19 DS2 Policy Agreement and Enforcement Module (PAE)

19.1 DS2 Policy Agreement and Enforcement Module (PAE)

Owner(s):	INDRA
DOA Task:	T4.1
Tier:	3
Nature:	Foundation
Result:	Outcome

This task will create a federation mechanism to enable different data spaces to interoperate. This task will orchestrate the lifecycle of data from data collection to data exchange, to data disposal/deletion across a federation of distributed data stores. The data lifecycle will include the establishment of a data contract between the data sources, the establishment of trust between entities in the data flow, and the adherence to data sovereignty and security requirements in the resulting federated data set. Other challenges will be addressed, such as accountability for use of purpose, the propagation of new domain-specific data restrictions (such as policy changes) across the federation, and methods for non-repudiable lineage across the lifecycle. Additionally, topics to reduce the latency involved in the transfer of huge amounts of data, such as caching, and data relocation will be investigated. This task will examine current, emerging technologies in this field, such as work being led by European Data Spaces and from the GAIA-X project as a basis for extension

19.1.1 Introduction

Purpose: The primary function of the Policy Agreement and Enforcement Module (DS2 PAE) is to ensure compliance with the established policies and regulations governing data exchange among users in different data spaces. Henceforth, policies, regulations, and agreements are synonymous with the term policy. The policies are evaluated as the control plane stage of data sharing in the Connector. The policies serve two main purposes: Access Control and for Usage Control. Access Control determines whether access to data is granted or denied. Usage Control dictates how the data can be used once access is granted.

Description: Policies in dataspaces define who can access the data and the restrictions on data use for those with access. A policy, in this context, is a set of rules governing data sharing within a dataspace and, more specifically for DS2, between dataspaces as well as their participants. The main function of this module is to enforce policies associated with a data-sharing contract, an agreement between a provider and a consumer. The agreed policy, the contract, specifies the rules both parties must follow. This module focuses on rules that can be automatically enforced by software. Policies are evaluated before data transmission begins. Rules that cannot be automatically enforced are logged alongside the contract agreement for accountability. There will be two types of rules, Access Control and Usage Control.

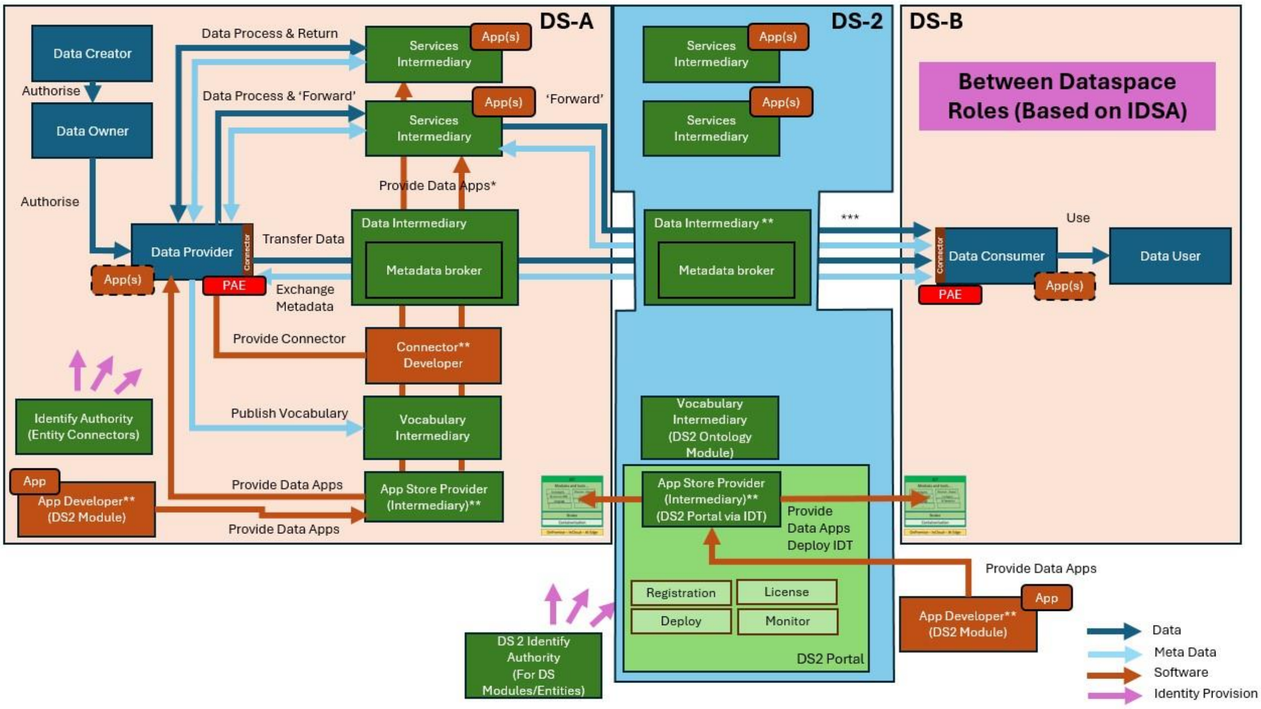
- Access Control rules define who is authorized to access the data. If these rules are not met, data sharing is not permitted.

- Usage Control rules define how the data can be used. Usage rules may require additional actions during or after data sharing, which might need to be executed by other software components or through human intervention.

The Policy and Agreement Enforcement Module will notify the relevant modules about the need for these actions.

19.1.2 Where this component fits

19.1.2.1 Big Picture



Where	Status
Within a single Dataspace for use between participants in that Dataspace only	Yes: The focus in DS2 will be on supporting the enforcement of rules defined for inter-dataspace use cases with this module acting at the participant in each dataspace to agree/enforce based on the other participant. In this sense whilst it is deployed by the participant it is always used in a Across Dataspaces without Service Intermediary scenario. But could be used in a single dataspace if they use EDC connector and ensure that all the rules that they need to enforce are supported.
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	N/A
Across Dataspaces without Service Intermediary	Yes: The Policy and Agreement Enforcement Module will be deployed in the DS2 Connector within the IDT of each dataspace.
Across Dataspace with Intermediary	N/A

Other Comments	N/A

19.1.2.2 Within a single Dataspace (where applicable)

The Policy and Agreement Enforcement Module may be used for policy enforcement within a single dataspace. It will be developed for compatibility with the EDC Connector, so any dataspace using the EDC Connector will potentially be able to utilize this module. However, in the context of the DS2 project, the Policy and Agreement Enforcement Module will provide support only for the policies and rules identified as necessary for scenarios involving two dataspaces. Consequently, if a dataspace wishes to use this module, it may need to extend it to support additional policies and rules.

19.1.2.3 Deployed and used by a single participant (where applicable)

N/A

19.1.2.4 Across Dataspaces without intermediary (where applicable)

The Policy and Agreement Enforcement Module will provide the capability to validate and enforce policies at all levels. Each policy is composed of rules that must be satisfied. The rules identified by DS2 as necessary for inter-dataspace scenarios will be supported and implemented by this module out of the box. There will be two types of rules supported: access control rules and usage control rules. Connectors may have their own policy enforcement mechanisms, which will be accommodated. This is represented by the 'Dataspace Policy Extension' in later diagrams. This does not limit the number of rules supported by DS2; rather, it adds flexibility

Access control rules define who can perform a data transaction. For example, they can specify that only participants with certain attributes (e.g., located in a specific country) can access a dataset. These rules are evaluated in the control plane of a connector. If the rules are not satisfied, data sharing will not commence. The Policy and Agreement Enforcement Module will be able to evaluate access control rules independently, without requiring other modules or applications.

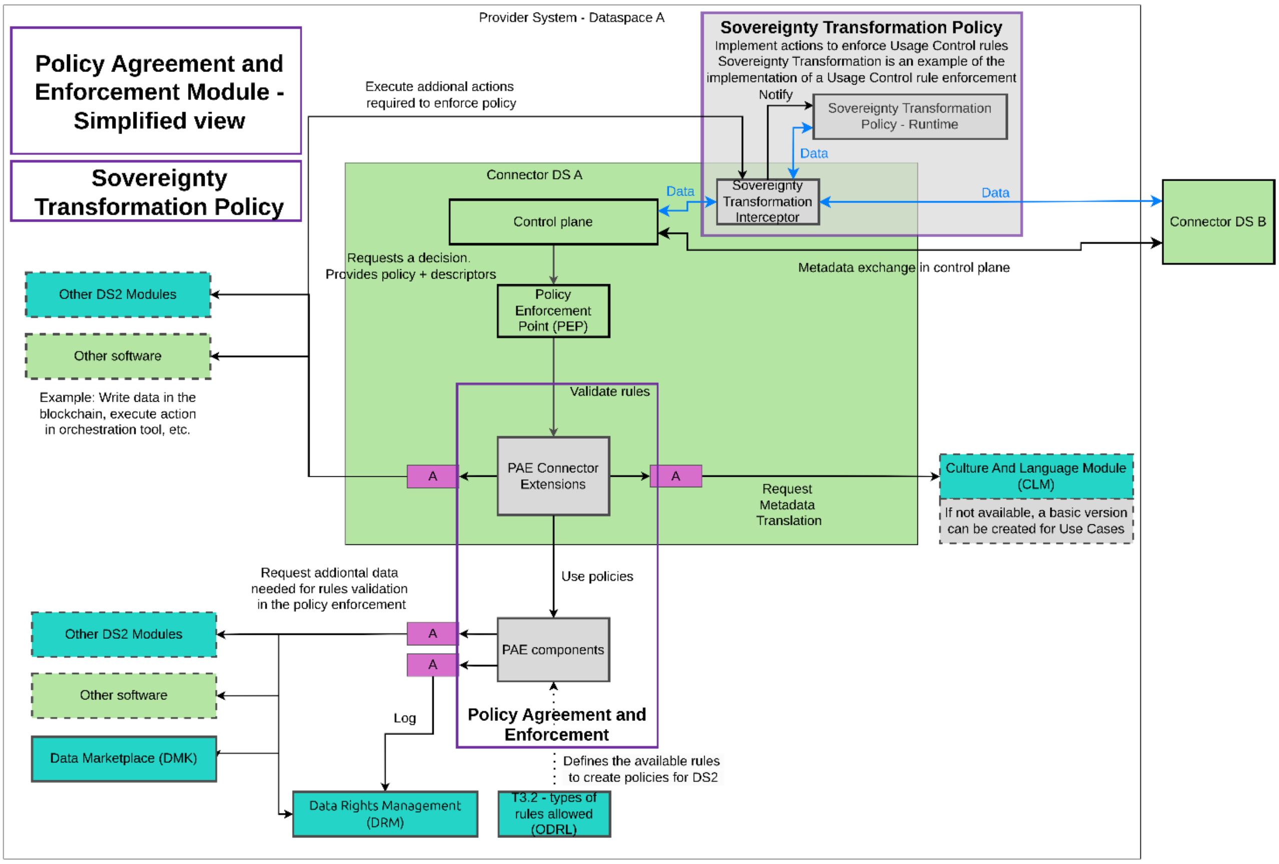
Usage control rules specify additional actions that must be performed during or after data sharing. Since the Policy and Agreement Enforcement Module acts before data sharing begins, it requires collaboration with additional modules to enforce these rules. Therefore, the Policy Agreement and Enforcement Module will define an interface for this purpose. Any module or application that supports the enforcement of usage control rules during or after data sharing will need to be compatible with this interface and execute the necessary software programs to enforce the policy. As an example of this capability, the Policy and Agreement Enforcement Module will handle the enforcement of data sovereignty-oriented transformations required by policies, such as removing specific data and anonymizing data. These rules can be used in conjunction with others to define policies. For example, if a consumer lacks a specific certification, a particular field (or column) in the data will not be shared.

19.1.2.5 Across Dataspaces with Intermediary (where applicable)

N/A

19.1.3 Component Definition

The figure below represents the actors, internal structure, primary sub-components, primary DS2 module interfaces, and primary other interfaces of the module.


Policy Agreement and Enforcement Module - Simplified view
Sovereignty Transformation Policy
Provider System - Dataspace A
Sovereignty Transformation Policy
Implement actions to enforce Usage Control rules
Sovereignty Transformation is an example of the implementation of a Usage Control rule enforcement
Notify
Sovereignty Transformation Policy - Runtime
Data
Execute addional actions required to enforce policy
Connector DS A
Data
Sovereignty Transformation Interceptor
Data
Connector DS B
Control plane
Requests a decision.
Provides policy + descriptors
Metadata exchange in control plane
Other DS2 Modules
Other software
Example: Write data in the blockchain, execute action in orchestration tool, etc.
Policy Enforcement Point (PEP)
Validate rules
PAE Connector Extensions
A
A
Request Metadata Translation
Culture And Language Module (CLM)
If not available, a basic version can be created for Use Cases
Request addiontal data needed for rules validation in the policy enforcement
Use policies
Other DS2 Modules
Other software
Data Marketplace (DMK)
A
A
PAE components
Log
Policy Agreement and Enforcement
Defines the available rules to create policies for DS2
Data Rights Management (DRM)
T3.2 - types of rules allowed (ODRL)

The following figure expands on the previous one by detailing the subcomponents for policy enforcement and their relationships with other modules.

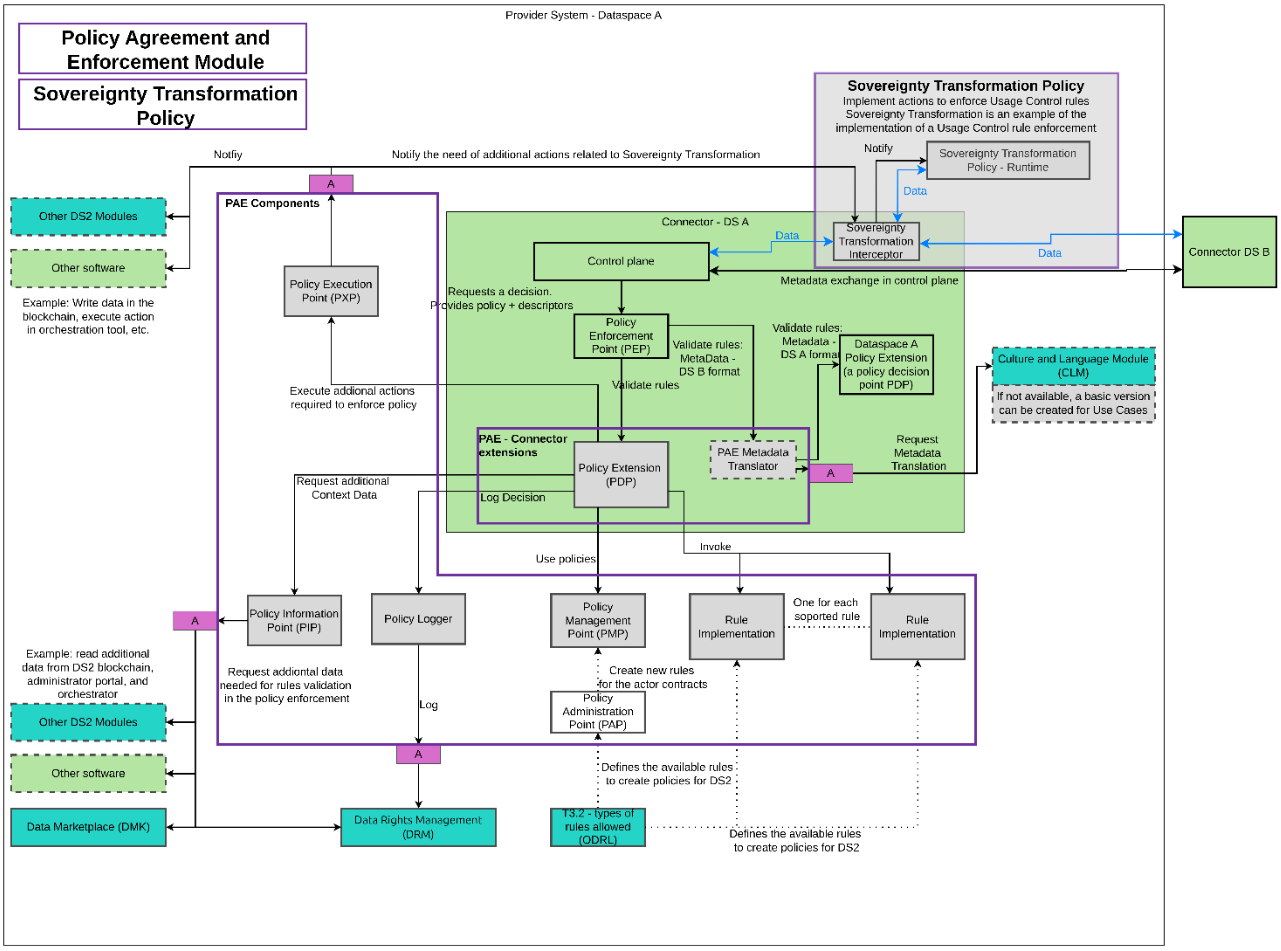

Provider System - Dataspace A
Policy Agreement and Enforcement Module
Sovereignty Transformation Policy
Notfiy
Notify the need of additional actions related to Sovereignty Transformation
A
PAE Components
Other DS2 Modules
Other software
Example: Write data in the blockchain, execute action in orchestration tool, etc.
Policy Execution Point (PXP)
Execute addional actions required to enforce policy
Connector - DS A
Control plane
Requests a decision. Provides policy + descriptors
Policy Enforcement Point (PEP)
Validate rules: MetaData - DS B format
Validate rules
Validate rules: Metadata - DS A format
Dataspace A Policy Extension (a policy decision point PDP)
PAE - Connector extensions
Policy Extension (PDP)
PAE Metadata Translator
Log Decision
Request additional Context Data
Request Metadata Translation
Sovereignty Transformation Policy
Implement actions to enforce Usage Control rules Sovereignty Transformation is an example of the implementation of a Usage Control rule enforcement
Notify
Sovereignty Transformation Policy - Runtime
Data
Sovereignty Transformation Interceptor
Data
Data
Metadata exchange in control plane
Connector DS B
Culture and Language Module (CLM)
If not available, a basic version can be created for Use Cases
Use policies
Invoke
Policy Information Point (PIP)
Policy Logger
Policy Management Point (PMP)
Rule Implementation
One for each soported rule
Rule Implementation
Create new rules for the actor contracts
Policy Administration Point (PAP)
Request addiontal data needed for rules validation in the policy enforcement
Log
Example: read additional data from DS2 blockchain, administrator portal, and orchestrator
Other DS2 Modules
Other software
Data Marketplace (DMK)
Data Rights Management (DRM)
T3.2 - types of rules allowed (ODRL)
Defines the available rules to create policies for DS2
Defines the available rules to create policies for DS2

Figure 1: Schema for the Module

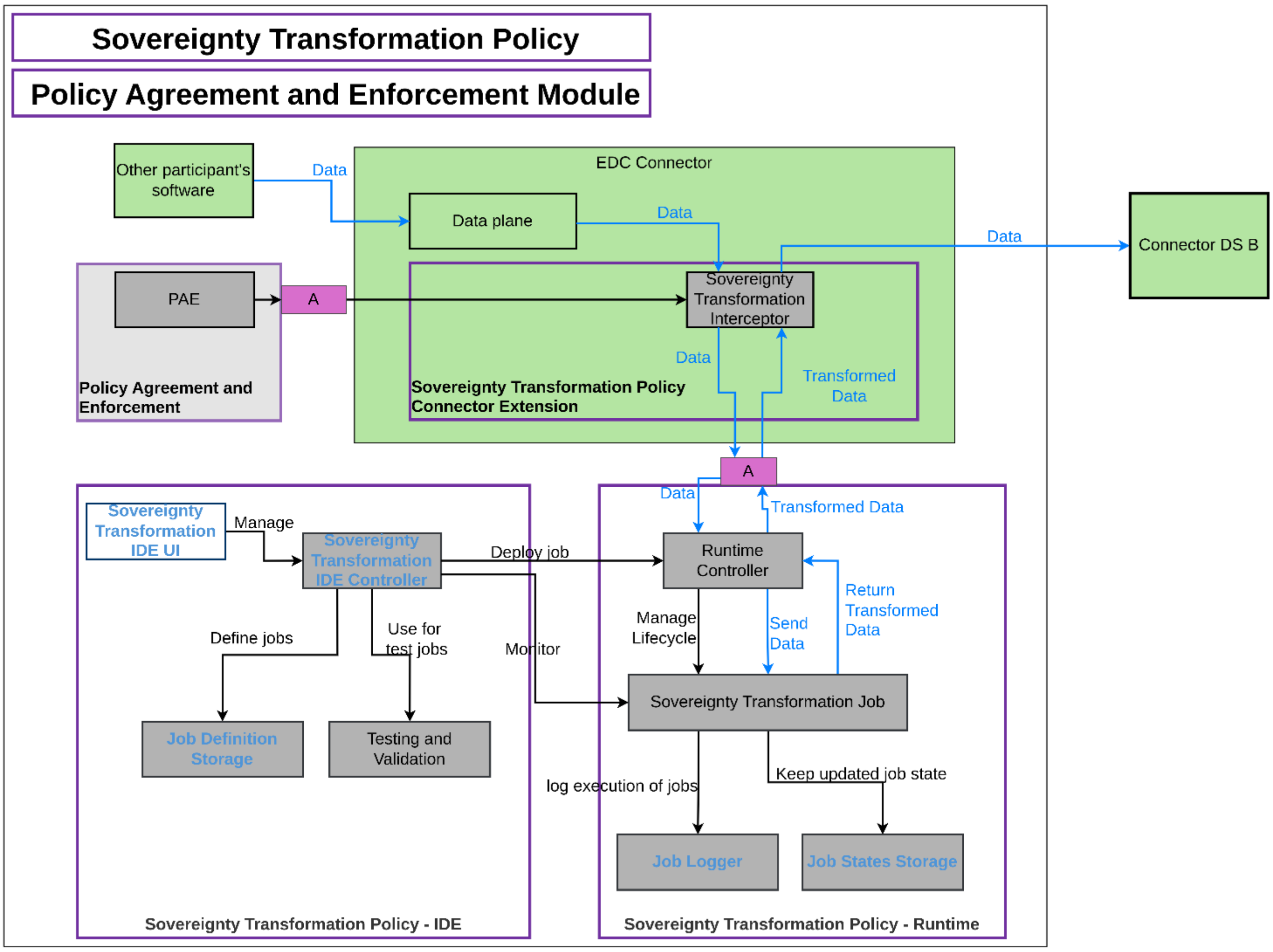
This module has the following subcomponent and other functions:

Note that the large green box represents a connector. It is depicted this way to clarify which components are connector extensions deployed within the connector, and which are deployed outside it.

- **Policy and Agreement Enforcement Module:**
  - **Policy Extension (PDP):** This extension will be integrated into the Policy Enforcement Point of the Connector. It will oversee the coordination of all policies supported by DS2.
  - **PAE Metadata Translator:** This optional extension will translate metadata to ensure compatibility with rule enforcement across different dataspaces with varying participant and service descriptors. As of this document's edition, it is to be decided if this will be handled by the Connector itself, through the CLM Module, or through the Catalog Module therefore, it will be implemented as needed.
  - **Sovereignty Transformation Interceptor:** This extension will enable policy-oriented data transformations during data exchange – for example anonymising data. It serves as the implementation of a Policy Execution Point. Note that the Data Transformation feature of DDT module is not used to do this since it is more generic
  - **Sovereignty Transformation Policy – Runtime:** This component provides an example of Usage Control rule enforcement. It is expanded and described in detail later in this section.

- **Policy Execution Point (PXP):** This component interfaces with the execution of additional actions required when a policy is enforced and after policy evaluation. An example is the Sovereignty Transformation Interceptor, which will be used to enforce certain usage control rules.
- **Policy Information Point (PIP):** This component is responsible for implementing interfaces with all relevant modules and external software (e.g., retrieving participant information from a CRM application) from which additional context data is needed for policy evaluation. A mandatory interface is with the Data Marketplace module to validate asset purchases.
- **Policy Logger:** This component will handle all logging related to policy enforcement. In addition to internal logging, policy decisions can be recorded in the Data Rights Management module.
- **Policy Management Point (PMP):** This component stores all the policies that must be enforced for each share of data. Specifically, it will store the contract offers, including the reference to the data offered and the associated policies.
- **Rule Implementation:** Each rule within a policy must have a corresponding software component that supports its evaluation. A Rule Implementation component may support several rules and might rely on an existing rule engine or be implemented from scratch, depending on the rules it needs to support. Access Control rules will be enforced based on these implementations. ODRL will be used by T3.2 for defining the types of rules allowed. Within PAE an example of Usage Control rules will be implemented using the Sovereignty Transformation Policy. In these cases, the Rule Implementation will evaluate the rule, but the needed actions to enforce the rule are delegated, in the mentioned example to the Sovereignty Transformation Policy. The responsibility of PAE is to notify to the required component. If there is not such a component to automatically handle rule actions, the evaluation will be logged with the Policy Logger, but any further verification or action is out of the scope of this module.
- **Policy Administration Point (PAP):** This is a UI that allows for the definition of policies to be used in data sharing. Policy definitions are stored in the PMP. Policies will be defined using templates developed based on the types of rules identified in WP3. The tool will assist users in easily defining rules and will generate machine-readable policies.

The next figure shows the subcomponents for Sovereignty Transformations.


Sovereignty Transformation Policy
Policy Agreement and Enforcement Module
Other participant's software
Data
EDC Connector
Data plane
Data
Data
Connector DS B
PAE
A
Sovereignty Transformation Interceptor
Policy Agreement and Enforcement
Data
Transformed Data
Sovereignty Transformation Policy Connector Extension
A
Data
Transformed Data
Sovereignty Transformation IDE UI
Manage
Sovereignty Transformation IDE Controller
Deploy job
Runtime Controller
Return Transformed Data
Define jobs
Use for test jobs
Monitor
Manage Lifecycle
Send Data
Sovereignty Transformation Job
Job Definition Storage
Testing and Validation
log execution of jobs
Keep updated job state
Job Logger
Job States Storage
Sovereignty Transformation Policy - IDE
Sovereignty Transformation Policy - Runtime

- **Sovereignty Transformation Policy:**
  - **Sovereignty Transformation Policy - IDE:** This group of components is shared with the T6.2 Data Inspector. Some of them will be developed based on existing technology, specifically the Dataflow component of the Onesait Platform open-source product. While T4.1 Policy Enforcement focuses on defining and executing transformation jobs for policy enforcement, the Data Inspector will enhance the current capabilities of Dataflow to include data inspection, monitoring, and notifications. T6.2 will lead this component development, and T4.1 will be built on the improved version from T6.2. Therefore, T4.1 has a dependency on T6.2, but the opposite is not true.
    - **Sovereignty Transformation IDE UI:** This graphical user interface allows users to define Sovereignty Transformation Jobs for enforcing policy rules that require data modifications. Examples include field removal and data anonymization. A Sovereignty Transformation Job definition is a data pipeline with an input, an output, and a set of transformation stages in between. This component is based on existing INDRA software, with updates needed to support new features for DS2.
    - **Sovereignty Transformation IDE Controller:** This component manages Sovereignty Transformation Jobs during design time and oversees their deployment and monitoring at runtime. It is based on current INDRA software but requires significant upgrades to split the tool into the IDE component and the Runtime component (potentially several).

        - **Testing and Validation:** This component will encompass test definitions and include the storage of small datasets specifically for automatic testing purposes as well as providing validation functionality. The automated tests will be designed to thoroughly validate the accuracy and correctness of the sovereignty transformation jobs. This validation process is essential to ensure that the jobs meet the required standards and perform as expected before they are deployed. By conducting these tests beforehand, potential issues can be identified and addressed, ensuring a smooth and reliable deployment of the sovereignty transformation jobs.
        - **Job Definition Storage:** This component stores definitions of Sovereignty Transformation Jobs, based on INDRA software, with extensions planned to improve version control of the definitions.
    - **Sovereignty Transformation Policy – Runtime:**
        - **Runtime Controller:** This component will manage the execution of Sovereignty Transformation Jobs during runtime. It will define the interface for integrating job execution with the Sovereignty Transformation Interceptor and will handle the job lifecycle: deployment, upgrade, removal, start, and stop.
        - **Sovereignty Transformation Job:** This component represents the runtime execution of a data pipeline definition. Each transformation supported will require a Sovereignty Transformation Job definition. One instance of this component will be created for each Sovereignty Transformation Job needed at runtime, even for the same definition. The creation of these instances will be managed by the Runtime Controller.
        - **Job Logger:** This component logs all relevant information about each job execution. Based on INDRA software, it will require minimal development to adapt to changes in other module components.
        - **Job State Storage:** This component stores the states of job executions throughout their lifecycle, enabling job resumption in the event of failures during execution. Based on INDRA software, it will require minimal development.

Interfaces with other modules:

- **T3.2 Types of rules allowed (ODRL):** This will define the types of rules that might be used in policy definition. The types of rules will be defined using the ORDL standard, which has been selected by IDSA for policy enforcement in dataspaces. Once these rules are defined, the Policy and Agreement Enforcement module will be responsible for providing software support to enforce all rules that can be evaluated without human intervention. As part of this work, WP3 will define the metadata required for policy enforcement. This metadata will primarily be incorporated during catalog data registration, making it available to the connector.
- **Data Right Management (DRM):** DRM will be used by the Policy and Agreement Enforcement to log the decisions about policy.
- **Data Marketplace (DMK):** The Data Marketplace will be queried by the Policy and Agreement Enforcement module to obtain additional information if a policy requires a purchase to obtain data.

- **Culture and Language Module (CLM):** Integration with the T5.1 Knowledge Base will be implemented to translate metadata, ensuring compatibility across different dataspaces. If it is determined later in the project that the Culture and Language Module is unsuitable for this task, simple mappings using the metadata from the project's use cases will be employed to prevent blocking the implementation.
- **Other Software / Other DS2 Modules:** The PAE module defines APIs to obtain additional information needed for the policy enforcement process, such as metadata about participants or services that may not be accessible via the connector. In such cases, PAE will acquire the necessary data. One example of this additional metadata is verifying through DRM whether a risk assessment has been previously conducted. Additionally, if any rules require actions to be executed by other software or DS2 modules, PAE will notify the requirement when evaluating the rules.

19.1.4 Technical Foundations and Background

The components involved in the transformation for policy enforcement of this module are based on an existing Onesait Platform component named Dataflow. Onesait Platform is an open-source modular platform, and consequently, its components, as Dataflow, also are open-source. The dataflow component will be upgraded as it is stated in the previous section.

Subcomponent/Component	Owner	License
Onesait Platform	INDRA	Apache 2.0

19.1.5 Interaction of the Component

The following table specifies the primary input/output controls/data to blocks which are not part of the module

With Module/Feature	Received From/Gives To	What
Data Rights Management Module	Receive From	Record the outcome of each policy enforcement process.
T3.2 - DS2 Regulatory Guidance for Practitioners	Receive From	T3.2 defines the types of rules that must be enforced. The PAE module will then provide support for enforcing those rules.
Data Marketplace	Receive From	The Data Marketplace will provide information about 'purchased datasets such that associated allowances can be enforced
Culture and Language Module	Give to	When performing policy enforcement a metadata translator is required, the CLM will be queried to obtain the equivalence between attributes in one dataspace and those in another dataspace.
Culture and Language Module	Receive From	The returned CLM query.
Other DS2 Modules (PIP)	Receive	If additional metadata must be queried from another module, the current module will use the available API to obtain it. Note: until there is a clearer picture of policy structural rules these modules and the ability to implement is not known
Other DS2 Modules (PXP)	Give	If another module need to implement policy enforcement actions, the current module will send notifications to the API provided. An example is the Sovereignty Transformation components included in

		this module.

19.1.6 Technical Risks

Risk	Description	Contingency Plan
Integration with non-production ready EDC Connector	The EDC Connector is an ongoing project, with the latest version currently at v0.8.0. Therefore, the interfaces may change during the project.	Design the solution with the minimum number of dependencies to minimize the impact of potential changes in the EDC Connector's internals.

19.1.7 Security

Security Issue	Description	Need
Authorization to add rules	Control is needed over who can publish assets and which policies are attached to those assets	The policies and contract are declared in the Connector using its own API. It is expected that the standard authorization and authentication mechanisms of the connector will address this issue
Enforcement of rules	A failure to enforce rules could allow unauthorized consumers to access data	Adherence to good coding practices is essential. Implementing tests that validate the correct functionality of the PAE Module is particularly important. For the specific rules of authorization, a "deny by default" strategy will be employed. Together with the application of the "least privilege principle", this approach will help reduce the risk of unintentional data exposure.

19.1.8 Data Governance

Data Governance Issue	Description	Need
PAE module does not storage or manage participants data	The PAE module does not manage participant data during exchanges; it only uses metadata.	N/A
Sovereignty Transformation Policy	The Sovereignty Transformation Policy example of Usage Control rules will read and transform data in transit. No data will be stored.	N/A

19.1.9 Requirements and Functionality

This module will be used in the following use cases:

City Scape	✓
Green Deal	✓
Agriculture	✓

Inter-Sector TBD

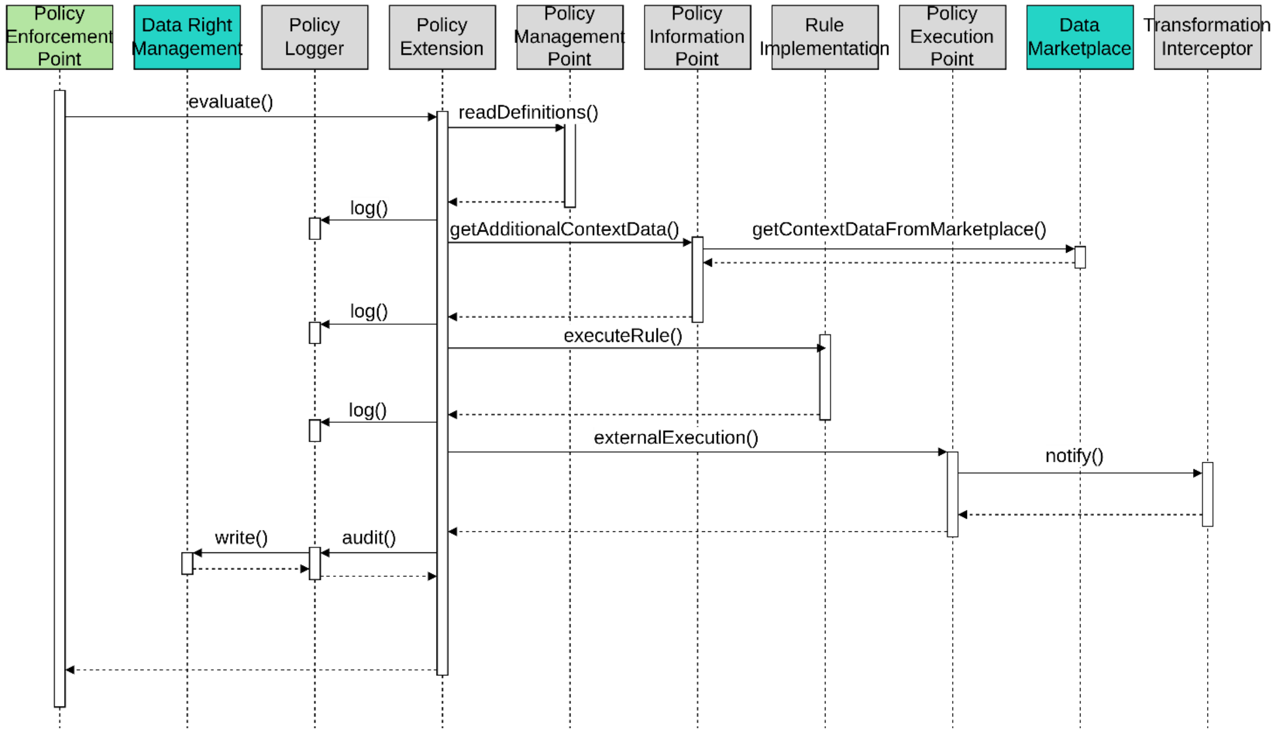
Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use case are as per the table below noting that in many cases further discussion might need to take placed between pilot partner and module partner to determine if a fit or the scope of the precise fit.

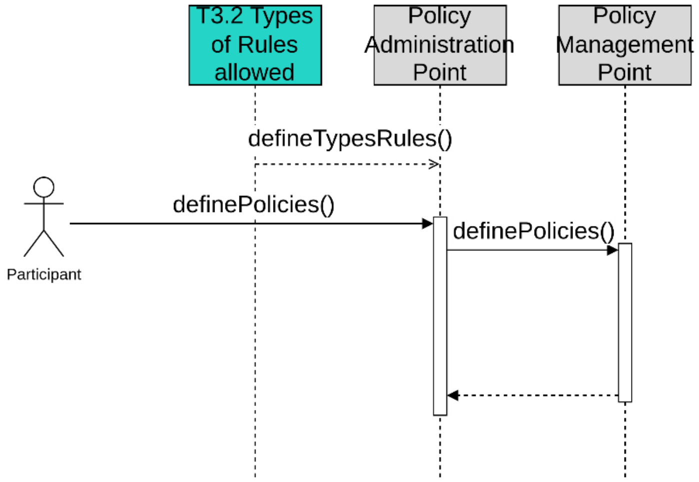
As can be seen all scenarios have inferred this aspect which is unsurprising. Regardless, as a foundation module of DS2 this module will always be deployed as a module within an initial IDT installation. It is noted it could also be deployed outside of IDT.

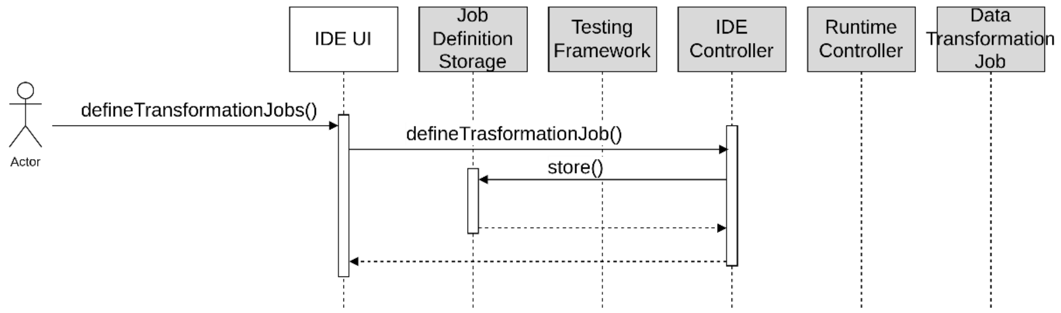
WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 2.2 UC1.1	"Confidentiality of the data, privacy of personal data (energy consumption) Risk: policies are not flexible"	Usage rule for anonymization will ensure these policies	R & D	M
Section 2.2 UC1.2	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC1.3	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC1.4	"Data access control"	Access rules will enforce these policies.	R & D	M
	Use Case 2: Green Deal			
Section 2.2 UC2.1	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC2.2	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC2.3	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC2.4	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC2.5	"Data access control"	Access rules will enforce these policies.	R & D	M
	Use Case 3: Agriculture			
Section 2.2 UC3.1	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC3.2	"Data access control"	Access rules will enforce these policies.	R & D	M
Section 2.2 UC3.3	"Data access control"	Access rules will enforce these policies.	R & D	M

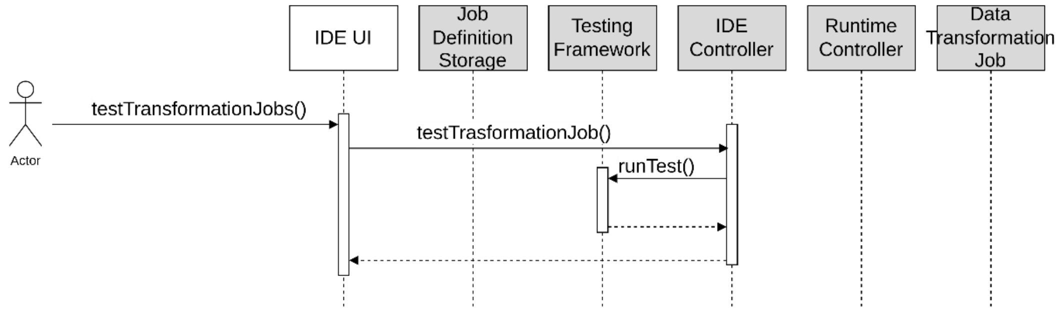
19.1.10 Workflows

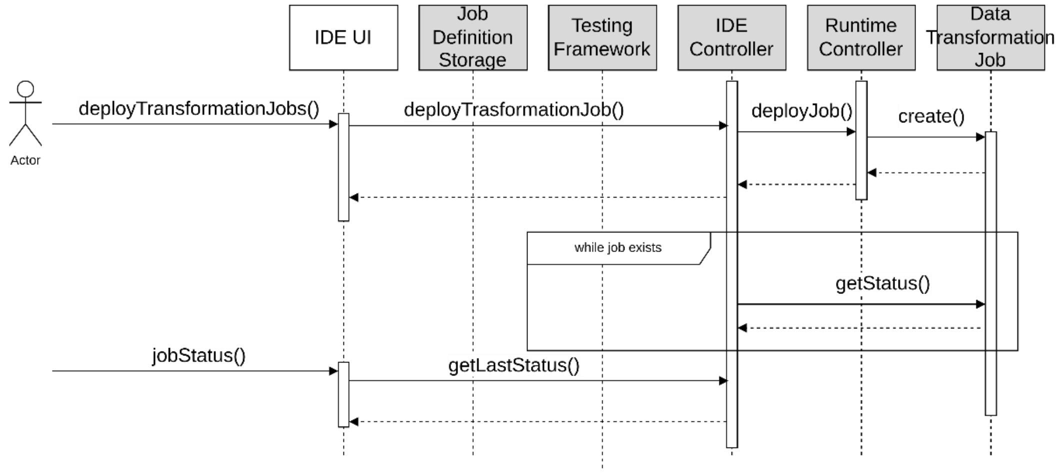
The following sub-sections describe the sequence diagrams of the Module






IDE UI
Job Definition Storage
Testing Framework
IDE Controller
Runtime Controller
Data Transformation Job
Actor
defineTransformationJobs()
defineTrasformationJob()
store()


IDE UI
Job Definition Storage
Testing Framework
IDE Controller
Runtime Controller
Data Transformation Job
Actor
testTransformationJobs()
testTrasformationJob()
runTest()


IDE UI
Job Definition Storage
Testing Framework
IDE Controller
Runtime Controller
Data Transformation Job
Actor
deployTransformationJobs()
deployTrasformationJob()
deployJob()
create()
while job exists
getStatus()
jobStatus()
getLastStatus()

19.1.11 Role, Resourcing, and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
Policy Extension (PDP)	New component implementation	■			
Metadata Translator	New component implementation			■	
Sovereignty Transformation Interceptor	New component implementation		■		
Policy Management Point (PMP)	New component implementation			■	
Policy Administration Point (PAP)	New component implementation			■	
Policy Logger	New component implementation	■	■		
Policy Information Point (PIP)	New component implementation	■	■		
Policy Execution Point (PXP)	New component implementation		■		
Rule Implementation	Implementation of rules supported by the policies.	■	■	■	
Sovereignty Transformation IDE UI	Minor updates, with primary development completed in the DINS module		■		
Sovereignty Transformation IDE Controller	Minor updates, with primary development completed in the DINS module		■		
Testing Framework	Minor updates, with primary development completed in the DINS module	■			
Job Definition Storage	Minor updates, with primary development completed in the DINS module		■		
Runtime Controller	Minor updates, with primary development completed in the DINS module		■		
Sovereignty Transformation Job	New implementation of transformation jobs for sovereignty transformation		■		
Job Logger	Minor updates, with primary development completed in the DINS module			■	
Job State Storage	Minor updates, with primary development completed in the DINS module			■	
Data Right Management (DRM)	Integration			■	
Data Marketplace	Integration			■	
Culture and Language Module	Integration			■	
Maintenance and evolution	Maintenance after release				■
Table Total/DOA Task Total/Resilience	**Comments:**				

19.1.12 Open Issues

The following table summarises open issues/uncertainties that need to be resolved during the next stages or implementation.

Issue	Description	Next Steps	Lead or Related Component
Connector	Be compatible with EDC Connector extensions	Creation of a test extension for experimentation	INDRA
Actual Enforcement	Access Control rules will be enforced by the PAE module. Usage Control rules require additional modules or software. This module includes an example of Usage Control enforcement implementation for Sovereignty Transformation Policies. If further enforcement is needed, it will be necessary to determine which additional modules, or software will implement these mechanisms.	Investigate where this could be done	TBD
Meta Data Broker Use and ontology conversion	It is to be decided if this will be handled by the Connector itself, through the CLM Module, or through the Catalog Module therefore, it will be implemented as needed.	Further discussion with TM and others to confirm nothing more is expected	INDRA, INTU, VTT