can apply for the processing of the *Data* and therefore, the sharing of the *Data*.

For the sharing of personal *Data* concerning people in the European Union, the General Data Protection Regulation (GDPR) defines 6 possible lawful bases, see Box 9.

Box 9
The General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is EU-legislation that contains provisions and requirements for the *Data* protection and privacy of natural persons in the EU. It imposes obligations onto any organisation which targets or collects *Data* related to people in the EU. Personal *Data* may only be processed if the *Data* controller (actor responsible for determining the purpose of *Data* processing) and *Data* processor (acting on behalf of the controller) have a lawful basis for processing the *Data*. The GDPR defines 6 lawful bases for the sharing of personal *Data*.

1. **Consent:** The individual has given clear consent for you to process their personal *Data* for a specific purpose.
2. **Contract:** *Data* processing is necessary for a contract you have with the individual, or because they have asked you to take specific steps before entering into a contract.
3. **Legal obligation:** *Data* processing is necessary for you to comply with the law (not including contractual obligations).
4. **Vital interests:** *Data* processing is necessary to protect someone's life.
5. **Public task:** *Data* processing is necessary for you to perform a task in the public interest or for official functions, and the task/function has a clear basis in law.
6. **Legitimate interests:** *Data* processing is necessary for a legitimate interest or the legitimate interests of a third party.

The GDPR does not apply to the processing (and sharing) of non-personal *Data*. In the future, the EU Data Governance Act[7] may define requirements for sharing non-personal *Data*. The Data Governance Act is still work in progress (expected Q2 2021) and should be considered when the *Trust Framework* is being developed.

For non-personal *Data*, it is up to the *Data Service Provider* to determine the lawful basis on which their *Data Service* will be based. For the sharing of business *Data*, two possible lawful bases are identified:

1. **Consent:** The organisation, or a natural acting person with the necessary rights on behalf of an organisation, has given clear consent for the processing of their non-personal *Data* for a specific purpose.
2. **Other lawful bases:** Any other legal agreements or ground can be used as basis for the processing of non-personal *Data* for a specific.

The *Trust Framework* should support all possible *Data Services* for personal *Data* and business *Data* and therefore, should be agnostic to the lawful basis used for *Sharing Data*. In the contracts that *Participants* will have with the *Trust Framework Authority* it should be clearly stated that it is the responsibility of each participant to establish an appropriate lawful basis for its *Data* processing activities. Furthermore, for transparency reasons the lawful basis on which a *Data* service is based should be part of the T*erms and Conditions* of that *Data Service* and should be included in the *Data Service Transaction Agreement*.

[7] – https://ec.europa.eu/digital-single-market/en/news/data-governance-act

7.3.4. Legal topics

Several legal topics have been identified which are relevant and should be covered in the *Trust Framework* to lower barriers for cross-*Domain Data Sharing*. These are categorised according to the separation of powers as shown in Table 7. The separation of powers is a governance structure which prevents the concentration of power at a single entity such that no single entity can abuse its power. A rule making power will establish and maintain the rules in the *Trust Framework* for its *Participants* to adhere to, the executive power will administer, monitor, and enforce the established rules, and the judicial power will settle *Disputes*. In practice, it is not always practical to fully separate the three powers, and the division of these roles may change with the maturity and scale of the scheme. For example, in iSHARE various executive responsibilities have shifted from the *Scheme* Owner role to the *Scheme* Administrator. The *Trust Framework* will need sufficient checks and balances so that it is clear to *Participants* that no single entity has disproportionate power it can abuse.

Table 7: Legal topics categorised by the separation of power

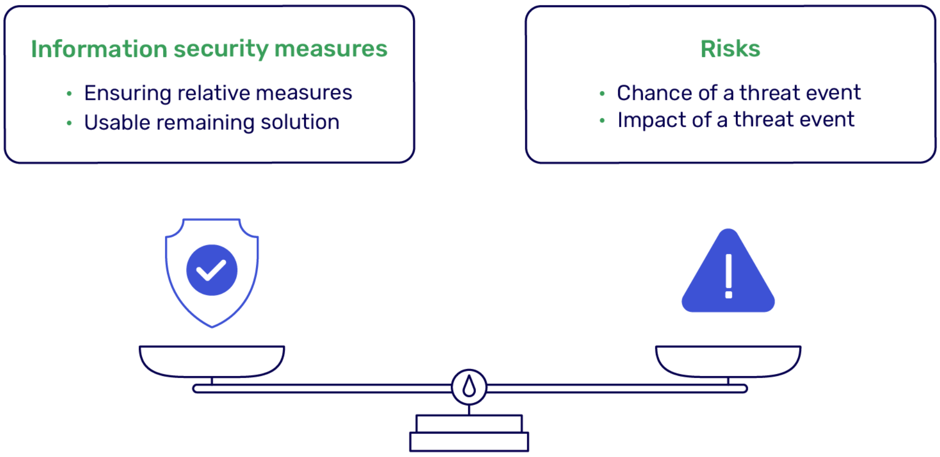
Rule Making power	Executive power	Judicial power
Relevant legislation	Supervising entities	Liabilities
Privacy	Acceptance criteria & KYC	Sanctions
Competition law	Governance structure oversight	Complaint & dispute management
Participant-scheme	Certification framework	Incident handling processes
Bilateral relations	Certification process	Escalation & decision making
Terms & Conditions	Change procedures & process	…
Governance Composition	Version management	
…	Monitoring and reporting	
	…	

08.
Information Security

8.1. Introduction

When sharing *Data*, organisations expose themselves to information security risks that need to be managed. To determine the risk of a *Data Service*, a risk analysis should be performed. This provides insights into the potential risks to which an organisation exposes itself with the *Data Service*. *Information Security* management involves the implementation of sufficient measures to balance the risks of possible threat events. Once the potential risks during *Data Sharing* have been determined, *Information Security* measures can be implemented to mitigate risks in line with the risk appetite of the organisation. A widely used model to discuss Information *Security* is the CIA triad, see **Box 10** for an overview. Examples of threat events include unauthorised access to *Data* or deletion of *Data*. Examples of Information *Security* measures include the encryption of communication or contracts defining restrictions. A balance between the risks and implemented measures must be found to reduce risks to an acceptable level while still providing a usable solution, see **Figure 24**.

Figure 24: Information security management is the balance between security risks and measures



Information security measures

- Ensuring relative measures
- Usable remaining solution

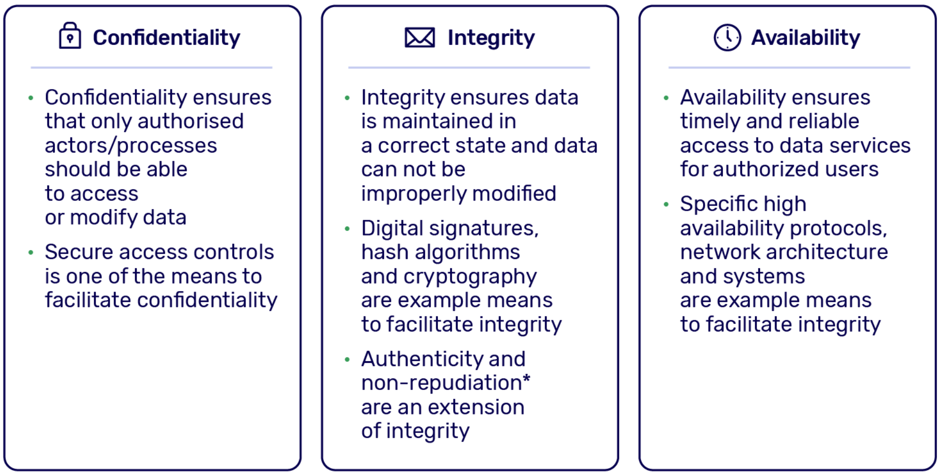
Risks

- Chance of a threat event
- Impact of a threat event

Box 10
The CIA Triad

The CIA (Confidentiality, Integrity and Availability) triad of *Information Security* is an *Information Security* model which can be used as a starting point for discussing Information *Security* topics and categorising security measures. Figure 25 gives an overview of the concepts within the CIA triad.

Figure 25: The CIA Triad: Confidentiality, Integrity, Availability



Confidentiality

- Confidentiality ensures that only authorised actors/processes should be able to access or modify data
- Secure access controls is one of the means to facilitate confidentiality

Integrity

- Integrity ensures data is maintained in a correct state and data can not be improperly modified
- Digital signatures, hash algorithms and cryptography are example means to facilitate integrity
- Authenticity and non-repudiation* are an extension of integrity

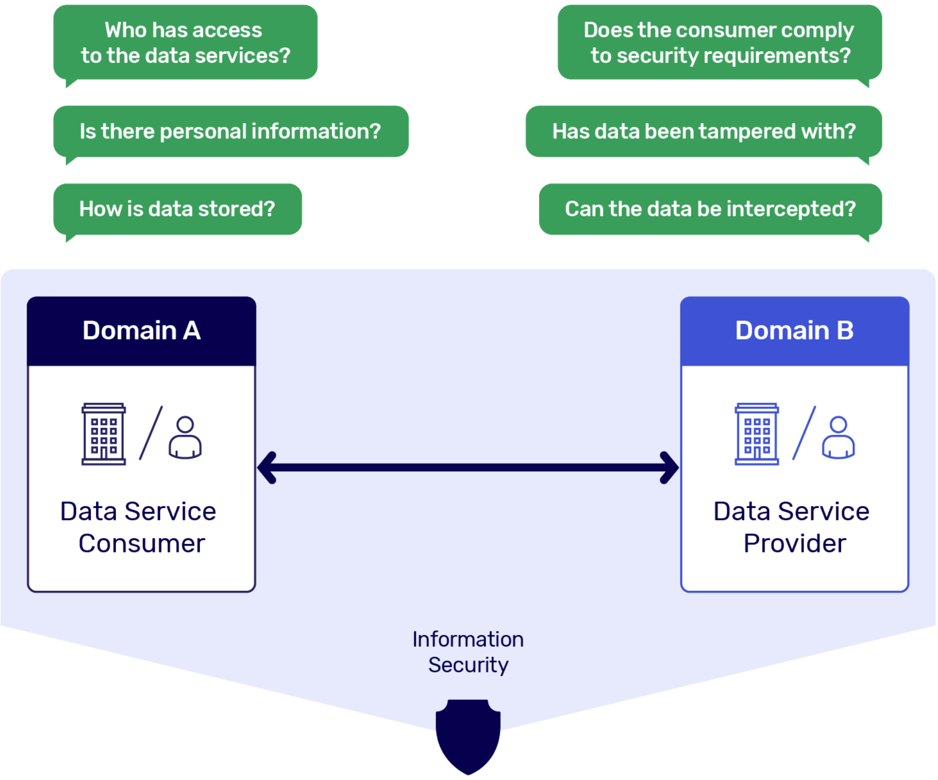
Availability

- Availability ensures timely and reliable access to data services for authorized users
- Specific high availability protocols, network architecture and systems are example means to facilitate integrity

8.2. Relevance

In the context of cross-*Domain Data Sharing*, *Information Security* concerns the risks and measures related to the end-to-end *Data Sharing* transaction between actors from different *Domains*. This includes not only what happens when sharing *Data*, but also what happens to the *Data* itself. See Figure 26 for a non-exhaustive view on topics related to *Data Sharing* across *Domains*.

Figure 26: Examples of questions related to Information security in cross-domain data sharing


Who has access to the data services?
Is there personal information?
How is data stored?
Does the consumer comply to security requirements?
Has data been tampered with?
Can the data be intercepted?
Domain A
Data Service Consumer
Domain B
Data Service Provider
Information Security

Therefore, *Information Security* includes measures implemented within the *Data Service Consumer Domain* (e.g. secure storage of *Data*) and the *Data Service Provider Domain* (e.g. validating implemented security measures), as well as the *Harmonisation Domain* (e.g. secure exchange infrastructure). *Information Security* is a basic prerequisite to enable *Trust*, as it contributes to reducing risks to sufficiently low levels required to share *Data*.

8.3. Description

To facilitate *Information Security* across domains, *Domain* A and B need to be able to communicate with each other on applicable Information *Security* concepts via a shared language and understanding. A shared language and understanding of Information *Security* is needed to allow for unambiguous communication on *Information Security* concepts, requirements and measures.

The main challenge for creating a shared language on *Information Security* is the large amount of variance in applicable security concepts between *Domains*. The *Information Security* risks, and risk appetite of *Domains* differ from one another, which in turn leads to a difference in implemented *Information Security* measures. In many cases these various measures aim to mitigate similar risks, and therefore achieve similar goals, but go about it in different ways. This hinders the understanding of implemented measures and levels of risks across *Domains*. To make communication about Information *Security* measures manageable and to lower barriers to interoperability, the clustering of security measures is a practical solution.

8.3.1. Information security clusters and levels

A security cluster can be defined as a set of *Information Security* measures which pursue the same objective. Clusters make it easier to communicate and understand the implemented security measures across *Domains*.

Depending on the use case, transactions may have higher or lower risk. For example, low-risk transactions, such as the sharing of personal preferences like shoe size, do not require the use of high amounts of *Information Security*. On the other hand, high-risk transactions, such as the sharing of personal medical *Data*, require a very high amount of *Information Security*. The *Trust Framework* should facilitate all types of use cases and therefore enable both high-risk and low-risk transactions. To reduce barriers for use of the *Trust Framework*, low-risk transactions should be facilitated though use of low *Information Security* levels and not be mandated to use high levels of *Information Security* measures. At the same time, the *Trust Framework* should allow high security where needed to enable high-risk transactions. Security levels are a practical solution to facilitate this as these can be defined such that the security level is based on the security cluster requirements. See Box 11 for example of security levels used in *Data Sharing* by the *International Data Spaces Association*, which could be used as a reference for the *Trust Framework*.

Box 11

Security levels within IDSA - DIN SPEC 27070

IDSA reference architecture defines requirements for a security gateway for *Data Sharing* in the DIN SPEC 27070 standard, which is based on the ISA/IEC 62443 international series of standards. The ISA/IEC 62443[8] standards originate from the industrial process sector, but have since been used in various contexts, including *Data Sharing*. ISA/IEC 62443 defines security levels which are mapped to different types of attacks based on the means, resources, skills, and motivation of attackers. See Figure 27 for an overview of the defined security levels.

Figure 27: Example of security levels in the ISA/IEC 62443 specifications

Security Level	Definition	Means	Resources	Skills	Motivation
1	Protection against casual or coincidental violation	simple	low	generic	low
2	Protection against intentional violation using simple means with low recources, generic skills, and low motivation				
3	Protection against intentional violation using sophisticated means with moderate resources, IACS-specific skills, and moderate motivation	sophisticated	moderate	IACS-specific	moderate
4	Protection against intentional violation using sophisticated means with extended resources, IACS-specific skills, and high motivation	sophisticated	extended	IACS-specific	high

ISA/IEC 62443 goes on to define seven Foundational Requirements, see Table 8 for an overview. All aspects associated with meeting a desired security level are achieved through meeting the requirements associated with the seven Foundational Requirements.

Table 8: Indicative overview of overview of Overview of foundational requirements as defined by ISA/IEC 62443

Foundational Requirements	Examples
Identification and authentication control	Human user Identification and authentication, multifactor authentication, etc.
Use Control	Authorisation enforcement, Permission mapping to roles, etc.
System Integrity	Communication integrity, input validation, error handling, etc.
Data Confidentiality	Information confidentiality, use of cryptography, etc.
Restricted Data Flow	Network segmentation, application partitioning, etc.
Timely Response to Events	Audit log accessibility, continuous monitoring, etc.
Resource Availability	Resource management, control system backup, emergency power, etc.

[8] – https://gca.isa.org/blog/download-the-new-guide-to-the-isa/iec-62443-cybersecurity-standards

Security levels based on requirements of security clusters facilitate different types of transactions. Security levels allow clear communication of various security requirements and support various implementations of *Information Security* measures. Furthermore, security levels reduce impact on *Domain Participants* which may have different security implementations as implementations can be easier understood, reducing required analysis of implementations. Furthermore, *Participant* implementations do not need to be adjusted to conform to specific standards.

The security levels and foundational requirements as defined by the ISA/IEC 62443 standards (see **Box 11**) can be used as a basis to define security levels and clusters in the *Trust Framework*. However, it has been identified that these clusters consider only software components. Other factors such as the physical security or the operational process are not included in these standards. Therefore, the ISA/IEC 62443 standards should be enriched with additional international standards to obtain a complete view on possible security clusters. One additional source will be the ISO/IEC 2700x series of standards, which provides insights into additional topics from an *Information Security* management perspective. ISO/IEC 27002 contains 14 security control clauses that collectively contain 35 main security categories. These include physical and environments security, operational security, incident management and business continuity management, among others, which are topics that are not covered in the ISA/IEC 62443 standards.

In the *Trust Framework* for cross-*Domain Data Sharing*, the security levels and clusters will be determined and the ISA/IEC 62443 and ISO/IEC27001 standards will be used as reference material.

The security implementations of *Domains* should adhere to the minimum requirements defined by the *Trust Framework*. For a specific *Data Service*, the *Data Service Provider* can determine the security requirements based on the security levels and clusters available in the *Trust Framework* agreements. The *Data Service Consumer* should consider these requirements before deciding whether to make use of the service. Therefore, the *Data Service* security requirements should be included in the *Data Service* description before a *Data Service* can take place (see **9.3.1 Data service discovery**). If a burden of proof is required for the adherence to the security requirements, this should be captured in *Logging* (see **Chapter 10.3.3**). The mechanisms to achieve this will be described in the *Trust Framework*. Furthermore, for transparency reasons the security requirements on which a *Data Service* is based should be included in the *Data Service Transaction Agreement*.

8.3.2. Information security principles

Several security principles have been identified which can be applied to the *Data Sharing Canvas* and future *Trust Framework* for cross-*Domain Data Sharing* to guide all *Information Security* discussions and decisions.

1. **Use of existing standards and consideration of best practices.**

   This is a generic design principle for the *Data Sharing Canvas* but is especially important for the complex topic of Information Security as standards provide a solid foundation of managing security.

2. **Fit-for-purpose security levels**

   This principle means facilitating low-risk transactions to use low information security measures to reduce barriers for use but allowing high security where needed to enable high-risk transactions.

3. **Organisational and technical security measures go hand-in-hand**

   Information Security relies on technical and organisational measures which complement each other to enable a best solution to facilitate *Trust*.

4. **Enable trust through security and privacy by design**

   Security and privacy are not only defensives mechanisms, but also enables *Trust*. Therefore, *Information Security* must be rigorously included in the design of the *Trust Framework*.

09.
Data Service Exchange

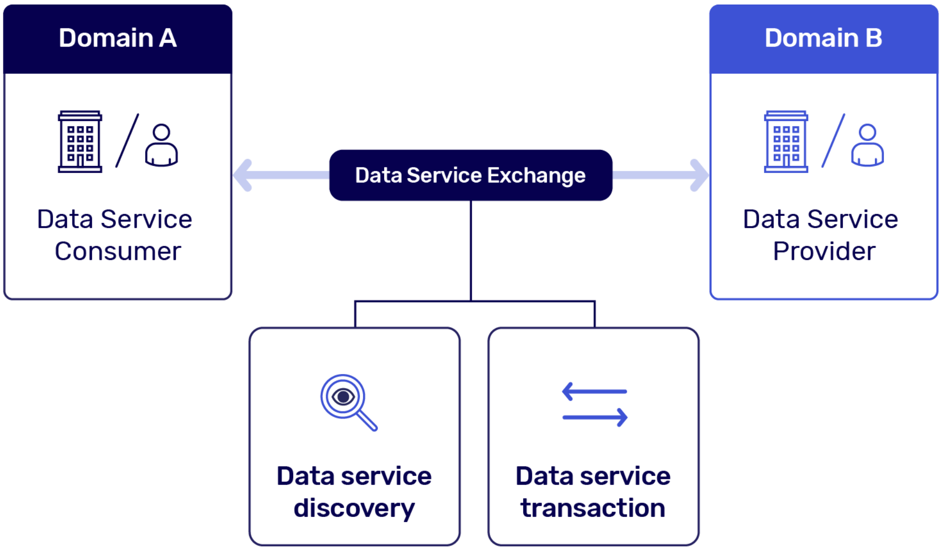
9.1. Introduction

To achieve interoperable *Data Sharing* across *Domains*, a technical communication standard (a so-called exchange protocol) should be defined in the *Trust Framework*. Therefore, the functional *Data Service* exchange requirements should be determined before standardisation and implementation decisions of an exchange protocol are made. This chapter explores some of the functional *Data Service* exchange requirements.

9.2. Relevance

The complete *Data Service* exchange can be split into two distinct steps: *Data Service Discovery*, and *Data Service Transaction*, as shown in Figure 28. These steps should be carried out sequentially and, where possible automatically, without human interaction. In order for a *Data Service Consumer* to perform a *Data Service Transaction* with a *Data Service Provider*, they must first know that the service exists, meets their needs and if so, where to find the service. A *Data Service Provider* must be discoverable to allow a *Data Service Consumer* to find the *Data Service Provider* and its service(s). Once the *Data Service Consumer* has discovered the *Data Service Provider*, they are able to perform a *Data Service Transaction* without the need for re-discovery for subsequent transactions.

Figure 28: Data service consumers must discover services before they can make use of them.



9.3. Description

9.3.1. Data service discovery

A *Data Service Discovery* mechanism should be facilitated in the *Trust Framework* and give answers to several different questions from the *Data Service Consumer* perspective, such as:

- What *Data Sharing Domains* are part of the *Trust Framework*?
- What *Data Service Providers* are available?
- What *Data Services* do the *Data Service Providers* offer?
- Do *Data Service Providers* have *Data* that is relevant for me?

A *Data Service Discovery* mechanism facilitates the *Fair Principle* "Findable" of *Data Services*, see **Box 13** for more details. The *Data Service Discovery* mechanism and should at least have the following characteristics:

- Allows services to connect without manual intervention,
- Allows *Data Service Consumers* to have access to all information needed to decide on whether to use the *Data Service*,
- Provides a clear communication from the *Data Service Provider* to the *Data Service Consumer* through a common language (*Metadata*).

A solution to enable *Data Service Discovery* is to maintain a *Service Registry* that contains service information for the purpose of discovery information. A *Service Registry* contains all the necessary information about all *Data Services* available and can be considered like a telephone book. All *Participants* are free to develop *Data Services* that adhere to the *Trust Framework* agreements and take the role of *Data Service Provider*. This is in line with the inclusive guiding principle, see **Chapter 3.3**. These *Data Services* can then be offered to *Participants* by making them discoverable. Since the *Trust Framework* network is dynamic by nature, as *Domains* and actors will change over time. Therefore, the *Service Registry* should be dynamic to facilitate this changing *Trust Framework* network.

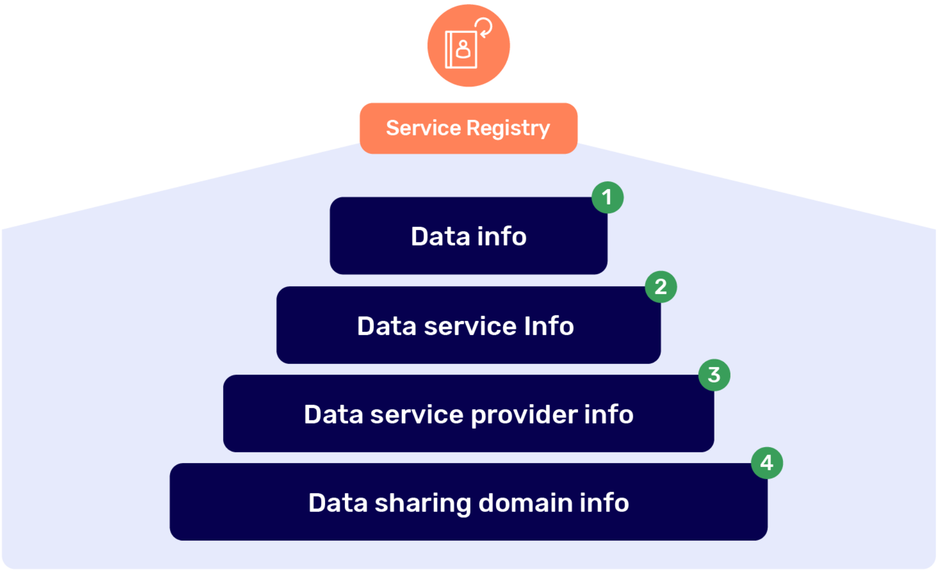
At minimum, the *Service Registry* should include information about the *Data Sharing Domains* which are participating in the *Trust Framework*. This allows *Data Service Consumers* to discover *Domains*, after which they still need to find answers to the rest of their questions elsewhere to be able to determine if they can and want to make use of the specific *Data Service*. However, this is not a practical solution, and does not allow services to connect without manual intervention. Therefore, additional information should be included in the *Service Registry* to simplify the process of discovering *Data Services* by the *Data Service Consumers*. The exact implementation choice of the *Service Registry* content will be made in designing the *Trust Framework*, but one can imagine the *Trust*

Framework Service Registry will contain information about relevant (see Figure 29):

- *Data* Information,
- *Data Service* Information,
- *Data Service Provider* Information,
- *Data Sharing Domain* information.

Initial discussions suggest that, practically, the *Service Registry* should contain at least *Data Sharing Domain* information and *Data Service Provider* information. For *Data Service Consumers*, this is the information needed for them to consider making use of the *Data Service*. If this information is included in the *Service Registry*, it relieves the *Data Service Consumer* of implementing complex discovery logic before making their consideration. In the development of the *Trust Framework* for cross-*Domain Data Sharing* the needed content of the *Service Registry* should be further investigated, and an implementation choice should be made.

Figure 29: The service registry can contain information about domains, service providers, services, and specific data



1 Information about data within a service, including:
quantitative data set descriptions, qualitative data set descriptions, etc.

2 Information about data services available by the provider, including:
service description, terms & conditions, security requirements, etc.

3 Information about data service providers within a domain, including:
service provider names, service provider capabilities, etc.

4 Information about domains, including:
names, locations, capabilities, implementation, method of communication, etc.

Data Service Providers require a mechanism to register their services in the *Service Registry*. The exact mechanism for the registration of *Data Services* will be detailed in the *Trust Framework* for cross-*Domain Data Sharing*. An assessment needs to be done whether to base this mechanism on a push or a pull model.

It may not be desirable for all *Data Service Providers* to provide the same level of information in the *Service Registry*. Furthermore, not all *Data Service Providers* may be able to or want to deliver all specified levels of information in the *Service Registry* as this may include sensitive *Data*. Therefore, an *Authorisation* mechanism could be considered for the *Service Registry* to facilitate that only authorised parties get access to specific discovery information. This will be further investigated in the development of the *Trust Framework* for cross-*Domain Data Sharing*. In the *Trust Framework* *Data Service Providers* should be able to register their services and be free to add information relevant to their services.

Based on industry standards several roles and functions have been identified that can facilitate *Service Discovery*. Two models are applicable for different perspectives in the *Trust Framework*. See **Appendix IV Data Service Discovery**, for more information. In 'Client' side discovery, the client is responsible for discovering services and performing transaction requests. For every request for discovery of a *Data Service*, the client will check a service registry to find relevant services. An alternative model is 'Server' side discovery in which the client makes a discovery request towards a discovery server. The server is responsible for discovering services and returns the discovery response to the client. An implementation choice based on a detailed analysis should be made for the type of implementation of the *Service Registry* and implementation mechanism. This analysis should include the assessment of the desired location and distribution of the *Service Registry*. This could be a single central implementation, or a decentralised distribution. Furthermore, possible actor(s) that could become responsible for the implementation and management of the *Service Registry* should be included in this analysis.

It is likely that the desired implementation of the *Data Service Discovery* mechanism and the *Service Registry* will change over time given the maturity and development of the *Trust Framework*. A basic implementation is likely to initially be sufficient, and this implementation could be further developed to support additional services in the future. Furthermore, it is possible that an actor may take up the role of a service broker to offer *Data Service Discovery* as a service to *Participants*. In the development of the *Trust Framework* for cross-*Domain Data Sharing* these possible implementation options should be considered in when making implementation choices for *Data Service Discovery*.

9.3.2. Data service transaction

Functional *Data Service* exchange requirements for the *Trust Framework* must be determined based on the *Data* transfer characteristics of desired use cases. *Data* transfer characteristics influence the *Data Service* exchange, for example, transferring a small amount of *Data* can be realised through sending the data in APIs, whereas transferring a large amount of *Data* is not possible through APIs. For large amounts of *Data* an FTP server could be used for example. Given the goal of the future *Trust Framework* to support an ever-changing number and variety *Data Sharing* use cases, several identified *Data* transfer characteristics should be supported.

The following have been identified and will be considered in the further development of the *Trust Framework*:

- Sharing of time-dependent *Data*,
- One-time sharing of *Data*,
- Continuous sharing of *Data*,
- Sharing large amounts of *Data*,
- Sharing small amounts of *Data*,
- Sharing of live *Data*,
- Sharing of static *Data*.

10.
Operational Agreements

10.1. Introduction

Within the *Trust Framework* operational agreements help to facilitate the trust between actors required for sharing *Data*. Operational agreements include topics such as:

- Service Level Agreements (SLAs),
- End user support,
- *Dispute Management*,
- *Logging*,
- Enrolment.

The *Data Sharing Coalition* concluded that SLAs and end user support do not need to be harmonised between *Domains* as these topics are part of domain-specific *Data Service* implementations without a cross-*Domain* component. SLAs and end user support are relevant topics which should be covered in the *Data Service Agreement*. The topics of *Dispute Management*, *Logging* and enrolment have been identified to have a component in the *Harmonisation Domain*, which requires agreements in the *Trust Framework*.

10.2. Relevance

The topics of *Dispute Management*, *Logging* and enrolment contain cross-domain components which therefore should be harmonised in the future *Trust Framework*. *Dispute Management* involves actors from different *Domains* and therefore, the *Dispute Management* process should be harmonised to a certain level, to enable *Trust* between *Participants* in the *Trust Framework*. *Logging* (or audit trails) by *Participants* is required for reporting purposes and to enable accountability within the *Trust Framework*. Furthermore, an enrolment process should be clearly defined for (potential) *Participants*.

10.3. Description

10.3.1. Dispute management

A core component to create *Trust* is setting clear expectations and requirements in the complete *Data Sharing* process, and subsequent compliance to these requirements for all actors involved. This includes creating transparency in all phases of *Data Sharing*:

- before sharing *Data* through *Trust Framework* agreements,
- during *Data Sharing* through *Data Service Transaction Agreements*,
- after *Data Sharing* through *Dispute Management*.

A *Dispute* occurs when actors within the *Trust Framework* cannot settle disagreements between them and *Dispute Management* is the process for managing all reported *Disputes*. It is unlikely that many *Disputes* will arise, and therefore the expectation is that a *Dispute Management* process will not be widely used. However, having a latent *Dispute* management process defined increases clarity and contributes to *Trust* between actors. The *Trust Framework Dispute Management* process will not replace existing judiciary systems but complement it for *Disputes* between *Participants*.

A *Dispute* arises when actors have a disagreement in which the actors cannot settle this between themselves. Within the *Trust Framework*, the *Trust Framework* agreements form the overarching rules (see Chapter 7.3.1) which will be used for the processing of *Disputes* between actors. Three types of *Disputes* have been identified which may occur within the *Trust Framework*. Therefore, the processing and management of these *Disputes* should be supported in the *Trust Framework* for cross-*Domain Data Sharing*.

1. A *Data Service Provider Disputes* an action from the *Data Service Consumer*. For example: The *Data Service Consumer* distributes or sells *Data* obtained via a *Data Service* and this commercial use of the *Data* goes against the *Terms and Conditions* of the agreement.
2. A *Data Service Consumer Disputes* an action from the *Data Service Provider*. For example: The *Data* provided to the *Data Service Consumer* by the *Data Service Provider* is not according to the *Data Service Consumers* expectations (e.g. *Data* quality is below what was advertised in the service description).
3. A *Dispute* between actors/domains and the *Trust Framework* for cross-*Domain Data Sharing*. For example: The *Trust Framework Authority* believes a *Domain* no longer adheres to certain *Trust Framework* rules, and the *Domain* disagrees.

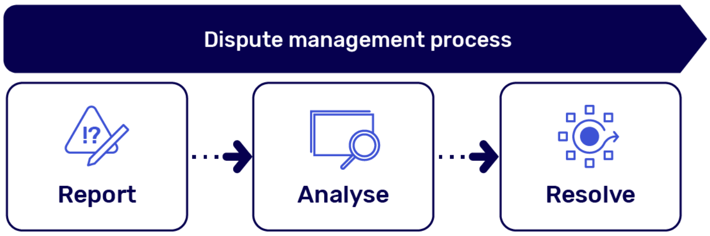
The settlement of *Disputes* should be facilitated by a neutral party to ensure that neither actors involved in a *Dispute* gains an unfair advantage. For the first two types of *Disputes*, the *Trust Framework Authority* can act as a neutral party to facilitate *Disputes* between *Participants*. When actors have a *Dispute* with the *Trust Framework Authority*, the *Trust Framework Authority* is no longer neutral, and should not facilitate the *Dispute* management process itself. Therefore, a separate entity is required to manage this type of *Disputes*.

Disputes within a single *Domain* should be processed in its respective *Domain*, only *Disputes* with a cross-*Domain* component should reported in the *Trust Framework*. For all cross-*Domain Disputes*, the *Trust Framework* agreements take precedence over the agreements within existing domains and *Data Service* agreements. See [Figure 19] for the hierarchy of laws and regulation that is also applicable to *Disputes*.

10.3.2. Dispute management process

The complete *Dispute Management* process can be split into three high-level steps as shown in Figure 30.

Figure 30: The three steps in managing a Dispute in the Trust Framework



Reporting the dispute

A *Dispute* is reported only when actors within the *Trust Framework* cannot settle disagreements between themselves. Actors involved in disagreements should attempt to resolve these between themselves via bilateral communication. For disagreements, the *Trust Framework* does not define a process for resolving them. A method for settling disagreements may be part of the *Data Service Terms and Conditions* of the specific *Data Service*.

The T*rust Framework* should define service level agreements for the process of solving disagreements to clearly define when a disagreement becomes a *Dispute*. If the actors cannot reach an agreement according to these service level agreements, they can report a *Dispute*. When a *Dispute* is reported to the *Trust Framework Authority*, a *Dispute Case Manager* should be assigned as a mediator to facilitate the *Dispute* management process for the actors involved in the *Dispute*. Depending on implementation choices made for *Dispute Management*, the *Dispute Case Manager* may be and external party, or may be available within the *Trust Framework*.

Analysing the dispute

In the next step of the *Dispute Management* process, a reported *Dispute* is managed by the *Dispute Case Manager* based on input provided by the actors. This is an iterative process which shall be managed by the *Dispute Case Manager*. Actors in the *Dispute* will provide input for the analysis and can provide evidence (e.g. audit trails from *Logging*, contracts, etc) and clarification on their position. The exact analysis process will probably not be defined in detail in the *Trust Framework* as this is dependent on the *Dispute*. Although the process is not fixed, the *Trust Framework* should define service level agreements for this process. This manages expectations of the actors involved and guides the process.

Resolving the dispute

The analysis leads to a decision on how to resolve the *Dispute*. The decision is made by the neutral *Trust Framework Authority*. The context of the *Dispute* influences the method of resolving *Disputes*. *Dispute* characteristics which impact the resolving of the *Dispute* include:

- Type of *Dispute*,
- Number of actors involved,
- Financial impact,
- Reputational impact.

The decision further includes the method to resolve the *Dispute*. Several possibilities for the resolving of *Disputes* have been identified. In the development of the *Trust Framework* for cross-*Domain Data Sharing*, the implementation of these possibilities will be further investigated, including the need for external proceedings for possible financial compensation. Possible methods to resolve *Disputes* could be (any combination of):

- The relevant party must update its implementation accordingly,
- (Financial) compensation,
- Warning, (temporary) suspension or removal of actor from the *Trust Framework*.

Depending on the *Dispute* and the decided method of resolving it, the result may be publicly published. A mechanism to facilitate this should be included in the *Trust Framework*.

If one of the actors involved in the *Dispute* does not agree with the *Dispute* resolution, they should be able to appeal the decision. The facilitation of an appeal process in the *Trust Framework* further adds towards building *Trust* required for *Data Sharing*. This appeals process will build on the existing judiciary systems and should be further described in the *Trust Framework*

The need for a detailed and operational appeal process will depend on the scale and maturity of the *Trust Framework* network. Therefore, when developing the *Trust Framework* possible solutions should be balanced against the need and costs of solutions implemented. In initial discussions, possible solutions have been identified through the instantiation of a neutral party or (external) arbitration committee, which can be considered a starting point for determining a solution.

10.3.3. Logging

All actors perform *Logging* at various points in time for many internal purposes. *Additionally*, *Logging* is required to enable actors to be able to provide proof of adhering to various requirements. This requires all actors involved in a transaction to perform *Logging* at all points in the *Data Service Transaction* lifecycle. The requirements in the *Trust Framework* can be split depending on for who they are applicable:

1. Requirements on *Trust Framework* level which are applicable to all *Participants*. This includes for example: Minimum logging requirements for activities as evidence for *Disputes*, clearing and settlement, reporting.
2. Requirements on *Data Service* level which are applicable only to specific actors involved in a *Data Service*. This includes for example: *Terms & Conditions*, reporting.

The *Trust Framework* should contain minimum *Logging* requirements to be used as evidence to validate compliance to the *Trust Framework* agreements. Existing *Logging* standards can be used as a basis for Logging specifications in the *Trust Framework*. An example of such a standard is the is the NEN-7513 standard, which includes Logging requirements used for healthcare applications that align with international standards. These standards provide a detailed overview of topics that should be included in logs for the healthcare *Domain*. The healthcare *Logging* requirements can be generalised for use outside of the healthcare *Domain* (see Table 9 for an indicative overview) and can be used as a basis to determine possible *Logging* specifications in the *Trust Framework*.

Table 9: Overview of generalised NEN-7513 logging requirements

Indicative

Topics	Details logged
Actions	Transactions, operational actions, special actions, actions impacting access controls, actions impacting logging, etc.
Actors	Identification, roles, action initiator, ID of authorised actor, type of access, Authorisation type, etc.
Object	ID of object (data), description, Authorisation protocol, consent required, privacy requirements, etc.
Generic logging details	Logging saved location, information source, security requirements: responsibility, availability, access, retention time, etc.

In the *Trust Framework* for cross-*Domain Data Sharing*, minimum *Logging* requirements for *Participants* should be detailed. To ensure alignment with current implementations of *Logging* within existing *Domains*, an exploration and analysis of current *Domain Logging* implementations should be performed. The results of this can be used as input for the minimum Logging requirements for the *Trust Framework*.

Note that *Logging* for reporting can be for 'unhappy flow' purposes, as described above, but can also be used for 'happy flow' purposes. For example, this can be used for dashboards to show the number of successful transactions in the past 24 hours, which can then be used for business development purposes.

10.3.4. Enrolment

As introduced in 7.3.1 Legal status of the Trust Framework, it is likely that to enable sufficient *Trust* between participants, the *Trust Framework* will require *Participants* to be certified to validate adherence to the binding *Trust Framework* agreements. To enable this, a clearly defined enrolment process should be available to potential participants and be included in the *Trust Framework*. An investigation should be conducted into the type of approval that may be required. This varies from a self-declaration of adherence to the *Trust Framework* agreements, to a certification process. In the creation of the *Trust Framework* for cross-*Domain Data Sharing*, this will be investigated, and an enrolment and certification process will be detailed. Possible types of approval which could be used in the *Trust Framework* are given by The Open Identity Exchange[9] and will be used as a reference when creating the *Trust Framework*.

[9] – https://openidentityexchange.org/guide-trust-frameworks-interoperability?page=governance

As the *Data Sharing Coalition* is publicly funded, all developed materials will be publicly available. This includes the *Trust Framework* agreements. Actors are free to use the agreements as they wish without participating in the *Trust Framework* to enable technical Interoperability. These are no-regret options that could be implemented by potential *Participants*. As explained above, simply complying to the *Trust Framework* agreements will often be insufficient to share data, as certification to verify adherence to binding agreements is required to create the needed *Trust* to share *Data* with *Participants*. Only *Data Sharing Coalition Participants* that are certified have scalable *Data Sharing* enabled between them.

11.
Business Models

11.1. Introduction

Business models for *Data Sharing* use cases describe how organisations create and capture value. Business models in the *Trust Framework* describe how the value of a *Data Service* is compensated for between actors. As the future *Trust Framework* should facilitate a wide variety of *Data Services*, multiple business models for cross-*Domain Data Sharing* should be facilitated in the *Trust Framework* agreements.

11.2. Relevance

Actors in a *Data Service* should agree to a business model before performing a *Data Service Transaction*. To this end, the *Data Service Provider* should communicate the relevant business model information to all potential *Data Service Consumers* during *Data Service Discovery* (see **Chapter 9.3.1**). Furthermore, once the financial compensation is agreed, a mechanism to settle this across domains is needed. Therefore, agreements to enable the communication of business models and facilitate financial clearing and settlement are required in the *Trust Framework*.

11.3. Description

A compensation mechanism is needed to facilitate the financial compensation between actors involved in the *Data Service Transaction* if applicable for the *Data Service*. Note that often, *Data Sharing* leads to new *Data Services*, in which case a compensation mechanism can be agreed upon between actors involved on how expected revenue is shared between them.

Examples of compensation mechanisms include, but are not limited to:

- Fees per transaction,
- Recurring fees,
- Flat fees,
- Fee per record of *Data*,
- Fees dependent on *Data* usage.

The compensation mechanism of a use case, is dependent on its characteristics, and could include factors such as:

- Actors involved,
- *Data Service* type,
- Value of the *Data Service*.

In practice, many of these compensation mechanisms seem realistic for cross-*Domain Data Sharing* use cases, and therefore these should be investigated for inclusion in the *Trust Framework*. Note that it is likely that there will be plenty of use cases that explicitly do not have business models or compensation mechanism implemented, and this possibility should also be included. See **Table 10** for examples of compensation mechanisms used in *Data Sharing Coalition* use cases.

In general, in *Data Services*, there should be value for both *Data Service Consumer* and *Data Service Provider* in every *Data Service Transaction*. Based on the specific cross-*Domain Data Service* and what actors aim to achieve through the *Data Service*, the value each actor perceives is not always obvious. In case of an imbalance of perceived value, one actor may need to compensate the other for the *Data Service*, as it could be expected that the actor who experiences the most value should financially compensate the other actor. Examples of the value experienced by actors in the *Data Sharing Coalition* use cases are shown in **Table 10**.

Table 10: Examples of value and compensation mechanisms used in Data Sharing Coalition use cases

Use case	Value for Data Service Consumer	Value for Data Service Provider	Compensation mechanism
Weed Robot	Farmers have guaranteed removal of weeds from land with minimal pesticide usage and damage to crops	Scanned *Data* can be used by weed whacking party to further train algorithms and provide better services	To be decided
Benchmarking for industry associations	Industry associations members can make strategic decisions based on benchmarks performed by the industry association	Industry association gains insights in and for the whole sector and can provide additional benchmarking services to its members	Annual membership fee paid by members to the industry association or a fee per benchmark
Green Loans	Financial domain obtains insights in customer energy usage to deliver advice and loans for sustainable measures to customers, driving new business	Energy system operators allow consumer to use energy *Data* in new contexts; fulfil their societal obligation of facilitating the use of energy *Data*	No additional fees, the use case is part of service offerings within each domain. This will be reassessed when transaction volume increases significantly
VODAN	Research institution realises Societal value; *Data* is being used for effectively battling COVID-19	Researchers' ability to analyse larger datasets, allowing algorithms to discover meaningful patterns in COVID-19 infections	None
Sharing shipment *Data* with insurers	Insurer receives structured and machine-readable *Data* that can be used in their services to enable improved processes and risk management	Logistics organisations can share their trade documentation in one click with control over their Data and without the administrative burden of paper-based documents	To be decided, as it is not clear what actor experiences the most value

Use case	Value for Data Service Consumer	Value for Data Service Provider	Compensation mechanism
Smart Cleaning	*Data* service processors provide additional value for their cleaning party through new insights obtained from the sensor *Data* so that the cleaning party can perform 'demand-based' services	Sensor providers can sell *Data* sensors to buildings and/or cleaning companies	Subscription fee per sensor and/or per batch of transactions from the sensor

In an ecosystem with many actors involved, the business model of a single *Data Service* cannot be determined or changed without considering the impact on the ecosystem. If partner organisations in the ecosystem make use of a *Data Service* and have their own business processes built around the *Data Service*, even a slight change to the business model of the original service can have a huge impact. Therefore, it is important to consider the business model at an early stage of *Data Service* development[10].

To enable trust needed for a *Data Service*, the *Data Service Consumer* must be aware of the business model of a *Data Service* before choosing to make use of it. To this end, the business model and compensation mechanism should be clear and transparent upfront and *Data Service Providers* should include the business model in *Data Service* information, as introduced in Chapter 9.3.1 Data service discovery. Note that the business models for *Data Services* are likely to be dynamic in nature to move with market developments. Therefore, the dynamic *Data Service Discovery* mechanism is suitable to support the potentially dynamic business models of *Data Services*.

Once the *Data Service Consumer* is aware of the business model of a *Data Service*, they can choose to accept that business model. After acceptance of the *Data Service* with accompanying business model in the *Data Service Transaction Agreement*, the *Data Service* can be consumed. Therefore, acceptance of the business model is conditional to making use of the *Data Service*.

Dependent on the business model, the financial compensation for consuming a *Data Service* should be settled between actors. The settlement of the financial compensation could be based on the actual usage. To enable financial compensation based on usage, transactions should be captured in *Metadata* which can be used in settlement calculations. For more information, see Chapter 14 Metadata.

[10] – https://www.researchgate.net/publication/330442460_Business_model_thinking_business_ecosyste ms_and_platforms_The_new_perspective_on_the_environment_of_the_organization

The process for clearing and settlement of the agreed financial compensation could still pose a hurdle for Interoperability and scale. If all *Domains* organise their payments in a non-standardised way this is not scalable as each *Domain* would need bilateral implementations to compensate each other. Therefore, a clearing and settlement mechanism can be considered in the *Trust Framework*. The need and costs of clearing and settlement services are dependent on the scale and maturity of the *Trust Framework*. This dependency of costs of clearing and settlement services on *Trust Framework* should be considered in the decision towards the use of a centralised or decentralised clearing and settlement mechanism within the *Trust Framework*.

Possible solutions for financial clearing and settlement have been identified and shall be further investigated for the *Trust Framework* for cross-*Domain Data Sharing*. One possibility includes that clearing and settlement is facilitated by a separate decentralised broker. The "context broker"[11] as defined by CEF Digital is an example of a decentralised broker. Within the *Trust Framework*, a decentralised broker role could be fulfilled by the *Trust Framework Authority*, or a separate service provider.

It could be that the *Proxy* will have a role in clearing and settlement to reduce the impact on *Data Service Consumers* and *Data Service Providers*. The exact mechanism for clearing and settlement and the role of the *Proxy* in this will be determined in the *Trust Framework*.

[11] – https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Context+Broker

12.
Governance

12.1. Introduction

The future *Trust Framework* agreements and network should be continuously managed and maintained to ensure alignment with future wishes and requirements of *Participants*. To achieve the management and maintenance of the *Trust Framework* agreements and network, a *Trust Framework Governance* is needed.

12.2. Relevance

Governance is needed for the development and subsequent management of the *Trust Framework*. These two phases can be considered separately:

1. **Trust Framework development**
   The initial development of the *Trust Framework* agreements is planned in the next phase of the *Data Sharing Coalition*, when the first version of the *Trust Framework* agreements is co-created in a project setting by members delegated by a so-called "coalition of the willing". This project has a typical co-creation governance, in which the delegates of the coalition of the willing will determine all the content of the *Trust Framework*.

2. **Trust Framework management**
   Once the first version of the *Trust Framework* has been developed and implemented, its agreements and network of *Participants* should be managed. *Participants* want to influence the future developments of the *Trust Framework* to ensure alignment with their future wishes and requirements, and to protect their investment during the development phase. This continuous management requires a neutral governing body which should be described in the *Trust Framework* agreements and thus be shaped and determined in the initial development phase.

12.3. Description

12.3.1. Trust Framework development

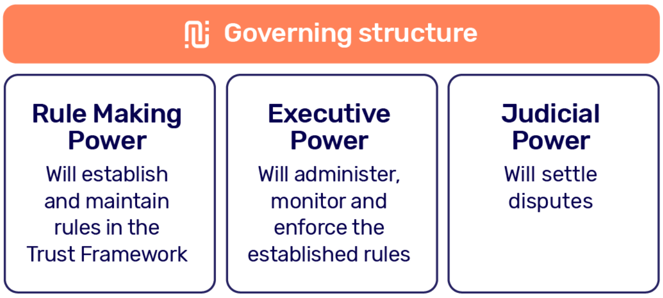
Through a co-creation project, the coalition of the willing shall develop the agreements in the *Trust Framework* for cross-*Domain Data Sharing*.

A project *Governance* structure will be instantiated for the initial development of the *Trust Framework* agreements. This project governance structure will be determined before starting the development of the *Trust Framework* for cross-*Domain Data Sharing*. The *Trust Framework* agreements should include a description of the *Governance* structure and *Governing Body* required for phase 2: *Trust Framework* management and maintenance.

12.3.2. Trust Framework management

The *Trust Framework* agreements will contain a description of the *Trust Framework Governing* structure, roles, and responsibility. The roles and responsibility will be described based on the separation of powers, see Figure 31. This separation of powers is useful in describing and categorising the *Trust Framework Governance* functionality and structure. However, it is likely not practical to realise a pure separate governance entity from the start, because financing separate entities is costly, as each power requires similar resources and capabilities. Furthermore, it is expected that there will not be many disputes in the *Trust Framework*, and therefore the judicial power will not have a large role. The implementation of the Governance is based on the level of maturity and size of the ecosystem, and therefore is subject to change over time. The exact realisation of the *Governing Structure* will be determined in the *Trust Framework* development phase.

Figure 31: The separation of powers in the Trust Framework Governing Structure


Governing structure
Rule Making Power
Will establish and maintain rules in the Trust Framework
Executive Power
Will administer, monitor and enforce the established rules
Judicial Power
Will settle disputes

Rule Making Power

The Rule Making Power establishes and maintains the *Trust Framework* agreements. The *Trust Framework* agreements need to be continuously maintained and updated to ensure alignment with future wishes and requirements of *Participants*. To facilitate this, the functionality of *Trust Framework* agreement management has been identified.

Executive Power

The Executive Power administers, monitors, and enforces the established *Trust Framework* agreements and contains all necessary functions to run and manage the *Trust Framework*. The *Trust Framework* network needs to be actively managed to enable cross-*Domain Data Services* for *Participants* and the enrolment of new *Participants* into the *Trust Framework*. Furthermore, the *Trust Framework* network should be monitored to ensure *Participants* meet and continue to adhere to the set rules and agreements in the *Trust Framework*. Additional roles may be needed to realise efficiencies within the *Trust Framework* network, such as providing standardised test tools. All these functionalities can be considered elements of the Executive Power.

Several functionalities have been identified which will be detailed in the *Trust Framework* for cross-*Domain Data Sharing*:

- Enforcement body,
- Monitoring body,
- Marketing,
- Service Registry management,
- *Trust Framework Participant* enrolment,
- Facilitating test tooling,
- Change and release management,
- Knowledge management.

Judicial Power

The Judicial Power plays a role in settling disputes. This includes the role of Dispute Case Manager, as described in 10.3.2 Dispute management process.

12.3.3. Trust Framework governance representation and financing

The *Governing Body* of the *Trust Framework* must be financed so that it has the resources to achieve its goals of developing and managing the *Trust Framework*. Financing is possible through various means such as:

- Subsidy,
- Recurring fees for *Participants*, or their *Domains*
- Fees based on *Trust Framework* usage.

The financing model of the *Governing Body* is dependent on the value and maturity of the complete *Trust Framework* ecosystem which impacts the willingness-to-pay of *Participants*. Initially, when the value of the *Trust Framework* is not clear to *Participants*, the willingness-to-pay may be low. However, once the *Trust Framework* has proven its value, the willingness-to-pay of *Participants* may increase. Therefore, the financing model of the *Trust Framework* Governance is subject to change over time, and this should be considered in the development of the *Trust Framework*.

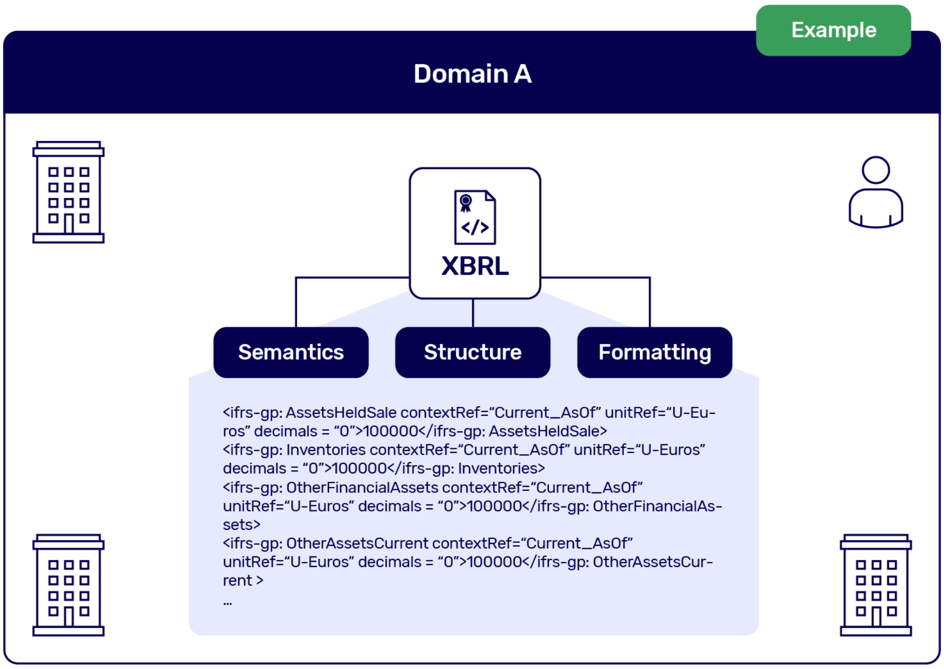
In governance structures, the *Participant* representation often has an impact on their influence. In practice, *Participant* representation is often closely linked to the financing of the *Trust Framework* and *Participant* contribution. In existing *Data Sharing Domains*, the link between financing and influence has been identified as an issue, as *Participants* who have the most influence may not act in the best interest of the complete ecosystem. Therefore, this issue should be addressed, and lessons learned by other Domains should be considered when determining the *Governance* of the *Trust Framework*. The financing of the *Trust Framework Governance* and *Participant* representation in the *Governing Body* will be determined in the *Trust Framework* development phase.

13.
Data Standards

13.1. Introduction

Data Standards are standards that provide the semantics, structure, and formatting of *Data*. *Data Standards* are used to ease communication and create a mutual understanding between actors sharing *Data*. See [Figure 32](#) for an example of the use of a *Data Standard* within a single *Domain*.

Figure 32: Example of XBRL used as a data standard within a domain



```
<ifrs-gp: AssetsHeldSale contextRef="Current_AsOf" unitRef="U-Eu-
ros" decimals = "0">100000</ifrs-gp: AssetsHeldSale>
<ifrs-gp: Inventories contextRef="Current_AsOf" unitRef="U-Euros"
decimals = "0">100000</ifrs-gp: Inventories>
<ifrs-gp: OtherFinancialAssets contextRef="Current_AsOf"
unitRef="U-Euros" decimals = "0">100000</ifrs-gp: OtherFinancialAs-
sets>
<ifrs-gp: OtherAssetsCurrent contextRef="Current_AsOf"
unitRef="U-Euros" decimals = "0">100000</ifrs-gp: OtherAssetsCur-
rent >
...
```

13.2. Relevance

Data Standards are used to create a mutual understanding on the semantics, structure and formatting of *Data* used in *Data* pull and *Data* push *Data Services*, as well as the *Data* exchange towards algorithms. See [Box 12](#) for a description of the differences between *Data Standards* and algorithm standards. For *Data* transfer in *Data Services*, *Data Standards* can be used to ensure a mutual understanding of the *Data* used.

Box 12

Algorithms

Algorithms differ greatly from *Data* when considering the standards used. *Data* in a specific *Data Standard* often can be mapped to another *Data Standard* and be useable. For example, an XBRL *Data* set can be easily converted to be represented in an XLSX file. This is not the case for algorithms. Algorithms are a sequence of instructions to perform a specific computation. Algorithms in computation are written in a certain software to perform their intended task. The algorithm cannot function within other software, and therefore the mapping of algorithms to other standards is not always possible without human interaction. For example, if an algorithm is written in Java, it cannot be easily converted to work in Python. In an academic research context, the mapping of algorithms is possible in theory, but this is not yet commercially viable for businesses.

In the context of the *Data Sharing Coalition*, an algorithm requires *Data* for it to function. This *Data* will be in a specific format and should be transferred to the algorithm for it to function. For this *Data* transfer, the mutual understanding of *Data Standards* applies.

Domains within the *Trust Framework* all make use of different *Data Standards*. Even within *Domains*, there is a variety of *Data Standards* used for a variety of specific use cases. Within a *Domain*, the *Data Service Provider* and *Data Service Consumer* are familiar with each other and can communicate about the *Data Standards* used for specific *Data Services* offered. For *Data Services* that operate across *Domains*, the *Data* used within *Domains* needs to be understandable to other *Domains*. To this end, the *Data Standard* used should be communicated across *Domains* to facilitate understanding of the *Data* by the *Data Service Consumer*.

13.3. Description

The *Data Standard* used in *Data Services* is dependent on several different factors such as actors involved, *Domains* involved and service offered, etc. For example, in some cases, the *Data Service Provider* determines the *Data Standard* used in their service. If the service is used by many different *Data Service Consumers*, they will likely not alter their standards used for a single *Data Service Consumer*. However, in some cases a single *Data Service Consumer* has sufficient power and influence that a *Data Service Provider* is willing to alter the *Data Standards* used in their service to accommodate their specific needs. Additionally, there are instances where a single *Data Service* supports the use of multiple *Data Standards*.

When a *Data Service* is implemented by a *Data Service Provider*, it will make use of a defined *Data Standard*. Once the *Data Service* is defined, the *Data Standard* is fixed, and *Data Service Consumers* must make use of it at the time of the transaction.

As there is a wide variety of *Data Standards* used across *Data Services*, every *Data Service* should explicitly communicate what *Data Standard* they use in the *Data Service Discovery* (see **Chapter 9.3.1**) process. This allows *Data Service Providers* to communicate their *Data* standards requirements before a *Data Service Transaction* can take place. To achieve this, a common language should be created to enable communication of the used *Data Standard* across domains.

To realise efficiencies and enable scalability within the *Trust Framework*, the communication of the used *Data Standard* should be implemented in a machine-readable way. Therefore, *Data Standards* should be communicated in *Metadata*, See Chapter **14 Metadata** for more information. Furthermore, this facilitates the Fair *Principal* "Reusable" of *Data*, see **Box 13** for more information. To enable all possible *Data Standards* to be used within *Data Services* in the *Trust Framework*, the *Trust Framework* should be *Data Standard* agnostic to support all *Data Standards* used in different *Domains*. There is a possibility to harmonise the semantics of *Data* standards across *Domains*. This will be further investigated in the creation of the *Trust Framework* for cross-*Domain Data Sharing*.

An alternative to describing used *Data Standards* in *Metadata* is to define a single *Data Standard* to be used by all *Domains*. It has been identified that it is not always possible to describe a single *Data Standard* that covers all requirements. Even within *Domains* it is often difficult to define a single *Data Standard* to be used. Due to the effort it would take to align all *Domains* on a single *Data Standard*, it is not feasible to create a *Data Standard* for the *Trust Framework*. Therefore, the standardisation of *Data Standards* is left out of scope for the *Trust Framework*. However, the *Harmonisation* of *Data* standards through bilateral agreements should remain possible to *Trust Framework Participants*.

14. Metadata

14.1. Introduction

Metadata describes everything about *Data*, *Data Services*, and *Data Service Transactions* in *Data Sharing* that cannot be assumed to be known by actors involved in *Data Service Transactions*. *Metadata* provides a common language through which actors can communicate with each other across domains in a machine-readable way, to create a shared understanding. Furthermore, *Metadata* may in itself have value to actors involved in a transaction. Within the future *Trust Framework*, *Metadata* is needed to achieve several different goals:

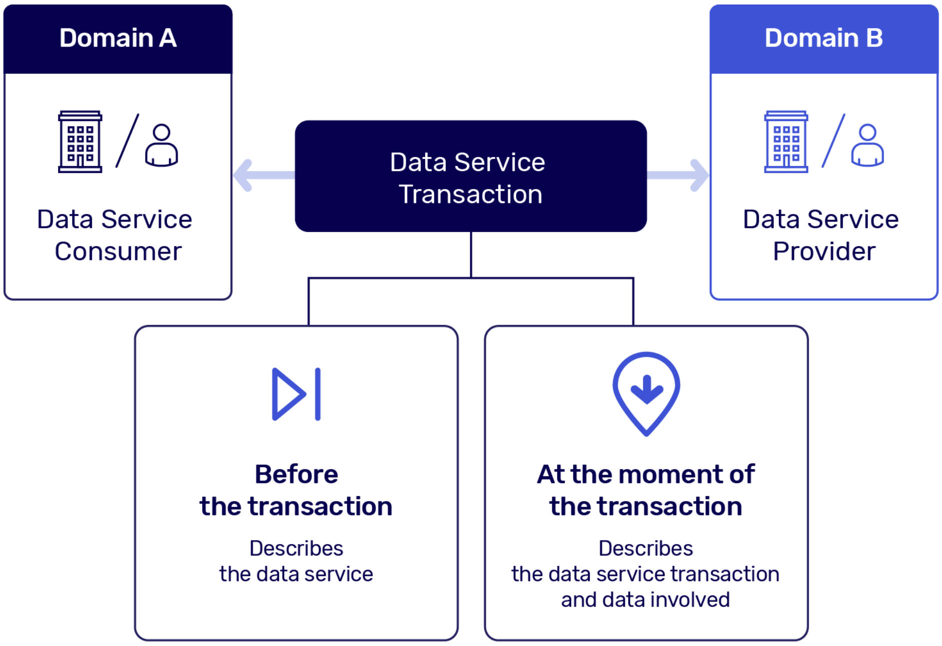
- Enable scalability and efficiencies by providing machine-readable information,
- Facilitate the discovery of *Data Services*,
- Provide input on the *Data Service* for post-transactional processes,
- Enable future developments of the *Trust Framework*, by being extensible by default.

Within the context of the *Data Sharing Coalition*, *Metadata* concerns the *Data Service Transaction* itself and does not include the logging that takes place afterwards.

14.2. Relevance

For a bilateral *Data Service* between two actors, *Metadata* is less relevant as the complete implementation of the *Data Service* is known to all actors involved. Once a *Data Service* becomes multilateral and more actors become involved, it is no longer obvious that all actors are aware of the *Data Service* and the agreements made to enable the *Data Service*. For multilateral cases, *Metadata* plays a role in clarifying all that cannot be assumed to be known. In a cross-*Domain Data* Service, *Metadata* is created at two distinct phases in the transaction lifecycle to achieve the goals described above. *Metadata* is created before a *Data Service Transaction* and at the moment of a *Data Service Transaction*, as shown in **Figure 33**. Before a *Data Service Transaction*, *Metadata* provides a *Data Service* description, which allows services to be discovered and actors to decide whether to engage in a *Data Service Transaction Agreement*. At the moment of a *Data Service Transaction*, *Metadata* is created to describe the *Data Service Transaction* and the *Data Service Transaction Agreement*. See **Chapter 9.3.2**, for an overview of the characteristics of *Data Service Transactions*.

Figure 33: Metadata is created before and at the moment of a data service transaction


Domain A
Data Service Consumer
Data Service Transaction
Domain B
Data Service Provider
Before the transaction
Describes the data service
At the moment of the transaction
Describes the data service transaction and data involved

One of the members of the *Data Sharing Coalition*, GO FAIR, have described several guiding principles for the reuse of digital assets for scientific *Data*. *Metadata* plays a large role in fulfilling the *Fair Principles*, which can also be generically applied to cross-*Domain Data Sharing* beyond the scientific *Domain*. See Box 13 for a description of the FAIR guiding principles.

Box 13
FAIR Data Principles

The FAIR Data Principles[12] provide guidelines for *Domains* and organisations to improve the findability, accessibility, interoperability, and reuse of digital assets. The principles are an extensive list that emphasises the need to make *Data* machine-actionable to deal with its increased volume, complexity, and speed of *Data* creation. The FAIR *Data Principles* indicate that *Data* needs to be:

[12] – https://www.go-fair.org/fair-principles/

Findable

The first step in (re)using *Data* is to find them. *Metadata* and *Data* should be easy to find for both humans and computers. Machine-readable *Metadata* are essential for automatic discovery of *Data* and *Data* services.

F1. (Meta)*Data* are assigned a globally unique and persistent identifier,

F2. *Data* are described with rich *Metadata* (defined by R1 below),

F3. *Metadata* clearly and explicitly include the identifier of the *Data* they describe,

F4. (Meta)*Data* are registered or indexed in a searchable resource.

Accessible

Once the user finds the required *Data*, they need to know how they can be accessed, possibly including *Authentication* and *Authorisation*.

A1. (Meta)*Data* are retrievable by their identifier using a standardised communications protocol,

- **A1.1** The protocol is open, free, and universally implementable,
- **A1.2** The protocol allows for an *Authentication* and *Authorisation* procedure, where necessary,

A2. *Metadata* are accessible, even when the *Data* are no longer available.

Interoperable

The *Data* usually need to be integrated with other *Data*. In addition, the *Data* need to interoperate with applications or workflows for analysis, storage, and processing.

I1. (Meta)*Data* use a formal, accessible, shared, and broadly applicable language for knowledge representation.

I2. (Meta)*Data* use vocabularies that follow *Fair Principles*

I3. (Meta)*Data* include qualified references to other (meta)*Data*

Reusable

The ultimate goal of FAIR is to optimise the reuse of *Data*. To achieve this, *Metadata* and *Data* should be well-described so that they can be replicated and/or combined in different settings.

R1. (Meta)*Data* are richly described with a plurality of accurate and relevant attributes,

- **R1.1.** (Meta)*Data* are released with a clear and accessible *Data* usage license,
- **R1.2.** (Meta)*Data* are associated with detailed provenance,
- **R1.3.** (Meta)*Data* meet domain-relevant community standards.

14.3. Description

14.3.1. Before the data service transaction

Before *Data* can be shared, relevant *Data Service* information needs to be clear to all actors involved in the *Data Service Transaction*. To this end, the potential *Data Service Consumer* first needs to discover the *Data Service*, as described in 9.3.1 Data service discovery. After the *Data Service Discovery*, the potential *Data Service Consumer* should have access to all *Data Service* information needed to come to a decision on whether or not to make use of the *Data Service*. Throughout the previous chapters of this document, several topics have been identified (see Table 11) which should be described in *Metadata* before the *Data Service Transaction*.

Table 11: Overview of categorised identified metadata topics

	Before the transaction	At the moment of the transaction
Actor information	• *Domain* information • *Data service provider* information • Role information	• *Data Service Provider* information • *Data Service Consumer* information • Role information
Data Service information	• Terms and conditions • Business model	• Negotiated Terms and conditions • Negotiated Business model
Data Service Transaction information	• Security level requirements	• *Data Service Transaction Agreement* • Security level • Consent • Transaction actions (for audit trails)
Data information	• *Data* description • *Data* standards • *Data* quality	• *Data* standards • *Data* quality

The identified topics actively contribute towards fulfilling the FAIR guiding principles (see Box 13) and can be categorised as shown in the left column of Table 11.

14.3.2. At the moment of the data service transaction

At the moment of a *Data Service Transaction*, *Metadata* is created by all actors involved in the *Data Service* to be used in processes after the *Data Service Transaction*. *Specific* actions during the *Data Service Transaction* should be captured in *Metadata* to be used for several different purposes, including:

- Register the accepted *Data Service*,
- *Data* analysis,
- Auditing,
- Clearing and settlement.

As shown in Table 11, topics have been identified which should be captured in *Metadata* at the moment of the *Data Service Transaction*. The topics can be categorised as shown in the right column of Table 11, and actively contribute towards fulfilling the FAIR guiding principles (see Box 13).

Optionally, the *Metadata* created about a dataset at the moment of the transaction could be captured and added to the *Metadata* before a transaction for future transactions. This could be relevant if a specific *Data Service* relies on knowledge of what has happened to a dataset in the past and the provenance of the dataset. This follows the FAIR Principle “Reusable”. This option should be investigated in the creation of the *Trust Framework* for cross-*Domain Data Sharing*.

14.3.3 Metadata in the Trust Framework

In the *Trust Framework* for cross-*Domain Data Sharing*, the *Metadata* implementation of the *Trust Framework* will be specified, based on the high-level business requirements described here. An investigation into existing *Metadata* implementations in *Domains* and by other *Data Sharing Initiatives* will be done to analyse where existing *Metadata standards* can be used in the *Trust Framework*.

15. Manifestation of Topics in the Trust Framework

The previous chapters present and investigate topics which have been identified to be relevant for cross-*Domain Data Sharing*. The insights presented will be used as an input for the future *Trust Framework* for cross-*Domain Data Sharing*. In the *Trust Framework*, legally binding agreements will be formed on all these topics to enable seamless cross-*Domain Data Sharing*. The *Trust Framework* for cross-*Domain Data Sharing* will be co-created with a coalition of the willing. Every topic that has been discussed in this *Data Sharing Canvas* will be covered in the *Trust Framework* and will be analysed across five disciplines: Business, Legal, Operational, Functional and Technical (BLOFT). Note that the order of these topics reflects the order in which they have been discussed in this document and does not necessarily correspond to the order they will be processed for the *Trust Framework*.

15.1. Data Service Terms and Conditions

The topic *Terms and Conditions* will be discussed in all BLOFT dimensions (Business, Legal, Operational, Functional and Technical) as it is connected to multiple different topics (e.g. IAA, *Metadata*, business model). The general outline of the topic will be discussed in the Functional part of the BLOFT dimensions of the *Trust Framework* for cross-*Domain Data Sharing*, as how organisations must deal with, and handle conditions is a functional aspect.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Make implicit *Terms and Conditions* more explicit,
- Finalise Terms and Conditions clusters,
- Create levels for Terms and Conditions clusters,
- Decide on Metadata language for Terms and Conditions.

15.2. Identification, Authentication and Authorisation

The general outline of the topic will be discussed in mainly the Functional and Technical part of the BLOFT dimensions of the *Trust Framework*, as these are the most important topics regarding how organisations must deal with and handle *Identification*, *Authentication* and *Authorisation*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Include explicit definitions for identifiers,
- Define standard LoAs based on eIDAS,
- Further investigate and define usage of Qualified Trust Services,
- Define interoperable UX standards,
- Define requirements needed to facilitate the distribution of *Authorisation* roles across *Domains*,
- Investigate and define a method of validating Pre-configured *Delegation*,
- Discuss and define the redirects and user interface requirements needed for interoperable human-to-machine *Authentication*.

15.3. Legal Context

Legal context is of vital importance to establish trust required to share *Data*. The general outline of the topic will be discussed in the Legal and Functional parts of the BLOFT dimensions of the *Trust Framework*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Formalise the legal basis of the *Trust Framework*,
- Specify the functionality of a chain of bilateral agreements,
- Investigate the role of a *Trust Framework Authority* functioning as monitoring and enforcement body,
- Investigate several open legal topics to ensure they are covered within the *Trust Framework*.

15.4. InformationSecurity

Managing *Information Security* risk is essential to establish trust required to share *Data*. The general outline of the topic will be discussed in mainly the Organisational and Technical part of the BLOFT dimensions of the *Trust Framework*, as these are the most important topics regarding how organisations implement *Information Security*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Define *Information* Security clusters
- Define security levels and requirements based on security clusters
- Specify how security levels can be communicated within *Metadata*

15.5. Data Service Exchange

The functional *Data Service* exchange requirements should be determined before implementation decisions of an exchange protocol are made as these have an impact on the functionality of the *Trust Framework*. The general outline of the topic will be discussed in mainly the Business and Technical part of the BLOFT dimensions of the *Trust Framework*, as these are the most important topics regarding how *Data Service* exchange can be realised.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Determining the contents of the Service Directory,
- Defining the Data Service Discovery mechanisms,
- Specifying Functional Data *Service* exchange requirements based.

15.6. Operational Agreements

Within the topic of Operational Agreements, Dispute Management is a key topic which should be harmonised in the *Trust Framework* to enable Trust. The general outline of the topic will be discussed in mainly the Operational part of the BLOFT dimensions of the *Trust Framework*, as this is the most important topic regarding a *Dispute Management Process*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Describe a Dispute Management Process,
- Define SLAs for the process of solving disputes,
- Define SLAs for the analyse of reported disputes,
- Determine the need and extent of an appeal process,
- Detail the minimum *Logging* requirements.
- Investigate and determine the enrolment (and potential certification) process for potential *Participants*

15.7. Business Models

The *Trust Framework* should support a wide variety of use cases with a variety in business models, therefore all possible business models should be facilitated. The general outline of the topic will be discussed in mainly the Business and Technical parts of the BLOFT dimensions of the *Trust Framework*, as these are the most important topics regarding use case business models and implementation of these.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Investigate the need to support all possible compensation mechanisms in the *Trust Framework*,
- Define a method to communicate use case business model across *Domains*,
- Investigate the need for a financial clearing and settlement function in the *Trust Framework*,
- Determine the role of the *Proxies* in Clearing and Settlement.

15.8. Governance

Governance is needed to develop and subsequently manage the *Trust Framework* agreements and network. The general outline of the topic will be discussed in mainly the Legal and Operational part of the BLOFT dimensions of the *Trust Framework*, as these are the most important topics regarding use case business models and implementation of these.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Determine a coalition of the willing who will decide on the content of the *Trust Framework*,
- Define a description of the *Governing Structure* in the initial *Trust Framework* agreements,
- Describe *Governance* functionality split by the separation of powers,
- Determine a *Governance* representation and financing model.

15.9. Data Standards

Data Standards are standards that provide the semantics, structure, and formatting of *Data*, and are used in the *Trust Framework* to create a mutual understanding between actors sharing *Data*. The general outline of the topic will be discussed in mainly the Technical part of the BLOFT dimensions of the *Trust Framework*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Ensure the *Trust Framework* is *Data* standard agnostic,
- Enable the communication of *Data* standards within *Metadata*.

15.10. Metadata

Metadata is needed in the *Trust Framework* to enable scalability and efficiencies by providing machine-readable information before and after *Data Service Transactions*. *Metadata* concerns all dimensions of the BLOFT framework, but the general outline of the topic will be discussed in mainly the Technical part of the BLOFT dimensions of the *Trust Framework*.

Steps to take to come to agreements for the *Trust Framework* for cross-*Domain Data Sharing* are/can be:

- Determine existing *Metadata* languages which can be used to describe all topics identified to be part of *Metadata*,
- Decide on the *Metadata* language used in the *Trust Framework*,
- Define a shared *Data* ontology that defines different levels for different *Data* constructs,
- Describe the technical implementation of *Metadata*.

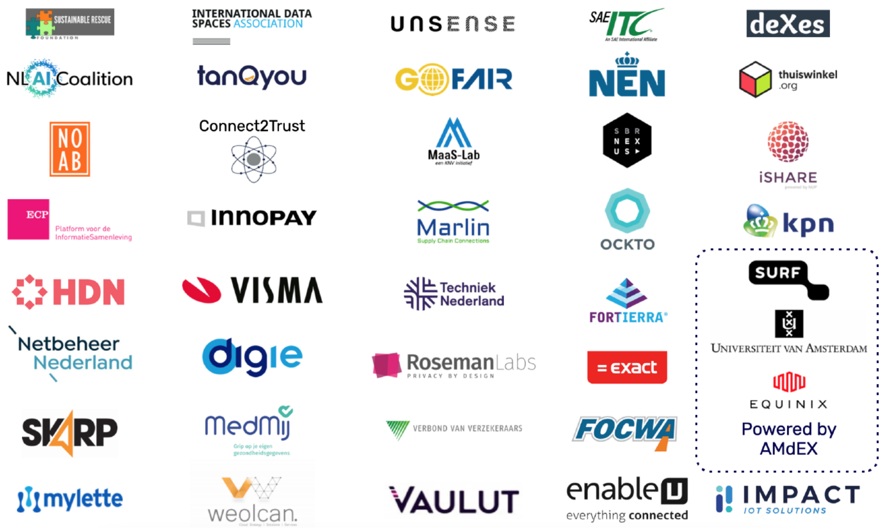
Appendix

I.

Data Sharing Coalition Overview

Figure 34: Overview of data sharing initiatives within the Data Sharing Coaltion as of April 2021

40 participating organisations...



...represent more than 100.000 organisations

Industry associations that represent their members

Data sharing initiatives and **software providers** that represent their users

Standards institutions that represent users of standards

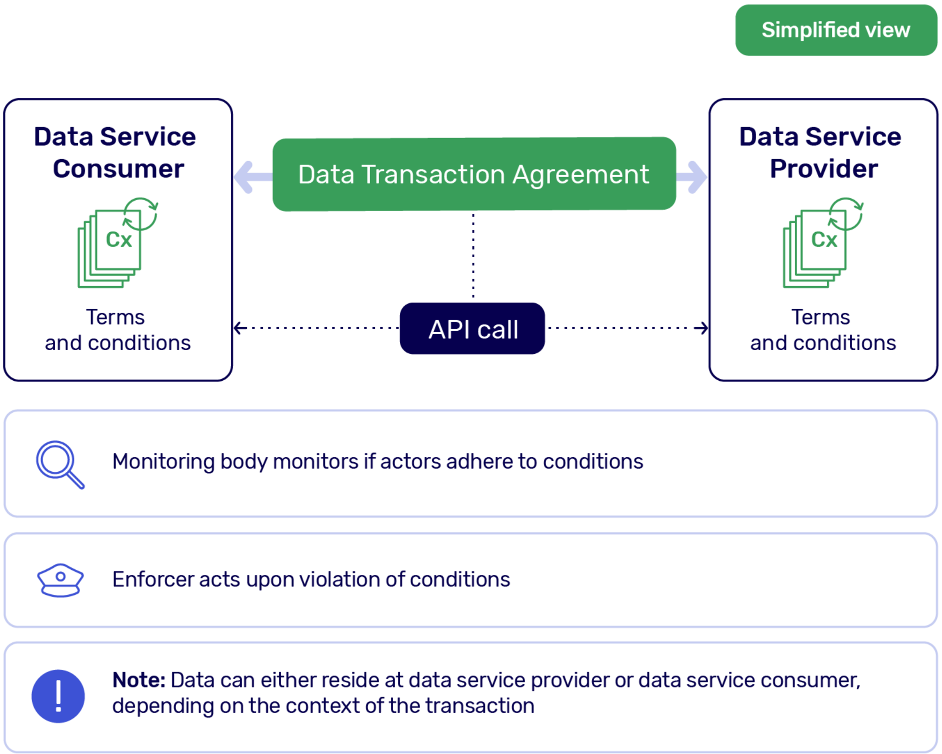
Companies that create value with data themselves

II.

Interoperability and Harmonisation

In a *Data Service Transaction Agreement* between a *Data Service Consumer* and a *Data Service Provider*, *Policies* apply. See Figure 35.

Figure 35: Terms and conditions in a data service transaction agreement.


Simplified view
Data Service Consumer
Cx
Terms and conditions
Data Transaction Agreement
API call
Data Service Provider
Cx
Terms and conditions
Monitoring body monitors if actors adhere to conditions
Enforcer acts upon violation of conditions
!
Note: Data can either reside at data service provider or data service consumer, depending on the context of the transaction

A *Data Service Transaction Agreement* is an agreement (handshake) between a *Data Service Consumer* and *Provider* on the *Terms and Conditions* associated with a specific data transaction. An agreement is achieved through the following five steps:

1. A *Data Service Provider* publishes its *Data Service* including all *Policies*.
2. A *Data Service Consumer* requests a *Data Service* (API call) and provides evidence of adherence to *Access Control Rules*.
3. The *Data Service Provider* evaluates the evidence and executes the requested *Data Service* based on the result of this evaluation.
4. The *Data Service Provider* confirms the *Data Service Transaction Agreement*.
5. The *Data Service Provider* executes the *Data Service* while both *Data Service Provider* and *Data Service Consumer* provide evidence of adherence *Obligations* and *Advice Policies*.

Note: Before a *Data Service Transaction Agreement* takes place, there may be a pre-contract phase where actors may negotiate the terms of a *Data Service*. For this phase, rules of engagement according to the respective *Domain* or consortia the actors are in may apply.

These steps hold for all types of *Data Services* (e.g. *Data* pull, *Data* push and *Data* visiting, see Table 3).

Box 14

Steps to reach a Data Service Transaction agreement in the energy domain

Within the energy *Domain*, the energy provider (*Data Service Consumer*) wants to make use of energy consumer *Data* (e.g. on energy usage), which is currently in possession of the DSOs (*Data Service Provider*). DSOs enable energy providers to access consumer *Data* through publishing their *Data Service*, including all *Policies* that the energy provider should adhere to. Only with consent of the consumer can the energy provider access the consumer's energy *Data*. The energy provider needs to identify the energy producer and the DSO authenticates the identity of the energy producer. In addition, the DSO evaluates the evidence of adherence to other *Policies* of the energy provider, before providing energy provider access to the consumer *Data*. Both the energy provider and the DSO have agreed on the *Policies* both should adhere to and access will be provided.

III.
Terms and Conditions

III.I. Terms and Conditions in DSC Use Cases

Box 15
Terms and conditions in DSC use cases

Different *Terms and Conditions* are relevant in the use cases in which the *Data Sharing Coalition* is involved. Below, indicative and non-exhaustive lists of *Terms and Conditions* (formalised into Policies) within these use cases are shown.

Example policies in 'Green Loans' use case (HDN – Netbeheer NL)

Access Control Rules:

- Identity of consumer must be verified at the appropriate Level of Assurance that matches the risk-context of the transaction
- There must be reasonable certainty that the EAN-code (smart meter identifier) for which *Data* is requested belongs to the consumer's smart meter
- Identity Intermediary must be certain
- Intermediary must have unique identifier
- DSO must be able to verify that intermediary is "Trustworthy"
- Consumer *Authorisation* must be linked to identifier of intermediary
- Purpose of *Data* requested must match the operations of the intermediary

Obligations and Advice:

- Scope of usage is the mediation process, which includes sending (subset of) *Data* to banks
- *Data* may not be altered and must maintain "seal of validity"
- Time to live is maximum of 24 months

Example policies in 'Sharing e-CMR data with insurers' use case (iSHARE – Verbond van Verzekeraars)

Access Control Rules:

- Access rights of the insurer must be registered by the claim issuer in an Authorisation Registry
- Authorisation is granted based on *Delegation* evidence provided by claim issuer to the e-CMR provider
- Parties must either be an organisation with delegated *Data* access or the owner of the *Data*.
- Parties must provide a qualified eIDAS (or PKIOverheid) certificate for *Authentication*

Obligations and Advice:

- Scope of usage is the claims handling process
- Licenses indicate for which purposes the (subset of) shipment *Data* may be used (e.g. no limitations, non-commercial use only, for own use only)
- Time to live of shipment *Data* points at insurer can be set to a maximum by the claim issuer

III.II. Initial Policy Clusters and Examples of Policies

Policy clusters are sets of *Policies*. The overview below shows preliminary *Policy* clusters. This overview is based on the input that is provided by the *Data Sharing Initiatives* in the *Data Sharing Coalition*. This overview of clusters is not exhaustive but serves as an example to be used as a starting point for the development of the *Trust Framework* for cross-*Domain Data Sharing*. This first set-up distinguishes clusters on its type of *Policies*: *Access Control Rules* and *Obligations and Advice* (both usage and other).

Table 12: Overview of clusters and types of policies

Cluster	Policies	Type
Scope	Time to live	*Obligations and Advice: Usage*
	Usage scope	*Obligations and Advice: Usage*
	Propagation restrictions	*Obligations and Advice: Usage*
	Third party use of Data	*Obligations and Advice: Usage*
	Usage based on geography	*Obligations and Advice: Usage*
	Target binding	*Obligations and Advice*
Authorisation	Access management	*Access Control Rules*
	Delegated rights	*Access Control Rules*
Authentication	Multi-factor Authentication	*Access Control Rules*
	Supported e-ID means	*Access Control Rules*
	Identity confirmation mechanism	*Access Control Rules*
Liabilities	Indemnification	*Obligations and Advice*
Privacy (pre)	Privacy Impact Assessments	*Access Control Rules*
	Risk analysis	*Access Control Rules*

Cluster	Policies	Type
Privacy (post)	Anonymisation	*Obligations and Advice*
	Right to be forgotten	*Obligations and Advice*
Information classification	Data classification scheme	*Access Control Rules*
Information access	Access management protocol	*Access Control Rules*
	Separation of functions	*Access Control Rules*
	User access rights audit	*Access Control Rules*
Operational conditions	Data minimalisation	*Obligations and Advice*
	Testing requirement	*Obligations and Advice*
	Data breach notification(s)	*Obligations and Advice*
Provenance	Obligated provenance	*Obligations and Advice*
Data storage	Data retention period	*Obligations and Advice*
	Data deletion evidence	*Obligations and Advice*
	Encryption of stored Data	*Obligations and Advice*
	Back-up retention period	*Obligations and Advice*
	Cryptographic key storage	*Obligations and Advice*
Non-repudiation	Digital signature requirement	*Obligations and Advice*
Laws and regulations	Declaration of adherence to law	*Access Control Rules*
	Applicable law	*Access Control Rules*
	GDPR compliance	*Access Control Rules*
Information security	Confidentiality	*Obligations and Advice*
	Integrity	*Obligations and Advice*
	Authenticity	*Obligations and Advice*
Geographical information	Data processing outside of EU	*Obligations and Advice*

Cluster	Policies	Type
Employee qualifications	IT officer assignment	*Access Control Rules*
	Employee competency declaration	*Access Control Rules*
	Employee screenings	*Access Control Rules*
Supervision	Monitoring	*All*
	Enforcement	*All*
	Arbitrage and dispute settlement	*Obligations and Advice*

III.II.I. Longlist of metadata languages for policies

Different *Metadata* languages exist of which some are specifically developed for *Terms and Conditions*. These *Metadata* languages enable coherent communication across sectors on *Terms and Conditions* and hence, examples (see below) are discussed in this chapter.

DCAT/ODRL

DCAT is a worldwide W3C *Metadata* standard, applied by the Dutch government among others. In the newest version of DCAT, datasets can be enriched with conditions for *Data Sharing*. ODRL is the standard for the description of these conditions.

eFlint

eFlint is a standard meant to make the structure and meaning of legal documents "machine readable".

Akomo Ntoso

Akomo Ntoso is an open standard meant to make the structure and meaning of legal documents "machine readable".

RDF

RDF (Resource Description Framework) is a standard for *Data* exchange, developed by W3C.

IV.
Data Service Discovery

IV.I. Industry Standards for Service Discovery

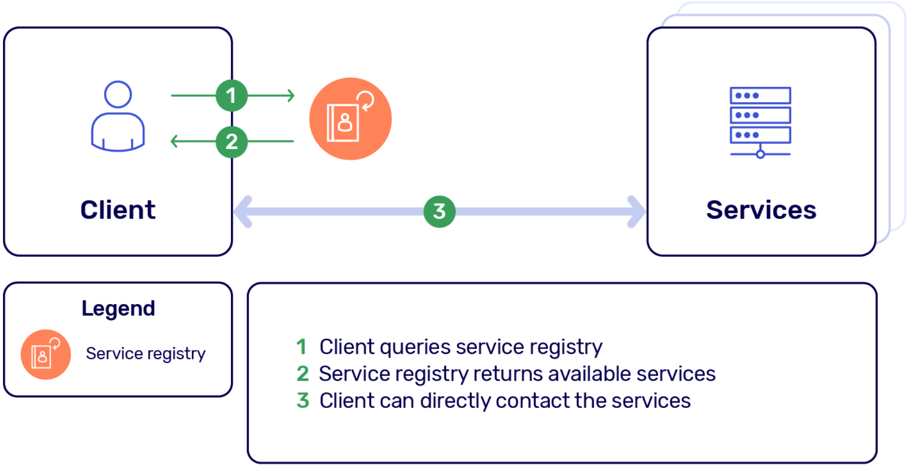
'Client' and 'Server' side discovery are industry standards for discovery using a service registry. From the perspective of cross-*Domain Data Sharing*, the Client can be considered either a *Data Service Consumer* or their *Proxy*. In this context, the services being discovered can be either the *Data Service Provider* or their *Proxy*.

IV.II.I. Client side discovery

In client side discovery, the client is responsible for discovering *Data Services*. For every discovery request, the client will check a *Service Registry*, see **Figure 36**. Main characteristics of client side discovery include:

- Straightforward interactions which do not require additional parties (i.e. discovery broker),
- Client implementation must contain intelligent logic and a coupling with the *Service Registry*.

Figure 36: Schematic overview of client side discovery



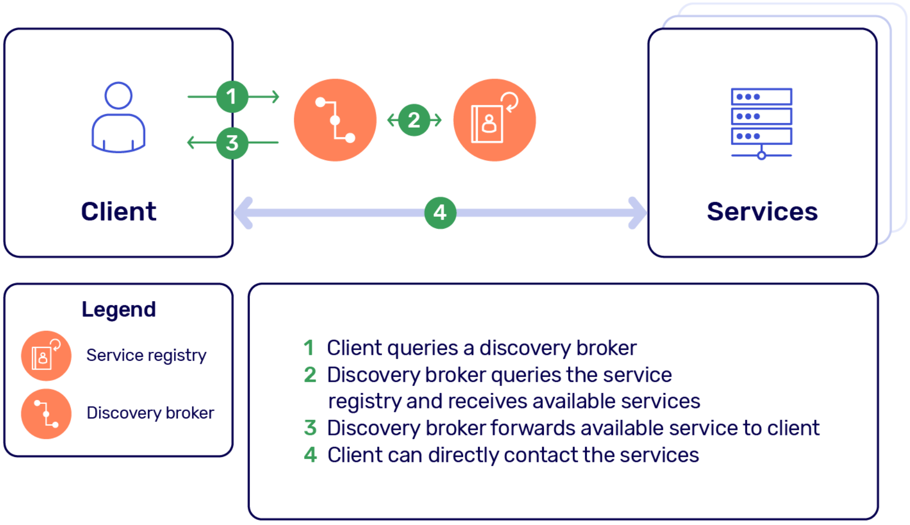
1 Client queries service registry
2 Service registry returns available services
3 Client can directly contact the services

IV.II.II. Server side discovery

In server side discovery, the client makes a transaction request towards a discovery broker. The discovery broker is responsible for discovering *Data Services*, see Figure 37. For every discovery request, the discovery broker will check a *Service Registry* and may perform additional services. Main characteristics of server side discovery include:

- Simple client implementation as discovery logic is handled by a broker,
- A discovery broker can deliver additional services,
- The role of discovery broker must be implemented and maintained, which comes with costs.

Figure 37: Schematic overview of server side discovery

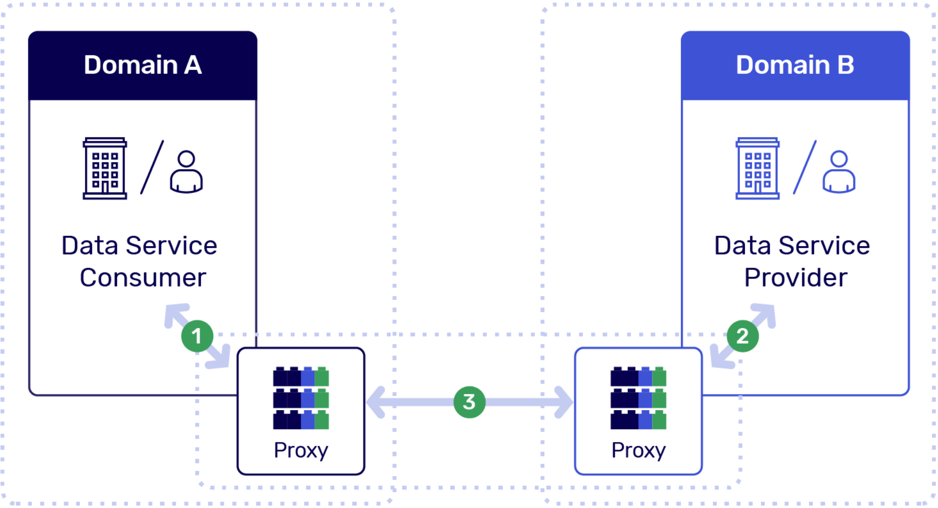


1 Client queries a discovery broker
2 Discovery broker queries the service registry and receives available services
3 Discovery broker forwards available service to client
4 Client can directly contact the services

IV.II. Data Service Discovery in the Proxy Model

Data Service Discovery applies to the complete end-to-end process of *Data Service Exchange*. In the *Proxy* model, *Data Service* discovery can be seen from a number of different perspectives. Once *Domains* are fully *Harmonised*, discovery can take place directly between *Data Service Consumers* and *Data Service Providers*. Before full *Harmonisation* is reached, each perspective of *Data Service* discovery must be considered separately, see Figure 38.

Figure 38: Various data service discovery perspectives within the proxy model



1 Data service consumer must discover services through their domain proxy
2 Domain B proxy must be discover data services from data service providers within Domain B
3 Domain A proxy must discover Domain B proxy in order to discover services available in Domain B

From a *Data Service Consumer* perspective, server side discovery reduces impact on the *Data Service Consumers*, (Discovery perspective 1 in **Figure 38**). A *Data Service Consumer* must discover the services that they want to make use of. To reduce impact on *Data Service Consumer*, the *Proxy* can perform this *Discovery* request for them. From the *Data Service Consumer* perspective, the *Proxy* has the role of discovery broker and this can be considered server side discovery.

The *Data Service Provider's Proxy* must be able to discover available *Data Services* within its *Domain* (Discovery perspective 2 in **Figure 38**). Depending on *Domain* implementations, both client and server side discovery solutions are viable as both solutions do not impact the *Data Service Provider*.

The *Data Service Consumer's Domain Proxy* must be able to discover *Data Service* providers within another *Domain* via their *Proxy* (Discovery perspective 3 in **Figure 38**). *Client* side discovery can be implemented in order for the *Proxy* to be able to perform its own discovery. Server side discovery can be implemented in order to facilitate discovery brokers to implement the discovery server. The design choices made will be applicable to *Data Service Consumers* and *Data Service Providers* once *Domains* reach full Harmonisation.

DATA SHARING
COALITION

✉ info@datasharingcoalition.eu
🌐 datasharingcoalition.eu