Enershare
The Energy Data Space for Europe

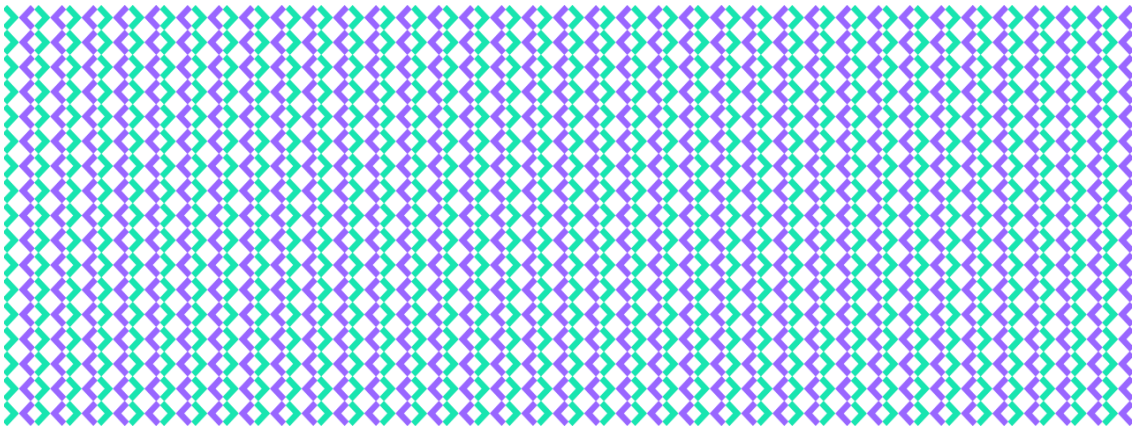
European Common Energy Data Space Framework Enabling Data Sharing - Driven Across – and Beyond – Energy Services

enershare.eu

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

D4.2 Enershare Trust and sovereignty building blocks

Beta version



Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Publication details

Grant Agreement Number	101069831
Acronym	Enershare

Full Title	European Common Energy Data Space Framework Enabling Data Sharing-Driven Across — and Beyond — Energy Services
Topic	HORIZON-CL5-2021-D3-01-01 'Establish the grounds for a common European energy data space'
Funding scheme	HORIZON-IA: Innovation Action
Start Date	Jul 1, 2022
Duration	36 months
Project URL	enershare.eu
Project Coordinator	Engineering
Deliverable	D4.2 – Enershare Trust and sovereignty building blocks. Beta version
Work Package	WP4 – Trust and sovereignty enabling framework and building blocks
Delivery Month (DoA)	M18
Version	1.0
Actual Delivery Date	February 28, 2024
Nature	Report
Dissemination Level	PU

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Lead Beneficiary	TNO
Authors	Sonia Jimenez (IDSA), Alessandro Rossi (ENGINEERING), Rizwan Mehmood (FhG), Simon Dalmolen (TNO), Maarten Kollenstart (TNO), Wouter van den Berg (TNO), Michiel Stornebrink (TNO)
Quality Reviewer(s)	Volker Berkhhout (FhG) and Aleksandr Egorov (NESTER)
Keywords	Data space connector, identity and access management, usage control, trust, sovereignty, full stack integrity, blockchain, ledger

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Document History

Ver.	Date	Description	Author	Partner
0.1	Nov 10, 2023	ToC	Michiel Stornebrink Simon Dalmolen	TNO
0.2	Dec 22, 2023	Draft	Rizwan Mehmood Sonia Jimenez Alessandro Rossi Maarten Kollenstart	FhG, IDSA, ENG, TNO
0.8	Jan 31, 2024	Draft Completed chapters 4 – 7	Rizwan Mehmood Sonia Jimenez Alessandro Rossi Maarten Kollenstart	FhG, IDSA, ENG, TNO
0.8	Feb 18, 2024	Consolidated version Integrated insights about connector infrastructure	Michiel Stornebrink Wouter van den Berg	TNO
0.9	Feb 23, 2024	Reviewed	Volker Berkhout Aleksandr Egorov	FhG NESTER
0.99	Feb 27, 2024	Review processed	Rizwan Mehmood Sonia Jimenez Alessandro Rossi Maarten Kollenstart Michiel Stornebrink	FhG, IDSA, ENG, TNO, TNO
1.0	Feb 28, 2024	Final version	Michiel Stornebrink	TNO

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Table of Contents

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

1 Introduction

1.1 About the project

The overall vision of Enershare is to develop and demonstrate a European Common Energy Data Space which will deploy an intra-energy and cross-sector interoperable and trusted energy data ecosystem. Private consumers, energy and non-energy business stakeholders and regulated operators will be able to access, share and reuse, based upon voluntary agreements or legal obligations where such obligations are in force: (a) large sources of currently fragmented and dispersed data; (b) data-driven cross-value chain services and digital twins for various purposes.

1.2 About this document

This deliverable, which is the second of a series of three (alpha, beta and final version), presents the (extended) designs for trust and sovereignty related components for the Enershare dataspace and describes the implementation and validation plans towards third technology release.

This deliverable accompanies the (extended) software components that are provided to WP8 for the integration in the energy dataspace. It is the intermediate result of joint activities of WP4, namely Task 4.1 Full stack integrity, Task 4.2 Federated Identity and Access Management framework, Task 4.3 Usage control technologies and Task 4.4 Decentralized DLT & smart contract for trusted data management.

1.3 Intended audience

The intended audience for this deliverable is:

- All project partners that are involved in WP4. This document contains the lower level designs of the building blocks for the third technology release
- All project partners from other WPs (especially WP5). This document contains the scope and building blocks that are considered part of the trust and sovereignty framework. It allows others to understand what building blocks and functionalities can and cannot be expected from WP4.
- All project partners involved in integration (WP8) and pilot implementations (WP9). This document addresses what building blocks are provided as part of the second technology release and how and which pilot requirements are addressed.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

- External stakeholders to the project that would like to be part of the Enershare energy dataspace and need to integrate and deploy trust and sovereignty building blocks.

1.4 Reading recommendations

The remaining document is structured as follows:

- **Chapter 2 Architecture** is called as such because it positions the topics of the different WP4 in the dataspace architecture. In addition, this chapter describes the alpha version deployment (MVP-1) and its components and processes.
- **Chapter 3 Dataspace protocol** discusses the recent developments around the Dataspace Protocol and explains the effect this has on the development of components in Enershare.
- **Chapter 4 Introducing decentralized identities** explores the shift towards decentralized identities in the Enershare dataspace, including the functional/logical view, technical design, and next steps to take towards the following technology release. Also, the alignment with sister projects and the importance of interoperability are discussed.
- **Chapter 5 Usage control enforcement** details the functional/logical view of usage control policies, including policy specification, classes, and components. It also describes the technical design and current developments in usage control enforcement, focusing on technology selection and pilot policy requirements
- **Chapter 6 Usage policy notarization on the blockchain** outlines how Distributed Ledger Technology (DLT) and blockchain can be leveraged to notarize usage policies, ensuring their immutability and non-repudiation, which are critical for establishing a secure and trusted energy dataspace. The chapter details the use of smart contracts, specifically designed for the Proof of Existence (PoE) of documents or policies. Finally, the current implementation of the PoE is laid out, though final implementation is planned for D4.3.
- **Chapter 7 Remote attestation for full stack integrity** discusses the process that allows one party (the verifier) to verify the integrity of the software and hardware environment of another party (the prover). The chapter goes on to discuss the technical design of the remote attestation mechanism, including the use of sidecar architecture to facilitate the attestation process.
- **Chapter 8 Conclusions** summarizes the updated set of software components for this beta version release, as well as the plans for the third and final technology release.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

2 Architecture

2.1 Conclusions from deliverables D2.5 and D4.1

The previous Enershare deliverable D2.5 – "Final Version of the Enershare Requirements, SSH-driven Approach and Reference Architecture" described the Reference Architecture for the Enershare project. The Reference Architecture incorporated currently existing initiatives, including IDSA, DSSC, BRIDGE, GAIA-X, OpenDEI, FIWARE, and DSBA, and was based on the use cases described in Enershare deliverable D2.1 – "Use cases' descriptions and list of minimum Data Space building blocks required for pilots". The Reference Architecture used the 4+1 Architectural View Model that contained the Logical View, Development View, Process View and Deployment View. It addressed the three pillars of the DSSC Blueprint Version 0.5, namely: Data Space interoperability, Trust and Sovereignty, and Data Service and Marketplace. A generic description of a minimal viable product (MVP) was included in the Development View, consisting of the identity provider (DAPS and CA), a Connector implementation, and the Metadata Broker.

Enershare Deliverable D4.1 – "Trust and sovereignty building blocks" addressed the pilot requirements in the energy dataspace in terms of identity- and access management and usage control policies and, additionally, discussed full-stack integrity and distributed ledger technology. The deliverable D4.1 described two essential components that were provided by WP4 for the first technology release of the energy dataspace, i.e., MVP-1. These components were the identity provider and the dataspace connector implementation.

It was concluded in deliverable D4.1 that WP4 would continue the work on trust and sovereignty by:

(1) integrating self-sovereign identity (SSI) concepts into the identity manager component (T4.2),
(2) implementing usage control policies that were identified in the pilots (T4.3),
(3) integrating full stack integrity in the (TSG) connector implementation (T4.1), and
(4) implementing a component using distributed ledger technology (T4.4).

Figure 1 shows how these components and tasks relate.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

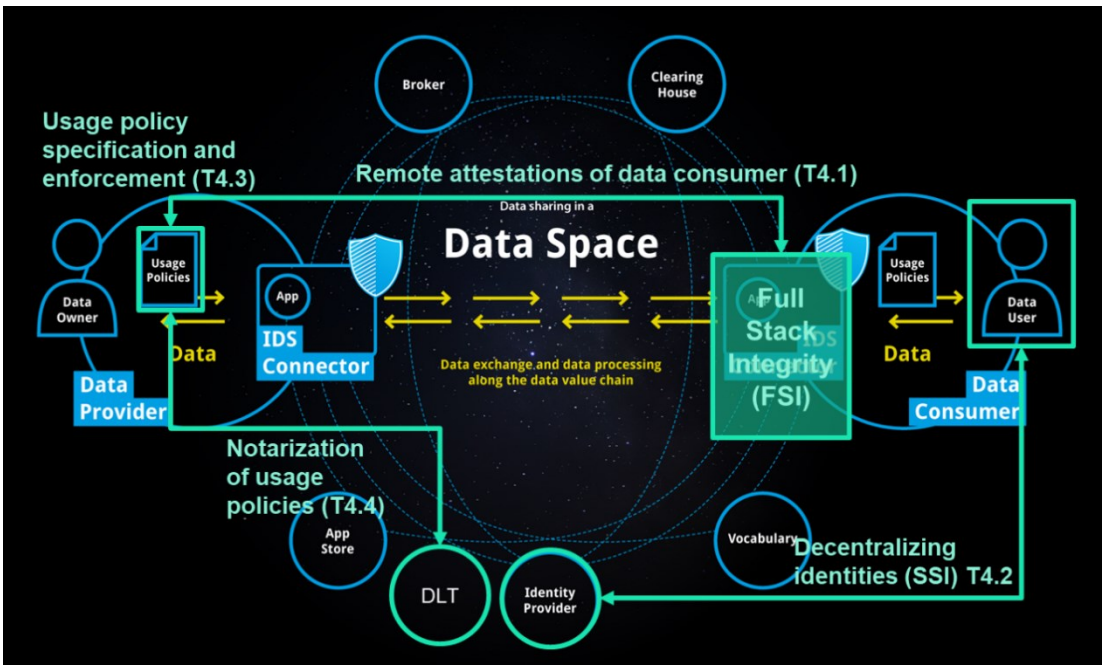


Figure 1: Extended trust and sovereignty functionalities

2.2 From OPEN DEI to DSSC building block taxonomy

The OPEN DEI building blocks have served as a reference for dataspaces over the past years. Currently, the Data Spaces Support Centre (DSSC) has taken on a more holistic and adaptable approach to the Data Spaces building blocks, resulting in the DSSC Blueprint version 0.5[1]. It includes (1) the involvement of a wider range of stakeholders, such as SMEs, startups, and public administrations in the DSSC's Blueprint to create a more robust and diverse data ecosystem; (2) a strong emphasis on interoperability between different dataspaces, leveraging insights from various Common Service Areas (CSAs) to facilitate smoother data exchange and integration across sectors, enhancing the overall utility of the dataspace; (3) an inherently higher focus on innovation and scalability, drawing from a variety of CSAs, each bringing unique insights and advancements; and finally, (4) the DSCC's governance model is more comprehensive, addressing issues like data sovereignty, privacy, and security in more depth.

Based on the above, the Enershare project has moved towards adoption of the DSSC building blocks. A mapping of the Enershare components to both the OPEN DEI building blocks and DSSC

[1] https://dssc.eu/space/BPE/179175433/Data+Spaces+Blueprint+%7C+Version+0.5+%7C+September+2023

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

building blocks was described in Enershare deliverable D8.2 – "Enershare Data Space (1st Technology Release)".

For the building blocks related to trust and sovereignty the building block taxonomy didn't change. The related DSSC building blocks are:

1. Access & usage policies and control
2. Identity Management
3. Trust

2.3 Report on alpha version deployment (MVP-1)

The first release of the Enershare dataspace (MVP-1) was described in Enershare deliverable D8.2 – "Enershare Data Space (1st Technology Release)". The provided alpha version components, Identity provider and Connector implementation, as described in D4.1 are part of the core of this dataspace infrastructure.

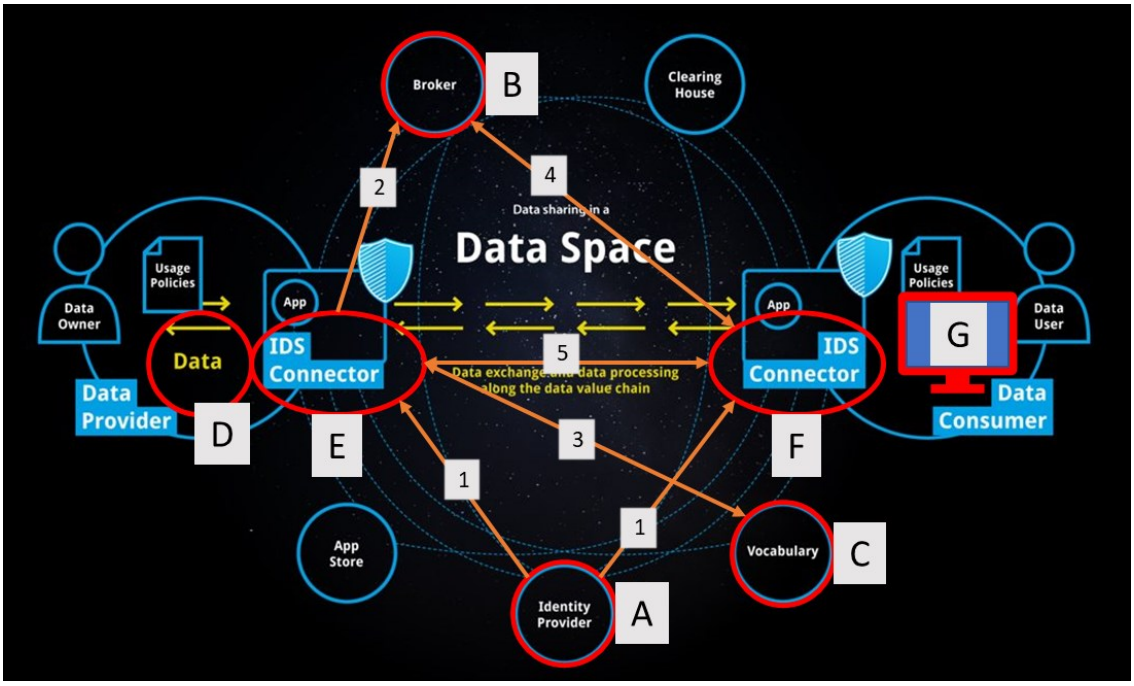


Figure 2: Components and processes included in MVP 1

Figure 2 shows the implemented components and processes of MVP-1. The components, marked in red, included the following:

A. A: Identity Provider
B. B: Metadata Broker
C. Vocabulary Hub

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

D. Data Set and Data Service
E. Connector of a Data Provider
F. Connector of a Data Consumer
G. User Interface (UI) to show Data Service response

All the core processes were implemented in MVP-1 (Figure 2, orange arrows), this includes:

1. Onboarding and identity provisioning
2. Data offering
3. Schema wizard to design API
4. Discovery of data services
5. Data exchange

Please see Enershare deliverable D8.2 - “Enershare Data Space (1st Technology Release)” for more details.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3 Dataspace protocol

In the past year, the development of the Dataspace Protocol converged to a stable version. This protocol will supersede the current IDS Communication Protocol in the near future. It is, therefore, important from the Enershare perspective to follow these developments and prepare for a move towards this protocol.

3.1 Introducing the Dataspace Protocol

"The Dataspace Protocol is a set of specifications designed to facilitate interoperable data sharing between entities governed by usage control and based on Web technologies. These specifications define the schemas and protocols required for entities to publish data, negotiate usage agreements, and access data as part of a federation of technical systems termed a dataspace."[2]

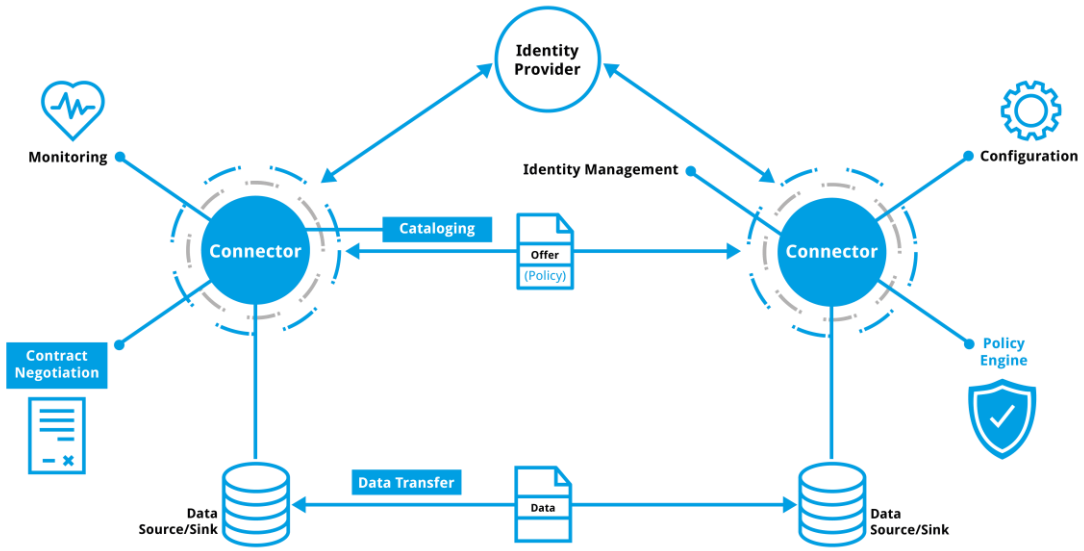


Figure 3: Overview of the dataspace protocol and its context (source: IDSA)

The Dataspace Protocol is formed around the fundamental concept of the differentiation between the control plane and the data plane. In this concept, the control plane assumes a crucial role in tasks such as cataloging, facilitating contract negotiations, and initiating as well as overseeing data transfers. On the other hand, the data plane is specifically dedicated to the

[2] https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol/overview/readme#abstract

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

execution of the data transfer processes. This division of responsibilities ensures an efficient management of the protocol, with the control plane handling the higher-level aspects of coordination and the data plane focusing on the actual implementation of data transfers.

The three aspects (cataloging, contract negotiation, and transfer process) have their own protocol and initial HTTP binding. The catalog protocol handles the metadata exchange of connectors like the Self-Description exchanges in the IDS Communication Protocol. However, in the Dataspace Protocol the catalogues are as close to DCAT[3] as they can be, without redefining concepts that already exist within DCAT. The advantage of this approach is that extensions to the description of assets and resources can be handled by modelling them in DCAT without requirement to synchronize this with the IDS Communication Protocol or Dataspace Protocol. The catalog protocol will be used within Enershare, especially for interactions with the Metadata Broker (terminology might need to be updated to better suit the new protocol), since the interactions with metadata brokers will follow the same catalog protocol. The negotiation protocol is very similar to the negotiation sequences in the IDS Communication Protocol, although similar to the catalog protocol the negotiation protocol uses the generic variant of ODRL[4]. The third aspect, the transfer process protocol, is a major change to the IDS Communication Protocol. This strongly relates to the distinction of control and data plane, in fact it creates the bridge between the control and data planes.

3.2 Adopting the Dataspace protocol

In D4.1 we selected TSG and TRUE as possible connectors (because of their implementation of the IDSA RAM). For MVP version 1 we deployed TSG related components. In MVP version 1.5 a separate dataspace infrastructure will also be deployed for TRUE connector based integrations. These two dataspace infrastructures are not (fully) interoperable with each other. This is why we need to continue the connector developments and adopt the Dataspace Protocol. Where needed, data services that are now provided with the current versions of the connectors need to be migrated to connector versions that implement with the Dataspace Protocol.

The risks of not migration during the duration of the Enershare project are:

- creating a dataspace that is not interoperable with other dataspaces within Europe after the project ends.
- support for existing technical solutions is expected to end, since connector providers will likely adopt the Dataspace Protocol.

[3] https://www.w3.org/TR/vocab-dcat-3/

[4] https://www.w3.org/TR/odrl-model/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

The impact of a possible migration towards Dataspace Protocol compliant connectors is expected to be minimal for participants in the Enershare dataspace. The usage of the OpenAPI data app in the first MVP can be adopted to allow a similar interaction from the perspective of backend systems, which allows an easy migration path. For participants involved in the infrastructure of the dataspace itself, the impact would be larger, and a more detailed migration plan is needed.

Similar to Figure 1, we can plot the trust and sovereignty functionalities as covered in WP4 and this report to the Dataspace Protocol overview. This is depicted in the figure below.

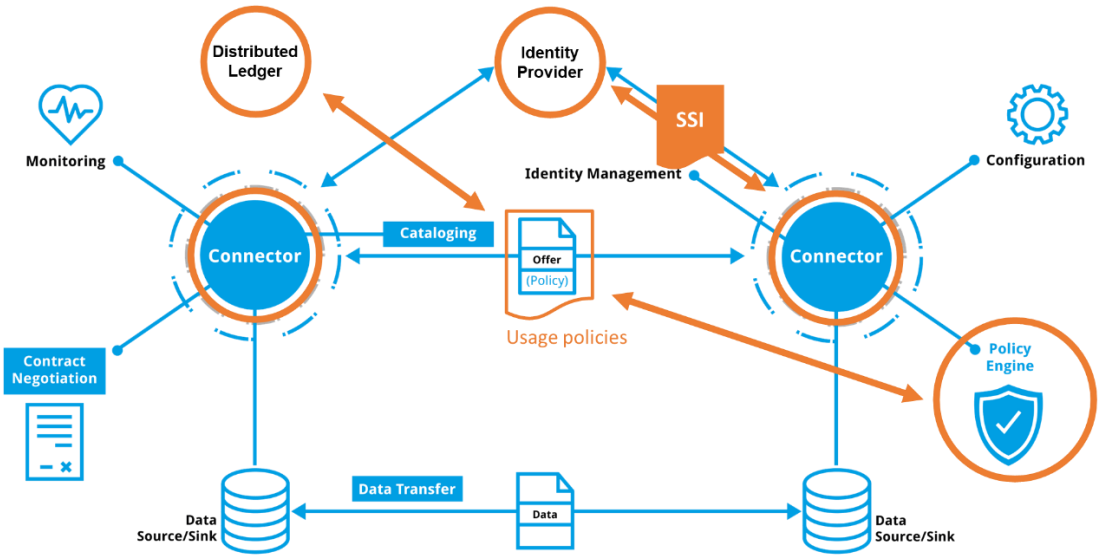


Figure 4: Trust and sovereignty functionalities plotted on Dataspace Protocol

3.3 Evaluation plan

3.3.1 Intra dataspace interoperability

The current TRUE and TSG connectors are not interoperable and require separate dataspace infrastructures as visualized on the left hand side in Figure 5. By adopting the Dataspace Protocol, we can demonstrate interoperability within the Enershare dataspace. Different components, like the identity provider, metadata broker and connectors, from different vendors or open-source initiatives can be combined to provide the required functionalities in a single dataspace infrastructure. This is visualized on the right hand side in Figure 5.

An updated version of TSG connector will come available in Q2. This implements the Dataspace Protocol. We can expect other implementations of the Dataspace Protocol as well, like the Eclipse Dataspace Components (EDC). This is outside the scope of our project.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

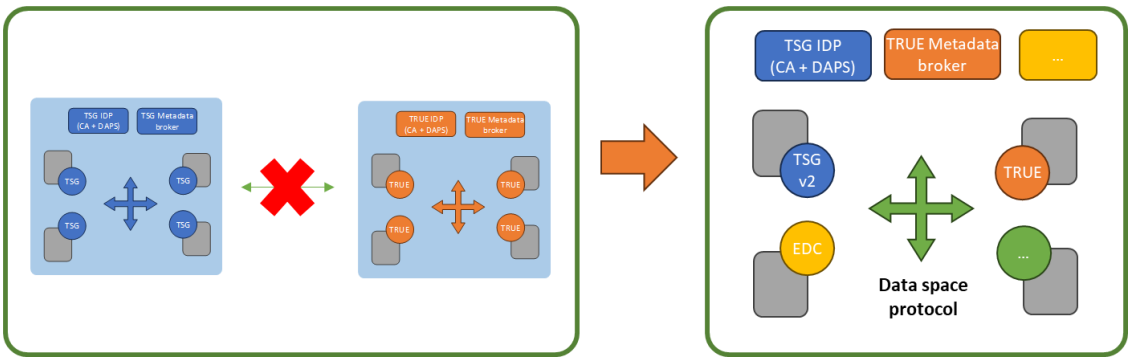


Figure 5: Towards intra dataspace interoperability

3.3.2 Inter dataspace interoperability

The work on interoperability in the system use cases as described in the CEEDS Blueprint of the Int:net project is relevant for our project as the interoperability will likely need to be demonstrated between Omega-X, Data Cellar and Enershare as depicted in the figure below. The projects EDDIE and Synergies are working on a different approach in identity management. Omega-X and Data Cellar are working based on the Eclipse Dataspace Components (EDC) so interoperability with the TSG and possibly TRUE infrastructure would be a very valuable demonstration. This needs to be explored in the upcoming period and reported on in the next deliverable (D4.3).

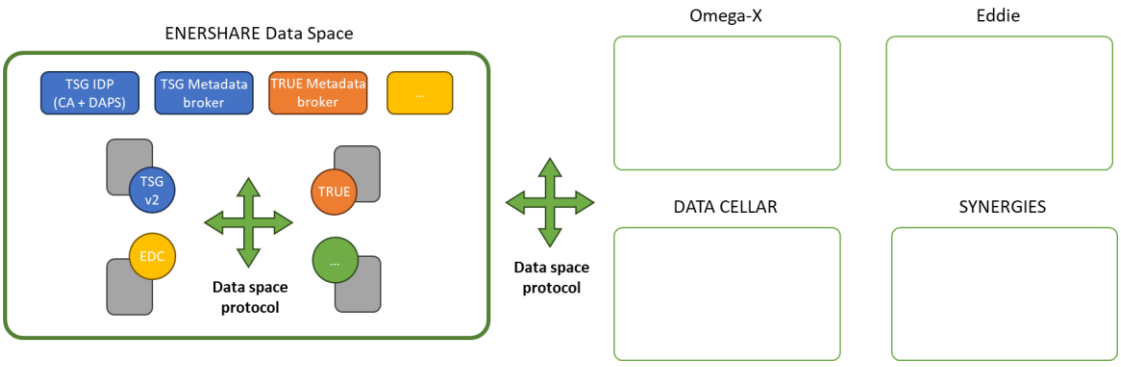


Figure 6: Towards inter dataspace interoperability

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

4 Introducing decentralized identities

4.1 Functional/logical view

The introduction of decentralized identities in favor of centralized identities goes hand-in-hand with the movement towards the Dataspace Protocol. In Figure 7 an overview of the related dataspace protocols is shown, where for the decentralized identities the important protocols are:

- **DID Resolvement Protocols** (blue): protocols used to retrieve relevant information of decentralized identities, including key material and services assisting the usage of decentralized identities.
- **Credential Issuance Protocols** (purple): protocols used to ensure issuers can issue verifiable credentials to holders and provide these credentials to the holders so that they can use these inside their own domain.
- **Verifiable Presentation Protocols** (orange): protocols used to present verifiable credentials to participants in the dataspace.

These protocols will form the basis to allow participants in a dataspace to receive the right credentials and use them in a proper way in the communication with other participants. It is important to ensure interoperability with projects and initiatives outside of the Enershare project. Since a lot of the related projects and initiatives are currently working on solutions around decentralized identities, deciding on the right set of protocols will be challenging. Therefore, keeping up to date with all developments and gathering new insights is key. This might result in changing the chosen protocols in the future to be aligned and to increase the chances of adoption of the Enershare results after the project.

These protocols are fundamentally different from the existing Identity Provider protocols based on public key infrastructure (PKI) and the Dynamic Attribute Provisioning Service (DAPS). The role of the Certificate Authority (CA), based on X.509, will move towards the role of an Issuer of verifiable credentials. And given the fact that the primary reason for the DAPS to exist is to provide a more dynamic way of attaching attributes to an identity, its role within a dataspace based around decentralized identities becomes largely obsolete. As this role can be taken over by creating multiple verifiable credentials, with varying lifetimes and revocation structures, that are linked to the same decentralized identity.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

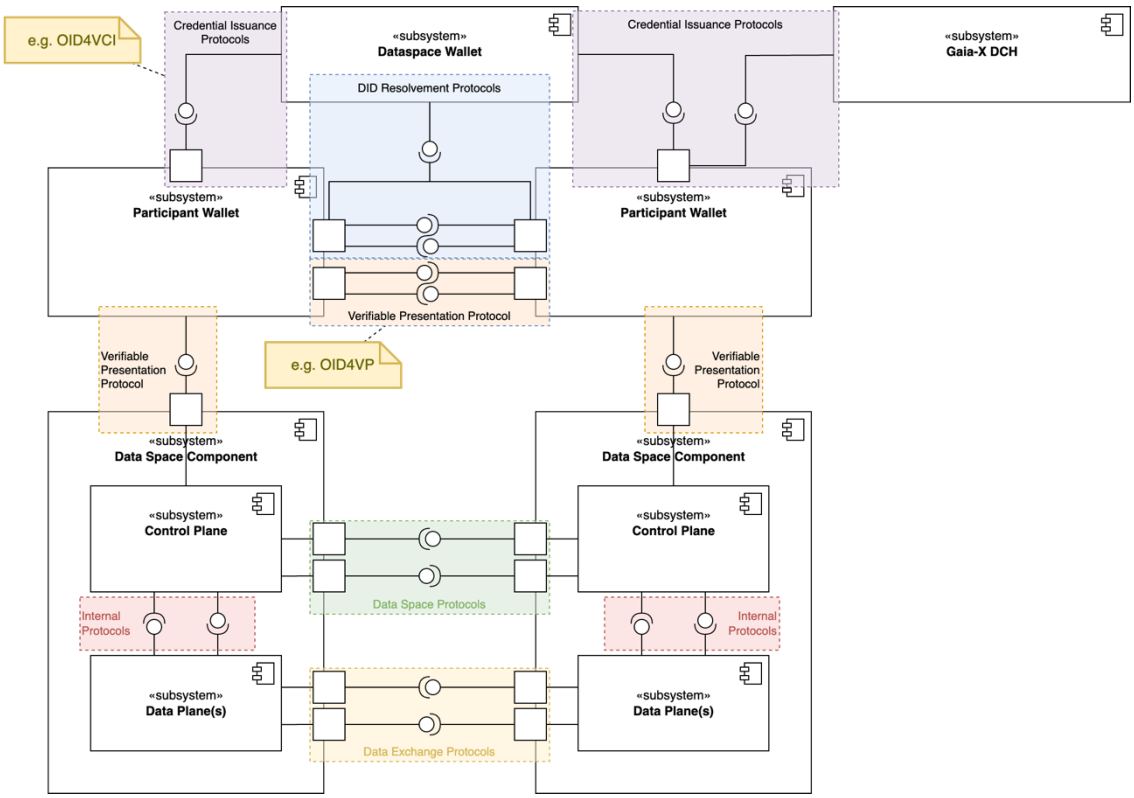


Figure 7: Overview of required protocols to support decentralized identities within dataspaces.

4.2 Technical design

The technical design for decentralized identities follows the protocols identified in the previous section. Starting with the DID Resolvement protocol, for which the initial supported DID method will be the Web DID method[5]. This method is chosen as initial DID method due to the simple nature of the method and the minimal implementation effort required to create the first proof of concepts. In a later stage, additional DID methods may be introduced that provide different characteristics. For example, blockchain-based DID methods like the Sovrin DID method[6] or the Indy DID method[7] might be considered due to their immutable ledger-based verifiable data registry.

4.2.1 DID method

The Web DID method relies on existing web domain reputations, based on DNS. The basis is that DID documents are hosted on a well-known endpoint, which can be resolved based on the

[5] https://w3c-ccg.github.io/did-method-web/

[6] https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html

[7] https://hyperledger.github.io/indy-did-method/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

DID. Examples of this resolving are shown in Table 1, where the process can be summarized to splitting the DID at colons omitting the first two parts ("did" and "web") the first part must be translated to the domain name and optional additional parts should be translated to path segments. For DIDs without path segments, the ".well-known" path should be used.

Table 1: DID Web Resolver

DID	DID Document location
`did:web:w3c-ccg.github.io`	https://w3c-ccg.github.io/.well-known/did.json
`did:web:w3c-ccg.github.io:user:alice`	https://w3c-ccg.github.io/user/alice/did.json

4.2.2 Credential Issuance protocol

Secondly, for the Credential Issuance protocol the initial choice has been made to focus on the OpenID for Verifiable Credential Issuance (OID4VCI)[8] specification. This specification builds upon OpenID and OAuth 2.0 and allows for several ways of requests and offers of Verifiable Credentials. Within Enershare, it is expected that the following flows will be supported:

- Requesting credentials: the "Authorization Code Flow" together with the Deferred Credential Endpoint flows. Since manual intervention at the credential issuer side is expected, additional details and verification might be required. It is expected that a user is involved in the process of requesting credentials at both the issuer and the requestor/holder.
- Offering credentials: used by issuers to issue new credentials in the case of key rotation or renewing credentials. The Credential Offer flows will be used, that will be handled without user interaction.

4.2.3 Verifiable Presentation protocol

Lastly, for the Verifiable Presentation protocol the initial idea is to leverage the OpenID for Verifiable Presentation (OID4VP)[9] specification. However, the suitability of this specification within dataspaces must be explored further. Since OID4VP has a focus on including human interaction at the verifier-side of the interactions, the applicability of this specification might not fully align with the use cases within dataspaces. In, for example, a traditional request-reply flow between two connectors it might be undesirable to require a user at the data provider

[8] https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html

[9] https://openid.net/specs/openid-4-verifiable-presentations-1_0.html

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

side. A solution to this might be to create a profile of OID4VP that allows for the verification of verifiable credentials without human intervention.

4.3 Evaluation plan

Given the move towards decentralized identities is only recently introduced, implementations within Enershare are still on-going.

The introduction of decentralized identities are part of next versions of the MVP. Initially, this new functionality will be placed next to the existing MVP infrastructure, allowing partners and pilots to test against the new workings.

In deliverable D4.3, we will report on the usage of the decentralized identities.

Furthermore, to establish a cohesive and interoperable approach within the Energy Data Space project cluster (int:net)[10], certain agreements need to be reached regarding software building blocks, APIs, and interoperability. Through discussions within the cluster, the sister projects have identified a fundamental subset of building blocks required for demonstrating interoperability. As a result, projects are expected to align their respective technologies to facilitate cross-project testing, ensuring uniformity and compatibility across the initiatives.

As an initial step toward achieving the objective of inter-dataspace interoperability, three pivotal system use cases (SUCs) have been defined:

- SUC1 Onboarding: process to generate, and check credentials to access an ecosystem
- SUC2 Data/service metadata interaction with the catalogue: pushing new metadata into the catalog and discover it)
- SUC3 Contracting: selecting a dataset and/or service to purchase it.

The introduction of decentralized identities strongly relates to SUC1. Onboarding describes the process to allow different companies/people/groups to participate in the different dataspaces with the maximum guaranties of identity, security and privacy. The most crucial aspect of the onboarding process is the type of identities that the dataspaces will be able to handle. The sister projects have agreed on using decentralized identities. Over the upcoming months, a comprehensive approach will be formulated, and the existing gaps will be addressed.

[10] https://intnet.eu/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5 Usage control enforcement

The chapter focuses on the implementation of usage control enforcement in dataspaces, building upon the developments mentioned in Chapter 3. The chapter is structured to detail the work done on TRUE connector and TSG connector within the dataspace, illustrating how they contribute to the overall enforcement of usage control. It begins by outlining the various components and their roles in the dataspace architecture which are based on standards defined by IDSA.

5.1 Introduction

Usage Control Policies allow to define the actions that the provider lets the consumer perform over the data consumed. These policies must be agreed upon between the provider and the consumer when they establish the contract agreement. Currently, these policies are written in ODRL language, which is a standard pattern defined by IDSA, and which is followed in the majority of developed data connectors.

Creating effective policies for data usage is a critical aspect of managing information utilization. Each participant in the dataspace must define their specific data usage control policies, even if they have diverse technical backgrounds. Data providers can choose a template defined by specific data connector, input the necessary details, and generate the relevant policy. Afterwards, they can employ a data usage control application to enforce the policy within the system, ensuring the protection of their data. To enhance the policy specification process, these policies are categorized into distinct classes, each corresponding to a specific template. Beyond simplifying policy creation, classifying these policies provides an additional advantage by allowing users to evaluate the scope of coverage offered by their data usage control policies.

The Data usage control policy is a set of rules that authorizes or restricts specific actions on a data asset. It may also require the execution of certain actions under defined circumstances. These actions can range from basic operations like displaying and printing to more fine-grained actions. For example, a policy may allow data reading without constraints but explicitly forbid the printing of data. Data usage control applications offer options such as whitelisting, blacklisting, or a combination to protect data, enabling the creation of policies that either permit or restrict data usage.

However, conflicts can arise between these policies, necessitating a strategy for conflict detection and resolution. For instance, conflicting policies may permit data usage but restrict the printing of data, highlighting the need for effective conflict resolution methods. In the International Data Space, conflict resolution approaches may consistently opt to prohibit

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

actions in case of conflicts. Nevertheless, the precise definitions of data usage actions and the methodologies for conflict detection and resolution are still evolving in the context of dataspaces. The data usage policies are categorized into 14 classes, as outlined previously in D 4.1.

5.2 Functional/logical view

5.2.1 Usage control components in TRUE connector

The figure below depicts the components for utilizing the existing usage control technologies and implementing them using the connectors in dataspaces. The usage control component in TRUE connector provides significant support in implementing this work. The Data App provides the implementation guidelines to map our use cases and fetch the required data from the source i.e. APIs, databases etc. The Usage Control app is used for verifying contract negotiations and verifies the usage policies. Once the usage policies are agreed the resource information is provided to the Data app interface which will return the required data.

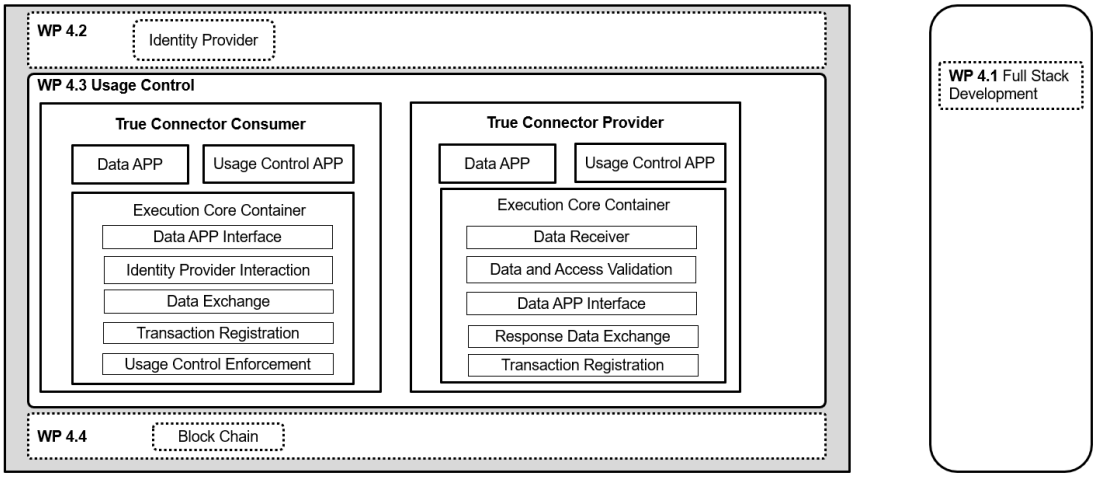


Figure 8: Subcomponents of the Trust & Sovereignty

5.2.2 Usage control components of TSG Connector

The Policy Enforcement Framework extends the data-flow model established by the XACML (Extensible Access Control Markup Language) standard, incorporating the following concepts and terminology [11]:

[11] Policy Enforcement Framework - TNO Security Gateway Documentation (tno-tsg.gitlab.io)

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Policy Enforcement Point (PEP): This component intercepts data requests and responses, carrying out access control to ensure that, in accordance with a decision, the request/response is either allowed or denied passage.

Policy Decision Point (PDP): This entity assesses a relevant policy and decides based on the context received from the PEP.

Policy Administration Point (PAP): This component administers policies within the PEF scope, likely offering CRUD-based (create, read, update & delete) support for administrators and enabling the PDP to request policies.

Policy Information Point (PIP): This entity supplies additional information that aids in the evaluation of policies.

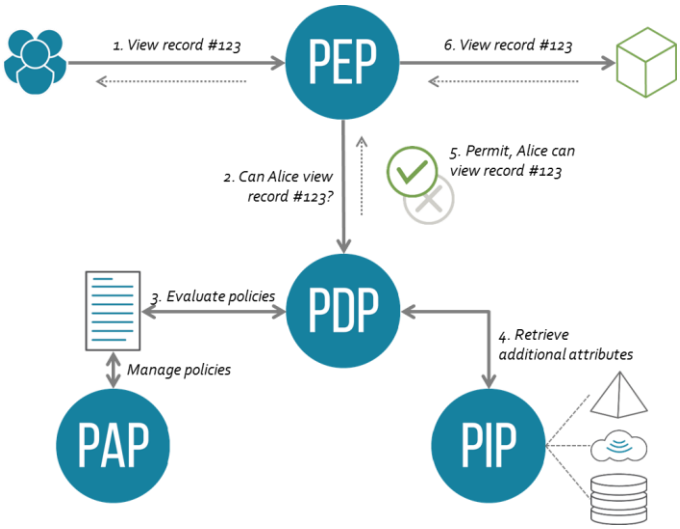


Figure 9: An overview of the interaction between components[12]

5.2.3 Pilot policy requirement

The policy pattern required from pilot 1 and pilot 2 are listed which has been discussed already in the previous deliverable and further pilot requirements will be added in future releases.

Table 2: Policy patterns from pilot requirements

No	Policy Pattern	Pilot 1	Pilot 2
1	Connector-restricted Data Usage	x	x
2	Interval-restricted Data Usage	x	x

[12] https://en.wikipedia.org/wiki/XACML

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

3	Purpose-restricted Data Usage Policy	x	x
4	Security Level Restricted Policy		x
5	Modify Data (in Rest)		x
6	Local Logging		x
7	Remote Notifications		x

5.2.4 Contribution for pilots

The policy requirements gathered from pilots as provided in the table above should be supported from the connector which is being used in Enershare project. The connectors which are currently chosen in the Enershare project are TSG connector and TRUE connector. In task 4.3, the Interval-restricted data usage and Purpose-restricted data usage policies were supported. We developed a policy implementation for Connector-restricted data usage and Security Level Restricted for TRUE connector. These new policies are developed in accordance with IDSA standards and now integrated in TRUE connector are available. For the final technology release the remaining required usage policies are implemented to fulfill the pilot requirements.

5.3 Technical Design

The overview of the technical components of the usage control policies and enforcement is discussed in the above sections. In the next part, the details of development which are to be implemented and would need to be extended to meet project requirements based on use-cases. The data consumer's connector sends a Contract Request to the Data Provider, which may either adopt the Contract Offer or propose modifications. Upon receiving the Contract Request, the Data Provider's IDS Connector validates its syntax, content, and signature. If deemed acceptable, a Contract Agreement is signed and confirmed with the Data Consumer. This agreement is then instantiated and deployed within both IDS Connectors for future reference in Policy Enforcement. However, if there's disagreement at any stage, the Contract can be rejected, leading to termination of the sequence and notification to relevant parties. It's important to note that a rejected negotiation sequence cannot be restarted, but a new one can be initiated at any time.

5.3.1 Use case for Enershare pilot 1

The components and technical details of the work which was done using the TRUE connector has been discussed. For use-case of Pilot 1 we proposed an architecture shown in figure 10 which utilizes the data service from Technalia. The use case for wind farm data is depicted in Figure 10. The steps of the use case are as following:

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

- Step 1: Utility Company request wind farm data from Service Provider.
- Step 2: Contract request is made to provider connector.
- Step 3: Usage policies defined in the contract are checked to the standard and constraints. Then these policies are saved in the H2 (in-memory) database in connector.
- Step 4: The provider connector then checks the requested resource in this case wind farm data.
- Step 5: The provider connector then provides an update to ECC (Execution Core Container) which then provides the information to Data App in provider connector.
- Step 6: Contract is accepted, and usage control policies are enforced.
- Step 7: The requested resource artifact is shared with the consumer connector.
- Step 8: The consumer connector then requested the provided artifact and data is provided.

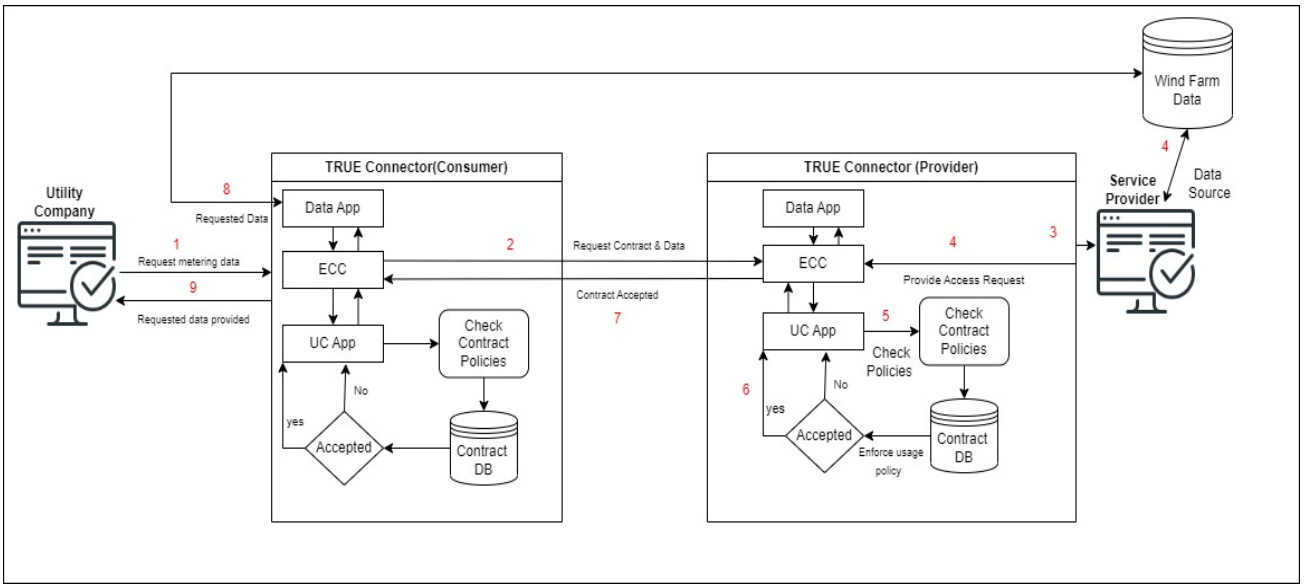


Figure 10: Pilot 1 Wind Farm Data Use Case

5.3.2 Communication perspective in TRUE connector

In figure 11, it shows technical overview of components communication for TRUE connector. As discussed previously, TRUE connectors are built upon three main components Basic Data App, Execution Core Container and Usage Control Data App. For communication among them interfaces are developed for connection. For the consumer and provider roles of TRUE connector the configuration parameters are set in property files. After the properties are set then the connectors are available on specified URL or IP addresses. The details of all the

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

components are available and how to configure the properties according to requirements and more configuration details are present on the website [13].

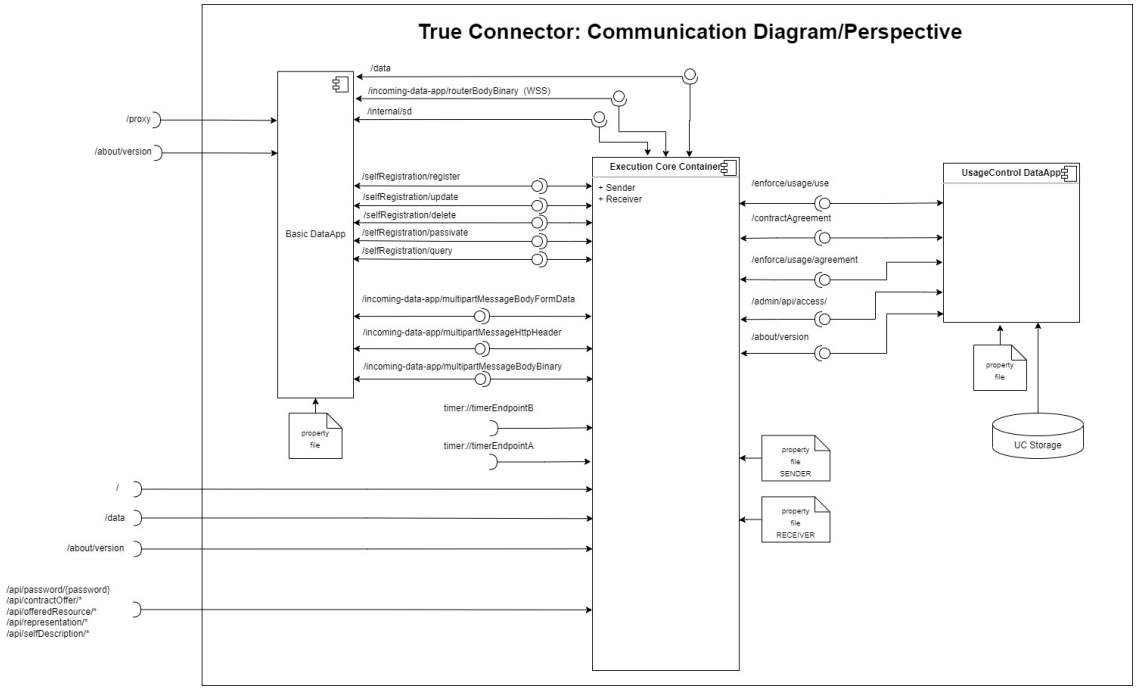


Figure 11: Communication diagram TRUE connector[13]

5.3.3 Sequence diagram in TRUE connector

The sequence diagram shown in the figure below shows the steps performed from consumer and provider using TRUE connector. In step 1 of the sequence diagram, identification of consumer and provider is verified using DAPS Server. In step 2 of the sequence diagram, contract offerings from the consumer connector are exchanged with the user. In step 3 of the sequence diagram, the contract negotiation is performed between the consumer and provider connector and then after the verification of usage contract and policies data is provided to the consumer.

[13] Engineering-Research-and-Development/true-connector: TRUE (TRUsted Engineering) Connector for the IDS (International Data Space) ecosystem (github.com)

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

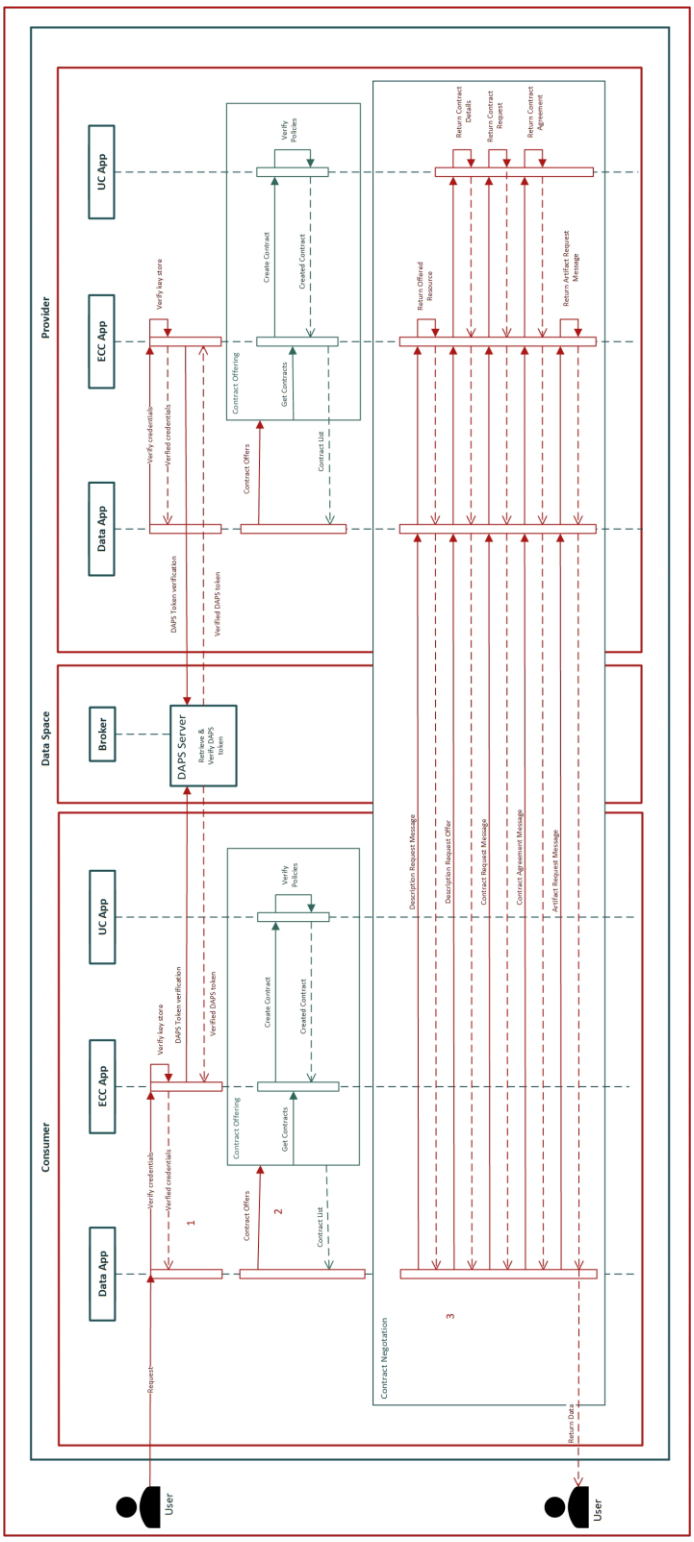


Figure 12: Sequence diagram TRUE connector

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5.3.4 Sequence diagram policy negotiation TSG Connector

The embedded Policy Enforcement Framework (PEF) streamlines the automated negotiation of policies, as depicted in the figure below. This empowers consumers to send an *ids:ContractRequest*, originating from an *ids:ContractOffer*, accessible through either the Metadata Broker or the Provider's Self-Description, for example. The rules used for automatic negotiation are as following:

- Properties such as *permission*, *prohibition*, and *duty* must have the same length.
- For each rule, the *assignee* must be an exact match.
- For each rule, the *assigner* must be an exact match, or if no assigner is provided in the *ids:ContractOffer*, the *ids:ContractRequest* may supply an assigner.
- For each rule, the *target* must be an exact match.
- For each rule, the *constraint* must be an exact match, or if no constraint is provided in the *ids:ContractOffer*, the *ids:ContractRequest* may supply a constraint.
- For each rule, the *action* must be an exact match, or if no action is provided in the *ids:ContractOffer*, the *ids:ContractRequest* may supply an action.
- For each rule, the *assetRefinement* must be an exact match.
- For each rule, the *preDuty* must match in the same way as *permission*, *prohibition*, and *duty* are allowed to match.
- For each rule, the *postDuty* must match in the same way as *permission*, *prohibition*, and *duty* are allowed to match.

Currently, the automated policy negotiation does not support prompting an Administrator in the User Interface to decide whether an *ids:ContractRequest* should be converted into an *ids:ContractAgreement*.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

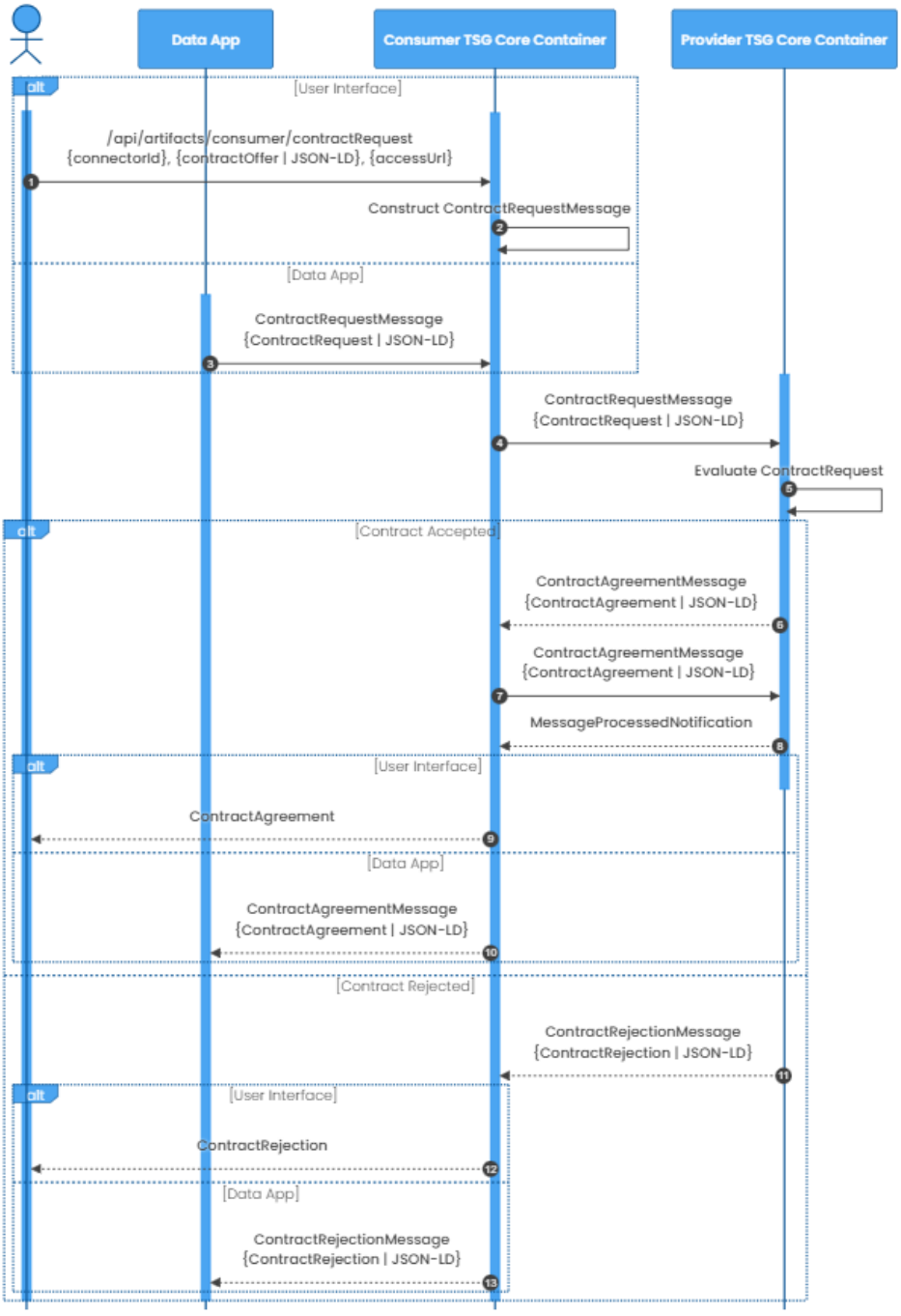


Figure 13: Policy negotiation sequence diagram of TSG connector[14]

[14] https://tno-tsg.gitlab.io/docs/communication/message-flows/#policy-negotiation

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

5.3.5 Key findings for connectors

Initially the project was planned to provide common infrastructure for Enershare project which would support both TRUE connector and TSG as both are developed according to IDS-RAM 4 architecture standard from IDSA. Currently, TSG connector is being used in the project due to which the usage policies which were developed for TRUE connectors needs to incorporate into TSG connector. Details of both the connectors has been provided in the previous section which would be helpful in development of interoperability between them, and the new dataspace protocol would also play a vital role in achieving these results. The deliverable D 2.3 would further reflect on the results for TSG connector and evaluation of results accordingly.

5.4 Current developments and deployments in Enershare

The component overview and contract negotiation process has been explained in detail for TRUE connector and TSG connector. The initial test data which was provided by Pilot 1 was accessible via a REST API. Using the provided data, the TRUE connector was tested on the above discussed use-case.

In the section 5.2.3 the usage policy required by the pilot has been discussed. Currently, two new policy type implementations were developed for the TRUE Connector. They are listed below.

5.4.1 Connector-restricted data usage policy

The Connector-restricted Data Usage policy restricts the usage of the data to a specific connector of a specific Data Consumer if the Data Consumer owns one or more Connector(s). The policy can be expressed in ODRL like so:

```
{
   "@context": [
      "http://www.w3.org/ns/odrl.jsonld",
      {
         "dc": "http://purl.org/dc/terms/",
         "ids": "https://w3id.org/idsa/core/",
         "idsc": "https://w3id.org/idsa/code/"
      }
   ],
   "@type": "Agreement",
   "uid": "http://example.com/policy/restrict-connector/312",
   "profile": "http://www.w3.org/ns/odrl/2/core",
   "dc:creator": "Data Provider 123",
```

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
   "dc:description": "The Connector-restricted Data Usage policy restricts the
usage of the data to a specific IDS connector of a specific Data Consumer assuming
that the Data Consumer owns one or more IDS Connector(s).",
   "dc:issued": "2022-05-19T12:00",
   "ids:provider": "http://example.com/ids/party/123",
   "ids:consumer": "http://example.com/ids/party/456",
   "permission": [
      {
         "target": "http://example.com/ids/data/789",
         "assigner": "http://example.com/ids/party/123",
         "assignee": "http://example.com/ids/party/456",
         "action": [
            "derive",
            "display"
         ],
         "constraint": [
            {
               "leftOperand": "idsc:CONNECTOR",
               "operator": "isAnyOf",
               "rightOperand": [
                  {
                     "@value": "?connector1URI",
                     "@type": "xsd:anyURI"
                  },
                  {
                     "@value": "?connector2URI",
                     "@type": "xsd:anyURI"
                  }
               ],
               "ids:pipEndpoint": [
                  {
                     "@type": "ids:PIP",
                     "ids:interfaceDescription": {
                        "@value": "?interfaceURI",
                        "@type": "xsd:anyURI"
                     },
                     "ids:endpointURI": {
                        "@value": "?endPointURI",
                        "@type": "xsd:anyURI"
                     }
                  }
               ]
            }
         ]
      }
```


Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
    ]
}
```

5.4.2 Security level-restricted data usage policy

The Security Level-restricted Data Usage policy restricts the usage of the data to a Specific IDS Connector when it is certified for a specified security level (i.e. *idsc:BASE_SECURITY_PROFILE*, *idsc:TRUST_SECURITY_PROFILE* and *idsc:TRUST_PLUS_SECURITY_PROFILE*)[15].

The policy can be expressed in ODRL as:

```
{
   "@context": [
      "http://www.w3.org/ns/odrl.jsonld",
      {
         "dc": "http://purl.org/dc/terms/",
         "ids": "https://w3id.org/idsa/core/",
         "idsc": "https://w3id.org/idsa/code/"
      }
   ],
   "@type": "Agreement",
   "uid": "http://example.com/policy/restrict-security-level/12",
   "profile": "http://www.w3.org/ns/odrl/2/core",
   "dc:creator": "Data Provider 123",
   "dc:description": "The Security Level-restricted Data Usage policy restricts
the usage of the data to a Specific IDS Connector when it is certified for a
specified security level (i.e. idsc:BASE_SECURITY_PROFILE,
idsc:TRUST_SECURITY_PROFILE and idsc:TRUST_PLUS_SECURITY_PROFILE).",
   "dc:issued": "2022-05-19T12:00",
   "ids:provider": "http://example.com/ids/party/123",
   "ids:consumer": "http://example.com/ids/party/456",
   "permission": [
      {
         "target": "http://example.com/ids/data/789",
         "assigner": "http://example.com/ids/party/123",
         "assignee": "http://example.com/ids/party/456",
         "action": [
            "derive",
            "display"
         ],
```

[15] https://github.com/International-Data-Spaces-Association/InformationModel/blob/develop/codes/SecurityGuarantee.ttl

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
        "constraint": [
           {
              "leftOperand": "idsc:SECURITY_LEVEL",
              "operator": "isPartOf",
              "rightOperand": [
                 {
                    "@value": "idsc:TRUST_PLUS_SECURITY_PROFILE",
                    "@type": "xsd:string"
                 },
                 {
                    "@value": "idsc:TRUST_SECURITY_PROFILE",
                    "@type": "xsd:string"
                 }
              ],
              "ids:pipEndpoint": [
                 {
                    "@type": "ids:PIP",
                    "ids:interfaceDescription": {
                       "@value": "?interfaceURI",
                       "@type": "xsd:anyURI"
                    },
                    "ids:endpointURI": {
                       "@value": "?endPointURI",
                       "@type": "xsd:anyURI"
                    }
                 }
              ]
           }
        ]
     }
  ]
}
```


5.4.3 Development of data app

The *data app*, which is a core component of the TRUE connector, is further developed according to the data service from the pilots. To integrate different data services provided by project partners and pilots in Enershare project a generalized method must be developed that is according to REST APIs services provided from them. The *data app* is configured accordingly to fulfill the requirement of the project. The *ArtifactRequestMessage* requested module in the *data app* is updated and the parameter payload with additional features would be utilized to map the parameters. The parameters in payload are depicted in Figure 14 and consist of:

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

- **Headers**: It is an array of parameters which are required for the REST APIs request.
- **Method**: It is a parameter which provides information of REST service request method is GET/POST based on the requirement of provided service.
- **TransferContract:** This the contract which was agreed between consumer and provider.
- **Request Artifact:** This is the target artifact or requested URL.
- **MessagType:** In this request we are requesting artificat so *ArtifactRequestMessage* as there is different other message which can see on the documentation of TRUE Connector.
- **Forward-To:** The parameters would forward the request provider connector.
- **Multipart:** There are multiple request parameters which are defined in TRUE Connector.

The updated *data app* is available in Docker Hub[16] and deployed with the TRUE connector docker configuration. Testing of this *data app* is completed internally and with the service provided by the pilot. The requirements of services for other pilots need to be evaluated once they are available.

[16] https://hub.docker.com/repository/docker/fiosbast/ids_be_data_app

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

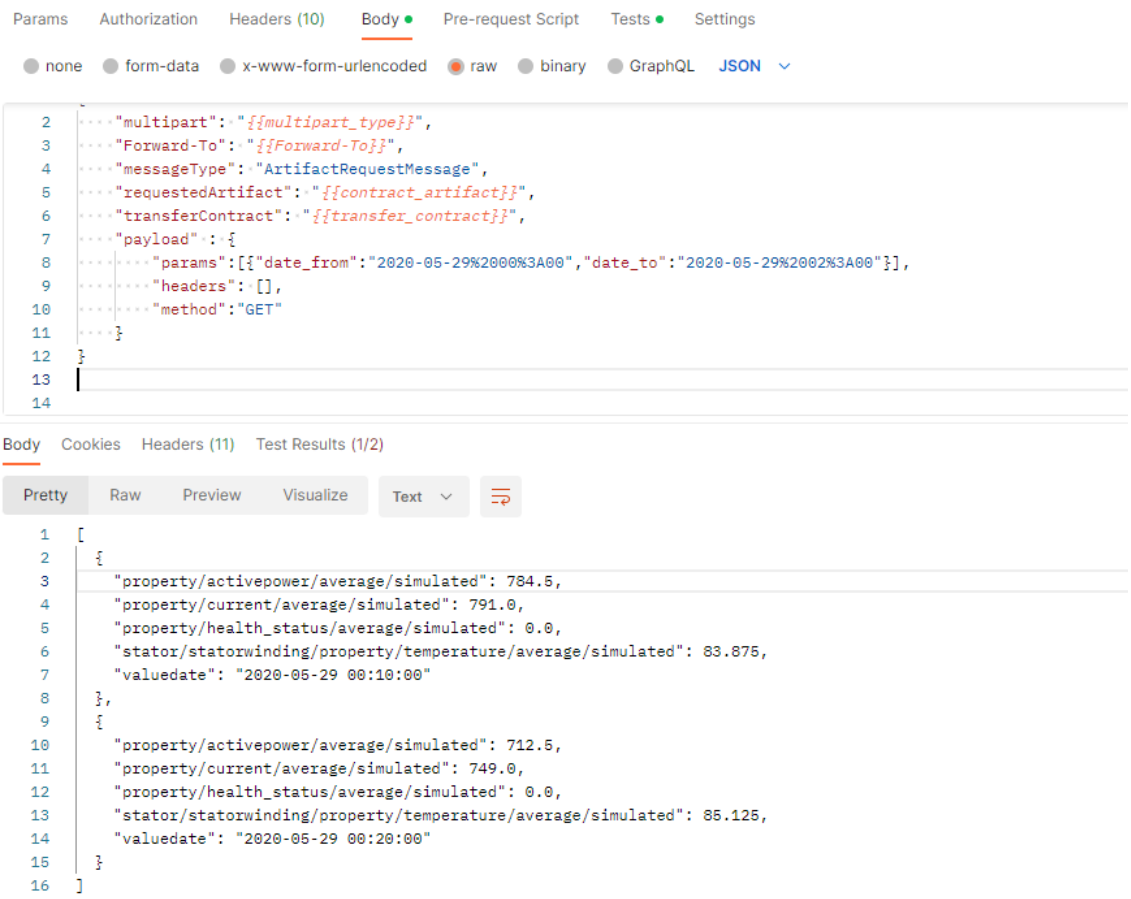


Figure 14: Exchanged data result between two connectors

5.5 Evaluation plan

In the development phase the TRUE connector was being used and deployed for both data consumer and data provider roles. It includes the enforcement of policies as specified as requirements by pilot 1. The planned was to finish the second technology release and deploying these in Enershare dataspace infrastructure and this will be available for project partners and pilots to integrate, use and evaluate.

Given the fact that current dataspace infrastructures for TRUE and TSG are not interoperable, as explained in Chapter 3, the usage policy interoperability cannot be realized between these two connectors. The usage control enforcement implementation, as realized in the TRUE connector and described in this Chapter 5, can be tested

Next step is to also implement the policy enforcement in the TSG connector by the team working on TSG to make it available for pilots using that implementation. Furthermore, the usage control component which was updated for the TRUE connector will be further

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

investigated and integrated in EDC connector and be evaluated considering different use-cases of Fraunhofer IOSB-AST and Enershare project. The usage policies which are worked in currently development can be used in implementations of use cases using OneNet connector as it would be based on TRUE connector architecture.

When the next generation connector(s) have adopted the Dataspace Protocol, the policy specifications can be exchanged between two (or more) connector implementations from different vendors and open-source initiatives. This is because the ODRL language which is used for the policy specifications is part of the Dataspace Protocol specification for contract negotiation.

In the next phase, which was planned considering TRUE connector, the notarization service which provided from WP 4.4 would be integrated and tested accordingly. Now as the TSG is the official connector which is to be used in Enershare but according to initial evaluation this integration would take more time in the TSG connector and would exceed the timeline for the project. In the next deliverable, the details of the development and pilots' requirements which are integrated would be discussed using TSG connector and working examples would be demonstrated in the report for evaluation.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

6 Usage policy notarization on the blockchain

6.1 Functional/logical view

To enforce the usage control with a higher level of security, blockchain and Distributed Ledger (DLT) technology has been adopted, to support the usage of Web 3.0 tools, able to ensure the integrity, immutability and non-repudiation of the data, and particularly the information regarding the usage control policies as described in the previous chapter.

The differences between Web 2.0 and Web 3.0 applications have been already reported in the previous deliverable D4.1, so for further reference on blockchain, DLT and Web 3.0 techniques please refer to that documentation. What is important to recall, is that to work with Web 3.0 applications, *Dapps* (Distributed Applications, i.e., software programs using the blockchain) must be developed using specific protocols (e.g., JSON-RPC[17]), tools (e.g., Metamask[18]) and libraries (e.g., web3.js[19] / web3.py[20] or ethers.js[21] / ethers.py[22]), which allow to interact with Ethereum-based chains using cryptographical wallet directly. Ethereum[23] has been selected for the possibility of implementing and using *smart contracts*, which can be viewed as software programs, written in specific languages (e.g., Solidity[24]) running inside the blockchain. For all these reasons, these approaches require specific expertise and in some cases are difficult to apply and exploit directly. To support the *notarization* (i.e., registration into the blockchain) of usage control policies, an API layer has been specifically designed and developed, to allow interacting with smart contracts using traditional HTTP requests, with standard and well-known Web 2.0 instruments (e.g., REST services).

In particular, in D4.1 the adoption of two different typologies of smart contracts was envisioned. With the advancement of the project, the smart contract which emerged as most promising and has been developed for early demonstration and tests, was the so called *PoE* (Proof of

[17] https://www.jsonrpc.org/specification

[18] https://metamask.io/

[19] https://web3js.org/

[20] https://web3py.readthedocs.io/en/stable/

[21] https://docs.ethers.org/v6/

[22] https://pypi.org/project/ethers/

[23] https://ethereum.org/

[24] https://soliditylang.org/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Existence), whose main functionalities are the notarization of data (i.e., registration into the blockchain) and verification of its *hash* (i.e., a footprint of the data, result of the application of a cryptographic function on it). The notarized data are the definitions of the usage control policy discussed in the previous chapter.

Various versions of the PoE have been implemented and discussed. The API layer already developed was designed for the usage of the first officially released PoE smart contract, i.e., PoE v5. This version provides the basic features and does not take into consideration other requirements emerged with the progress of the work. In this smart contract, data are notarized and can be verified by their hash, without registering clear data into the blockchain. This is particularly indicated in case of sensible information. Moreover, data cannot be unregistered once notarized. In this case, the API layer permits to:

- register data inside the Blockchain invoking the *registerData* method of the PoE smart contract,
- prove data already registered inside the blockchain invoking the *validateData* method of the PoE smart contract ,
- prove a hash of data already registered inside the blockchain invoking the *validateProof* method.

Inside the blockchain technology, there is a distinction between *transactional* and *view* methods. The first type entails the modification of the state of the blockchain (i.e., modification of a wallet balance or a global variable of the smart contract) and require *gas*, i.e. the payment of a fee to the blockchain nodes for its maintenance. The second type are read-only methods, which does not require any payment of gas, since they are just query on the current status of the blockchain. The *validateData* and the *validateProof* are non-transactional view methods and for this reason it is not necessary to use a private key to sign the transaction; on the contrary, the *registerData* is a transactional method, so it is necessary to define a private key inside the API layer to sign the transaction.

The same strategy will apply for the usage of the final version of the PoE smart contract. At writing time, the current version is v7. It differentiates from the v5 basically for the introduction of two methods, *lookup* and a *nullify*, non-transactional and transactional respectively, to implement a wider range of use cases and fulfill the emerging needs coming from task 4.3 for the usage control policy registration and retrieval.

The UML class diagram of the PoE v7 smart contract is depicted in the figure below.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

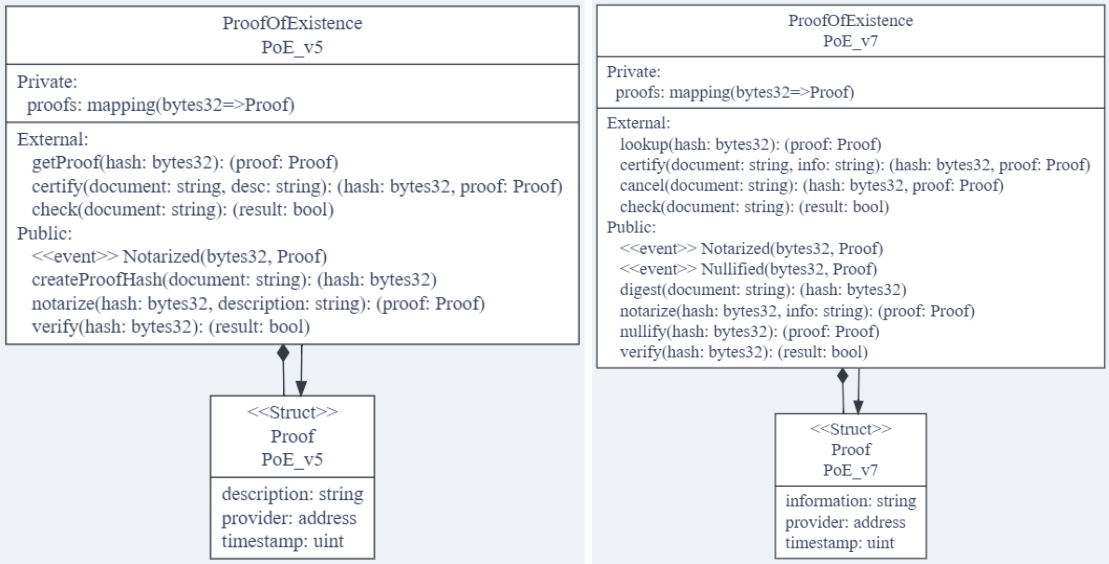


Figure 15 - Smart contract class diagram

In the diagram, the properties and methods directly available from the smart contract are shown. Starting from this design, the API layer will implement similar functionalities of the APIs for the PoE v5 but also the possibility to delete already registered notarization inside the blockchain. This will be crucial for usage policies not yet valid or in use and is a valuable extension of the classic PoE concept. In addition, in the PoE v7 case, the notarization will attach to a proof object that contains information provided by the user, the address of the user and the timestamp when the document was notarized.

The smart contract is part of the general software architecture, which includes the API layer, which is represented as UML component diagram. The architecture envisioned for PoE v7 is depicted in the following figure.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

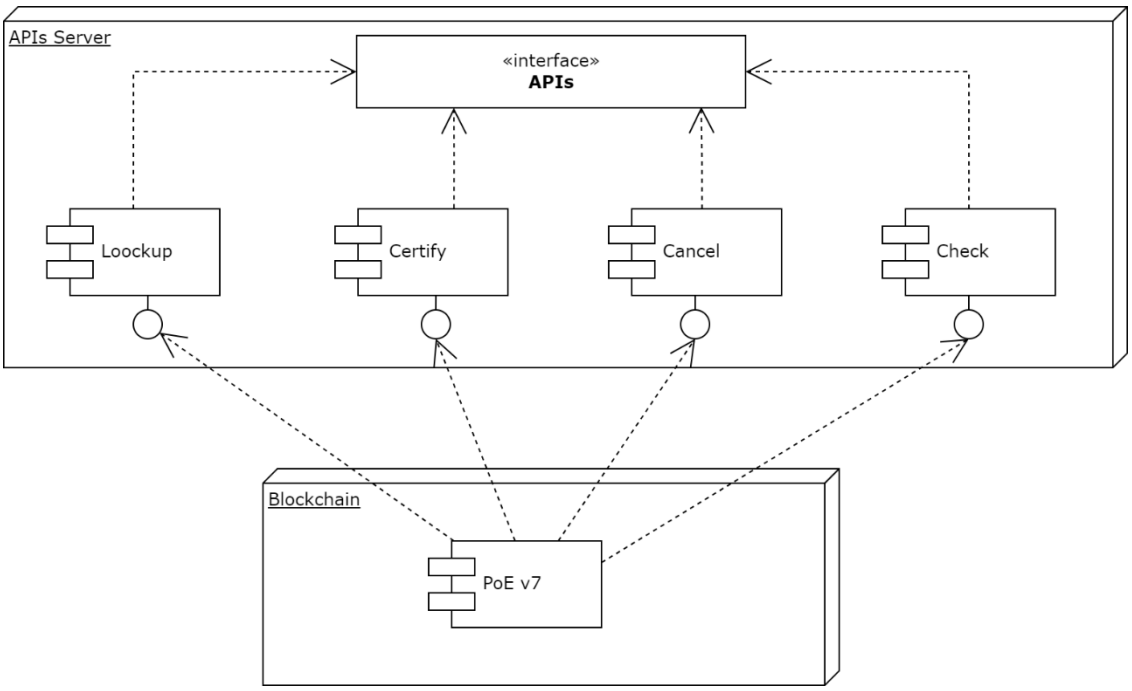


Figure 16 – API Layer component diagram

The diagram depicts the relations between the different components of the Blockchain infrastructure and the API layer and completes the logical/functional view of the proposed solution.

6.2 Technical design

In the previous paragraph, the main functionality of the usage policy notarization was described, along with its specifications, to enforce the security and integrity of the information by means of the usage of DLT/blockchain and smart contracts.

In order to maintain the highest level of generality, the smart contract has been designed to be interoperable and independent on the specific network where it is deployed. The smart contract has been developed in Solidity, an object-oriented, high-level language for implementing smart contracts that govern the behavior and the state of Ethereum-based blockchains.

When dealing with DLT/blockchain technology, the other main technical choice regards the type of blockchain to adopt, and the implementation to exploit. To maintain the objective of generality of the development, three different cases have been studied and tested, as concrete blockchain solutions for the deployment:

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

- A Ganache[25]-based private blockchain;
- A private permissioned Hyperledger Besu[26] installation;
- The Ethereum-based public testnet Sepolia[27].

The choice of where to deploy the contract is then left to the user, since all the three solutions have their own different pros and cons. The Ganache blockchain is simple to install and useful for tests, Hyperledger Besu can be the right choice for private consortium while the Sepolia public blockchain mimics the Ethereum mainnet ensuring that smart contracts also work on that blockchain. Most of these aspects have been already addressed in details in D4.1, whose inspection is suggested as reference to obtain further information about.

The API layer for the interaction with the smart contract was developed using Node.js and Espress.js; this choice was made mainly for the libraries to interact with the blockchain that are written mostly in JavaScript as well as because Node.js is a simple, scalable, speed framework with single-threaded and non-blocking execution of threads. The API server was wrapped inside Docker to permit an easy installation and configuration.

6.3 Current developments and deployments in Enershare

The current implementation of the PoE contract is reported in this section. It is worth noting that an evolution of the contract is in progress, as described in the previous section. The source code is reported as it is since it has been designed and developed to be self-descriptive with comments and easy-to-use for programmers.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

/**
 * @title Proof of Existence
 * @author 0x7efC9d38581c7Cb8372325c349Ec5095D027B3Dd
 * @dev Notarization of documents into the Blockchain through the hash of their
content.
 */
contract ProofOfExistence {

    struct Proof {
        string description; // unencrypted metadata
```

[25] https://github.com/trufflesuite/ganache-ui

[26] https://github.com/hyperledger/besu

[27] https://sepolia.dev/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
        address provider;  // the sender's address
        uint timestamp;   // notarization instant
    }

    /// @dev The contract state
    mapping (bytes32 => Proof) private proofs;

    event Notarized(bytes32, Proof);

      // ================= //
     // Utility functions //
    // ================= //

    /// @dev Calculate the document proof as the SHA-3 hash of its content
    function createProofHash(string memory document) public pure returns (bytes32
hash) {
       hash = keccak256(bytes(document));
    }

    /// @dev Get the document proof data associated to a notarized hash, if any
    function getProof(bytes32 hash) external view returns (Proof memory proof) {
        proof = proofs[hash];
        // return proof;
    }

      // ====================== //
     // Notarization functions //
    // ====================== //

    /// @dev Calculate and notarize the proof of a document
    function certify(string calldata document, string calldata desc) external
returns (bytes32 hash, Proof memory proof) {
        hash = createProofHash(document);
        proof = notarize(hash, desc);
    }

    /// @dev Store the proof of a dacument in the contract state
    function notarize(bytes32 hash, string calldata description) public returns
(Proof memory proof) {
        require(proofs[hash].timestamp == 0 , "Proof already notarized.");
        proof = Proof(description, msg.sender, block.timestamp);
        proofs[hash] = proof;
        emit Notarized(hash, proof);
```

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
    }

      // ====================== //
     // Verification functions //
    // ====================== //

    /// @dev Calculate and verify if the proof of a document has been notarized
    function check(string calldata document) external view returns (bool result) {
        result = verify(createProofHash(document));
    }

    /// @dev Verify if a hash has been notarized as proof of a document
    function verify(bytes32 hash) public view returns (bool result) {
        result = proofs[hash].timestamp != 0;
    }
}
```

The API server for PoE v5 exposes the following three endpoints. The /register endpoint registers data from the request body by invoking the *registerData* method of the PoE v5 smart contract. The *signedTransactionMethod* function signs and sends a transaction to the blockchain using the parameters provided as input, such as the private key of the account, the web3 library functions, the data to be authenticated, and others. in both the /prove and /prove-hash endpoints are used non-transactional methods, *validateData* and *validateProof*, respectively, which take as input only the public key and not the private key.

```
app.post('/register', async function (req, res) {

    var data_req = req.body.data;

    var data = PoEContract.methods.registerData(data_req).encodeABI();

    signedTransactionMethod(
        account, data, SmartContractAddress, PRIVATE_KEY, web3, CHAINID,
        [{ type: "bytes32", name: "hash"}]
    ).then(result => {
        res.status(200).send(result);
    }).catch(err => {
        res.status(400).send(err);
    });
```

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

```
})

app.post('/prove', async function (req, res) {

    var data = String(req.body.data);

    var method = PoEContract.methods.validateData(data);

    view(method, PUBLIC_KEY).then(result => {
        res.status(200).send(result);
    }).catch(err => {
        res.status(400).send(err);
    });

});

app.post('/prove-hash', async function (req, res) {

    var data = String(req.body.data);

    var method = PoEContract.methods.validateProof(data);

    view(method, PUBLIC_KEY).then(result => {
        res.status(200).send(result);
    }).catch(err => {
        res.status(400).send(err);
    });

});
```

A Swagger interface is planned to be provided as well, as testing environment and interactive documentation for the Open API provided.

6.4 Evaluation plan

As the software components here described are designed to be support to the Usage Control, As previously reported in Section 5.5, the pilot site where the blockchain tool will be deployed will be pilot 1, since the application is the same, with no variations to what already described in the previous section.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

In particular, the enforcement of policies will be provided with the notarization service implemented by means of the PoE smart contract. In this way, it will be possible to register and verify the immutability and integrity of the policy definitions though the characteristics of the blockchain. Once notarized, the hash of the policy will be verified by any call to the blockchain. This will be enabled for both JSON RPC direct calls (Web 3.0 style) and classical REST interface, with the API layer developed (Web 2.0 style).

On the other hand, any attempt of verifying corrupted or tampered data would result in a missed verification, meaning that the tested data are not validated by the blockchain. For this reason, the address of the last notifier and nullifier is always tracked to maintain the transparency of the network.

The implementation details and documentation will be given in D4.3.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

7 Remote attestation for full stack integrity

The technical design of remote attestation relies on the IDSCPv2[28] in which the establishment of a secure channel provided, with the ability to include drivers that support parts of the channel creation. Different types of drivers can be used:

7.1 Functional/logical view

The functional view of remote attestation can be described with the generic roles *Verifier* and *Prover*. Where the verifier asks the prover to provide a proof of the environment the prover is acting in. To setup a secure connection between two parties, both parties will play both roles in order ensure both parties have provided proof of their environments. In Figure 17 the processes of both roles are depicted to clearly show the process flow as well as the moments the actor of the process changes.

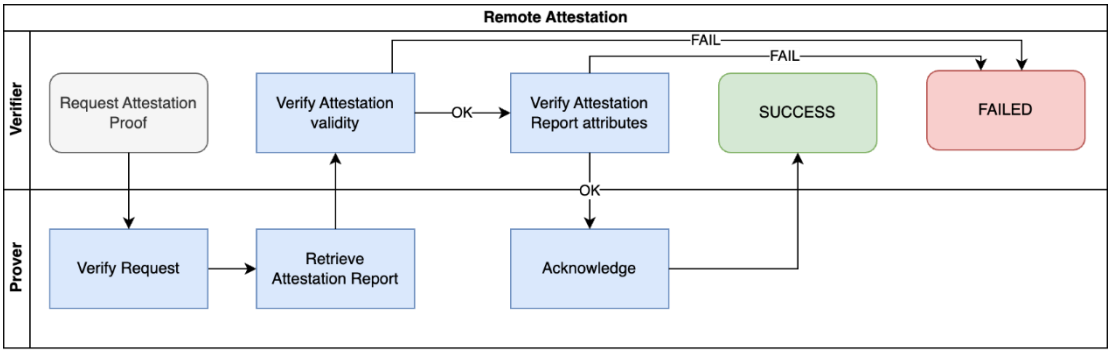


Figure 17: Remote Attestation functional view

In the Enershare dataspace, the only impacted components are the dataspace connectors. The process of remote attestation plays a role during the establishment of a secure connection between two connectors. In Figure 18 an overview of the context is shown, highlighting that the only component relevant for remote attestation process would be the core container of the connector. However, the scope of remote attestation would need to cover also, at least, the data app since the data communicated between the two parties will flow to the data app. Optionally, a third component might come into play in a later stage, namely a service that provides the connectors with a list of approved environments. This component would be used during the *Verify Attestation Report attributes* process by the verifier to ask whether the provided attestation of an environment is known and deemed ok to interact with. This is,

[28] https://github.com/industrial-data-space/idscp2-jvm/wiki

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

however, optional as both parties are also capable of keeping a list of trusted environments themselves.

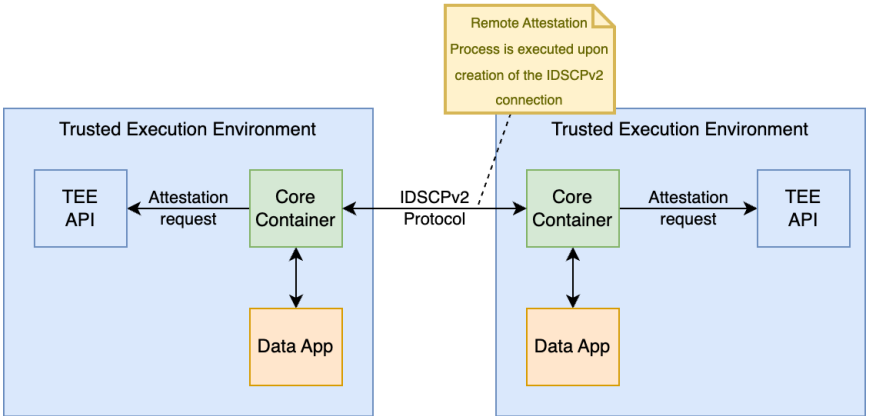


Figure 18: Context overview of remote attestation

A move towards the Dataspace Protocol, as described in Chapter 0, would make remote attestation a bit simpler. In that case, it would probably be sufficient to only execute remote attestation across the data planes instead of requiring the full stack to be attested. Thus, after the Dataspace Protocol is adopted and implemented within the Enershare dataspace, the primary focus of this work will also shift to remote attestation on data plane level.

7.2 Technical design

The technical design of remote attestation relies on the IDSCPv2[29] in which the establishment of a secure channel provided, with the ability to include drivers that support parts of the channel creation. Different types of drivers can be used:

- DAPS Drivers: for retrieving and verifying tokens from DAPS instances. The default DAPS driver is based on the interactions with the Omejdn DAPS, with which the current version of the TSG DAPS is interoperable with.
- Secure Channel Drivers: for creating secure channels. The default channel is a plain TLSv1.3 socket, but additional drivers can be created to use other persistent channels.
- Remote Attestation Drivers: for executing the Remote Attestation process described in Figure 17.

Primarily the Remote Attestation Driver will be important for the remote attestation process, but the other drivers might need changes to allow for easier deployment in pilots.

[29] https://github.com/industrial-data-space/idscp2-jvm/wiki

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

In the technical design of remote attestation, a sidecar architecture is used. This is an additional application that runs close to the connector, e.g. in the same Kubernetes pod, that can assist in the actions required for retrieving *attestation reports* for the running application. The reason for this sidecar approach is to allow easy replacement of the sidecar with a different one to allow different types of remote attestable environments to be supported. Since the way an application is able to retrieve an *attestation report* might differ across cloud/bare metal vendors.

The sequence diagram in Figure 19 shows the sequence of messages between the *verifier* (being an IDSCPv2 Remote Attestation Driver), the *prover* (also an IDSCPv2 Remote Attestation Driver), and the *prover sidecar*. The messages between the verifier and prover follow the IDSCPv2 message structure. In steps 3, 6, and 7 in the sequence diagram, the actual logic of retrieving and verifying the attestation proof will be handled.

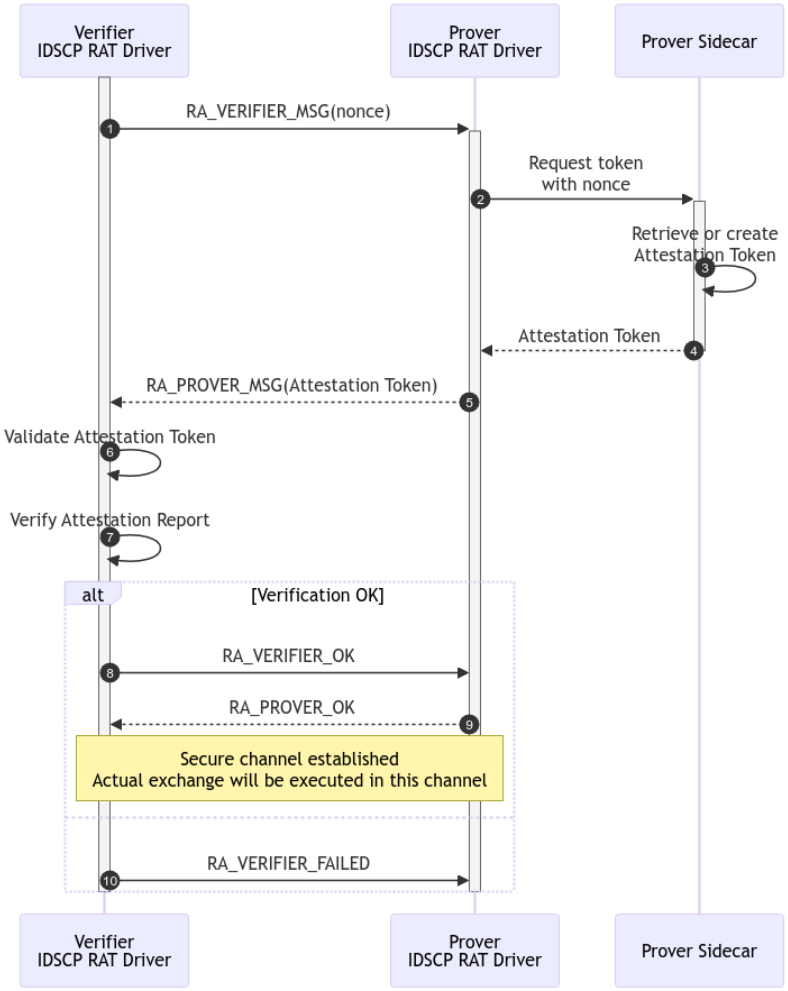


Figure 19: Remote Attestation sequence diagram

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Tests for remote attestation based upon IDSCP have since been executed, with confidential computing based on AMD Secure Encrypted Virtualization-Secure Nested Paging (AMS SEV-SNP). These tests are, due to the complexity of the setups given the link between hard- and software, executed in a straightforward setup.

Given the recent move towards the Dataspace Protocol, the evaluation of remote attestation should be executed preferably on this new protocol. The connectors that comply to this protocol are not yet available; this is part of the development, integration and evaluation work as specified in Chapter 3. Therefore evaluation of remote attestations in the context of the actual deployments of the pilots is not possible within the timeframe of the project, and in particular the timeline of the WP4 activities.

However, the evaluation of the remote attestation functionality can be done in a lab environment, demonstrating the functionality based upon the pilot data services and requirements. In deliverable D4.3, we will report on this evaluation.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

8 Conclusions

For the first technology release, WP4 provided two essential building blocks for the energy dataspace as reported in D4.1: a dataspace connector implementation and an identity provider. Together with the metadata broker (WP5), these components allowed for a minimal viable dataspace and were used for the deployment of the MVP version 1 of the Enershare dataspace according to the architecture as described in Chapter 2. At the moment of writing this report, a total of 15 connectors are deployed in the Enershare dataspace.

For this second technology release (beta version), the TRUE connector is extended with support for additional policy types, as described in Chapter 5 and an additional component is made available; the notarization component as described in Chapter 6. The beta version components are listed in the next section. The technical design for remote attestation functionality is provided in Chapter 7 and will be part of the third technology release .

Furthermore, we concluded that connector interoperability is best reached via the new Dataspace Protocol. The current TRUE and TSG connectors are not fully interoperable. In the past year the new Dataspace Protocol has emerged and is internationally recognized[30] [31] [32] as the way to achieve interoperability amongst different connector implementations. Within the Energy Data Space cluster project (int:net)[33], Enershare's approach and commitment to realize dataspace interoperability is based on the Dataspace Protocol.

This chapter concludes with a section about the consolidated plans for the third and final technology release.

8.1 List of (software) components for beta version

8.1.1 Connector implementations

The dataspace connector is the basis for trust and sovereignty. It provides secure peer-to-peer data exchange with IAA (Identification, Authentication, Authorization).

[30] https://dssc.eu/space/BBE/178422298/Control+plane+vs.+Data+plane

[31] https://internationaldataspaces.org/dataspace-protocol-ensuring-data-space-interoperability/

[32] https://projects.eclipse.org/projects/technology.edc

[33] https://intnet.eu/

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

TNO Security Gateway (TSG)	
Description	IDS-based HTTP Multipart communication. See: Message Flows, Deployment. And the Playground to play around with the connector
Software details	Written in Kotlin, built into Docker images. Default deployment based on Kubernetes & Helm.
Codebase	• https://gitlab.com/tno-tsg/core-container • https://gitlab.com/tno-tsg/helm-charts/connector
ADDED Deployed for Enershare dataspace	At the moment of writing this report, a total of 15 connectors are deployed in the Enershare dataspace. To get an overview of the deployed connectors, see: https://daps.enershare.dataspac.es/#connectors

TRUE Connector	
Description	IDS-based connector implementation consisting of a execution core container, a data app and usage control component.
Software details	Written in Java, built into Docker images. Docker compose and kubernetes manifests available for deployment.
Codebase	• https://github.com/Engineering-Research-and-Development/true-connector
ADDED Extended functionality	Support for additional policy types, as described in Chapter 5

8.1.2 Identity provider (CA + DAPS)

The certificate authority (CA) issues identity certificates for connector instances by signing Certificate Signing Requests (CSRs) that have been handed in by valid connector instances. It revokes certificates that become invalid and, for higher trust levels, assure that private keys are properly stored in hardware modules (such as a TPM or HSM).

The DAPS component provides dynamic, up-to-date attribute information about Participants and Connectors in form of signed claims and embeds them into Dynamic Attribute Tokens (DATs).

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

TSG DAPS (includes CA)	
Description	Implementation of an IDS Dynamic Attribute Provisioning Service (DAPS) v2, combined with certificate authority capabilities to sign certificate signing requests (CSR)
Software details	Written in Typescript. Both backend and generic frontend application.
Codebase	• https://gitlab.com/tno-tsg/daps • https://tno-tsg.gitlab.io/docs/daps/
ADDED Deployed for Enershare dataspace	Accessible via: https://daps.enershare.dataspac.es/ See section 2.3 and D8.2 – Enershare Data Space (1st Technology Release) for more information.

8.1.3 ADDED - Notarization service

Proof of Existence and API Layer	
Description	Smart contract running into an integrated Ethereum-based blockchain, and a Dapp providing a REST API Layer for easy notarization and verification of the integrity of data and documents.
Software details	Written in Solidity and Javascript, built into Docker images. Docker compose available for deployment.
Codebase	https://gitlab.com/dt-iot/enershare/proof-of-existence
Deployed for Enershare dataspace	Designed for the Usage Control to enforce policy integrity. Shared also with the Slovenian pilot. Indirectly affecting the Energy dataspace, as a supporting tool.

8.2 Plan for third technology release

An updated version of the TSG connector will become available in Q2. This implements the Data Space Protocol. At this moment it is not clear if the TRUE connector with support for the Dataspace Protocol will also become available within the timeframe of the project. We can expect other implementations of the Dataspace Protocol as well, like the Eclipse Dataspace Components (EDC). This is outside scope of our project.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831

Furthermore, related to tasks T4.1 to T4.4 this report provided the functional and technical components designs. In short:

- For identity and access management (T4.2) we'll integrate self-sovereign identity (SSI) concepts by means of introducing decentralized identities. For more details see Chapter 4.
- For usage control policies (T4.3) different policy enforcements will be implemented and demonstrated in different connector implementations. For more details see Chapter 5.
- Regarding distributed ledger technology (T4.4) the notarization component will be integrated with the policy enforcement component(s) of the dataspace connector(s) to notarize access and usage policies. This is described in Chapter 6
- Finally, the concept of full stack integrity (T4.1) will be demonstrated by means of integrating remote attestations in the next generation of TSG connector. This is described in Chapter 7.

Enershare has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101069831