15 DS2 Data Inspection Module (DINS)

15.1 DS2 Data Inspection Module (DINS)

DS2 Data Inspection

Owner(s):	INDRA
DOA Task:	T6.2
Tier:	2
Nature:	Optional
Results:	Outcome

This task will enable an edge-to cloud connectivity through applications and devices capable of collecting, processing data and interconnecting this data with the cloud infrastructure of T6.1. It will be receiving, storing, and processing enormous amounts of data and management tools need to extract insights and make data-driven decisions based on data. This task will implement the DLC ecosystem that ensures data security and privacy and implement appropriate measures such as encryption, access control and anonymization. All relevant adapters, interfaces and UIs are developed within this task that allow the use, maintenance, and full control of the ecosystem. This task will establish concepts of open data that permit public data availability and accessibility for use and reuse without restrictions. It will further ensure that individuals and organizations will stay in control of their data, allowing them to control their own data and decide how and when it is shared. Once the federated IDT environment is operational through T6.3 amongst other task, it will be necessary to monitor it continuously at execution time (aka T4.2). Based on a data quality lifecycle, tools will be developed to define and constantly monitor data quality and establish a framework for automatic checks, eg of loss detection.

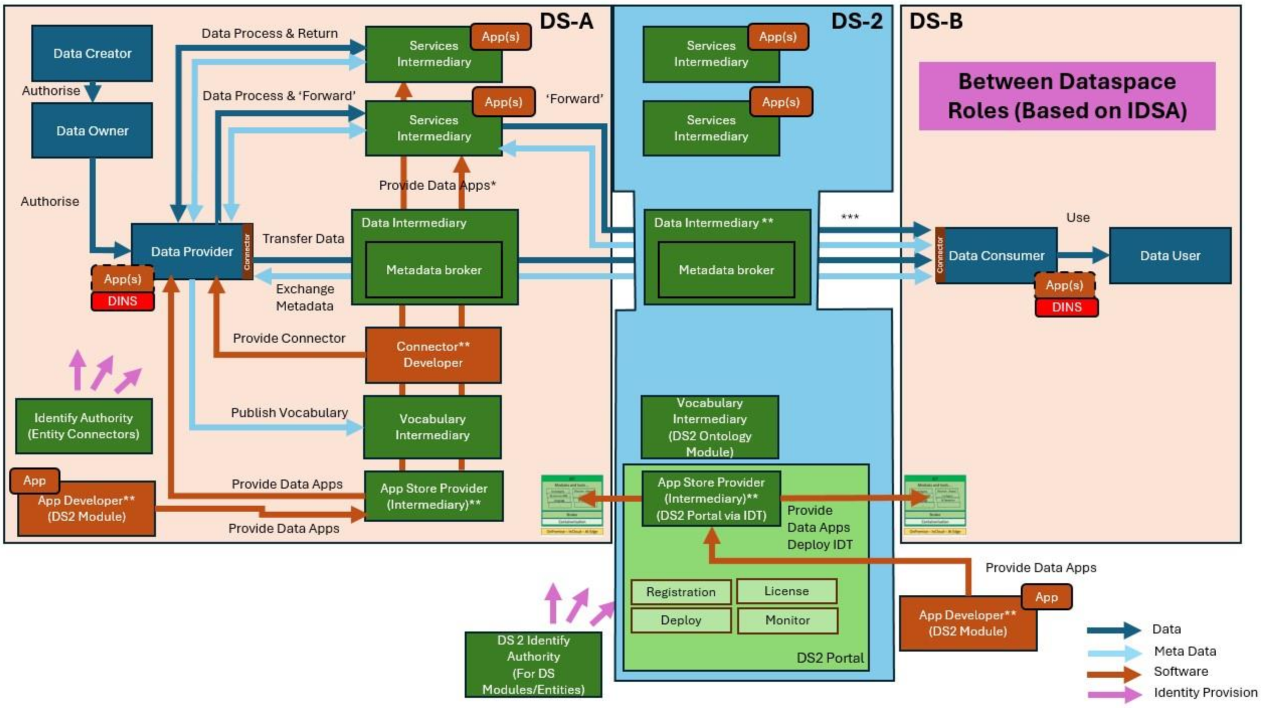
15.1.1 Introduction

Purpose: The Data Inspection Module (DINS) facilitates the configuration and deployment of processes for real-time data analysis, ensuring data quality and compliance with thresholds set by the parties involved. It performs several key functions: generating notifications based on the values of the exchanged data, executing reactions such as sending requests and notifications to external tools, and integrating with models developed to enhance its capabilities. It is a complement to the Data Share Controller which focuses on control information with both modules using the Data Interceptor.

Description: The Data Inspection is responsible for analysing the data provided and/or consumed by participants, ensuring the quality of the data shared amongst them. For example, a consumer can use DINS to set up analysis jobs that define rules for data validation. This includes establishing thresholds for specific data values; when these thresholds are exceeded, issues are generated and recorded for monitoring. Additionally, the DINS module can trigger notifications to other modules and components when issues are detected. For instance, it can notify the Data Share Controller Module, which may respond by blocking data sharing if certain limits are reached.

15.1.2 Where this component fits

15.1.2.1 Big Picture



Where	Status
Within a single Dataspace for use between participants in that Dataspace only	N/A
Deployed and used by a single participant to enable the participant in either an In-Data space or Inter- Data space scenario	Yes, the Data Inspection is integrated into the participant's data application, allowing it to operate in both single dataspace and cross-dataspace scenarios.
Across Dataspaces without Service Intermediary	N/A
Across Dataspace with Intermediary	N/A
Other Comments	N/A

15.1.2.2 Within a single Dataspace (where applicable)

N/A

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

DINS is a module designed for use by both data providers and data consumers to inspect data before or after sharing. It is intended to be integrated into the participants' data management operations, making it suitable for both single-dataspace scenarios and data sharing across dataspaces. The primary goal of the Data Inspection is to extract metrics and generate alerts based on the analysis of shared data values. To achieve this, DINS allows participants to define their own rules for data inspection. The outcomes of these rules are stored for further analysis. Additionally, DINS includes alert management capabilities to generate notifications and alerts when specific rules are triggered. Users can define and customize alerting criteria, such as thresholds or specific data value occurrences. This flexibility ensures that alerts are meaningful and relevant to the specific data being analysed. DINS includes dashboards to manage the alerts generated and the capability to define webhooks to notify external systems, for example, send an email or push notifications to platforms like Slack or Microsoft Teams.

15.1.2.4 Across Dataspaces (where applicable)

N/A

15.1.2.5 Dataspace Intermediary (where applicable)

N/A

15.1.3 Component Definition

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

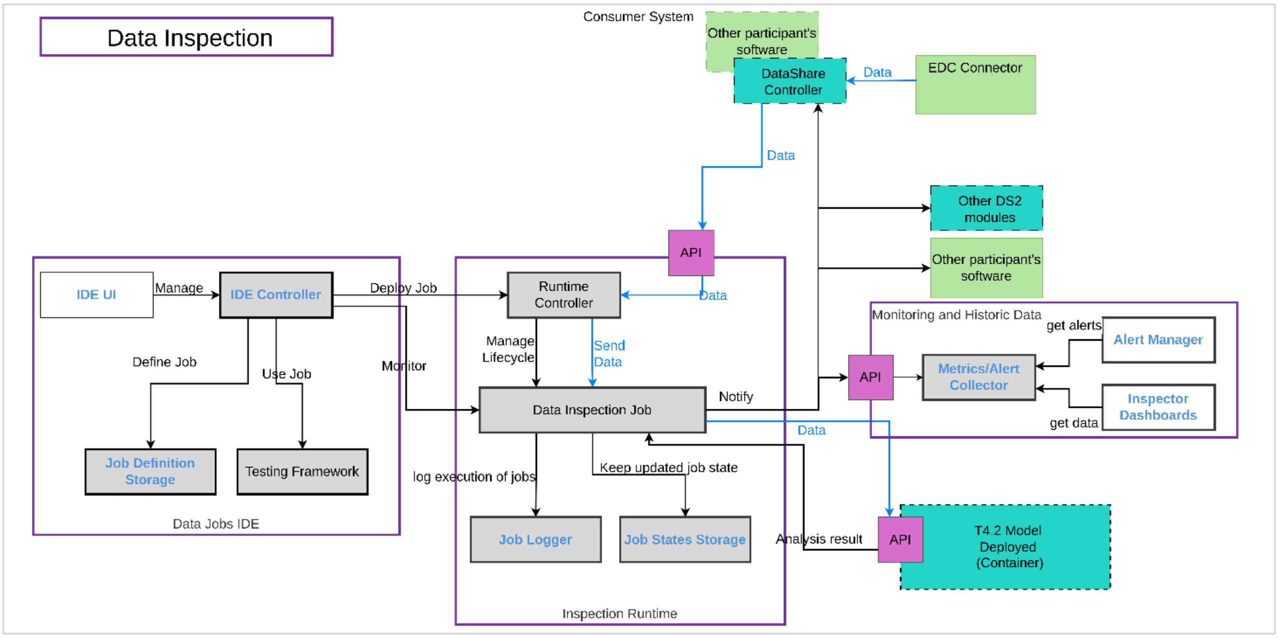


Figure 1: Schema for the Module

This module has the following subcomponent and other functions:

- **Data Jobs IDE:** This group of components is shared with the T4.1 Policy Enforcement Modules. 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 data transformation jobs, the Data Inspection module 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.

  - **IDE UI:** This graphical user interface allows users to define Data Inspection Jobs for monitoring data and triggering notifications and alerts. For example, it can evaluate values against specified thresholds. This component is based on the existing INDRA software, with updates needed to support new features for DS2.
  - **IDE Controller:** This component manages Data Inspection 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).
  - **Job Definition Storage:** This component stores definitions of Data Inspection Jobs, based on INDRA software, with extensions planned to improve version control of the definitions. The definitions are created through a graphical user interface, enabling users to design Data Inspection Jobs as data pipelines using a drag-and-drop approach. Each data pipeline will include, at a minimum, the configuration of data sources (e.g., data formats) and the rules for inspecting the data (e.g., detecting specific fields and value thresholds). In the end, each data pipeline definition is a JSON document and a set of parameters. They will be stored in a distributed persistence engine.
  - **Testing Framework:** This component will include test definitions and the storage of small datasets for automatic testing. This component will enable users to define automated tests that are executed to validate Data Inspection Jobs before deployment.

- **Inspection Runtime:**
  - **Runtime Controller:** This component will manage the execution of Data Inspection Jobs during runtime. It will define the interface for integrating job execution with external components and will handle the job lifecycle: deployment, upgrade, removal, start, and stop.
  - **Data Inspection Job:** This component represents the runtime execution of a data inspection definition. Each type of data inspection supported will require a Data Inspection Job definition. One instance of this component will be created for each Data Inspection Job needed at runtime, even for the same definition. The creation of these instances will be managed by the Runtime Controller. The Data Inspection Job includes the definition of an SDK and interfaces that facilitate the extension of capabilities, such as supporting additional data formats.
  - **Job Logger:** This component logs all relevant information about each job execution. Based on INDRA infrastructure, it will require minimal development to adapt to changes in other module components.
  - **Job States 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 infrastructure, it will require minimal development.

- **Monitoring and Historic Data:** This component will store historic metrics gathered by the analysis jobs. This set of elements will be based on the open-source stack of monitoring tools Grafana.

  - **Metrics/Alert Collector:** This component will collect all the data from the Analysis Jobs.
  - **Alert Manager:** It collects alerts, manages them by categorizing and prioritizing, and allows for the configuration of notification channels such as email and Teams. It also provides a centralized interface for tracking and managing active alerts.
  - **Inspector Dashboard:** It allows the visual analysis of the data inspected.
- **Model Deployed:** If a complex analysis is required for data inspection, the Data Inspection Jobs will have the capability to use data models deployed with the T4.2 Data Model Development Toolkit Module. For example, a field value could be used as input in a prediction model, and based on the result, a decision can be made whether an alert should be generated.
- **Data Share Controller:** The Data Interception part of the Data Share Controller connects data in the data pipeline with the Data Inspection. If the Data Share Controller is not present, any other software that implements the Data Inspection API can provide the data. The Data Inspection will trigger notifications to the Data Share Controller Module (and other modules or components, if required) based on rules defined by the participants. For example, "stop data sharing".
- **Other DS2 Modules / Other Participant's Software:** The Module can notify any software that needs to receive these notifications, provided that the software is compatible with the notification mechanisms implemented by DINS. This capability facilitates the integration of systems that can react to data shared between participants, such as the Data Share Controller.

15.1.4 Technical Foundations and Background

Some components involved in the Data Inspection 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

15.1.5 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	Receive From/Gives To	What
Data Share Controller	Receive From	Data Share Controller will send the data to be inspected. DINS will define an API to receive the data.
Data Share Controller	Gives To	The Data Inspection will send notifications to the Data Share Controller and will be compatible with the API defined by that module.
Model Development Toolkit	Receive From	The Data Inspection Jobs will have the capability to use models developed and deployed by the Model Development Toolkit Module. This will be done optionally by Data Inspection Jobs that require complex operations.

Other DS2 modules	Gives To	The Data Inspection will send notifications to any DS2 modules that are subscribed, allowing them to react to data shared between participants if needed.

15.1.6 Technical Risks

Risk	Description	Contingency Plan
Data formats	The Data Inspection Module need to work with the data shared, and therefore needs to support the encoding and format of the data analysed.	The Data Inspection Jobs will be defined with clear interfaces and SDK that facilitate the creation of plugins to support additional data formats.
Accuracy	Accuracy in data inspection is crucial when potential issues detected in data are notified to other systems because accurate data ensures reliable and correct information is communicated.	The Testing Framework, along with well-designed patterns in the data pipelines for data inspection, are key elements in ensuring this.

15.1.7 Security

Security Issue	Description	Need
Authentication / Authorization of IDE UI	The IDE UI requires user management capabilities. As a development tool, it necessitates that user’s be managed by participants similarly to other internal user accounts, such as those for databases or internal documentation.	N/A

15.1.8 Data Governance

Data Governance Issue	Description	Need
Testing Framework	The testing framework will use datasets to validate data pipelines.	The datasets used for testing can be created using synthetic data.
Inspection of data	Due to the nature of this feature, the module will have access to data in transit.	No data will be stored permanently in the module. Only temporary storage in memory will be used. In any case, data managed by an instance of this module within a participant’s environment are data that the participant already has authorization to view or use. It is up to the participants to define authorizations for their staff members.
Handling of personal data	Personal data may be processed by this module.	Consumers and providers should ensure that personal data is transferred in accordance with relevant regulations.

15.1.9 Requirements and Functionality

This module will be used in the following use cases:

City Scape ✓

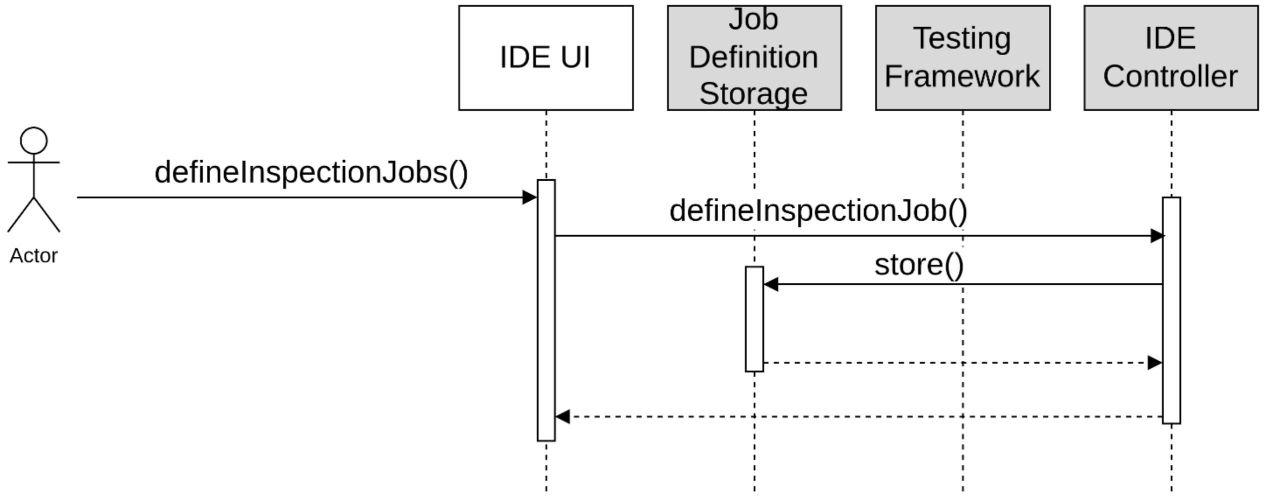
Green Deal	✓
Agriculture	✓
Inter-Sector	TBD

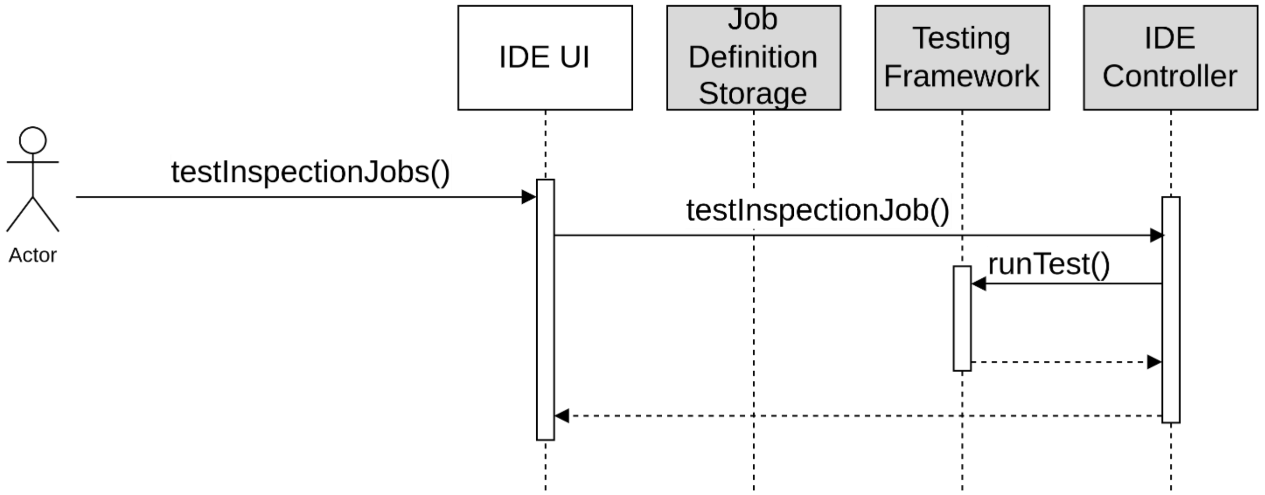
Their requirements and functions/extensions to achieve them relative to this module, specifically extracted from the use case are as per the table below noting that in many cases further discussion might need to take placed between pilot partner and module partner to determine if a fit or the scope of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

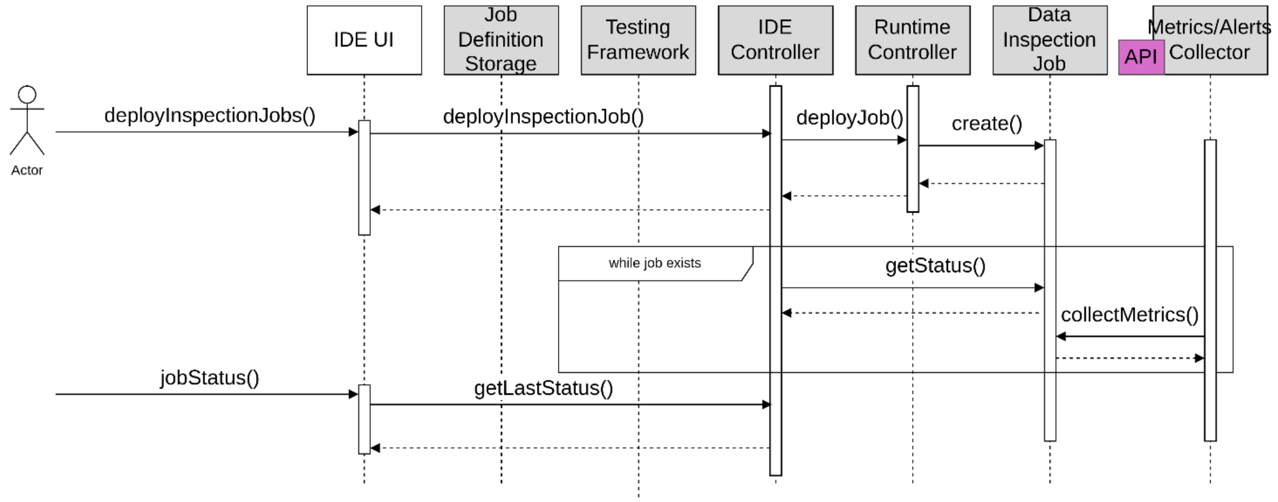
WHERE	WHAT	WHY	Run/Design Time	Priority
	Use Case 1: City Scape			
Section 2.2 UC1.1	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC1.2	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC1.3	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC1.4	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
	Use Case 2: Green Deal			
Section 2.2 UC2.1	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC2.2	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC2.3	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC2.4	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
Section 2.2 UC2.5	Data quality	Data Inspection allows the continuous monitoring of the data quality	R & D	M
	Use Case 3: Agriculture			
Section 2.2 UC3.1	Data quality	Access rules will enforce these policies.	R & D	M
Section 2.2 UC3.2	Data quality	Access rules will enforce these policies.	R & D	M
Section 2.2 UC3.3	Data quality	Access rules will enforce these policies.	R & D	M

15.1.10 Workflows

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







15.1.11 Resourcing and Milestones

Sub-component	Main Activity	M18	M24	M30	M36
IDE UI	Upgrade of versions and development to support the new capabilities from the user interface		■		
IDE Controller	Development of new features, especially communication with remote Runtime Controllers and Data Inspection Jobs.		■		
Job Definition Storage	Upgrade and small improvements		■		
Testing Framework	New development	■			
Runtime Controller	New development		■		
Data Inspection Job	New development		■		
Job Logger	Upgrade and small improvements			■	
Job States Storage	Upgrade and small improvements			■	
Metrics/Alert Collector	Upgrade and configurations			■	
Alert Manager	Upgrade and configurations			■	
Dashboard	Upgrade and development of new dashboards			■	
T4.2 Model Deployed	Develop component to use the API of this module			■	
DataShare Controller	Define, develop and integrate with this module			■	
Bug Fixing and Maintenance	Maintenance after release				■
Table Total/DOA Task Total/Resilience	**Comments:**				

15.1.12 Open Issues

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

Issue	Description	Next Steps	Lead or Related Component
N/A			