RE4DY

MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	**CERTH Sovereign Data Transformation Service**
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

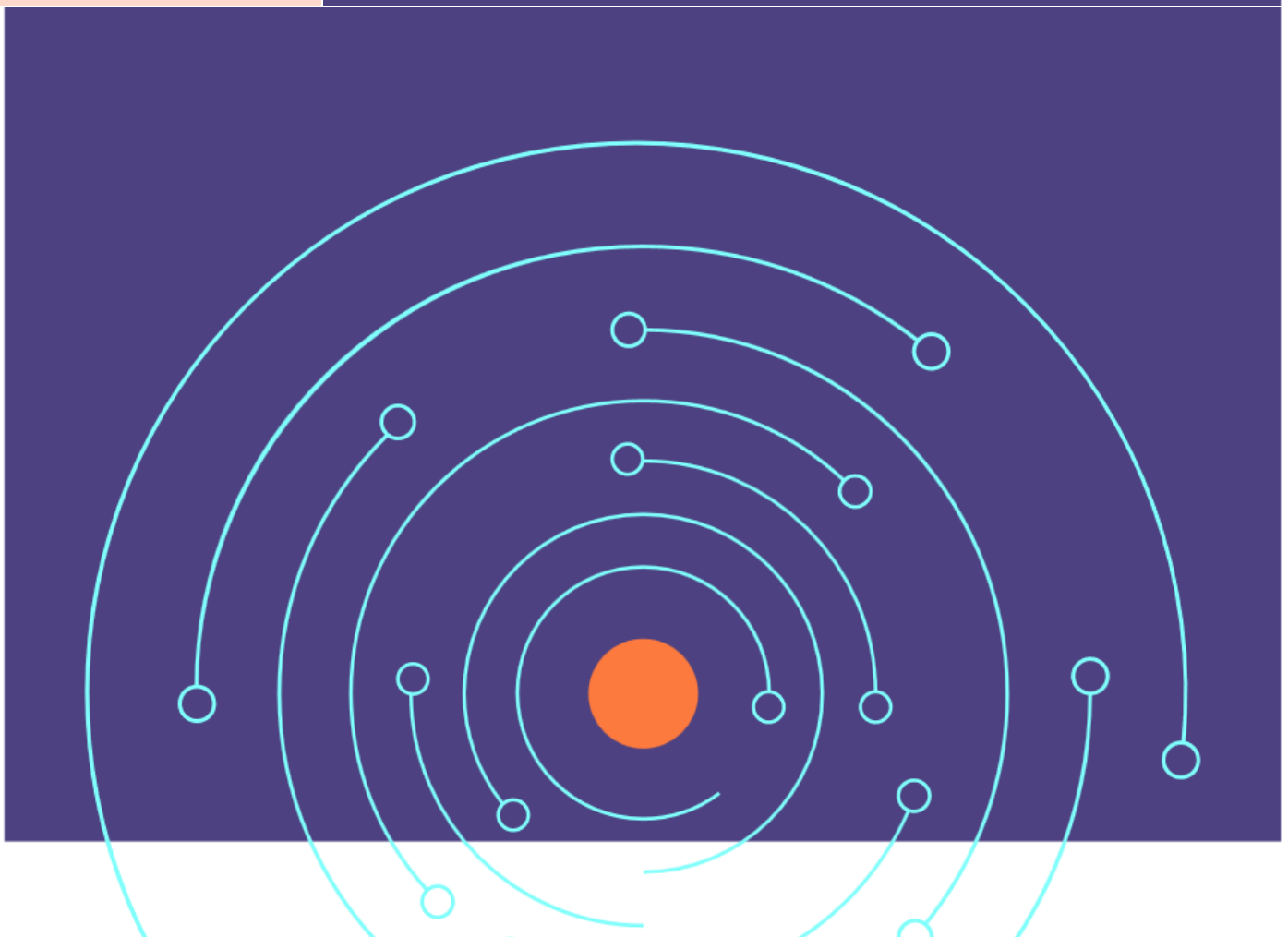

Table of contents

1. Component Description

In the scope of the RE4DY project, the paramount importance of robust ETL (Extract, Transform, Load) services has necessitated the development of specialized Data Transformation Services. Central to these services is the use of Apache NiFi[1], a recognized leader in the realm of data flow technologies, which will underpin the foundational architecture of our ETL processes.

Apache NiFi's versatile capabilities allow our Data Transformation Services to accommodate a broad spectrum of file formats, including but not limited to Json, CSV, and plain text. This adaptability ensures that our services can seamlessly integrate into diverse data ecosystems, especially in scenarios where file format restrictions are minimal or non-existent. Moreover, in instances where the incoming file format might not align with preferred standards, our services are still equipped to carry out the necessary transformations efficiently.

Leveraging the power of processors within Apache NiFi, we aim to holistically address the three core pillars of ETL – extracting data from its source, transforming it into the desired format, and loading it to the target system or database. This approach ensures that our Data Transformation Services are not only robust and reliable but also highly adaptable to the unique requirements presented by different use-case providers in the RE4DY project.

To further ensure sovereign and secure data communication the RE4DY data transformation services across the digital fabric are coupled with data space connectors. The connectors are used are based on IDSA RAM [2]and provides all the necessary capabilities for setting rules regarding data access and sharing.

2. Input

Main Inputs for the Data Transformation Component are:

- **Source Data:** This encompasses a variety of file formats that the Data Transformation Component is designed to handle, including but not limited to Json, CSV, plain text, etc.
- **Configuration Parameters:** These might include settings related to data extraction, transformation logic, or loading processes. They could be specific parameters for Apache NiFi processors, settings for JOLT[3] transformations, or custom parameters for other tools and processes within the component.
- **Transformation Rules:** These define how the source data should be transformed. Given that JOLT technology is being utilized, this could refer to specific JOLT transformation specifications.

[1] https://nifi.apache.org/
[2] https://internationaldataspaces.org/publications/ids-ram/
[3] https://github.com/bazaarvoice/jolt

- **Ontological Mappings:** If the Data Transformation Component interfaces with the Ontology Repository for data transformation, then mappings or references to standardized data models and ontologies are essential inputs.
- **Target System or Database Specifications:** Information on where the transformed data needs to be loaded, whether it's a type of database, a cloud storage location, or another system altogether.
- **Provenance Data:** Historical data detailing the lifecycle of each FlowFile, which can be used for data lineage, auditing, or debugging purposes. This data can come from the Provenance Repository.
- **Custom Processor Settings:** If there are any customized Apache NiFi processors developed specifically for the RE4DY project, their configurations, scripts, or any other related settings would be vital inputs.

3. Output

Main Outputs for the Data Transformation Component are:

- **Transformed Data:** Post-processed data ready for ingestion into target systems. This data would be in the desired format (Json, CSV, plain text, etc.) as required by downstream applications or storage solutions.
- **Data Load Logs:** Reports or logs that provide a summary of the data load processes, detailing successes, failures, or any discrepancies encountered during the data transformation.
- **Provenance Records:** Enhanced records that indicate how each Flow File was processed, transformed, and loaded. These records, stored in the Provenance Repository, help trace the journey of each piece of data through the transformation process.
- **Transformation Audit Trails:** Detailed logs that keep track of all transformation rules applied, any data mapping done using ontologies, and any exceptions or errors encountered.
- **Error Reports:** In cases where data fails to transform or load correctly, error reports detailing the reason for failure, the data source, and any other relevant metadata.
- **Ontological Data Mapping Reports:** If data is mapped using ontologies from the Ontology Repository, a report detailing these mappings, the ontologies used, and the resultant structure of the transformed data.
- **Performance Metrics:** Metrics detailing the performance of the Data Transformation Component, including processing times, throughput, efficiency, and other relevant KPIs.
- **Data Quality Reports:** Post-transformation, these reports can provide insights into the quality of the transformed data, detailing any inconsistencies, missing values, or other potential issues.

4. Information Flow

Data Mapping Using Ontology

The use case "Data Mapping Using Ontology" describes the process where a User interacts with the components of the "Data Transformation Services" to map raw data to a standardized ontology model. This process includes the provision of input data, selection of the desired ontology model from the Ontology Repository, data mapping based on the selected ontology, and retrieval of mapped data by the User. The outcome is structured data in alignment with the chosen ontology, ensuring semantic coherence and compatibility across systems.

Primary Actor: User (Data Engineer, Data Scientist)

Secondary Actors: Ontology Repository, Data Transformation Subcomponent

Preconditions:

- The Data Transformation Services are properly configured and operational.
- The desired ontology model is available in the Ontology Repository.
- Raw data, along with any necessary parameters and configurations, are available.

Main flow:

- **Input Data Provision** - The User provides raw data and specifies the desired ontology model for mapping.
- **Ontology Fetching** - The Ontology Repository retrieves the specified ontology model.
- **Data Mapping** - The Data Transformation Subcomponent maps the raw data based on the retrieved ontology.
- **Results Retrieval** - The User accesses the mapped data and any associated metadata or visualizations.

Optional steps:

- **Data Transformation Feedback** - Users can provide feedback on the mapping results, suggesting improvements or corrections, which can be utilized for refining mapping rules or ontology models.

Exception paths:

- If a chosen ontology model is not available, the system returns an error notification to the User.
- If mapping errors occur, the system logs details and notifies the User.

Post-conditions:

- Mapped data, conforming to the chosen ontology, is generated and available to the User.
- Feedback from users might initiate refinements in mapping rules or ontology models.

Trigger:

The User initiates the data mapping process by providing raw data and specifying the desired ontology model.

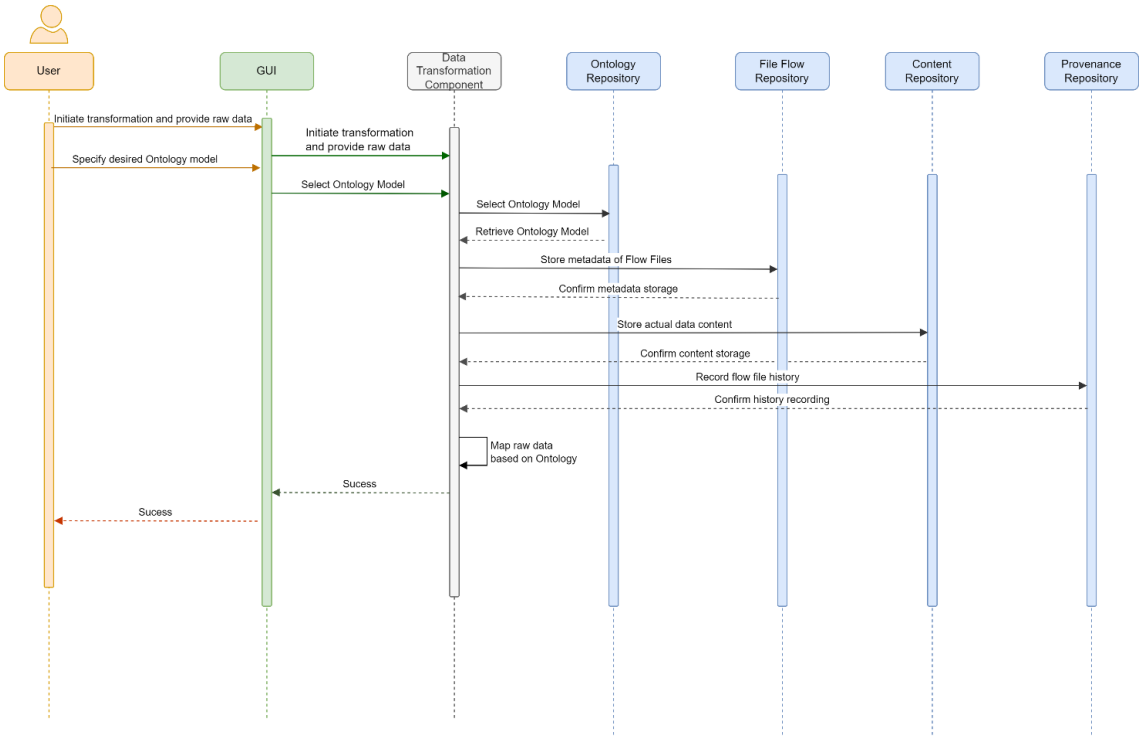

Figure 1: Data flow regarding ontological mapping of raw data

Transforming Varied File Formats

The use case "Transforming Varied File Formats" delves into the process by which a User interacts with the components of the "Data Transformation Services" to transform diverse input file formats like JSON, CSV, and text into a desired standardized format. This transformation ensures data consistency and prepares data for further downstream processing or integration with different systems.

Primary Actor: User (Data Integration Specialist, Data Scientist)

Secondary Actors: Data Extraction Subcomponent, Data Transformation Subcomponent

Preconditions:

- The Data Transformation Services are properly configured and operational.
- Necessary configurations or transformation rules for the intended format conversion are defined and available.
- Raw data in its initial format is available for processing.

Main flow:

- **File Submission** - The User submits the data file in its initial format (e.g., JSON, CSV, text) and specifies the desired output format.
- **Data Extraction** - The Data Extraction Subcomponent extracts content from the provided file, preparing it for transformation.
- **Data Transformation** - The Data Transformation Subcomponent processes the extracted content, converting it into the specified format.
- **Results Retrieval** - The User accesses the transformed data file, which is now in the desired standardized format.

Optional steps:

- **Transformation Feedback** - Post transformation, Users can provide feedback on the results, potentially suggesting improvements, refinements, or corrections which can then be used to refine future transformations.

Exception paths:

- If the provided file format is unsupported or corrupt, the system returns an error notification to the User.
- If transformation errors occur due to incorrect configurations or unexpected content, the system logs details and notifies the User.

Post-conditions:

- Data is transformed into the desired format and is ready for further processing, storage, or integration.
- Feedback mechanisms may instigate refinements in transformation rules or methods.

Trigger:

The User initiates the data transformation process by submitting a data file and specifying the desired output format.

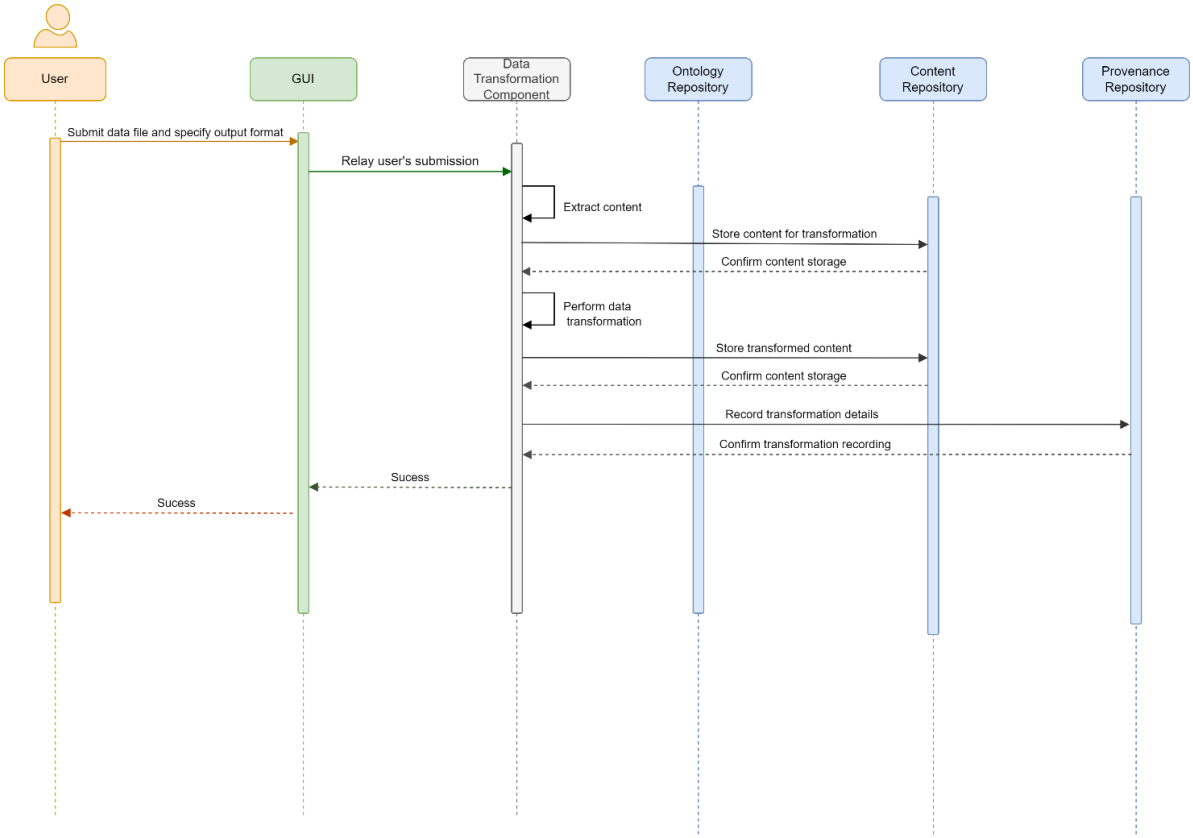

Figure 2: Data flow regarding data transformation example

5. Internal Architecture

The architecture of the platform is structured into three distinct layers: the Presentation Layer, the Service Layer, and the Persistence Layer. The delineation into these three layers embodies a standard architectural design aimed at logically organizing the system's components. This segregation promotes a clear separation of concerns, paving the way for easier maintenance and scalability of the platform.

- The **Presentation Layer** is engineered for user interaction and information presentation.
- The **Service Layer** encapsulates the core business logic, acting as a conduit between the presentation and persistence layers, mediating their interactions.

- The **Persistence Layer** is devoted to data storage and retrieval, providing a robust foundation for the platform's data-centric operations.

Each layer, with its set of dedicated functionalities, interacts cohesively to deliver the comprehensive capabilities of the CERTH Data Transformation Component. This layered design also augments the platform's adaptability, ensuring that modifications or extensions in one layer have minimal ripple effects on the others.

Presentation Layer:

- For RE4DY end-user or other components that will consume data transformation services there are no UIs available. All the calls will be done through Data Space Connectors between data providers and data consumers.
- **In general** User Interface in Apache NiFi serves as the primary interaction point for users, enabling them to design, monitor, and manage their data transformations but this will be used during development phase only. This UI is flexible, and efficient for diverse data integration needs and is going to support a lot the delivery of data transformation services.

Service Layer:

- **Flow Controller:** Is a core component of architecture. Essentially, it is the "brain" behind Data Transformation component operations, orchestrating the processing of data and ensuring that all NiFi and RE4DY custom processors work in harmony.
- **Data Loading sub-component:** Is a custom designed component for data loading, tailored to address unique RE4DY requirements. The component supports functions such as source integration, flexible data ingestion, and error handling.
- **Data Extraction sub-component:** Is a custom designed component for data cleaning and filtering (if needed).
- **Data Transformation sub-component:** Is a custom designed RE4DY data transformation processor, built on top of the Apache NiFi data transformation processors. Some of the envisioned functionalites so far are:
    - **Data Conversion:** Transform data from one format or structure to another. JOLT processors are used as well at this part.
    - **Data Enrichment:** Augmenting data with additional information.
    - **Data Aggregation:** Summarizing or grouping data.
    - **Business Logic Application:** Applying specific logic or rules.
    - **Data Mapping:** Mapping the data to the standardized data models and ontologies selected to be used by the RE4DY use case providers.
- **Data Space Connectors**[4]**:** They are enabling trusted and sovereign communication between two parties that are involved in a data transformation service. A data consumer

[4] https://international-data-spaces-association.github.io/DataspaceConnector/

that needs a data source with a specific data format will setup a data space connector. This connector will communicate with the relevant data space connector of the data provider. The data from provider data sources will be transformed through the Data Transformation Processors and the result will be transmitted to the consumer.

Persistence Layer:

- **Flow File Repository:** It is responsible for storing the metadata of the Flow Files that are currently being processed by the system. In essence, the Flow File Repository is crucial for ensuring data integrity, system recoverability, and providing a snapshot view of the current state of data.
- **Content Repository:** Is a core component of Apache NiFi that manages the actual content or data associated with the FlowFiles being processed in the system. Unlike the FlowFile Repository, which handles metadata, the Content Repository deals with the data payload.
- **Provenance Repository:** The Provenance Repository is responsible for recording and preserving a comprehensive history of each Flow File that flows through the system. This detailed log enables users to trace the lineage and lifecycle of the data as it is processed.
- **Ontology Repository:** Ontology Repository in the context of data transformation provides a structured, semantic framework that ensures that data is not just transformed in structure but also in meaning. By mapping data to standardized ontologies, it provides a way to ensure data consistency, integration, and meaningful representation across diverse RE4DY use cases.

- **External Data Sources:** The actual data sources from RE4DY partners that is going to be transformed.

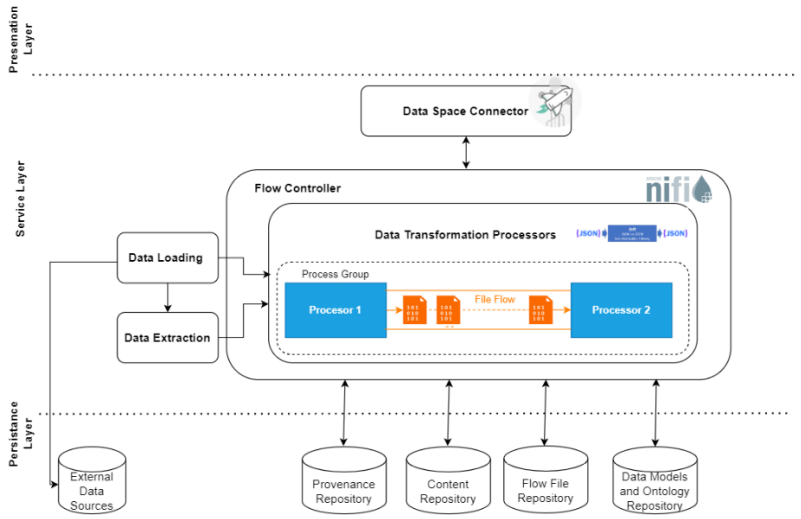

Figure 3: Architecture of RE4DY Data Transformation Services

6. API

TBA in the updated version of the deliverable.

7. Implementation Technology

The combination of IDSA Connectors, Apache NiFi and JOLT for our custom data extraction component ensures that we have both a robust and scalable data extraction framework, as well as specialized capabilities for JSON data transformations. Apache NiFi offers us the infrastructure and environment to manage large-scale data flows, while JOLT provides the specificity and flexibility needed for JSON data manipulations. Together, they form a potent tech stack for our custom data extraction needs.

8. Comments

None.