RE4DY

MANUFACTURING DATA NETWORKS

Title	D3.3: Qualified digital continuum 4.0 open toolkit & pan-European resilient data space TEF
Document Owners	INTRA
Contributors	WP3 partners
Dissemination	Public
Date	11/12/2024
Version	V1.0

Document Status

Deliverable Leader	INTRA
Internal Reviewer 1	INNO
Internal Reviewer 2	SANDETEL
Work Package	WP3: Continuity Management Toolkit for Industrial Digital Thread & Cognitive Twin Fabrics
Deliverable	D3.4: Qualified digital continuum 4.0 open toolkit & pan-European resilient data space TEF
Due Date	31/11/2024
Delivery Date	11/12/2024
Version	V1.0

Version history

09/05/2024	Table of Contents
25/10/2024	Tools descriptions
30/10/2024	Experiments descriptions
29/11/2024	TEFs Contributions
02/12/2024	Version ready for review
11/12/2024	Final version

Project Partners

#	Acronym	Participant organisation name
1	INNO	ASOCIACIÓN DE EMPRESAS TECNOLÓGICAS INNOVALIA
2	CHAL	CHALMERS TEKNISKA HOGSKOLA AB
3	IDSA	INTERNATIONAL DATA SPACES EV
4	VWAE	VOLKSWAGEN AUTOEUROPA, LDA
5	CEIT	ASSECO CEIT AS
6	UNI	UNINOVA-INSTITUTO DE DESENVOLVIMENTO DE NOVAS TECNOLOGIAS-ASSOSIACAO
7	FILL	FILL GESELLSCHAFT MBH
8	AVL	AVL LIST GMBH
9	VIS	VISUAL COMPONENTS OY
10	UMH	UNIVERSIDAD MIGUEL HERNANDEZ DE ELCHE
11	ATLANTIS	ATLANTIS ENGINEERING AE
12	DATA	DATAPIXEL SL
13	CORE	CORE KENTRO KAINOTOMIAS AMKE
14	UiO	UNIVERSITETE I OSLO
15	AVIO	GE AVIO
16	ENG	ENGINEERING-INGENIERIA INFORMATICA SPA
17	POLIMI	POLITECNICO DI MILANO
18	AtoS	ATOS IT SOLUTIONS AND SERVICES IBERIA SL
18.1	AtoS-ES	ATOS SPAIN SA
19	KU	KATHOLIEKE UNIVERSITEIT LEUVEN
20	INTRA	NETCOMPANY-INTRASOFT SA
21	NOVA	NOVA ID FCT - ASSOCIACAO PARA A INOVACAO E DESENVOLVIMENTO DA FCT
22	ICF	INDUSTRY COMMONS FOUNDATION (INSAMLINGSSTIFTELSE)
23	CERTH	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS
24	S21SEC	GRUPO S 21SEC GESTION SA
25	UPV	UNIVERSITAT POLITECNICA DE VALENCIA
26	CNR	CONSIGLIO NAZIONALE DELLE RICERCHE
27	SANDETEL	SOCIEDAD ANDALUZA PARA EL DESARROLLO DE LAS TELECOMUNICACIONES SA
28	SSF	SWITZERLAND INNOVATION PARK BIEL/BIENNE AG
29	GFMS ADVMAN	GF MACHINING SOLUTIONS AG
30	Fraisa SA	FRAISA SA
31	SIE	SIEMENS SCHWEIZ AG
32	MIRA	MIRAITEK

Executive Summary

This deliverable, "Qualified Digital Continuum 4.0 Open Toolkit & Pan-European Resilient Data Space TEF", showcases the RE4DY project's progress in advancing Industry 4.0 technologies through a set of specialized toolkits tailored to specific industrial challenges. Building upon the foundational work delivered in D3.2, this document extends the toolkits' functionalities and features fostering seamless data sharing, federated analytics, digital twin implementations and advanced visualization. Each tool included in the toolkit is aligned with the RE4DY architecture, addressing the emerging needs of the diverse use cases and experimental scenarios.

D3.3 demonstrates how these tools were integrated and validated through experimentation. Each experiment leverages a combination of tools to address a real-world industrial challenge and providing quantified KPIs illustrating the results of these end-to-end cases such as predictive maintenance, digital continuity, and secure data exchange etc. Additionally, this deliverable introduces the concept and implementation of a network among Testing and Experimentation Facilities (TEF) as a foundational platform for fostering industrial innovation and collaboration across diverse ecosystems.

This document includes updated descriptions of toolkit components, emphasizing their modular design and integration capabilities within the RE4DY architecture. Additionally, it provides recommendations and a roadmap for ensuring the toolkit's sustainability and broader adoption across the European industrial landscape.

By addressing industrial challenges and aligning with interoperability standards, D3.3 ensures that the RE4DY Digital Continuum 4.0 Toolkit can serve as a cornerstone for advancing digital and physical system integration in Europe.

Contents

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth Generation
5QI	5G QoS Identifier
AAS	Asset Administration Shell
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
API	Application Programming Interface
ARP	Allocation and Retention Priority
BART	Bidirectional and Auto-Regressive Transformers
BDA	Big Data Application
BERT	Bidirectional Encoder Representations from Transformers
BFO	Basic Formal Ontology
BLER	Block Error Rate
CAD	Computer Aided Design
CGIs	Cell Global Identifiers
CI/CD	Continuous Integration/Continuous Delivery
CMM	Coordinate Measuring Machine
CN	Core Network
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
CQI	Channel Quality Indicator
CSV	Comma-Separated Values
DaaP	Data as a Product
DAPS	Dynamic Attributes Provisioning Service
DB	Database
DC	Data Container
DCI	Data Container Invoker
DCP	Data Connection Profile
DIDI	Dataspace for Industrial Data Intelligence
DL	Downlink
DN	Data Network
EDC	Eclipse Data Connector
EDM	Electric Discharge Machining
eIDAS	electronic Identification, Authentication and Trust Services
EPO	European Patent Office
ETL	Extract Transform Load
FAIR	Findability, Accessibility, Interoperability and Reusability
FEDMA	Federated Maintenance for Milling Machines

FML	Federated Machine Learning
FPdM	Federated Predictive Maintenance
GDPR	General Data Protection Regulation
GFBR	Guaranteed Flow Bit Rate
gNB	Next Generation Node B
GPSI	Generic Public Subscription Identifier
GPSI	Generic Public Subscription Identifier
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
HTTP	HyperText Transfer Protocol
IAM	Identity and Access Management
ICCID	Integrated Circuit Card ID
ID	Identifier
IDS	International Data Spaces
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IMSI	International Mobile Subscriber Identity
iOS	iPhone Operating System
IoT	Internet of Things
IP	Internet Protocol
IP	Intellectual Property
IPC	Industrial PCs
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IT	Information Technology
JS	JavaScript
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Tokens
K8S	Kubernetes
KC	Keycloak
KG	Knowledge Graph
KPI	Key Performance Indicator
LLM	Large Language Model
MAC	Media Access Control
MCSs	Modulation Coding Schemes
MFBR	Maximum Flow Bit Rate
ML	Machine Learning
MLOPS	Machine Learning Operations
MQTT	Message Queuing Telemetry Transport

MRD	Meta Repository Demonstrator
MSISDN	mobile subscriber ISDN number
NC	Numerical Control
NEF	Network Exposure Function
NLP	Natural Language Processing
NoSQL	Not Only SQL
NOVAAS	NOVA Asset Administration Shell
NPN	Non-Public Network
NS	Network Slices
NW	Network
NWDAF	Network Data Analytics Function
OAuth2.0	Open Authorization 2.0
OPC	Open Platform Communications
OPC-UA / OPCUA	Open Platform Communications United Architecture
OT	Operational Technology
PCB	Printed Circuit Board
PCF	Policy Control Function
PDB	Packet Delay Budget
PDU	Packet Data Unit
PDU	Packet Data Unit
PEI	Permanent Equipment Identifier
PER	Packet Error Rate
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
POC	Proof of Concept
QFI	QoS Flow ID
QIF	Quality Information Framework
QoS	Quality of Service
RAN	Radio Access Network
RDTM	Realtime Digital Twin Model
Redis	Remote Dictionary Server
ResNet	Residual Neural Network
REST	Representational State Transfer
RF	Radio Frequency
ROS	Read-Only Storage
RQA	Reflective QoS Attribute
RRC	Radio Resource Control
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality

RSSI	Received Signal Strength Indicator
RUL	Remaining Useful Life
SCRO	Supply Chain Reference Ontology
SEAL	Service Enabler Architecture Layer
SFTP	Secure File Transfer Protocol
SIM	Subscriber Identity Module
SINR	Signal Interference Noise Ratio
SLA	Service Level Agreements
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SOAP	Simple Object Access Protocol
SPN	Service Provider Name
SQL	Structured Query Language
SSF	Smart Systems Factory
SSF	Swiss Smart Factory
SSO	Single Sign-On
SST	Slice/Service type
TEF	Testing and Experimentation Facilities
THT	Through-Hole Technology
TLS	Transport Layer Security
TSC	Time Sensitive Communications
TSCAI	TSC Assistance Information
TSN	Time Sensitive Networking
TVT	Transactions on Vehicular Technology
UA	Unified Architecture
UE	User Equipment
UI	User Interface
UL	Uplink
UML	Unified Modeling Language
UPF	User Plane Function
URL	Uniform Resource Locator
UX	User Experience
V2N2V	Vehicle-to-Network-to-Vehicle
V2X	Vehicle-to-Everything
VLAN	Virtual Local Area Networks
WIPO	World Intellectual Property Organization
WP	Work Package
XAI	Explainable Artificial Intelligence
xCDT	Executable Cognitive Digital Twin
XG-Boost	eXtreme Gradient Boosting

XML	Extensible Markup Language
Yolov3	You Only Look Once Version 3
Yolov5	You Only Look Once Version 5

1 Introduction

1.1 Purpose and scope

Deliverable D3.3 concludes the efforts initiated in WP3, "Continuity Management Toolkit for Industrial Digital Thread & Cognitive Twin Fabrics," which seeks to establish a network of Testing and Experimentation Facilities (TEFs) and European Data Spaces supported by a set of tools comprising the RE4DY toolkit.

Building on the achievements of D3.2, which introduced the first-generation Digital Continuum 4.0 Open Toolkit and its experimentation plan, this deliverable, D3.3, advances the toolkit by incorporating feedback from earlier phases of the RE4DY project. While D3.2 contained 26 tools, this deliverable narrows its focus to 14 tools—some of which are newly introduced, while others are upgraded or extended versions of tools from D3.2. Tools not covered in this deliverable remain part of the broader RE4DY toolkit, with their functionalities and roles either unchanged since D3.2 or outside the primary scope of this iteration. This deliverable leverages the experimentation frameworks and methodologies established in WP3 and provides an in-depth exploration of the advancements made in the design, integration, and experimentation of the tools.

It starts by highlighting the architectural improvements, enhanced APIs, and expanded interoperability features for the upgraded tools. This leads to extended functionalities, and support for seamless integration across various systems addressing the increasingly complex demands of the industrial ecosystems.

Additionally, it presents experimental results and insights derived from applying the toolkit in diverse industrial scenarios, demonstrating its impact on addressing critical challenges in industrial digitalization. The validation within the Testing and Experimentation Facilities (TEFs) emphasize the adaptability and applicability to real-world industrial scenarios. This serves as a foundational step toward large-scale trials and broader adoption within the RE4DY framework, laying the groundwork for the next phases of the project.

1.2 Relation to other project WPs and Tasks

Deliverable D3.3 plays a pivotal role in the RE4DY project by bridging the outputs of foundational work packages with the implementation and validation activities in large-scale trials. It builds upon the architectural designs, resiliency strategies, and experimental frameworks developed in earlier stages while preparing the groundwork for pilot and trial phases under WP4 and WP5.

Within WP3, D3.3 represents a significant advancement from the first-generation toolkit introduced in D3.2. It serves as a key input to WP4 and WP5, which focus on large-scale trials and process optimization in smart connected factories. The toolkit's validation within TEFs, as described in this deliverable, supports WP4's activities related to pilot setup and experimentation. Similarly, WP5 leverages the advancements described in D3.3 to optimize distributed industrial processes, including predictive maintenance and collaborative manufacturing workflows.

In summary, Deliverable D3.3 reflects the iterative and collaborative nature of the RE4DY project, bridging the foundational work of WP2 and WP3 with the validation and implementation phases of WP4 and WP5.

1.3 Structure of the document

The document is structured as follows:

Section 1 serves as an introduction, providing the purpose and scope of the current deliverable, its structure and its relationship with other deliverables and other project work packages and tasks.

Section 2 describes the tools included in the Digital Continuum 4.0 Toolkit, highlighting both upgrades to tools introduced in D3.2 and new tools added in D3.3, along with their functionalities, integration and technical details.

Section 3 explains how the toolkit aligns with the RE4DY Reference Architecture and contributes to achieving the project's digitalization objectives.

Section 4 presents the experiments, the methodology, scenarios, and the results validating the upgraded and new toolkit components.

Section 5 discusses the role of the Testing and Experimentation Facilities (TEFs) in validating the toolkit and fostering industrial collaboration.

Section 6 summarizes the key outcomes of this deliverable and provides the final conclusions.

2 RE4DY Data Space and Toolware Catalogue

This chapter provides an updated catalogue of tools included in the RE4DY Digital Continuum 4.0 Toolkit, focusing on 14 tools presented below. These tools fall into two categories:

- New Additions: Tools that were not part of D3.2 but introduced in D3.3.
- Upgraded Tools: Existing tools from D3.2 that have undergone enhancements, modifications, or extensions to expand their functionalities.

To facilitate comparison, tools retained from D3.2 are cross-referenced with their corresponding subsection numbers in the earlier deliverable in Table 1 below.

Table 1: Mapping of Tools Between D3.2 and D3.3 (Status and References)

TOOL (D3.3)	NEW (N)/ UPDATED (U)	D3.2 SECTION REFERENCE
NOVA Asset Administration Shell (NOVAAS)	U	2.10
FEDMA - Federated Maintenance for Milling Machines	U	2.8
Data Container	U	2.24
Sovity Open-Source EDC Connector	N	-
ICF Knowledge Graph Visualisation Environment	U	2.1
ICF Meta Repository Demonstrator	U	2.2
ICF IPscreener Integration	U	2.15
5G Digital Twin (5G AAS)	U	2.25
Realtime Digital Twin Model (RDTM)	N	-
ALIDA	U	2.9
CERTH Sovereign Data Transformation Service	U	2.5
CERTH XAI and Active Learning Platform for Defect Detection	U	2.6
Federated Predictive Maintenance (FPdM)	U	2.4
Analysis Center	N	-

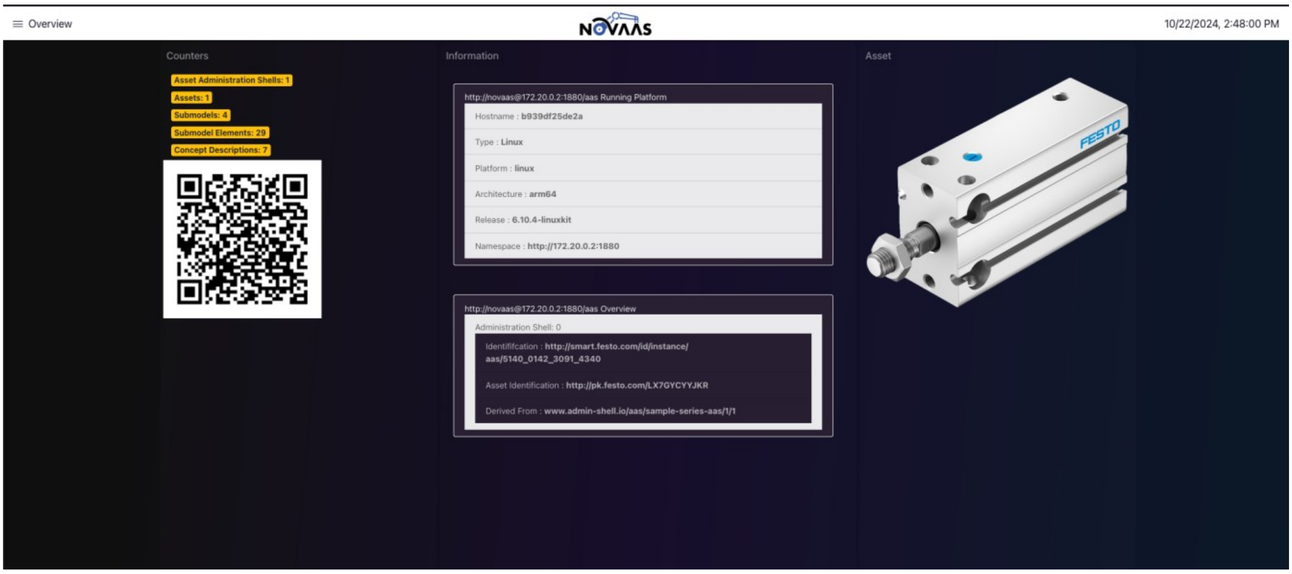
2.1 NOVA Asset Administration Shell (NOVAAS)

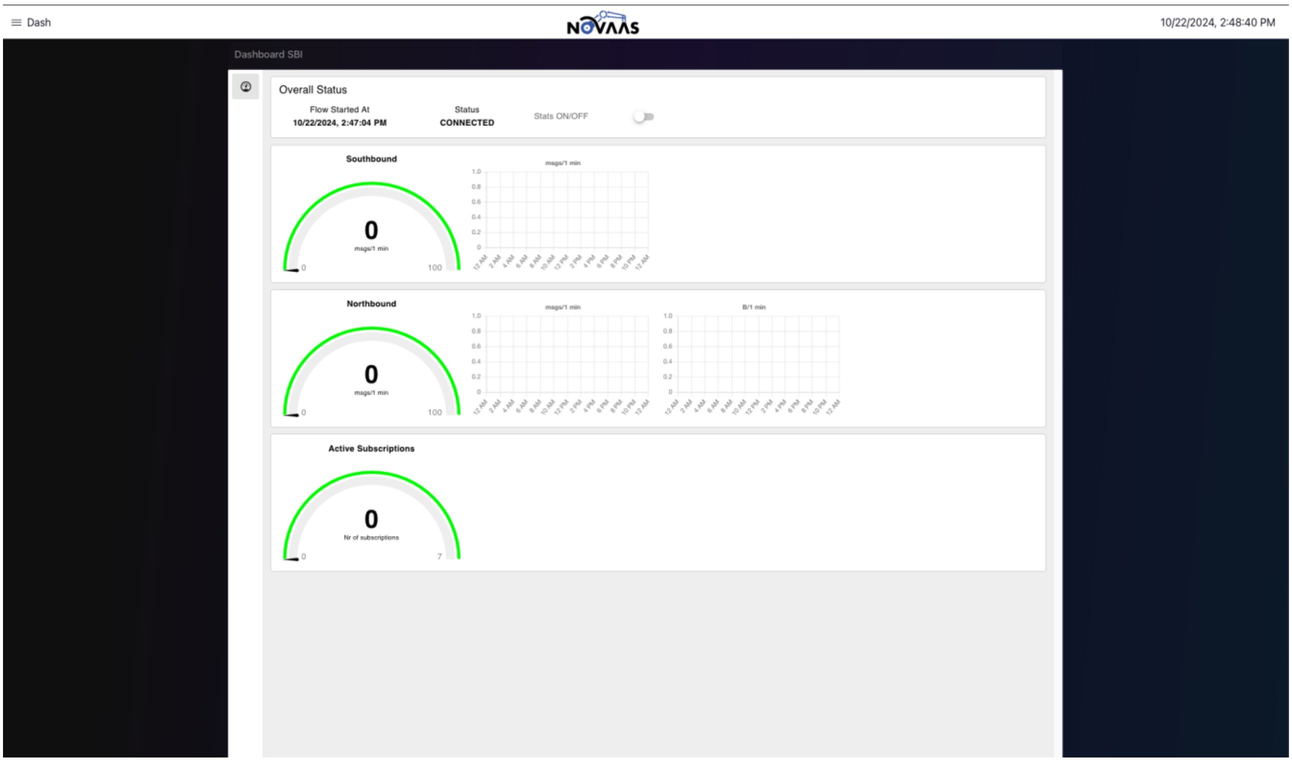
2.1.1 Description

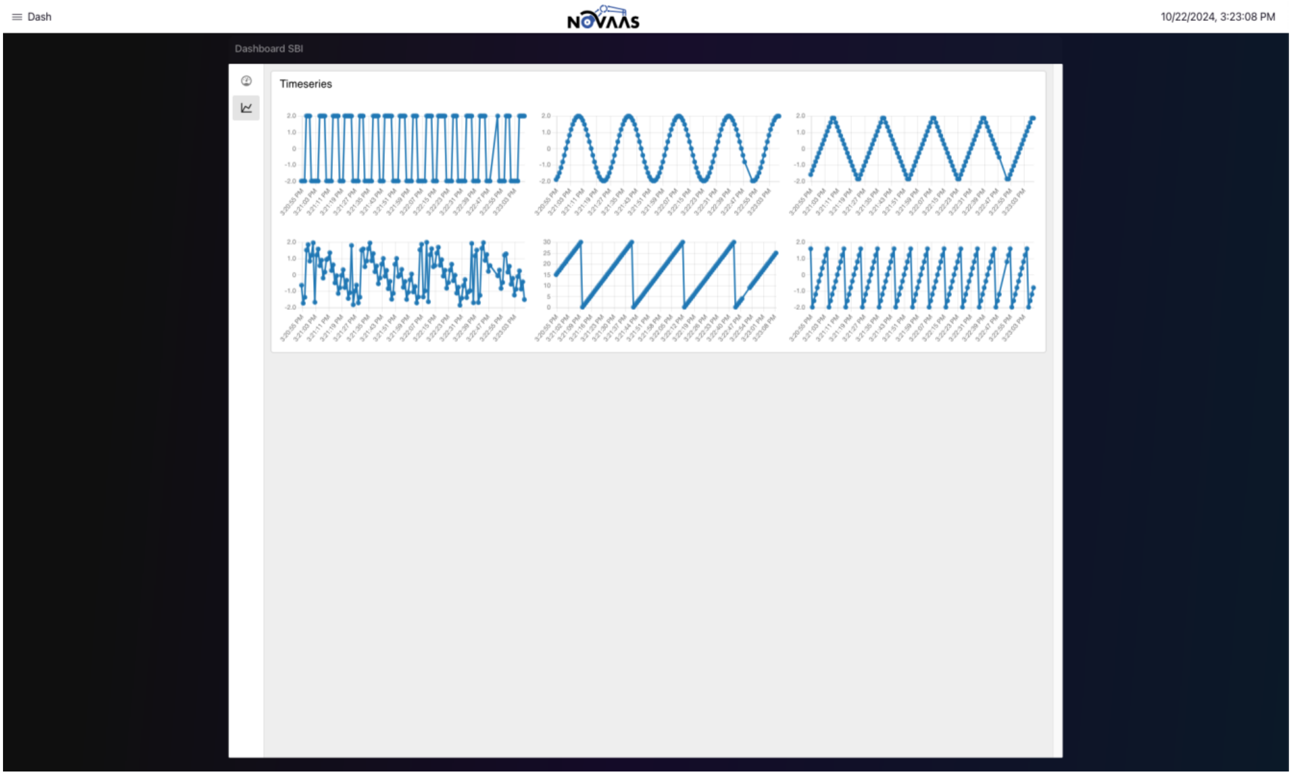
NOVAAS is an open-source implementation of the RAMI4.0 Asset Administration Shell concept. The Asset Administration Shell (AAS) is a key concept within the context of the Industrial Internet of Things (IIoT) and Industry 4.0. It is a standardized framework for describing and managing industrial assets and their digital representations ("data image" of the asset) in a way that enables seamless interoperability between various components and systems in industrial environments. The AAS is used to digitalize any physical asset in order to be integrated seamlessly into and I4.0 compliant system. Briefly, the AAS implements the Digital Twin concept for Industry 4.0. NOVAAS is currently compliant with the V2 specification of the AAS as well as latest version V3 (Developed in the context of the RE4DY project). The tool is implemented using the next-generation software development principles i.e. using low/no code platforms, microservices and APIs and containers. In the context of RE4DY the NOVAAS will be used for designing and developing I4.0 compatible Digital Twins. It will be used to provide and harmonized and standardized connection to the physical asset within a manufacturing production system. Moreover, NOVAAS can be used to run AI algorithms at the edge for several applications such as quality inspection, predictive maintenance, dashboarding etc.. Finally, the NOVAAS (as an implementation of the AAS concept) will enable the creation of data pipelines for seamless data flows across the production lifecycle (Digital Thread).

2.1.2 Input

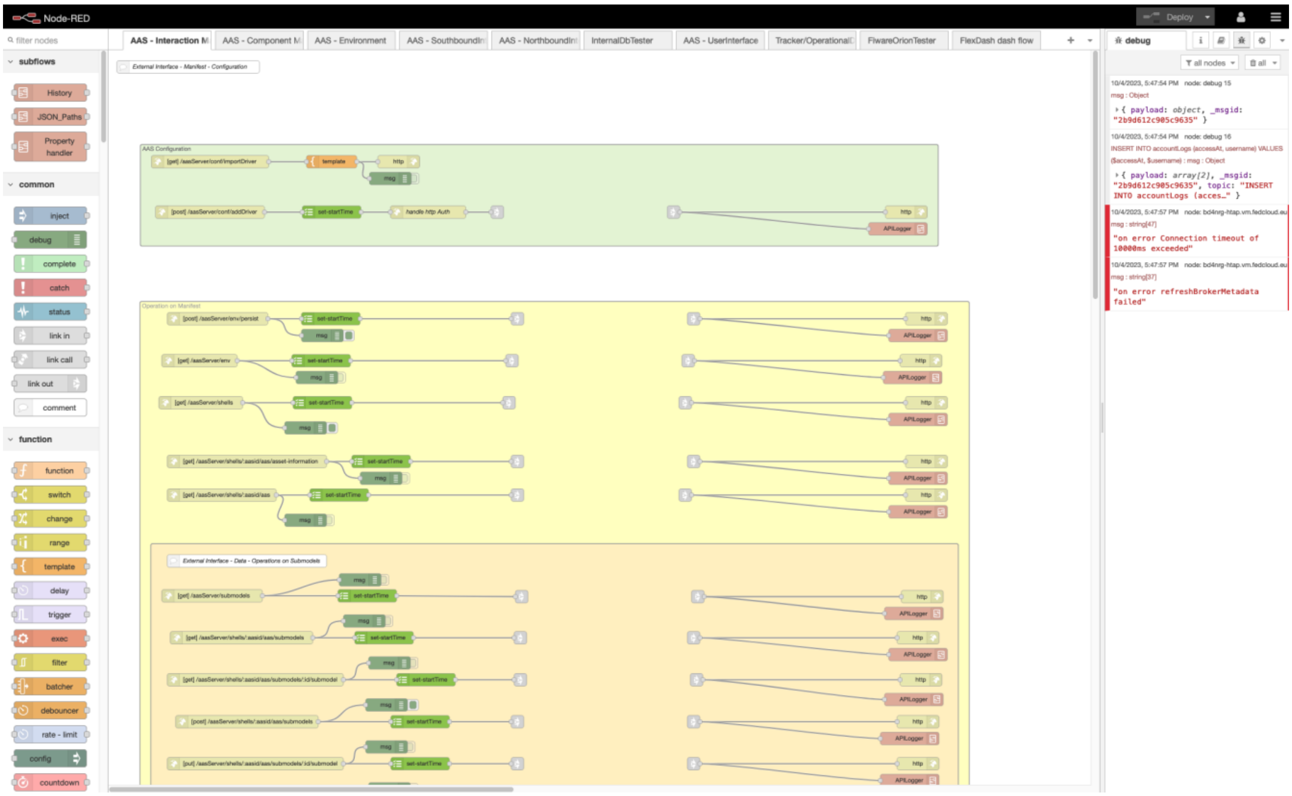
- The tool has a user interface available on ip:port/ui that shows the description of the physical asset in terms of submodels and related properties. The submodels and properties can be used to model both static and dynamic information. Each property can be semantically describe using the concept description class (this is also shown within the UI). The UI also provides plotting capabilities. It allows to plot the real time data extracted from the physical asset.


Overview
NOVAAS
10/22/2024, 2:48:00 PM
Counters
Asset Administration Shells: 1
Assets: 1
Submodels: 4
Submodel Elements: 29
Concept Descriptions: 7
Information
http://novaas@172.20.0.2:1880/aas Running Platform
Hostname : b939df25de2a
Type : Linux
Platform : linux
Architecture : arm64
Release : 6.10.4-linuxkit
Namespace : http://172.20.0.2:1880
http://novaas@172.20.0.2:1880/aas Overview
Administration Shell: 0
Identififcation : http://smart.festo.com/id/instance/aas/5140_0142_3091_4340
Asset Identification : http://pk.festo.com/LX7GYCYYJKR
Derived From : www.admin-shell.io/aas/sample-series-aas/1/1
Asset
FESTO


Dash
NOVAAS
10/22/2024, 2:48:40 PM
Dashboard SBI
Overall Status
Flow Started At
10/22/2024, 2:47:04 PM
Status
CONNECTED
Stats ON/OFF
Southbound
0
msgs/1 min
Northbound
0
msgs/1 min
B/1 min
Active Subscriptions
0
Nr of subscriptions



- The backend of the tool is accessible by using the ip:port link (see the image below).

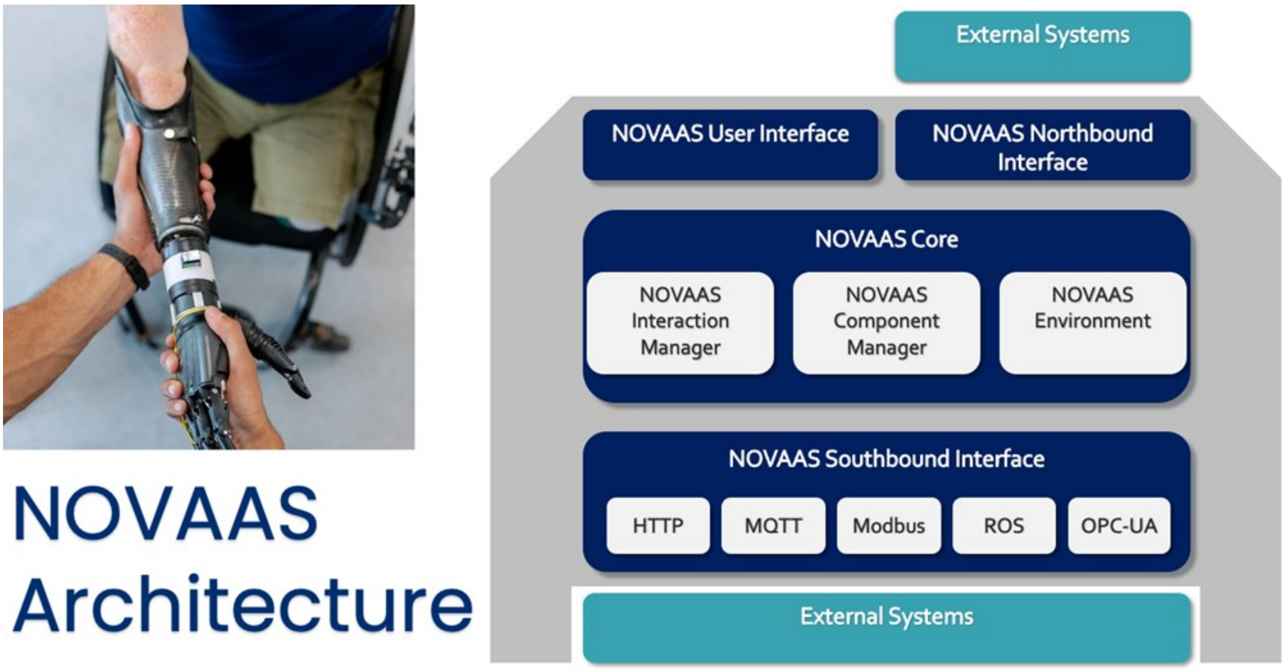


- The tool is totally generic; however it requires the development of a simple connector to enable the connection with the asset. The technology used depends on the asset, however the tool supports a large number of connectors (ROS, Modbus, OPC-UA, MQTT, HTTP, etc) basically the whole ecosystem provided by Node-Red (that is the development platform).

2.1.3 Output

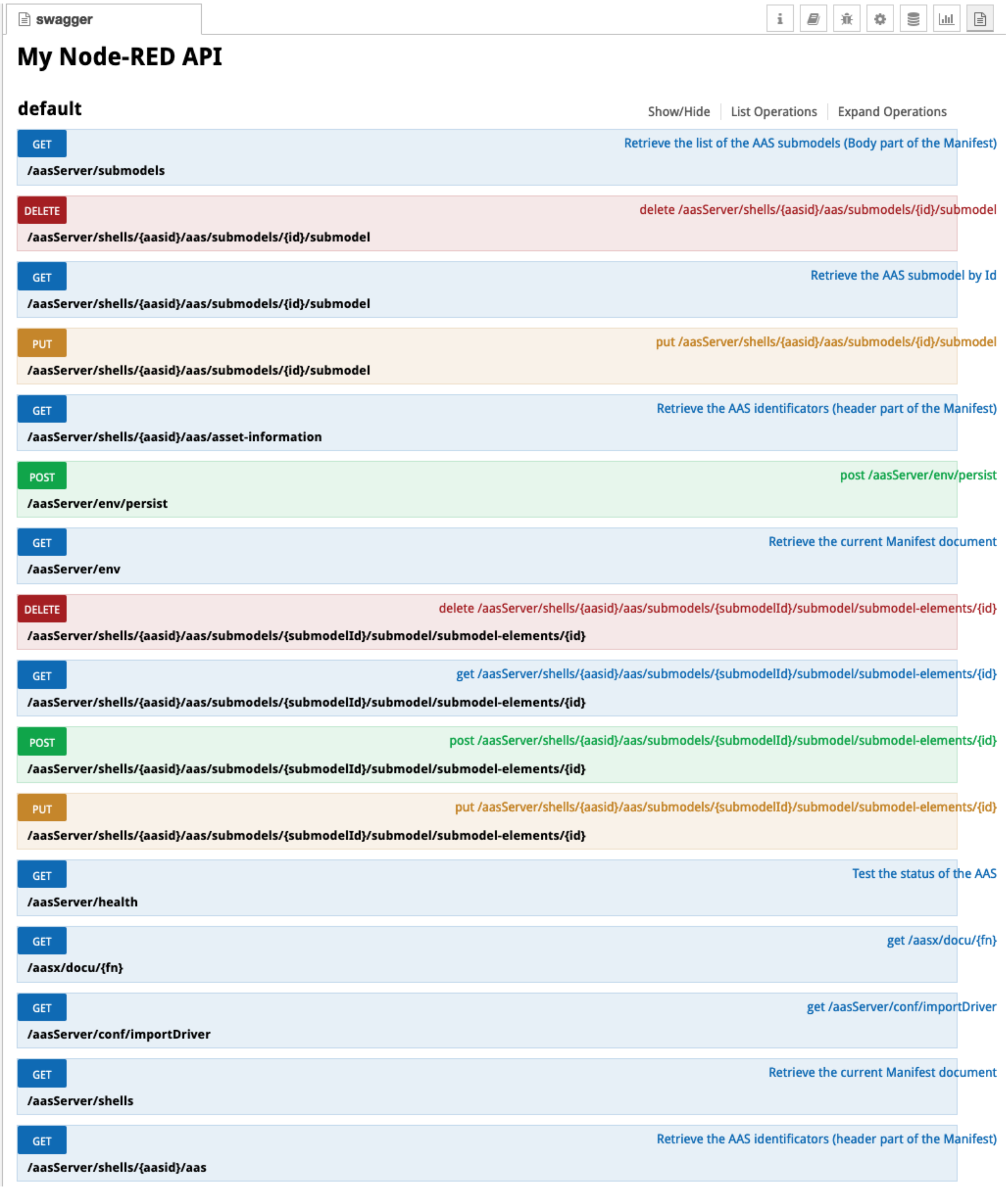
- All the data from the asset can be accessed by using the REST API (standardized API).
- Event based communication is also supported. The technology adopted is MQTT and the events are following the V2/V3 AAS specifications.

2.1.4 Internal Architecture



2.1.5 API

- The API is documented in V2 AAS specifications that can be retrieved from: https://industrialdigitaltwin.org/en/
- Internally the API is also documented using a swagger plug in


swagger
My Node-RED API
default
Show/Hide | List Operations | Expand Operations
GET /aasServer/submodels — Retrieve the list of the AAS submodels (Body part of the Manifest)
DELETE /aasServer/shells/{aasid}/aas/submodels/{id}/submodel — delete /aasServer/shells/{aasid}/aas/submodels/{id}/submodel
GET /aasServer/shells/{aasid}/aas/submodels/{id}/submodel — Retrieve the AAS submodel by Id
PUT /aasServer/shells/{aasid}/aas/submodels/{id}/submodel — put /aasServer/shells/{aasid}/aas/submodels/{id}/submodel
GET /aasServer/shells/{aasid}/aas/asset-information — Retrieve the AAS identificators (header part of the Manifest)
POST /aasServer/env/persist — post /aasServer/env/persist
GET /aasServer/env — Retrieve the current Manifest document
DELETE /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id} — delete /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id}
GET /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id} — get /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id}
POST /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id} — post /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id}
PUT /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id} — put /aasServer/shells/{aasid}/aas/submodels/{submodelId}/submodel/submodel-elements/{id}
GET /aasServer/health — Test the status of the AAS
GET /aasx/docu/{fn} — get /aasx/docu/{fn}
GET /aasServer/conf/importDriver — get /aasServer/conf/importDriver
GET /aasServer/shells — Retrieve the current Manifest document
GET /aasServer/shells/{aasid}/aas — Retrieve the AAS identificators (header part of the Manifest)

2.1.6 Implementation Technology

- Node-RED: Node-RED is an open-source flow-based development tool and runtime environment for connecting hardware devices, APIs, online services, and various software applications. It provides a visual programming interface that allows users, often referred to as "makers" or developers, to create and deploy Internet of Things (IoT) applications and automation tasks without the need for extensive coding. Node-RED is a flow-based, visual development platform and runtime environment that simplifies the creation of IoT applications and automation workflows. It offers a web-based, drag-and-drop interface for building and connecting nodes (blocks) that represent different functions and devices. These nodes can include data sources, data processors, output actions, and more. Users can create complex workflows by connecting these nodes in a visual manner, and Node-RED handles the underlying logic and communication between components.
- Javascript: most of the more complex logic is implemented using javascript.

Platform:

NOVAAS has been built since the very beginning to be cloud-native.

- Kubernetes
- Docker
- Swarm

2.1.7 Comments

None

2.2 FEDMA - Federated Maintenance for Milling Machines

2.2.1 Description

FEDMA's primary objectives focus on predicting both the Wear and Remaining Useful Life (RUL) of FRAISA's milling tools while simultaneously detecting anomalies in operational data. These goals are closely aligned with FRAISA's operational procedures and are supported by data and metadata from GF's My rConnect program. Central to this solution is federated learning, an advanced technique that ensures data privacy and security while reducing latency and minimizing bandwidth usage by limiting data transfers to a central server, fully compliant with GDPR regulations. This approach allows the model to learn from data contributed by multiple machines, significantly enhancing its predictive accuracy. As a result, FEDMA's components streamline operations by providing valuable insights into tool wear, optimizing RUL predictions, and improving anomaly detection.

2.2.2 Input

The FEDMA components receive milling operation data, including:

- CSN files with raw signals data such as vibrations, forces, and temperatures.
- JSON files with metadata like job name, program and duration.
- Essential insights, machining strategies (axial infeed depth (ap), radial infeed depth (ae)), tool geometry, and workpiece material.

Each FEDMA AI model is triggered by specific conditions or actions upon receiving new data, initiating the training process. Once local model training is complete, only the model's parameters are transmitted to the server for aggregation. With this approach, the global model receives only the parameters from the local models, ensuring that no initial data is transmitted. This federated learning framework enhances data privacy and security by keeping sensitive information localized, while still allowing the global model to improve through aggregated knowledge from multiple sources.

2.2.3 Output

The output of the FEDMA component will provide precise predictions for the current wear of each milling tool, including both the mean wear (Vb) and maximum wear (Vbmax) values, along with the tool's Remaining Useful Life (RUL).

Additionally, the output will feature a detailed visualization of the tool's wear history, showing how the Vb and Vbmax values have fluctuated across different operational jobs.

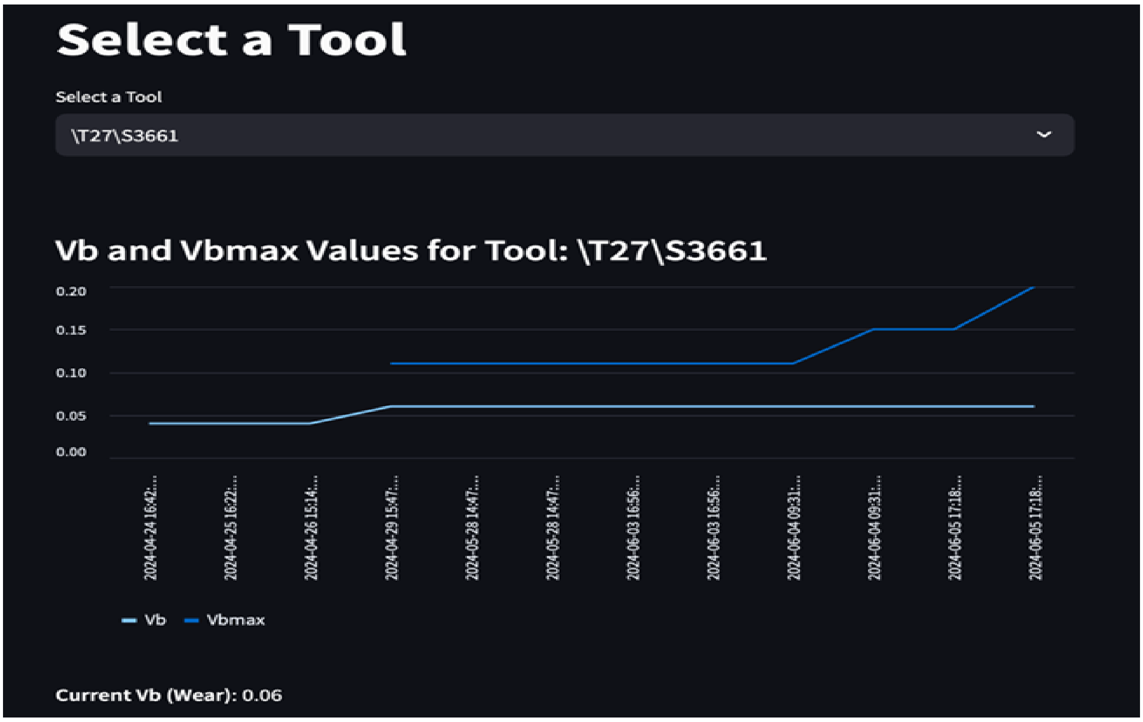


Figure 1: Tool Wear Predictions: Vb & Vbmax Visualization Across Operational Jobs

The RUL is calculated based on the highest wear threshold observed for each tool type during experiments or set by expert guidelines.

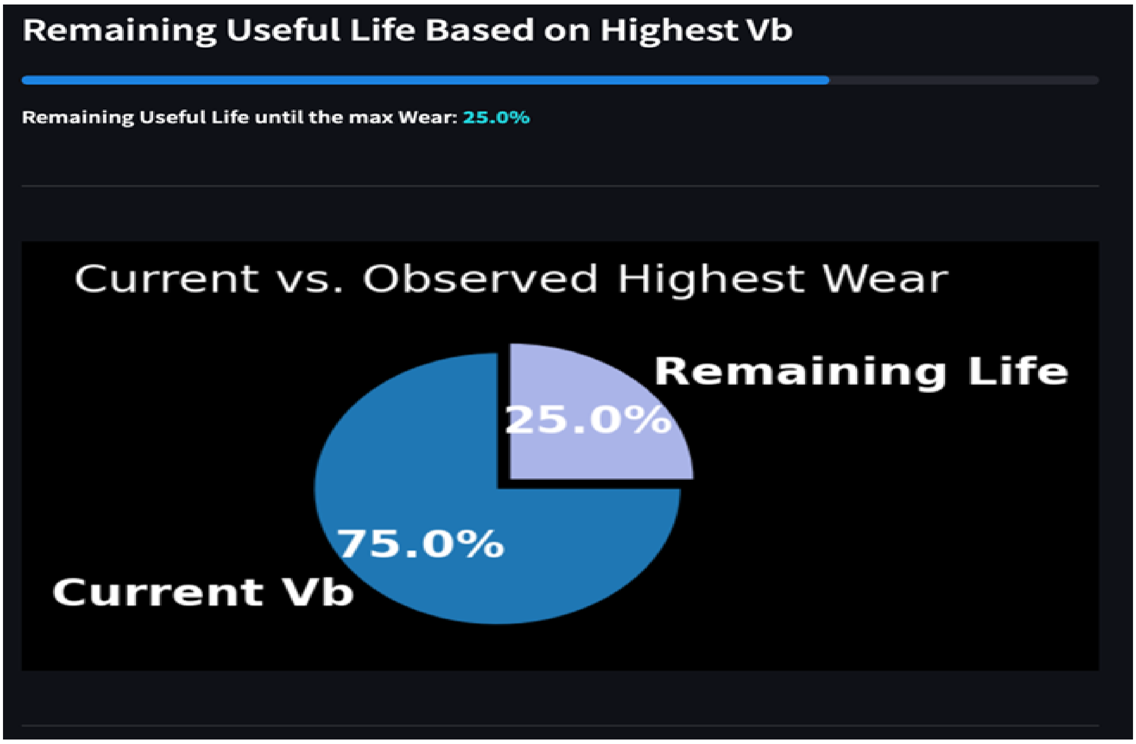


Figure 2: Remaining Useful Life (RUL) Prediction: Tool Longevity Until End of Usage

By leveraging advanced algorithms, this approach streamlines operations by delivering actionable insights into tool wear and RUL, enabling operational teams to make informed decisions without the need for frequent tool removal and manual wear assessments. This optimizes time and resources, facilitates better maintenance scheduling, reduces downtime, and maximizes each tool's productive life, contributing to more efficient and cost-effective operations.

2.2.4 Information Flow

The FEDMA component follows a systematic, privacy-preserving approach to predict the wear and the remaining life of milling tools through federated learning. By leveraging operational data from CNC machines and deploying advanced models at the edge, FEDMA ensures seamless integration and accurate predictions without compromising data privacy The following steps outline the process, from data collection to model deployment and global aggregation, ensuring seamless and accurate predictions across all connected CNC machines.

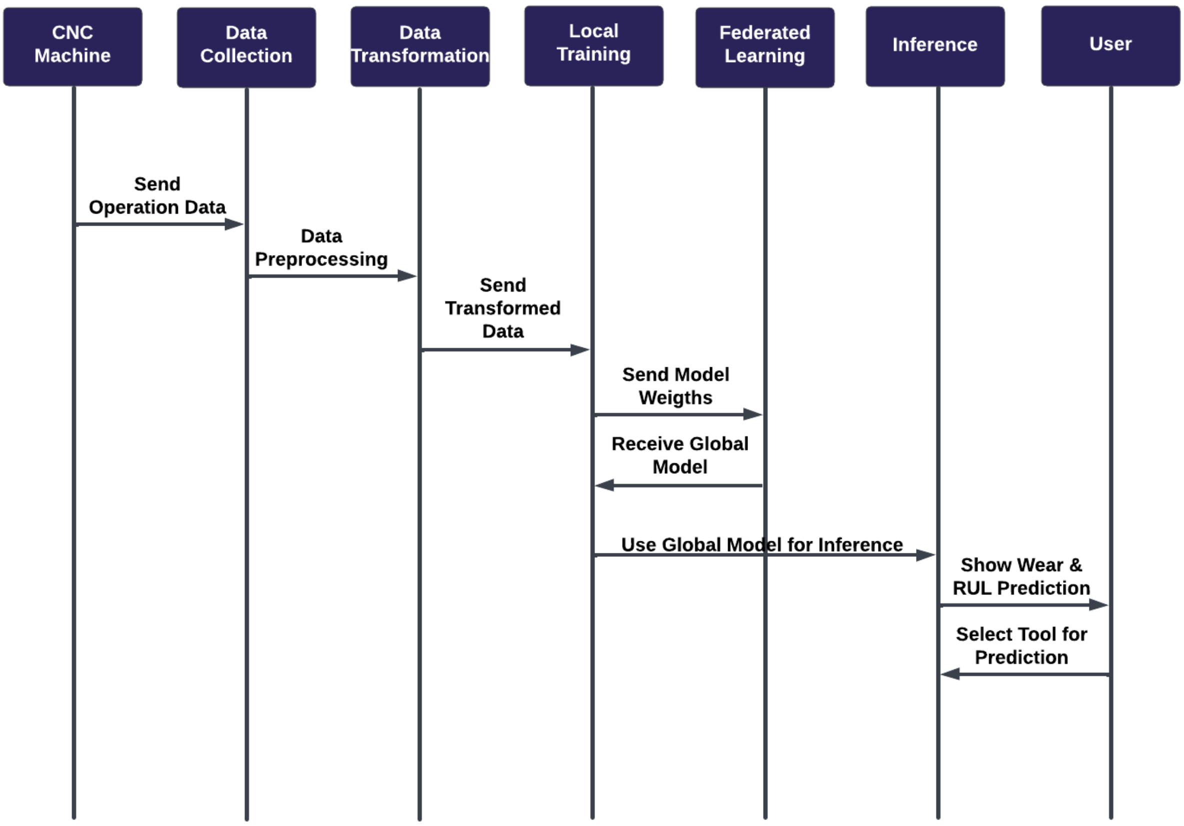


Figure 3: FEDMA's UML information flow

The diagram above (Figure 3) illustrates the UML Sequence Diagram for FEDMA's main information flow. It outlines how data moves through each module in the system:

1. CNC Machine sends operational data to the Data Collection module via a shared volume.
2. Data Collection preprocesses the data and sends it to the Data Transformation module for structuring.
3. Data Transformation prepares the data and passes it to the Local Training module for model training.
4. After training, the local model weights are sent to the Federated Learning module, where model aggregation takes place.
5. The Federated Learning module updates the global model and sends it back to the Local Training modules.
6. The Inference module uses the updated global model for making predictions, such as tool wear (Vb, Vbmax) and Remaining Useful Life (RUL).
7. The User can view the predictions, interact with the visualizations, and select tools for further analysis or maintenance.

This sequence ensures data privacy while enhancing predictive accuracy in CNC machine operations.

2.2.5 Internal Architecture

Docker Image:

The Internal Architecture will be containerised within a Docker image ensuring that the application and all dependencies are encapsulated within a portable and isolated environment. This image will be used across different CNC machines, ensuring consistency in deployments.

Docker Image Modules:

- Data Collection (Shared Volume) Module:

  Each CNC machine collects operational data (such as vibrations, forces, and temperatures) through a shared volume mechanism, enabling seamless data sharing between CNC machines and FEDMA clients. A dedicated data collection mechanism is established for each machine, with a shared volume mounted for each client. This setup facilitates the exchange of both operational data (e.g., vibrations, forces) and metadata (e.g., tool geometry, machining strategies), allowing clients to make informed inferences based on the combined information from the host machine and FEDMA clients.

- Data Transformation Module:

  This module automatically transforms the raw operation data into a structured format that the models can work with. It maps each milling tool type with its operational data, jobs performed, and the material of the workpiece.

- Local Training Module:

  Once the data is transformed, this module trains a local machine learning model using the CNC machine's operational data. This training occurs on each device to avoid the transfer of raw data, preserving privacy. Each CNC machine trains its local model on-site, using both operational and metadata to make predictions about tool wear and Remaining Useful Life (RUL). This local training is crucial to ensure that sensitive data never leaves the machine.

- Federated Learning Module:

  a. Share Model Weights with Central Server (Flower Server):
     After local training is completed, the model's updated weights are transmitted to a central server, commonly referred to as the *Flower Server*. The Flower Server aggregates model weights from all connected machines.

  b. Aggregate Global Model:
     The *Flower Server* aggregates the updated weights received from the individual machines, creating an updated global model. This aggregation leverages the diverse data from all machines, enhancing the model's predictive accuracy and robustness.

  c. Global Model Distribution:
     The updated global model is redistributed back to each edge device (CNC machine). This global model replaces the previous local model, incorporating insights from all participating machines to improve the predictive performance.

- Inference Module:

  This module uses the global model to perform predictions. It takes the latest operational data and runs the model to predict tool wear (Vb, Vbmax) and Remaining Useful Life (RUL). It generates visualizations for wear history and predictions for user interaction.Operators can view the wear and RUL predictions for each tool and select tools for further analysis or scheduling maintenance.

As the machines continue operations, the cycle repeats—new data is collected, local models are updated, and the global model improves with each iteration. The FEDMA clients continually produce inferences using operational data, allowing for real-time predictions of tool wear and RUL.

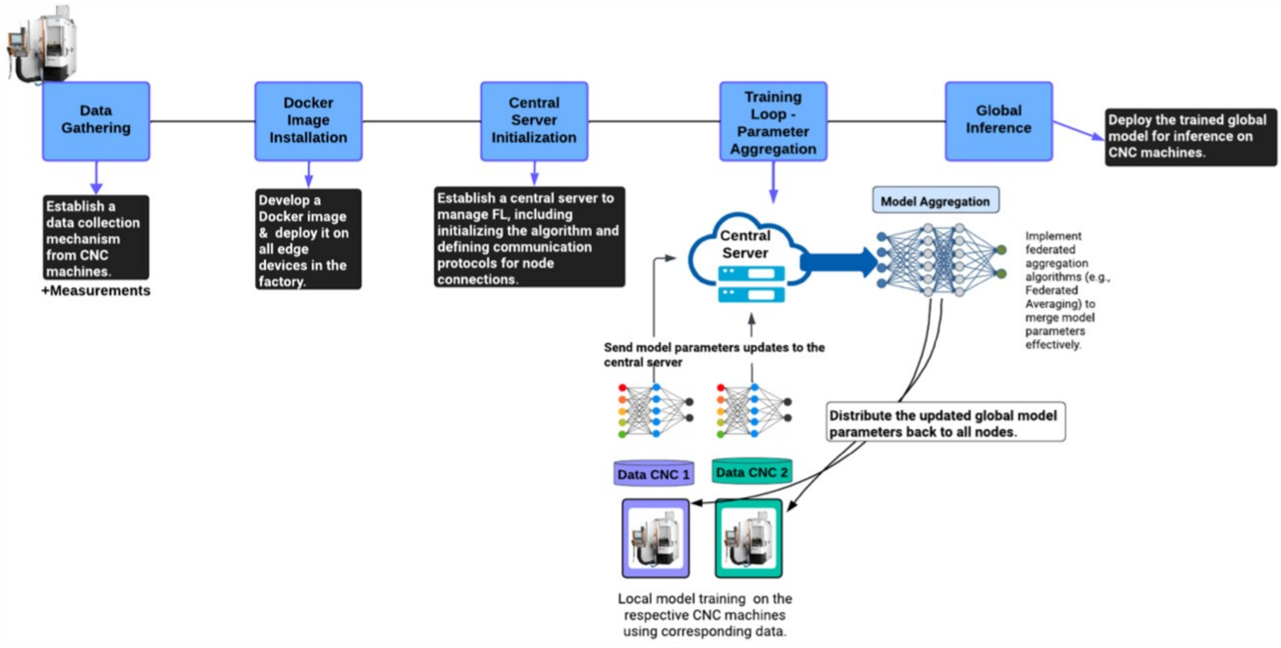


Figure 4: Federated Learning Flow for FEDMA Component

2.2.6 API

APIs for data acquisition as well as inference model output aggregation will be developed to provide input and output for the FEDMA component.

2.2.7 Implementation Technology

The FEDMA solution is primarily implemented in Python, due to its versatility in AI, data processing, and ease of integration with various libraries. Key technologies and platforms include:

Libraries:

- PyTorch: PyTorch is the primary deep learning framework used for building and training both local and global machine learning models. Its flexibility and ease of use make it well-suited for federated learning and model deployment.
- Scikit-Learn: Used for traditional machine learning tasks and preprocessing steps, aiding in model evaluation and validation during local training phases.
- Pandas: Employed for data manipulation, helping to organize and structure operational data from the CNC machines.
- NumPy: Supports numerical operations required for data processing during the model training and inference phases.
- Matplotlib: Generates visualizations for the wear history, plotting Vb and Vbmax values to provide insights into tool wear and performance over time.

Federated Learning Framework:

- The system relies on Flower for federated learning orchestration. Flower facilitates the aggregation of model weights across all participating CNC machines, ensuring the global model benefits from decentralized data while maintaining privacy.

Containerization:

- Docker is used to containerize the entire FEDMA solution. This guarantees consistent environments across all edge devices and allows for easy management of updates, dependencies, and scaling across different hardware configurations.

2.2.8 Comments

None

2.3 Data Container

2.3.1 Description

The Data Containers (DC) provide access to the data following the Data as a Product approach while, at the same time, creating an abstraction layer that will hide all the underlying technical complexity from the users. DC allow the applications to access the data in a trusted and reliable way regardless of the location (Edge or Cloud) and particularities of the data sources. The DC will provide the data according to the consumer's requirements in terms of format and data quality, with the proper considerations about security, privacy, and performance.

2.3.2 Input

The DC should be able to obtain the data from the sources (directly or indirectly) and call the necessary components of the RE4DY architecture in order to serve the data following the Data as a Product approach.

Inputs of the DC API:

- Dataset id
- Format: defines the output data format (for example, JSON, Parquet or CSV).
- Rules: allow the data consumer to select a subset of the available data. Rules can also be used to remove outliers from the data. A rule consists of three elements:
    - Subject: variable where the rule is applied.

  - Operator: the allowed operators are the following: “<=“, “>=“, “<“, “>”, “==”, “!=“, “or”, “++” (include only the columns whose names are specified in the “object” value), and “—“ (do not include the columns whose names are specified in the “object” value).
  - Object: value.
- Notification URL: endpoint in the client application to receive a notification when the data is available.

2.3.3 Output

The Data Containers will provide data and metadata.

Metadata request: The metadata will be sent in the HTTP response in JSON format.

Historical data request:

- API response: indicates whether the request was completed successfully or not.
- Notification: is sent to the specified HTTP endpoint once the process is completed and data is available. Example:

```
{
  "message": "Data sent successfully to FTP server",
  "success": "true",
  "filename": "2b4bd036-f7ac-4fca-89fd-f902968e3424"
}
```

- Data: The requested data will be served following the consumer’s requirements in terms of format and data quality. The current version makes the data available through the GET /data endpoint using the “filename” value from the notification as input parameter.

Streaming (subscription)

- API response: indicates whether the request was completed successfully or not and returns the subscription id if the request was successful. The subscription Id value can be used to cancel the subscription.
- Notification: is sent to the specified HTTP endpoint whenever new data is available. It contains the requested data, following the consumer’s requirements in terms of format and data quality.

2.3.4 Information Flow

Obtaining cleansed and preprocessed historical data from storage:

The flow shown in Figure 5 describes a particular scenario in which a cleansed and preprocessed dataset is available in the data storage. In that case, the DC can retrieve the data from the storage, apply the requested filtering rules and format and serve it to the client. To facilitate the management of big datasets, the data is sent to a repository (SFTP server) and the client receives a notification when the data is available. Then, the client can retrieve the data either through the corresponding request to the DC API or directly from the output repository.

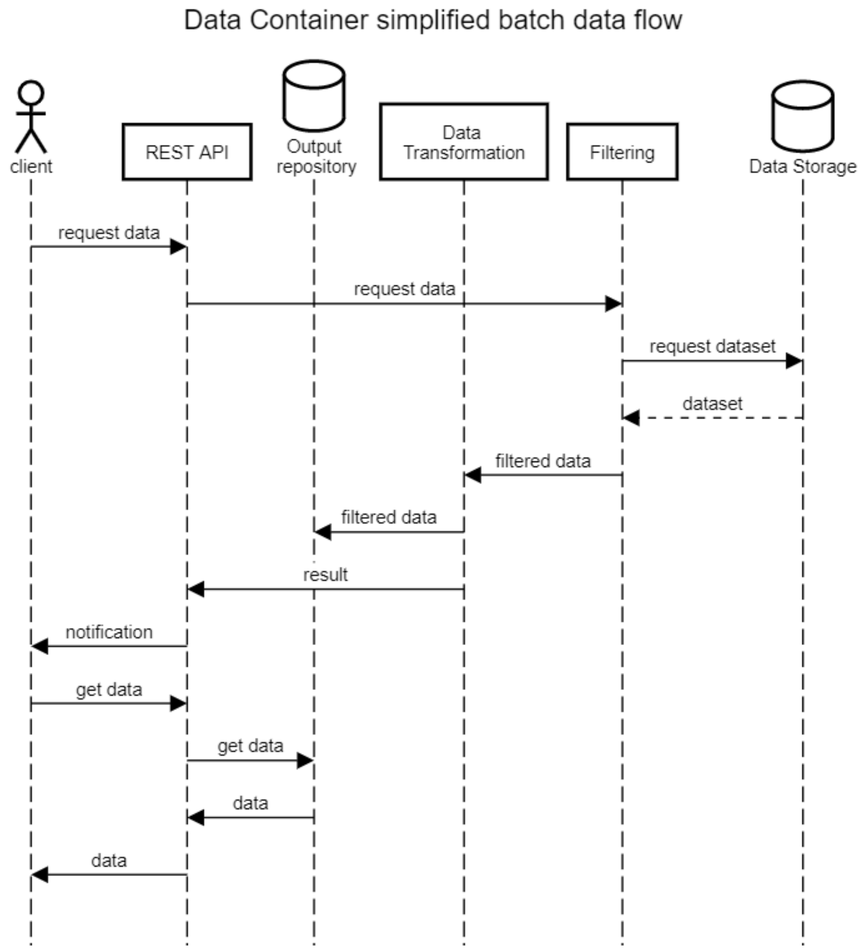


Figure 5: UML diagram of the process of retrieving data from the storage and serving it to the client

Obtaining historical data from the source:

The flow shown in Figure 6 illustrates the overall process to request historical data from the source via DC. Hence, it includes calls to other components of the RE4DY toolkit, such as the Data Transformation Services. To facilitate the management of big datasets, the data can be sent to a repository (such as an FTP or SFTP server). In that case, the DC sends a notification to the client when the data is available. In addition, the preprocessed dataset can be stored temporarily in the data storage to speed up the process if the dataset is requested again.

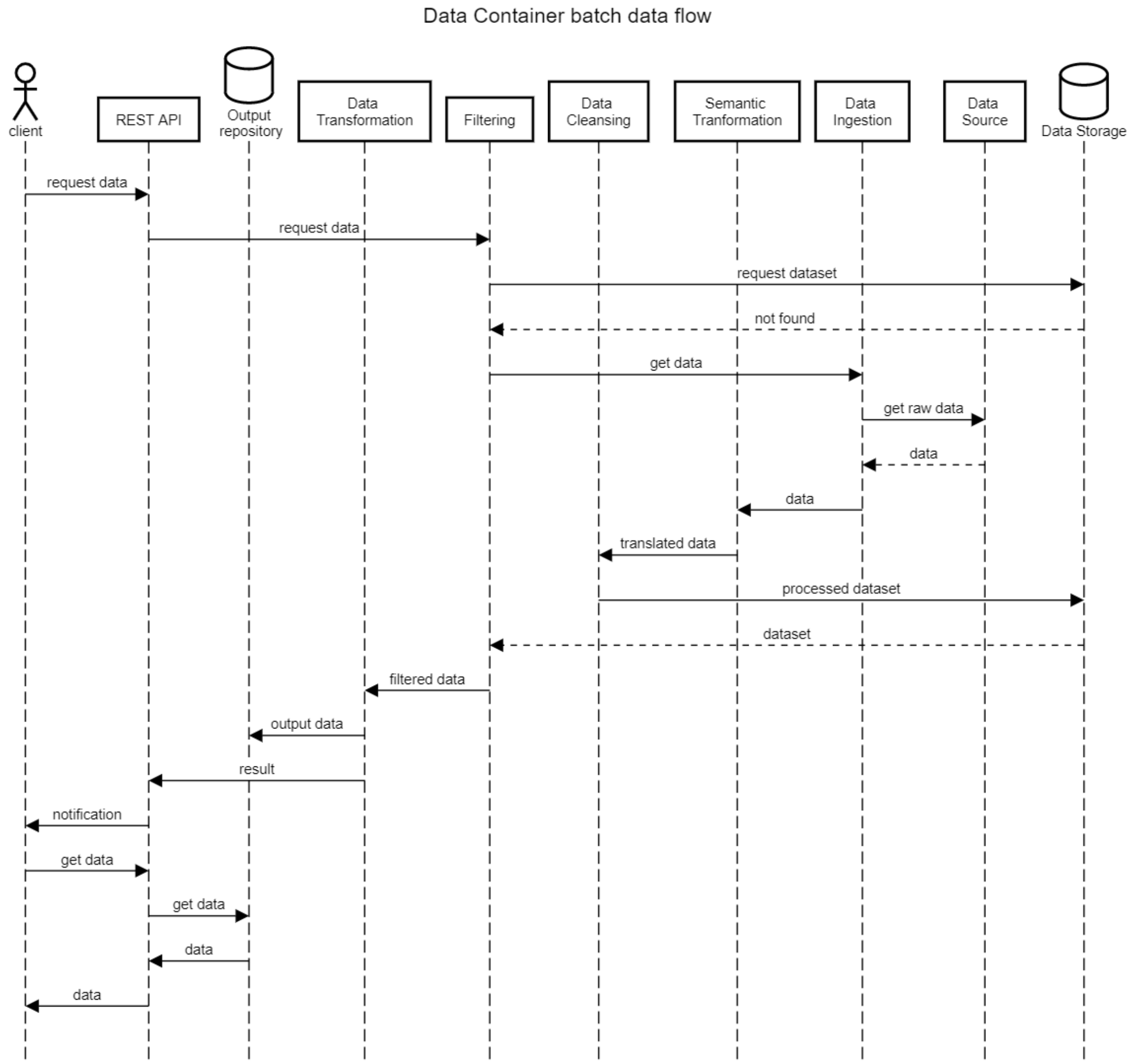


Figure 6: UML diagram of the complete process of retrieving and serving batch data in a generic Data Container

This flow shows the overall process. Hence, it includes calls to other blocks of the RE4DY architecture, such as Data Ingestion and Semantic Transformation.

Obtaining historical data via IDS connector:

The flow shown in Figure 7 describes a scenario where the data from the DC is shared in a Data Space through an IDS connector. In this case, a new component called DC Client, which acts as a mediator between the DC and the IDS connector, has been added to facilitate this integration. In the first step of this scenario, the data provider registers the data from the DC with the IDS connector. The DC and DC client provide methods in their APIs to assist in this step. Once the dataset has been registered, a data consumer with the proper access rights can negotiate the access and request the data using an IDS connector.

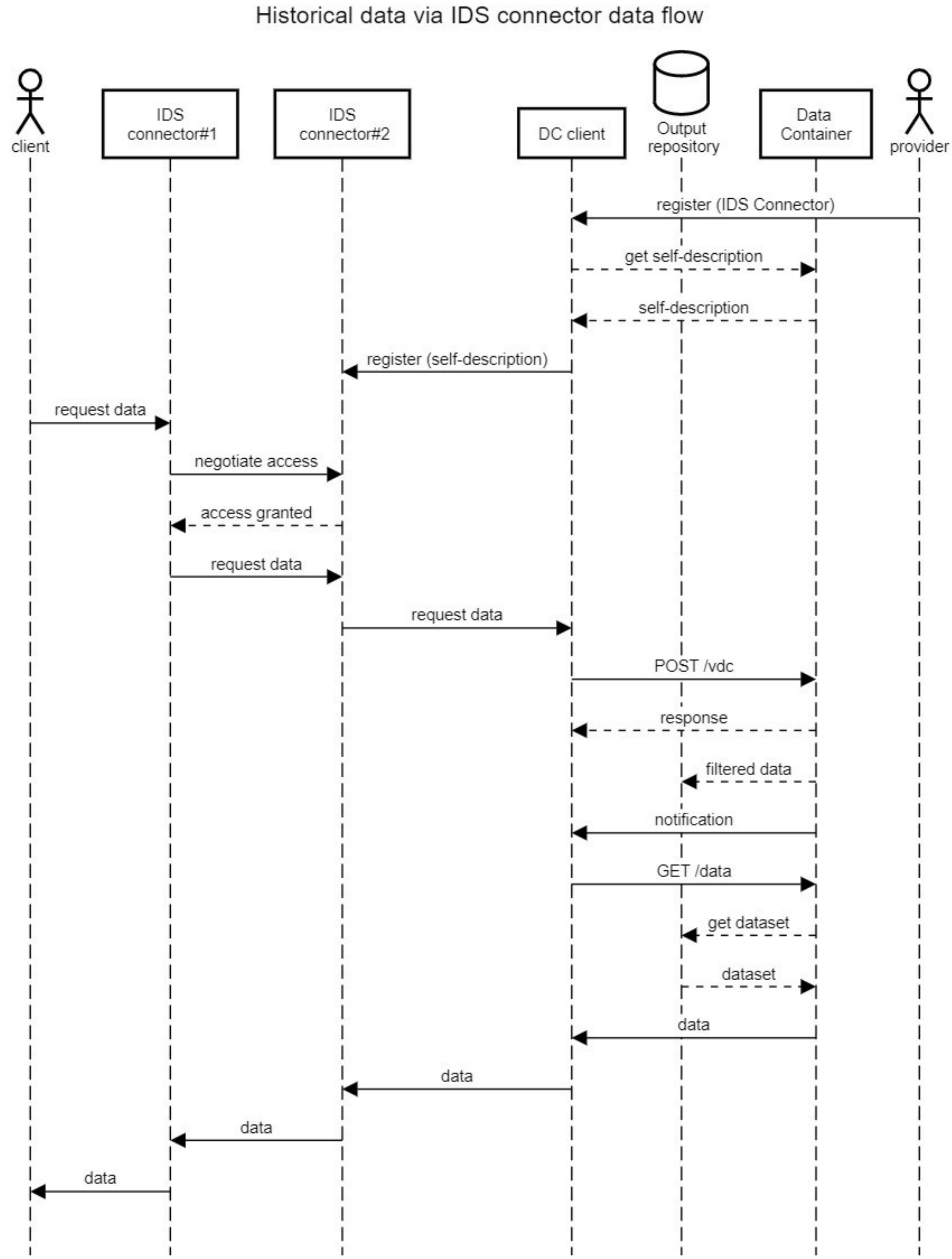


Figure 7: UML diagram of the process of serving batch data in a generic Data Container via IDS connector

Streaming (subscription)

The flow shown in Figure 8 illustrates how near-real time data can be accessed using a Data Container. In this case, the client subscribes to the data from the DC. Whenever new data that matches the client's requirements is available, a notification is sent to the endpoint provided in the subscription request with the new data, which can be pre-processed and converted to the required format in the DC.

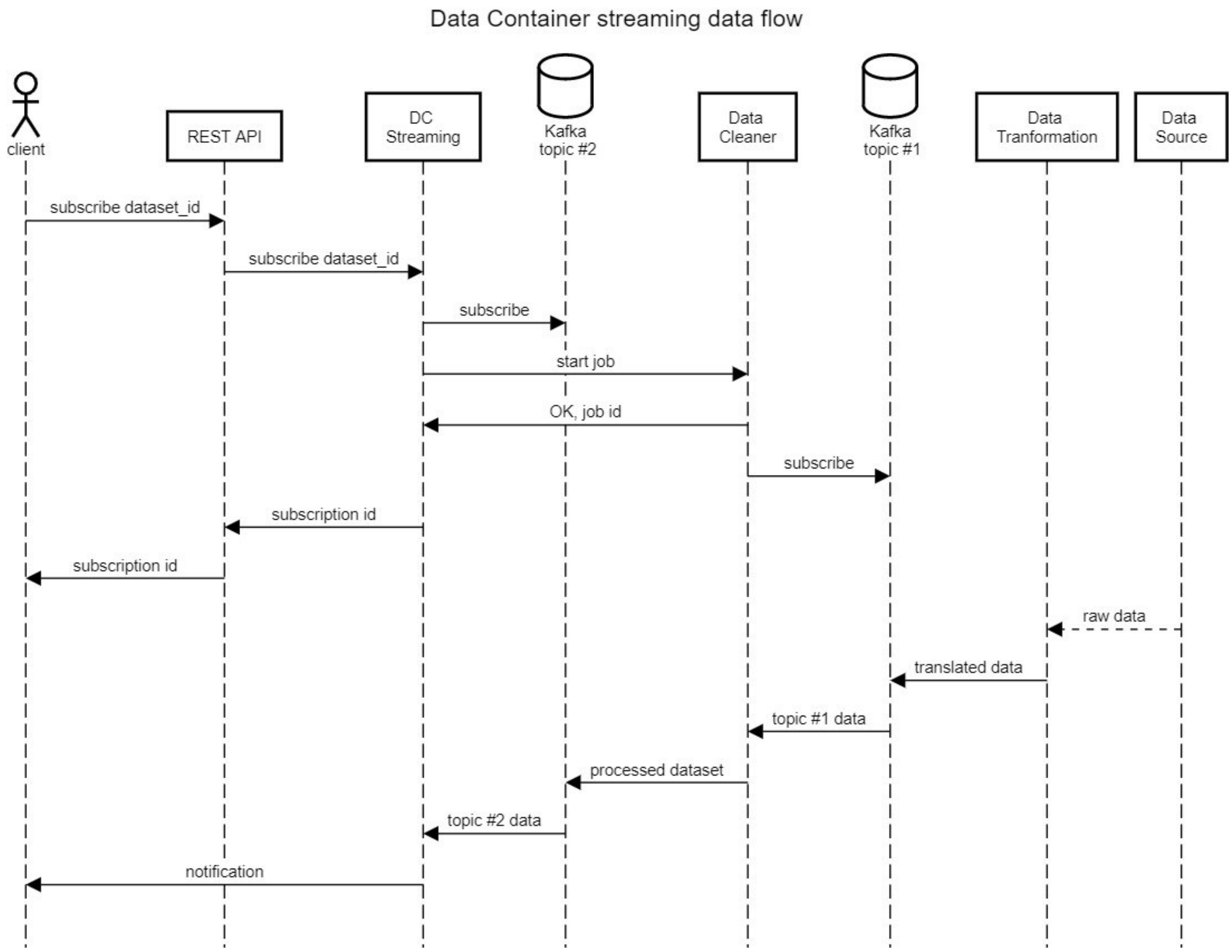


Figure 8: UML diagram of the complete process of subscribing to data in streaming in a generic Data Container and receiving notifications

2.3.5 Internal Architecture

Data Containers expose a REST API to enable access to the data and metadata. The control module manages the DC configuration and ensures that the conditions defined in the DCP are applied. When a request is received, the Data Processing module performs the necessary calls to other components of the RE4DY architecture to retrieve the requested data and ensure that the client's requirements are met. This module can also apply filtering rules to select only a subset of the available data. Finally, the Data Transformation module prepares the data and converts it into the format requested by the client.

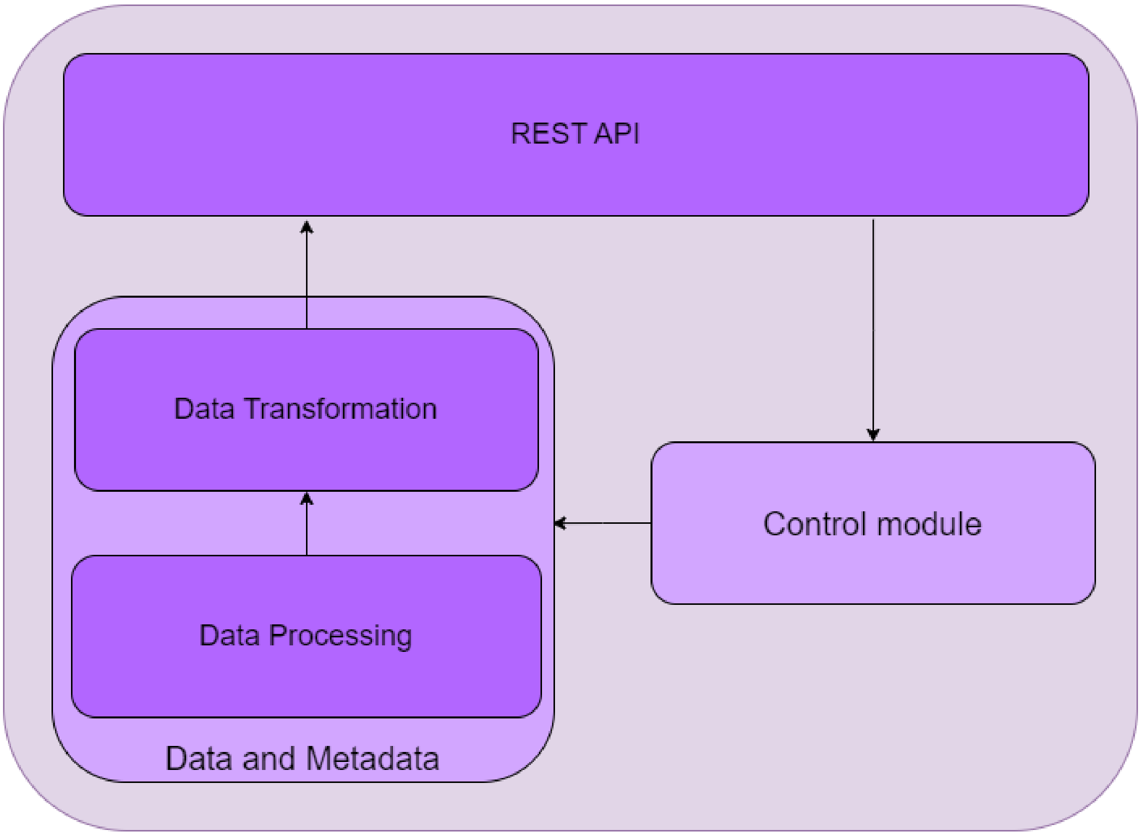


Figure 9: Internal architecture of a generic Data Container

2.3.6 API

- `GET /metadata`
  - **Description**
    This endpoint provides metadata associated to the datasets provided by the DC. The input parameter "dataset_id" identifies the dataset for which the metadata is requested.
  - **Input**
    - dataset_id
  - **Output**

```
{
    "variables": 11,
    "rows": 1,
    "size": 1291,
    "lastUpdate": "2023-06-05T15:31:16Z",
    "variablesNames": ["var1", "var2",...]
}
```

- `POST /vdc`
  - **Description**
    This endpoint is used by the data consumer to request data from the DC. The input parameter "dataset_id" identifies the dataset. The parameter "format" defines the selected

output format. The JSON array "rules" defines the filtering rules that will be applied to the selected dataset. A notification will be sent to the URL specified in the parameter "notification_url" when the dataset is available in the output repository.

- **Input**

```
{
    "dataset_id": "FaultySteelPlates",
    "rules": [
        {
        "name": "test-rule",
        "rule": {
          "subject_column": "TypeOfSteel_A300",
          "operator": "==",
          "object": 1
        }
      }
    ],
    "format": "json",
    "notification_url": "http://example.com/notify"
  }
```

- **Output**

```
{
    "result rows": 777,
    "message": "Request received successfully. Preparing results...",
    "success": true
}
```

- GET /data
  - **Description**
    This endpoint is used by the data consumer to retrieve a dataset from the DC once it has been pre-processed and converted to the proper format. The input parameter "id" identifies the dataset and corresponds with the "filename" attribute of the notification.
  - **Input**
    The "id" parameter is specified as a URL parameter. Example: GET /data?id=2b4bd036-f7ac-4fca-89fd-f902968e3424
  - **Output**
    The dataset, as specified in the previous call to the Data Container.

- GET /selfdescription
  - **Description**
    This endpoint is used by the data provider to retrieve a template for the asset description needed to register de

data from the DC with the marketplace or with an IDS connector.

- **Output**
  Asset description template (JSON-LD format).

- GET /dcp
  - **Description**
    This endpoint can be used to retrieve the content of the current DCP.
  - **Output**
    DCP description in JSON format.

- POST /dcp
  - **Description**
    This endpoint can be used to configure the Data Container by applying a new DCP.
  - **Input**
    New configuration values (JSON format).
  - **Output**

```
{
    "message": "DCP successfully applied",
    "success": "true"
}
```

- POST /subscription
  - **Description**
    This endpoint is used by the data consumer to subscribe to data in streaming. The input parameter "dataset_id" identifies the data source. The parameter "format" defines the selected output format. The JSON array "rules" defines the filtering rules that will be applied to the selected dataset. Whenever new data that matches the requirements of the consumer is available, it will be sent to the URL specified in the parameter "notification_url".
  - **Input**

```
{
    "dataset_id": "5gTest",
    "rules": [
      {
      "name": "test-rule",
      "rule":
      {"subject_column": "Timestamp", "operator": ">",
      "object": "2023-11-09T00:00:00.000Z"}
      }
            ],
    "format": "json",
    "notification_url": "http://example.com/notify "
}
```

- **Output**

```
{
    "message": "Subscription submitted successfully",
    "subscription_id": "20240603115305-0000",
    "success": "true"
}
```

- DELETE /subscription
  - **Description**
    This endpoint is used by the data consumer to cancel a subscription to data from the DC.
  - **Input**
    The "subscription_id"" parameter is specified as a URL parameter and identifies the subscription. Example: DELETE /subscription? subscription_id =20240603115305-0000
  - **Output**

```
{
    "message": "Unsubscribed successfully",
     "subscription_id": "20240603115305-0000",
     "success": "true"
}
```

2.3.7 Implementation Technology

Through the Data Containers, it should be possible to define flows that connect different elements of the RE4DY architecture to enable access to the data following the DaaP principles. Thus, the preferred implementation for the DCs is based on flow-oriented programming frameworks such as Apache NiFi or Node-RED. The current implementation has been created using Apache NiFi to define the data flows and Apache Spark for pre-processing. In addition, MongoDB and Apache Kafka have been integrated as part of the historical data and streaming pipelines, respectively, and the output repository in the historical data flows has been implemented using a SFTP server. The current implementation is deployed on Docker.

2.3.8 Comments

None

2.4 Sovity Open-Source EDC Connector

2.4.1 Description

The Sovity Connector is a component designed to facilitate secure and efficient data exchange within data spaces, such as Gaia-X or Catena-X, enabling organizations to exchange data safely while maintaining control over how their data is used.

The Sovity Connector is built on top of the Eclipse Dataspace Connector (EDC), an open-source framework from the Eclipse Foundation, which is implemented around modular and standardized data exchange protocols. It provides the underlying architecture for secure data transfers, allowing for interoperability between different systems. Sovity's implementation of the EDC extends this functionality with a focus on ease of use and compliance. It supports multiple authentication mechanisms like OAuth2.0 and offers data transfer modes like HttpData and HttpProxy.

2.4.2 Input

To interact with the Sovity Connector, there are several inputs that need to be provided. These inputs are necessary for defining data assets, managing access, and integrating with other systems or data spaces. The most relevant ones are:

Asset Information, which comprehends:

- *Asset Metadata*: including the name, description, data type, and format of the asset. Metadata helps describe what the data asset contains, making it easier for other data space participants to understand its purpose and usage.
- *Data Source Connection*: The connector requires information about where the data is stored. This could be a database connection string, API endpoint, or cloud storage location.

Usage Policies and Access Rules:

- *Access Control Policies:* Asset owners must define who is allowed to access the asset and under what conditions. This includes specifying which parties or roles can read or write to the data, time-based access restrictions, and rate limits.
- *Data Usage Conditions:* These define how the data can be used once accessed. For example, restrictions could be applied to prevent data from being resold or to ensure it is used for specific purposes like research.

Contracts:

- *Data Sharing Agreements:* When publishing data offerings, the connector requires information such as the duration of access, pricing models if applicable, and terms related to data handling and privacy.
- *Licensing Information:* licensing is important to ensure that data is used according to the rights and limitations defined by the provider, such as Creative Commons licenses or proprietary terms.

2.4.3 Output

The Sovity Connector generates various outputs:

Data API Endpoints: When a data asset is published, the Sovity Connector can provide secure API endpoints for other participants to access or consume the data. These endpoints abide by the policies and conditions set during the asset creation process, ensuring that only authorized users can interact with the data.

Data Transfer Logs: As data is accessed or transferred through the connector, logs detailing these transactions are generated. These logs can include information like data request timestamps, requester identities, and data transfer volumes, which are usedl for auditability and monitoring.

2.4.4 Information Flow

The information flow between two Sovity Connector instances, where one is offering an asset and the other is consuming it, follows a structured process that ensures secure and controlled data sharing. Here's a description their interaction:

1. Discovery and Request Phase:

- Asset Publishing: The data provider uses their Sovity Connector instance to publish a data asset. This asset includes metadata such as a description, data type, and the terms of use, which are registered in the data space's catalog.
- Discovery by Data Consumer: The consumer's Sovity Connector instance searches the data space catalog and discovers the published asset. The metadata provided by the offering connector helps the consumer understand the nature of the data, the conditions for access, and whether it meets their needs.

2. Negotiation Phase:

- Data Access Request: The consumer's connector sends a request for access to the asset.
- Policy Validation and Contract Negotiation: The provider's connector evaluates the request against the predefined usage policies of the asset. If the request aligns with the conditions set by the provider, a contract negotiation begins. This contract is digital and details how the data will be accessed, for how long, and under what restrictions.
- Contract Agreement: Once both parties agree on the terms, the Sovity Connectors automatically generate a digital contract. This contract is stored within the respective connectors for compliance tracking, and it serves as the basis for the subsequent data exchange.

3. Data Transfer Phase:

- Secure Data Exchange Initialization: With a contract in place, the provider's connector prepares to deliver the data according to the agreed-upon terms. The data consumer's connector sends a request to start the data transfer, and the provider's connector verifies this against the contract to ensure compliance.
- Data Transfer: The actual data transfer takes place over a secure channel, typically using HTTPS or a similar protocol. Depending on the configuration, this could involve direct data streaming or file downloads.

2.4.5 API

The connector provides a RESTful API layer, which allows users to interact with the connector programmatically. This API is used for managing assets, setting up policies, handling user permissions, and initiating data exchange processes.

Assumed base url of the Sovity Connector instance: http://{sovity-connector}/api/management/v2

Assets

- Create Asset
  - POST /assets

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/Asset",
    "https://w3id.org/edc/v0.0.1/ns/properties": {
```

```
            "https://w3id.org/edc/v0.0.1/ns/id": "http-source-
1",
            "http://www.w3.org/ns/dcat#version": "1.0",
            "http://purl.org/dc/terms/language":
"https://w3id.org/idsa/code/EN",
            "http://purl.org/dc/terms/title": "test-document",
            "http://purl.org/dc/terms/description": "my test
document",
            "http://www.w3.org/ns/dcat#keyword": [
                "keyword1",
                "keyword2"
            ],
            "http://purl.org/dc/terms/creator": {
                "http://xmlns.com/foaf/0.1/name": "My Org"
            },
            "http://purl.org/dc/terms/license":
"https://creativecommons.org/licenses/by/4.0/",
            "http://www.w3.org/ns/dcat#landingPage":
"https://mydepartment.myorg.com/my-offer",
            "http://www.w3.org/ns/dcat#mediaType":
"text/plain",
            "https://semantic.sovity.io/dcat-
ext#httpDatasourceHintsProxyMethod": "false",
            "https://semantic.sovity.io/dcat-
ext#httpDatasourceHintsProxyPath": "false",
            "https://semantic.sovity.io/dcat-
ext#httpDatasourceHintsProxyQueryParams": "false",
            "https://semantic.sovity.io/dcat-
ext#httpDatasourceHintsProxyBody": "false",
            "http://purl.org/dc/terms/publisher": {
                "http://xmlns.com/foaf/0.1/homepage":
"https://myorg.com/"
            }
        },
        "https://w3id.org/edc/v0.0.1/ns/privateProperties": {},
        "https://w3id.org/edc/v0.0.1/ns/dataAddress": {
            "https://w3id.org/edc/v0.0.1/ns/type": "HttpData",
            "https://w3id.org/edc/v0.0.1/ns/baseUrl":
"https://api.github.com/repos/sovity/edc-extensions/events"
        }
    }
```


- Delete Asset
    - DELETE /assets/http-source-1

- Request Assets

```
o POST http://{sovity-
  connector}/api/management/v2/assets/request

  {
      "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
      "https://w3id.org/edc/v0.0.1/ns/offset": 0,
      "https://w3id.org/edc/v0.0.1/ns/limit": 10
  }
```

Policies

```
• Create Simple Policy
    o POST /policydefinitions
    {
        "@type":
    "https://w3id.org/edc/v0.0.1/ns/PolicyDefinition",
        "@id": "policy-1",
        "https://w3id.org/edc/v0.0.1/ns/policy": {
            "@type": "http://www.w3.org/ns/odrl/2/Set",
            "http://www.w3.org/ns/odrl/2/permission": [
                {
                    "http://www.w3.org/ns/odrl/2/action": {
                        "http://www.w3.org/ns/odrl/2/type":
    "USE"
                    },
                    "http://www.w3.org/ns/odrl/2/constraint":
    []
                }
            ]
        }
    }


• Create Time Policy
    o POST /policydefinitions
    {
        "@type": "PolicyDefinitionDto",
        "@id": "policy-1",
        "https://w3id.org/edc/v0.0.1/ns/policy": {
            "@type": "http://www.w3.org/ns/odrl/2/Set",
            "http://www.w3.org/ns/odrl/2/permission": [
                {
                    "http://www.w3.org/ns/odrl/2/action": {
                        "http://www.w3.org/ns/odrl/2/type":
    "USE"
                    },
                    "http://www.w3.org/ns/odrl/2/constraint": [
```

```
                        {
                            "http://www.w3.org/ns/odrl/2/leftOp
erand": "POLICY_EVALUATION_TIME",
                            "http://www.w3.org/ns/odrl/2/operat
or": {
                                "@id":
"http://www.w3.org/ns/odrl/2/gteq"
                            },
                            "http://www.w3.org/ns/odrl/2/rightO
perand": "2022-05-31T22:00:00.000Z"
                        },
                        {
                            "http://www.w3.org/ns/odrl/2/leftOp
erand": "POLICY_EVALUATION_TIME",
                            "http://www.w3.org/ns/odrl/2/operat
or": {
                                "@id":
"http://www.w3.org/ns/odrl/2/lt"
                            },
                            "http://www.w3.org/ns/odrl/2/rightO
perand": "2030-06-30T22:00:00.000Z"
                        }
                    ]
                }
            ]
        }
}
```

- Create Participant Policy
  - POST /policydefinitions

```
{
    "@type": "PolicyDefinitionDto",
    "@id": "policy-1",
    "https://w3id.org/edc/v0.0.1/ns/policy": {
        "@type": "http://www.w3.org/ns/odrl/2/Set",
        "http://www.w3.org/ns/odrl/2/permission": [
            {
                "http://www.w3.org/ns/odrl/2/action": {
                    "http://www.w3.org/ns/odrl/2/type":
"USE"
                },
                "http://www.w3.org/ns/odrl/2/constraint": [
                    {
                        "http://www.w3.org/ns/odrl/2/leftOp
erand": "REFERRING_CONNECTOR",
                        "http://www.w3.org/ns/odrl/2/operat
or": {
```

```
                                        "@id":
        "http://www.w3.org/ns/odrl/2/eq"
                                    },
                                    "http://www.w3.org/ns/odrl/2/rightO
        perand": "my-edc2"
                                }
                            ]
                        }
                    ]
                }
            }
```

- Delete Policy
    - DELETE /policydefinitions/policy-1

- Request Policies
    - POST /policydefinitions/request

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
    "https://w3id.org/edc/v0.0.1/ns/offset": 0,
    "https://w3id.org/edc/v0.0.1/ns/limit": 10
}
```

Contract Definitions

- Create ContractDefinition
    - POST /contractdefinitions

```
{
    "@id": "contract-definition-1",
    "@type":
"https://w3id.org/edc/v0.0.1/ns/ContractDefinition",
    "https://w3id.org/edc/v0.0.1/ns/accessPolicyId":
"policy-1",
    "https://w3id.org/edc/v0.0.1/ns/contractPolicyId":
"policy-1",
    "https://w3id.org/edc/v0.0.1/ns/assetsSelector": [
        {
            "@type": "CriterionDto",
            "https://w3id.org/edc/v0.0.1/ns/operandLeft":
"https://w3id.org/edc/v0.0.1/ns/id",
            "https://w3id.org/edc/v0.0.1/ns/operator": "=",
            "https://w3id.org/edc/v0.0.1/ns/operandRight":
"http-source-1"
        }
    ]
}
```

- Delete ContractDefinition
  - DELETE /contractdefinitions/contract-definition-1

- Request ContractDefinitions
  - POST /contractdefinitions/request

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
    "https://w3id.org/edc/v0.0.1/ns/offset": 0,
    "https://w3id.org/edc/v0.0.1/ns/limit": 10
}
```

Catalog

- Request Catalog
  - POST /catalog/request

```
{
    "@type":
"https://w3id.org/edc/v0.0.1/ns/CatalogRequest",
    "https://w3id.org/edc/v0.0.1/ns/protocol": "dataspace-
protocol-http",
    "https://w3id.org/edc/v0.0.1/ns/counterPartyAddress":
"http://edc:11003/api/dsp",
    "https://w3id.org/edc/v0.0.1/ns/querySpec": {
        "@type":
"https://w3id.org/edc/v0.0.1/ns/QuerySpec",
        "https://w3id.org/edc/v0.0.1/ns/offset": 0,
        "https://w3id.org/edc/v0.0.1/ns/limit": 10
    }
}
```

Contract Negotiations

- Start Negotiation
  - POST /contractnegotiations

```
{
    "@type":
"https://w3id.org/edc/v0.0.1/ns/ContractRequest",
    "https://w3id.org/edc/v0.0.1/ns/consumerId": "my-edc2",
    "https://w3id.org/edc/v0.0.1/ns/providerId": "my-edc",
    "https://w3id.org/edc/v0.0.1/ns/connectorAddress":
"http://edc:11003/api/dsp",
```

```
    "https://w3id.org/edc/v0.0.1/ns/protocol": "dataspace-
protocol-http",
    "https://w3id.org/edc/v0.0.1/ns/offer": {
      "https://w3id.org/edc/v0.0.1/ns/offerId":
"Y29udHJhY3QtZGVmaW5pdGlvbi0x:aHR0cC1zb3VyY2UtMQ==:ZjM4ZTJl
MTItN2RmMC00ZjU3LTgwNDMtYjM0MzMwYTVkMDA3",
      "https://w3id.org/edc/v0.0.1/ns/assetId": "http-
source-1",
      "https://w3id.org/edc/v0.0.1/ns/policy": {
              "@id":
"Y29udHJhY3QtZGVmaW5pdGlvbi0x:aHR0cC1zb3VyY2UtMQ==:ZjM4ZTJl
MTItN2RmMC00ZjU3LTgwNDMtYjM0MzMwYTVkMDA3",
              "@type": "http://www.w3.org/ns/odrl/2/Set",
              "http://www.w3.org/ns/odrl/2/permission": {
                  "http://www.w3.org/ns/odrl/2/target":
"http-source-1",
                  "http://www.w3.org/ns/odrl/2/action": {
                      "http://www.w3.org/ns/odrl/2/type":
"USE"
                  },
                  "http://www.w3.org/ns/odrl/2/constraint":
[]
              },
              "http://www.w3.org/ns/odrl/2/prohibition":
[],
              "http://www.w3.org/ns/odrl/2/obligation": [],
              "http://www.w3.org/ns/odrl/2/target": "http-
source-1"
          }
      }
  }
```

- Request Contract Negotiations
    - POST /contractnegotiations/request

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
    "https://w3id.org/edc/v0.0.1/ns/offset": 0,
    "https://w3id.org/edc/v0.0.1/ns/limit": 20
}
```

- Cancel Negotiation
    - POST /contractnegotiations/ac49b1a0-aa76-4370-8b56-64168c8adad9/cancel

- Decline Negotiation

- POST /contractnegotiations/88687cb0-1d97-40c5-86c2-ad744afed538/decline

- Request Provider side
  - POST /contractnegotiations/request

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
    "https://w3id.org/edc/v0.0.1/ns/offset": 0,
    "https://w3id.org/edc/v0.0.1/ns/limit": 50
}
```

- Get Negotiation
  - POST /contractnegotiations/cecc96ae-01fc-4c0e-ab7e-1fec68acd650

Contract Agreements

- Request Contract Agreements
  - POST /contractagreements/request

```
{
    "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
    "https://w3id.org/edc/v0.0.1/ns/offset": 0,
    "https://w3id.org/edc/v0.0.1/ns/limit": 10
}
```

Data Transfer

- Start Data Push
  - POST /transferprocesses

```
{
    "@type":
"https://w3id.org/edc/v0.0.1/ns/TransferRequest",
    "https://w3id.org/edc/v0.0.1/ns/assetId": "http-source-
1",
    "https://w3id.org/edc/v0.0.1/ns/contractId":
"Y29udHJhY3QtZGVmaW5pdGlvbi0x:aHR0cC1zb3VyY2UtMQ==:MWZhMDk2
YzEtODcwNi00NjBiLWJlMmYtZmQyNDFkZWQxYjE3",
    "https://w3id.org/edc/v0.0.1/ns/connectorAddress":
"http://edc:11003/api/dsp",
    "https://w3id.org/edc/v0.0.1/ns/connectorId": "my-edc",
    "https://w3id.org/edc/v0.0.1/ns/dataDestination": {
        "https://w3id.org/edc/v0.0.1/ns/type": "HttpData",
```

```
            "https://w3id.org/edc/v0.0.1/ns/baseUrl":
    "https://webhook.site/a418c986-299d-4e22-a1e1-bf532631913a"
        },
        "https://w3id.org/edc/v0.0.1/ns/properties": {},
        "https://w3id.org/edc/v0.0.1/ns/privateProperties": {},
        "https://w3id.org/edc/v0.0.1/ns/protocol": "dataspace-
    protocol-http",
        "https://w3id.org/edc/v0.0.1/ns/managedResources":
    false
    }
```

- Request Transfer Processes
  - POST /transferprocesses/request

```
    {
        "@type": "https://w3id.org/edc/v0.0.1/ns/QuerySpec",
        "https://w3id.org/edc/v0.0.1/ns/offset": 0,
        "https://w3id.org/edc/v0.0.1/ns/limit": 10
    }
```

- Cancel Transfer Process
  - POST /transferprocesses/c715355b-1e4b-49a9-9ef0-956405e88fe3/terminate

2.4.6 Implementation Technology

As mentioned at the beginning this section, the Sovity Connector is built on top of the Eclipse Dataspace Connector (EDC), and therefore implemented in Java.

OAuth2.0 is used to manage authentication for API access, allowing secure, token-based interactions between the connector and other services. JWT (JSON Web Tokens): is used to facilitate secure communication between components, ensuring that data and identity information is exchanged in a tamper-proof manner.

The connector is typically deployed as a set of microservices, which enables scalability and isolated management of different functionalities like data access, policy enforcement, and data transfer.

2.4.7 Comments

None

2.5 ICF Knowledge Graph Visualisation Environment

2.5.1 Description

The ICF knowledge graph visualisation environment aims to visualise knowledge graphs of different types and supports the presentation of flows and interconnected data (an example is supply chains). The JS/TypeScript components can present significant data volumes through interactive graphs.

2.5.2 Input

The input to the JS/TypeScript components is data formatted as JSON text and CSV, which can be extracted from databases and web services. The software runs locally on the user's computer and does not offer services to external systems.

2.5.3 Output

The output is an interactive graphical model describing the knowledge graphs and flows the input data represents. Network charts and Sankey charts are among the graph types supported. Sankey charts represent supply chain flows, and network charts represent ontologies.

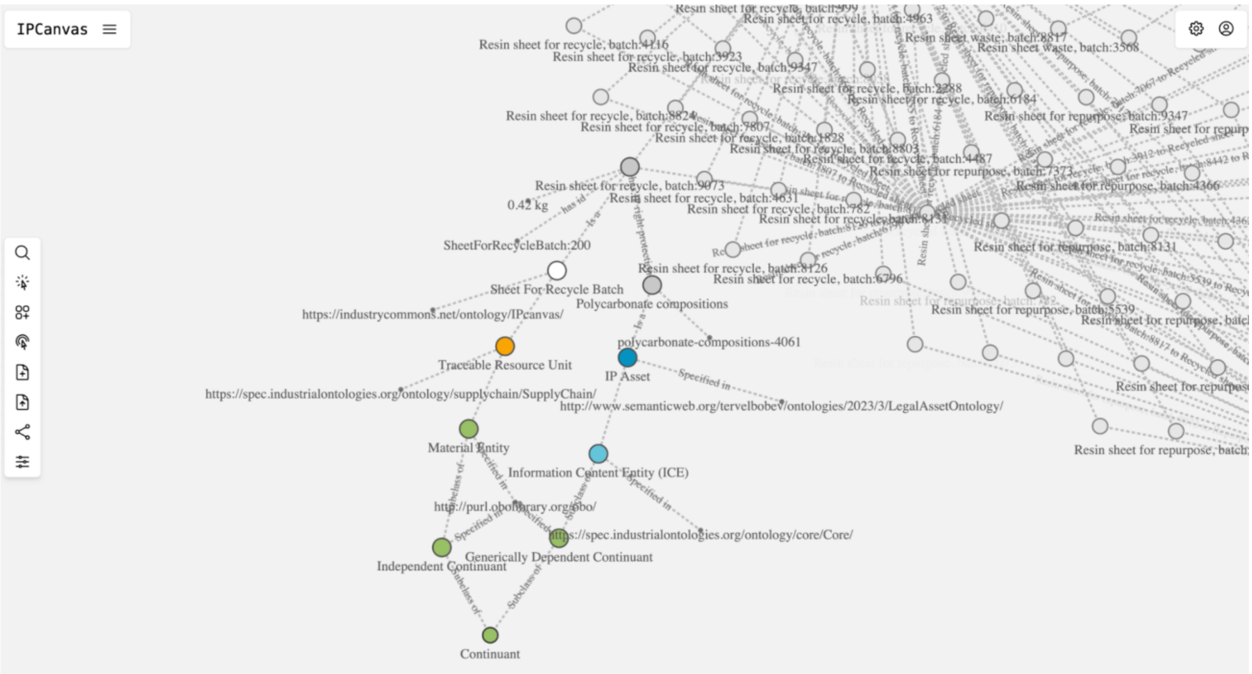


Figure 10: The IP ontology (in blue) and the supply chain ontology (in orange) are connected from the application level up to the BFO top-level ontology (in green)

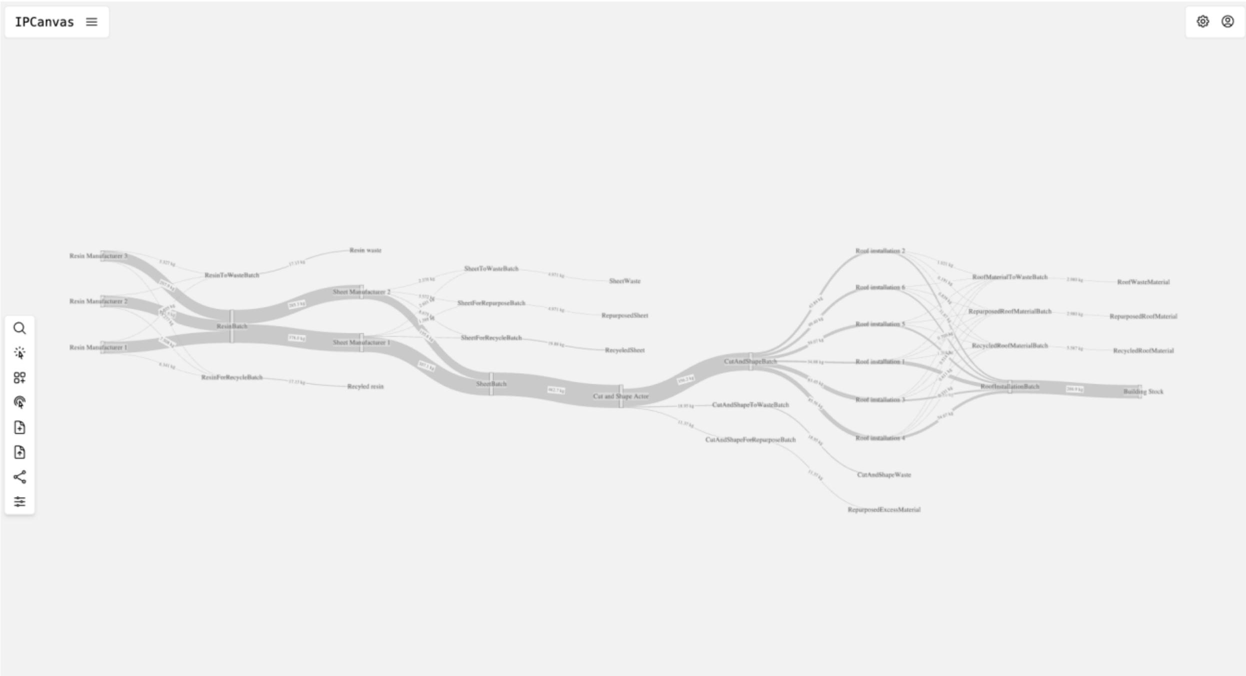
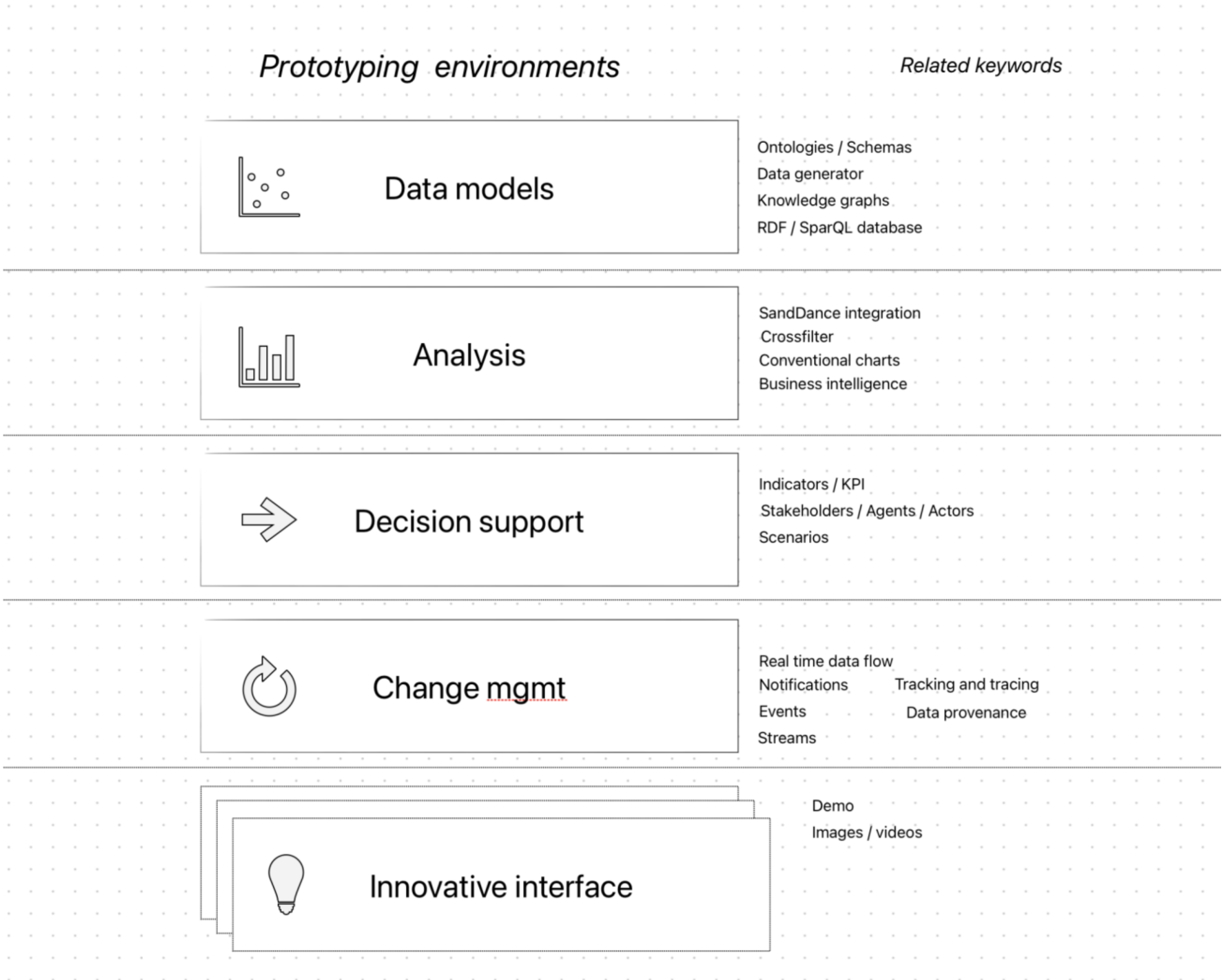


Figure 11: The Sankey chart is useful to show the flow from raw materials through the life cycle of the manufacturing process. In each station the side streams are also shown distributed between waste and different circular flows

2.5.4 Information Flow

To use the environment after installation, the user loads and prepares the data from external sources and adapts it to JSON or CSV files/variables. These are sent to the JS/TypeScript component that uses the data to create the knowledge or flow graphs. In the resulting component graph window, the user can instruct the component to print or copy the graph for external documentation. The graph allows for interactive operation, allowing the user to interact with data points in the graph to connect to other data sources, such as systems for storing attributes, etc.

The prototyping in RE4DY started with a straightforward methodology to tackle the challenges in production, product and supply chain data. The methodology steps include data models, data analysis, decision support, and change management before exploring and evaluating different interfaces.



The first step is identifying the data models, preparing for data that could come from industrial processes, and figuring out how to bring semantics into the data. This can be related to how data pipelines are created between data lakes and data warehouses, inspired by newer concepts such as data lakehouses, which can move more freely between unstructured and structured data using interoperability and metadata layers.

In the data models stage, different data formats, schemas, and ontologies are explored to move from industry knowledge to raw data and map the data coming into existing structures.

The data analysis step is a broad section that involves the whole team being able to explore data in any preferable way, such as through business intelligence tools. The prototyping environment should here start to load the data for purely explorative and statistical purposes using conventional visualisations such as charts and diagrams, as this kind of visualisation can be brought into the innovative interfaces.

One interesting concept when analysing data at this stage and connecting it to manufacturing data is to use unit visualisation. This concept is well defined in visualisation technology and will give a view where all items can be identified and rearranged according to different settings to give insights. Sanddance is an open source library used in our data analysis to bring in synthetic IP assets, to be sorted by relevance, location, category, and so on.

Figure 12: Unit visualisation in 3D using Sanddance

Working with data analysis in an explorative fashion, looking at existing solutions, but also in the prototyping environment, it should be possible to bring out interesting patterns to be used for innovative views. With innovative interfaces comes the challenge of novel solutions. UX experiments are recommended to try the solution on a suitable target group.

In this next step, we summarise the indicators we want to communicate in the result. In this way, we can, at a very early stage, evaluate the interfaces' usefulness with stakeholders in the context of their business operations. This can be considered part of a decision support system setup where KPIs will be the initial part of a workflow assessing the operation compared to different scenarios. The more indicators involved, the more challenges will be put on the input and output of the prototype to communicate different trade-offs of the various measures.

In a decision support process, both existing collected data from the operation must be input, and different measures and catalogue data must be used to create scenarios. This needs to be mapped to the ambitions of the stakeholders as the trade-offs commonly result in the need to adjust the ambition level. Ambitions can come from executives opting for more sustainable solutions and from the economic department mapping this to the cost, but also from different external stakeholders affected by the decisions that have meaningful say in the matter.

The change management perspective is often complicated and overlooked in data pipelines, resulting in much work that must be done repeatedly. Changes in the information models, ontologies, raw data, indicator settings, etc. It is recommended to build a timeline from the start to imagine not only how data is changing but also how revisions of documentation and models can be dealt with smoothly. This would be favourable if change management could be built into a pipeline, such as those inspired by CI/CD. If not, the prototypes will be expensive to keep updated. It should also be possible to discuss change management as part of the problem and solution in innovative interfaces connected to manufacturing. This is especially important in the context of digital twins, as these systems are defined by being "digital replicas". If the digital twin needs to be updated manually as the physical phenomenon is changing, this can be considered a warning signal that the twin is not sustainable. However, for prototyping and POCs that is perfectly normal, just as long as it is clear that change management is not taken into consideration.

2.5.5 Internal Architecture

The knowledge graph visualisation environment consists of a JS/TypeScript application with a defined API for data and results management and presentation. The application uses several external tools to visualise the data listed below.

2.5.6 API

The environment is a visualisation environment that provides visualisation components in different solutions. It does not contain an API for external systems access; it has only an API for local use. The user has to search and export the data either through programming or manually from the database system containing the data to be presented.

2.5.7 Implementation Technology

- GUI: JavaScript, https://react.dev/, https://vuejs.org/. https://headlessui.com/, https://tailwindcss.com/
- Backend: https://nextjs.org/, https://expressjs.com/
- Graphics programming: https://threejs.org/, https://deck.gl/, https://luma.gl/

2.5.8 Comments

None

2.6 ICF Meta Repository Demonstrator

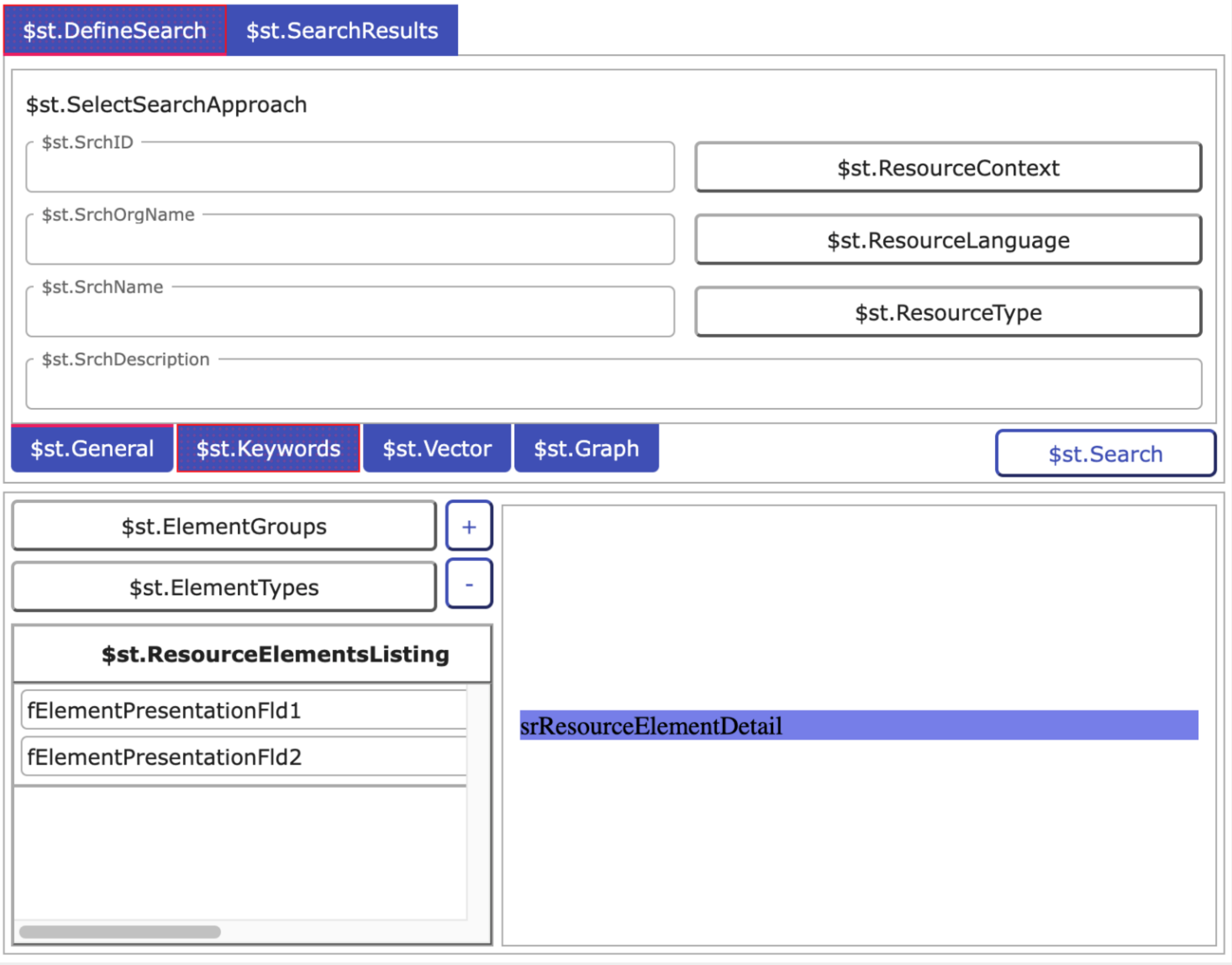
2.6.1 Description

The ICF Meta Repository Demonstrator (MRD) is a management infrastructure for related data and meta-data for objects, knowledge resources, interconnectivity resources, and relationships. It allows the management of many information resources of different types and contents that can be extended over time to meet future challenges. The MRD can be used to demonstrate a simple Digital Twin/curator support management system in the RE4DY project.

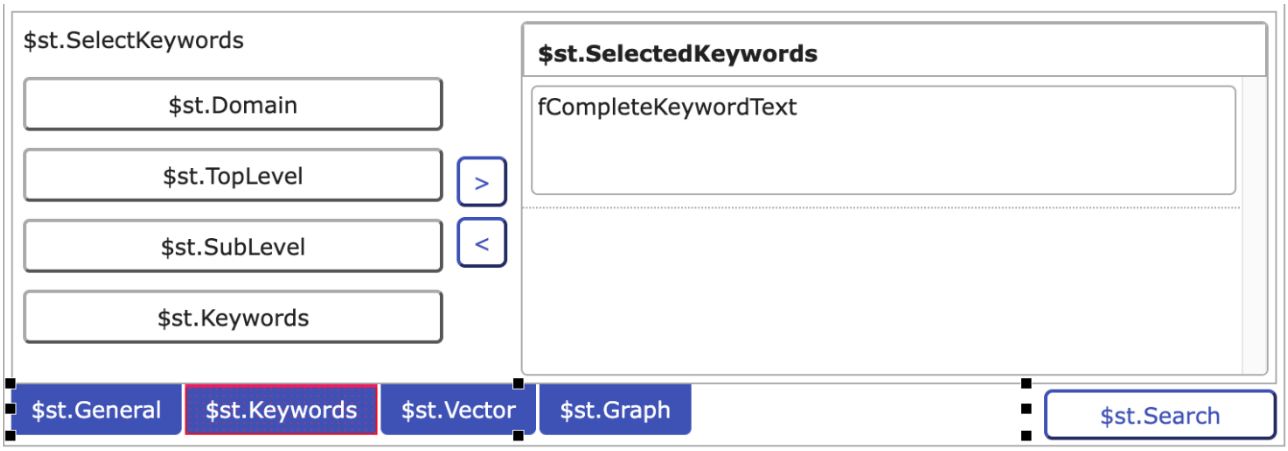
The system is modular, and modules can represent individual records/information entities as "mini-apps" that specialise in performing specific actions and operations, such as interacting with external computational services.

The system is prepared to integrate data from several databases and services into a specific context, acting like a hub. The planned system will include search mechanisms such as multilingual hierarchical keyword search approaches and vector database searches across multiple domains.

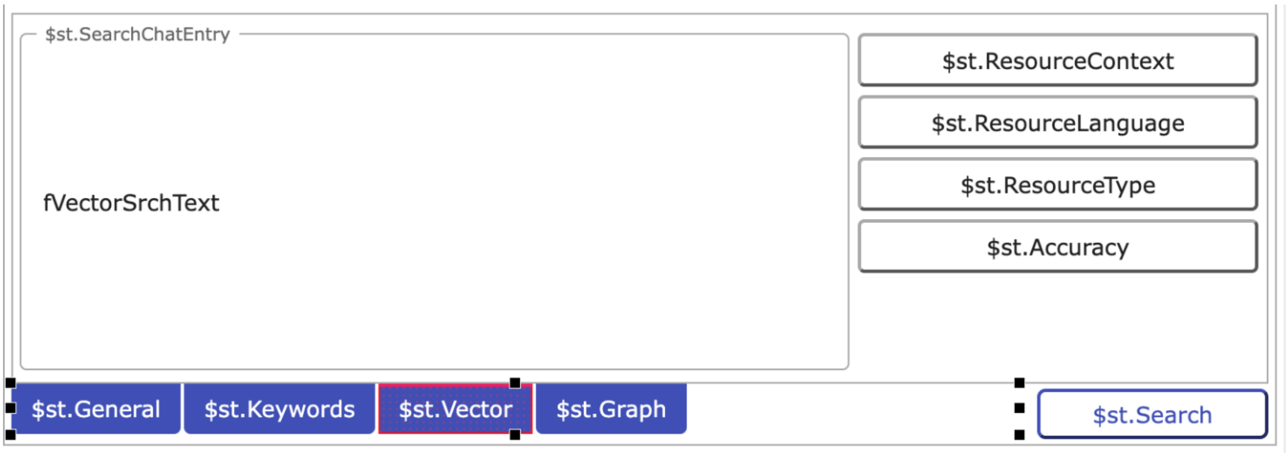
Any resource retrieved in the system can be in parallel connected to different types of resources, including multimedia and web resources and interfacing tools to qualified external services and functions. A dataset can be connected to resources like FAIR evaluation tools, instructions and documentation and a "mini-app" to interact with an external analysis service to evaluate the dataset if it is dynamically updated. Below the search screen for the system, it allows the search to include resources from several domains and types of resources. The text labels in the system are prepared for automatic localisation of the GUI depending on the language settings of the end-user computer system – supports Unicode and thus languages such as Chinese, Arabic, Hebrew, etc.



Below, the module for keyword searches has multilingual support, allowing the user to use different languages for the keywords but retrieve the same result set—a search on German "Tisch" results in the same result set as the English "bench." The keyword system is prepared to support a relatively high level of granularity and to include keyword structures for use contexts, environmental factors, etc. The keyword system is ready for shared use. Other actors can use the central PostgreSQL database system for the keyword catalogue, and the developer of the end-user systems must develop the mapping processes on the client side for the system to work.



The vector search option allows records in the system to be searched based on the vector representation of the data if the data is prepared for that type of search. Vector search can be compared to more straightforward implementations of ChatGPT (excluding analysis functionalities), giving the results as record listings from the database. Supporting vector searches demands the creation of "embeddings" in the database representing the vector representation of the data. Depending on the implementation, vector and graph data can be set in parallel to update external databases, which is helpful in external analysis and reporting. A search interface for local LLM searches or services such as ChatGPT allowing for AI-based analysis of datasets is possible but demands more planning and adaptation work.



The Graph option is currently under evaluation. To minimise clutter, graph data management might start with qualified searches that lower the number of nodes managed—which is vital for larger datasets. At a later stage, integrated visualisations can be supported by Apache eCharts graphics libraries.

2.6.2 Input

As a Digital Twin/Curator Support Platform, the resources during the demonstration are manually added (an option, if needed, is to import existing data) and linked to other relevant objects. Beyond industry-specific information types, data can be entered in the form of media, external links to web-based data, and external databases and services in specific cases. External data storage (measurement and time-series data) are examples of external services.

2.6.3 Output

End-user web-based GUI for desktop systems is covered above. Access to data via REST calls and direct database access calls is optional. The end-user environment can interface with other externally connected systems—report and export functionalities.

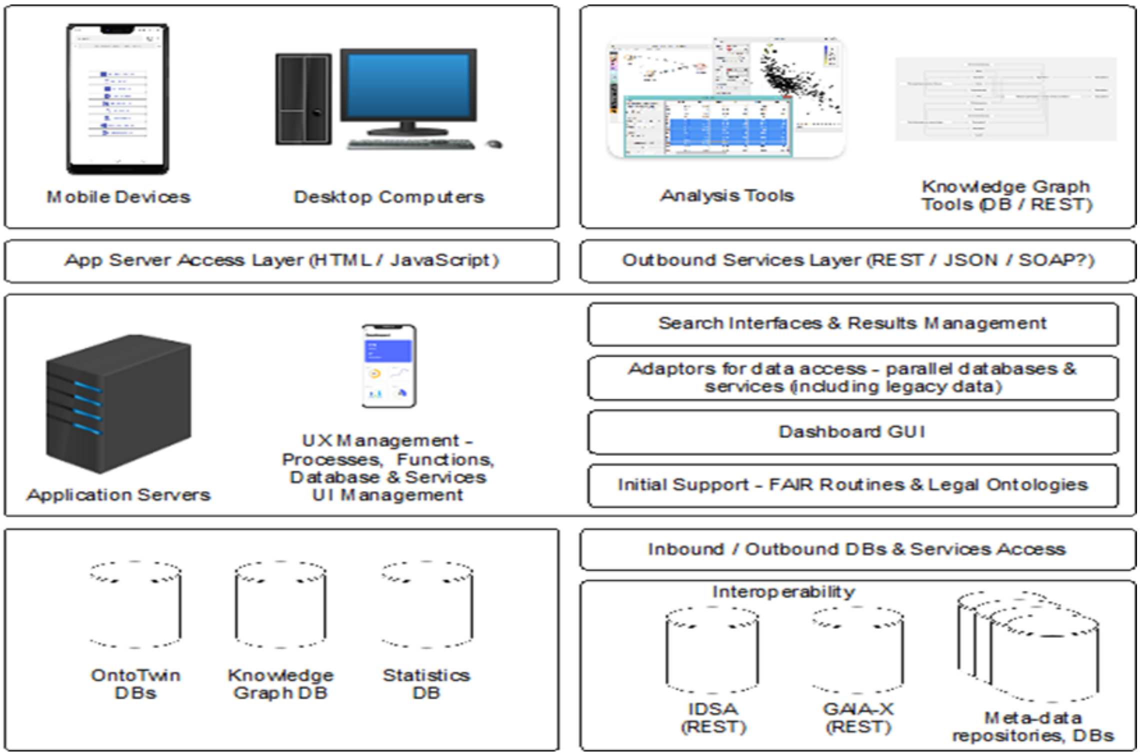


Figure 13: GUI of Meta Repository Demonstrator

2.6.4 Information Flow

MRD can be regarded as a resource hub architecture. It stores and manages a multitude of data and information that can be linked freely between themselves and with data stored in external (also legacy) data. It also can act as a service provisioning hub.

2.6.5 Internal Architecture

The system is separated into three main functionalities: database management, application server and interconnected end-user devices of different types.

The database management layer allows parallel connections and interactions with several different types of databases and services, often allowing them to show data in parallel in the different modules of the system. The chosen database platform for this implementation is PostgreSQL, but close to all relevant databases of various types are supported, either natively or by middleware. Access to data, information, and services can also be achieved through REST, SOAP, and other standard approaches on the market.

The application server acts as a coordinating hub, allowing interaction between multiple data sources, business logic, and end-user deployment modes.

Deployment modes supported range from web-interface clients (stationary during the test cases) to web services provisioning.

2.6.6 API

Beyond the end-user GUI devices of different types, the main APIs for access are REST and, in some cases, direct-controlled DB access or SOAP/XML. The core data for the information types and resources can be based on relevant JSON-LD schemas for different domains and use-cases.

2.6.7 Implementation Technology

- End-user systems: webpage-based GUI for desktop and mobile devices (option), native applications for Android and iOS (option - mobile phones and tablets).
- REST access (JSON)
- Application server: Omnis Studio application server (omnis.net) for managing end-user devices and accessing internal/external databases and services.
- Database server: PostgreSQL database server, which is used for both core database systems and for supplementary/supporting databases such as Knowledge Graph (pgGraph), Vector database (pgVector) and statistical support/(optional) time series database systems.

2.6.8 Comments

Due to its effective pilot development processes, Omnis Studio was chosen as a development tool for application server environments and end-user systems. The application server layer can later be developed in other development environments using the now-developed system as a specification example.

2.7 ICF IPscreener Integration

2.7.1 Description

The ICF IPscreener integration allows for effective exploration and understanding of the knowledge hidden in patents. Using a text description the AI-assisted software presents an instant dashboard of the innovation landscape. The results show trends on main competitors, R&D activities and business opportunities, they identify similar solutions, assist reading, interpretation as well as collaboration and automated reporting. IPscreener can help to validate and improve decision making strategies.

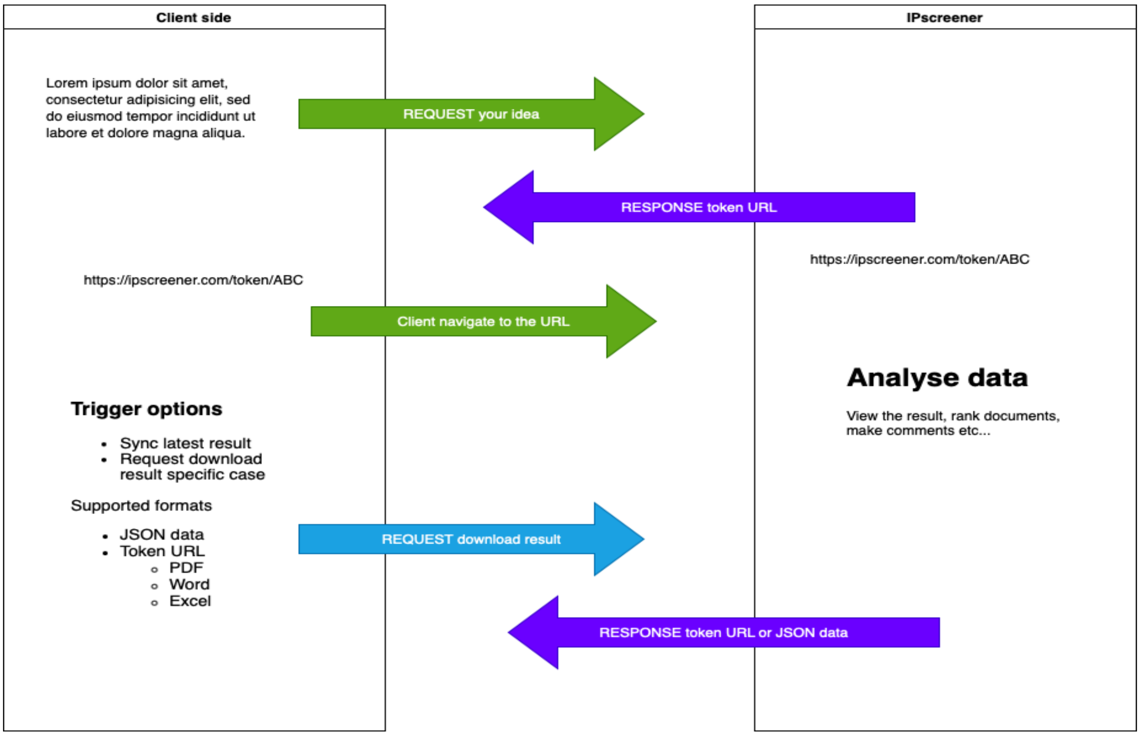
2.7.2 Input

Input is in text or voice, where main world languages are possible to use.

2.7.3 Output

Resulting data is output via a ticketing system according to the set API parameters tagged with the query, e.g. including identified document data, statistics, trend analysis, and comments made.

2.7.4 Information Flow



2.7.5 API

Official documentation describing the API framework and associated parameters is available and continuously updated at https://docs.ipscreener.com/developer

2.7.6 Implementation Technology

The general IPscreener platform is based on an engine hierarchy of Docker containers. This Docker engine configuration enables easy hosting of the main frame infrastructure both in a cloud environment as well as behind a firewall in local setup virtual space. Furthermore, the IPscreener core functionality is based on API interacting endpoints, providing easy integration as well as function sharing of the different services included.

The data sources are also interacting via APIs, based on a network of different database engines depending on the task. Examples of those are SQL, NoSQL, Redis, Elastic search, Lucene/Solr.

Training of different Machine Learning is engineered via Python libraries and frameworks including such as Tensor flow, BART, BERT and other LLM models. Adapted training is performed on the fly to adapt specific analysis to a topic domain.

2.7.7 System comments

The IPscreener platform is operating both as an independent fully compliable WEB service or via remotely integrated via REST access points. Data points as well as GUI/UX components are accessible for integration via both options.

2.7.8 Functionality

The connectivity is adapted to dynamically respond to system requests in the meta data hierarchy model of descriptive functional modules integrated in the chain of include meta definitions along a knowledge graph.

2.7.9 Scenario Checkpoint

Scenario A: IP Mapping of modules within a Knowledge Graph hierarchy – Development of innovation landscape KPIs

In this scenario, the intellectual property (IP) landscape is analysed to map and evaluate existing rights in view of supply chain descriptive ontologies, which serve as references for showing an innovation landscape of highlighting IP dependencies, show alternative solutions and identify technological trends. This ontology is tagged into a Knowledge Graph repository, enabling the testing of analytical functionalities within the broader IP framework upwards as well as downwards in the functional hierarchy branch analysed. The evaluation process culminates in key recommendations and a detailed record of the experiences encountered, which aids decision making strategies.

Key outcomes include:

- An initial version of an internal hierarchy ontology system, informed by IP considerations.
- An IP assets hierarchy aligned with a Knowledge Graph representation, integrated with visualization tools developed in subsequent potential Scenarios.
- Comprehensive documentation of lessons learned, along with recommendations for future IP-driven development and associated implementation strategies.

Participation units	• Meta data model • Knowledge Graph system • RE4DY ontology visualisation environment • IPscreener backend AI analysis and IP extraction module
Framework and data modeling	Analyzing production steps and components highlights potential bottlenecks where intellectual property (IP) rights might be crucial. Identifying these areas enables a thorough evaluation of IP protections, ensuring key assets are safeguarded and optimizing the management of related dependencies throughout the process. The knowledge graph components (modules and layers) shall include relationships / dependencies.
Proposed Gains	• Better awareness; Understanding of IP implications and possibilities • Enhanced analytical capabilities: Integration of IP-ontologies and dependencies along the knowledge graph hierarchy • Informed decisions; Increased accuracy of analysis of IP relations to and improves the quality of decisions.
KPIs	• Main KPI: IP aligned Knowledge Graph visualization • End goal KPI: Component hierarchy model documented IP innovation landscape data and IP-strategy insights.
Partners Participation	Lead: ICF; Partners: KU Leuven, Chalmers (related ontologies)

2.7.10 Comments

None

2.8 5G Digital Twin (5G AAS)

2.8.1 Description

This component represents the digital twin or Asset Administration Shell of a complete 5G system. An AAS is a digital model or a set of submodels that provides relevant information and features of an asset. AASs describe the technical functionalities exposed by the assets, ensuring interoperability between systems managing manufacturing processes. Asset Administration Shells (AAS) are then standardized digital representations or digital twins of assets and play a crucial role in the management and administration of assets within manufacturing environments.

The digital twin or AAS of a 5G system is integrated by two main components: the digital twin of the 5G UE, which is the endpoint of a 5G link, and the digital twin of the 5G network that encompasses all the nodes and functions within the 5G Radio and Core networks. This component contains both the 5W NW and UE digital twins, providing then the digital twin or AAS of a complete 5G system.

Figure 14: Representation of the main components of a 5G Digital Twin or AAS

The implemented 5G system digital twin follows 5G-ACIA[1] recommendations and guidelines to facilitate the integration of 5G with industrial systems and applications. It is also 3GPP-standard compliant and follows the current AAS standards defined by Plattform Industrie 4.0[2]. The 5G system digital twin is made of a 5G UE digital twin and a 5G NW digital twin that provide access to, and expose, the main parameters and capabilities of 5G networks to support Industry 4.0 systems and applications. The designed 5G UE and NW digital twins include passive properties and define active operations for the interaction between 5G components as well as between 5G and industrial devices, systems, and applications. The 5G system digital twin corresponds to a type 2 or reactive AAS and includes a message interface that supports interaction between AASs, and between AASs and software applications. The interface enables controlled access to AAS data and operations. The

[1] *5G-ACIA, Using Digital Twins to Integrate 5G into Production Networks, 5G-ACIA White Paper, Feb. 2021.*

[2] *Platform Industrie 4.0, Details of the Asset Administration Shell - Part 1, Specification, v3.0RC02. May 30, 2022*

AASs have been implemented using the AASX Package Explorer[3] and BaSyx[4] (in particular its SDK for Python).

2.8.2 Input

The 5G system digital twin obtains information from the 5G network and UEs. The information obtained shows the current status of the 5G network and UEs, as well as the 5G connections established in the network. Providing access to this 5G information facilitates the efficient integration of 5G with industrial or manufacturing processes. Such integration requires the capacity of industrial applications to access information about the 5G capabilities and performance.

To this aim, the 5G digital twin will utilize the interfaces and exposure functions enabled by the 5G systems to make this data available, including the Network Data Analytics Function (NWDAF), the Network Exposure Function (NEF), and the Service Enabler Architecture Layer (SEAL). Both digital twins can also receive input data from other digital twins. Specifically, they can provide information about the status and operation of each other. Moreover, the digital twin of the industrial device to which the UE is related will provide input data about the traffic generated by the industrial application and the status of the device.

2.8.3 Output

The 5G digital twin exposes and simplifies access to 5G data, allowing for easier management and utilization of the 5G technology within manufacturing systems. The 5G digital twin structures and organizes the data obtained from the 5G system and makes it accessible following and standardized approach that facilitates its seamless integration with the digital twin of manufacturing systems. The 5G NW and UE digital twins or AASs provide the capacity to retrieve data on demand, periodically or on an event-triggered basis following the exposure interfaces in 5G. The AASs utilize event subscriptions to expose or receive from other AASs data that may change dynamically. An event may be an on-demand request for data or periodic reports to monitor the evolution of the target data. It is also possible to trigger events when specific conditions occur, for example, when a parameter that is monitored exceeds a predefined threshold or a UE enters a certain area in the factory plant. To manage the event subscriptions, the submodels within our AASs contain a list of requested event subscriptions, and a list of notified events for each subscription.

[3] *AASX Package Explorer. Github repository:* https://github.com/admin-shell-io/aasxpackageexplorer

[4] *Eclipse-basyx. Basyx-python-sdk. Github repository*: https://github.com/eclipsebasyx/basyx-python-sdk

2.8.4 Information Flow

We present an example to illustrate the use of the defined 5G digital twin to support industrial processes and the information flow between different entities. To this end, we consider an evolution of a use case proposed by 5G-ACIA that deals with QoS management in a 5G network supporting a production process. The use case considers a factory shopfloor where there are two 5G-capable industrial robots that have active connections with the 5G NW using their 5G UE (UE1 and UE2). Each of these active connections has a guaranteed QoS profile. A new production process starts and requires high-resolution image inspection. A 5G inspection camera is activated to transmit the images to an image analysis unit on the cloud for automated inspection. Each 5G UE and the 5G NW implement their 5G UE and NW digital twin, respectively. The UE associated to the 5G camera (UE3) requests a new connection (or PDU session) with a high-bandwidth QoS profile (NewConnectionRequest operation in the 5G UE3 digital twin). Upon reception of the new connection request, the 5G NW digital twin initiates the QosMapping operation to determine the QoS profile that can be guaranteed for the new connection based on the aggregated information about the current state of the network, the QoS requested (including its priority), and the QoS profiles of the active connections. The 5G NW digital twin informs the 5G UE3 digital twin of the guaranteed QoS profile, and the 5G NW digital twin establishes the new connection for UE3 (EstablishConnection operation). The activation of UE3 changes the network status, and the 5G NW digital twin sets new RAN parameters for UE2 (using the SetRANConfiguration operation in the 5G UE digital twin) to maintain its guaranteed QoS profile, e.g. it increases the number of retransmissions. In addition, the 5G NW digital twin needs to change the QoS profile for the UE1 using the ModifyConnection operation in 5G NW digital twin. After all changes, the 5G NW digital twin estimates the performance of all three active connections to assess whether the QoS requirements are met (PerformanceOfAPDUSession operation). The QoS performance experienced is updated in the 5G UE digital twins (*QosPerformance* submodel) and 5G NW digital twin (*QosMonitoring* submodel), and is made accessible to industrial applications or devices through the corresponding subscriptions and events.

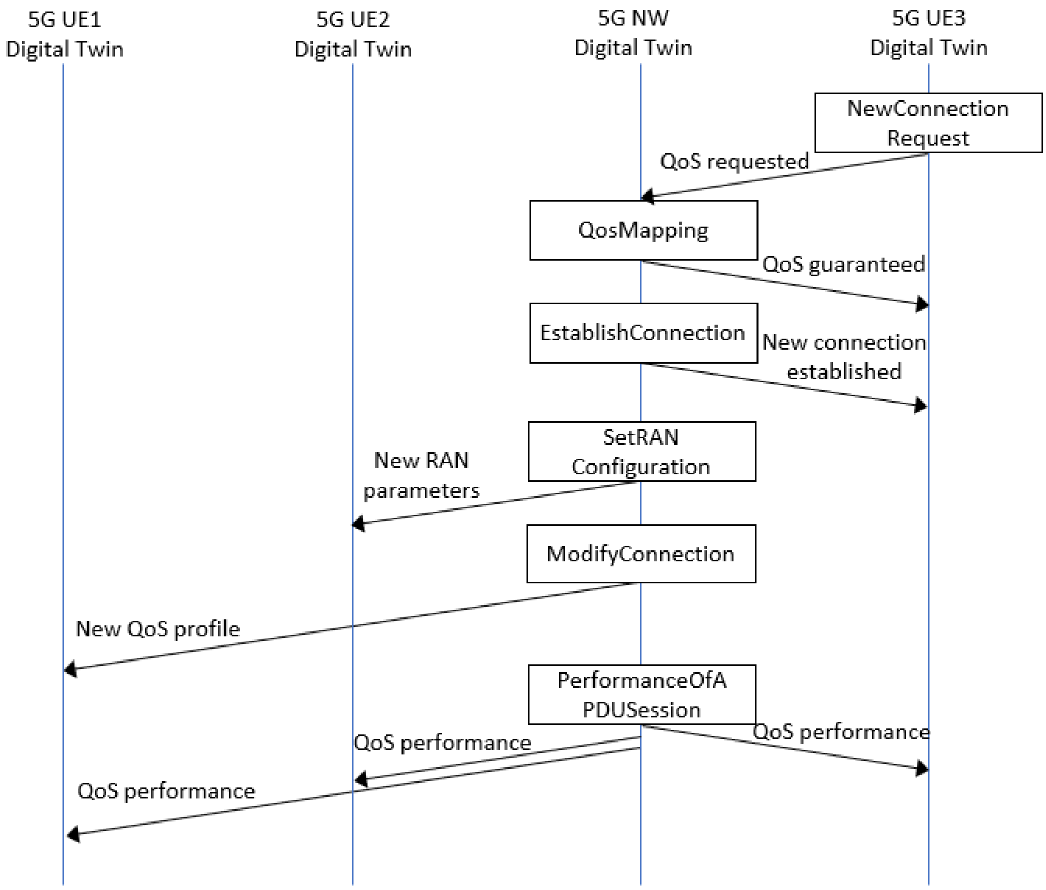


Figure 15: QoS management use case using the 5G digital twin

2.8.5 Internal Architecture

The information and functions within the 5G digital twin are organized in different submodels including all the passive data along with the active operations that we propose should be included in the submodels. Submodels are technically isolated from each other. Each submodel comprises submodel elements, which serve as the components suitable for describing and distinguishing assets. These submodel elements can be properties, operations, and collections, enabling a hierarchical structure for asset differentiation.

The designed 5G NW and UE digital twins follow the recommendations of 5G-ACIA about what data and properties should the digital twin of a 5G system include based on 5G definitions within 3GPP. In addition, the 5G NW and UE digital twins have been designed following the specifications of Plattform Industrie 4.0 for the definition of AASs. In this context, they include, in addition to all 5G submodels that will be presented in the following sections, the following five default submodels for interoperability of the AASs: *Nameplate, Identification, Documentation, Service* and *TechnicalData* submodels:

- *Nameplate* submodel provides identifying information about an asset, such as the manufacturer's name, model type, and serial number.

- *Identification* submodel is utilized for property recognition. It comprises elements such as the manufacturer's name, supplier's name, asset ID, manufacturing date, device revision, software revision, hardware revision, etc. Despite some overlapping elements with the Nameplate submodel, these serve different functions in varied scenarios.
- *Documentation* submodel organizes and categorizes pertinent documents, making it easier to locate them. This encompasses items like data sheets, maintenance manuals, and user guides.
- *Service* submodel contains support and maintenance information. It offers contact details for necessary support. Elements within this submodel may include supplier name, contact info role, email, or phone number.
- *TechnicalData* submodel aims to provide interoperable technical data that describes the asset of the respective Asset Administration Shell.

Standardized templates are available in the repository of Plattform Industrie 4.0.

2.8.5.1 Internal Architecture of the 5G NW Digital Twin

The definition of a 5G NW digital twin can be a challenging task due to the complexity of 5G networks that are integrated by many physical (antennas, base stations, network servers, etc.) and logical elements (virtual RAN and CN network functions). The designed 5G NW digital twin then follows a functional design where information is structured and grouped by functions or operations of a 5G network rather than by network nodes. This facilitates an easier evolution and scaling of the 5G NW digital twin as the implementation can be tailored to the specific functionalities that are necessary or more relevant for each implementation of the 5G system digital twin. Figure 16 shows the internal architecture of the designed 5G NW digital twin and its submodels which are next described.

Figure 16: Submodels of the 5G Network digital twin or AAS

- *NPN5GNWIdentity* submodel: This submodel includes information to identify the network, in particular, the Public Land Mobile Network (PLMN) ID or Non-Public Network (NPN) ID.

- *AssetServiceRegistry* submodel: This submodel contains information about the characteristics of the 5G network at planning and deployment phases, including the Asset Service description, the identification of the Integrator Company and planning references. It also provides 5G coverage maps for the factory where the network is deployed, and information about the Service Level Agreements (SLA) between the 5G network operator/provider and the factory operator. The SLA includes the metrics by which the level of service is measured, and the expected performance per service type.

- *Network5GDataSheet* submodel: This submodel includes information that characterizes the technical capabilities of the deployed 5G network, including the supported 3GPP release, the supported and used spectrum bands, the maximum data rate achievable in downlink (DL) and uplink (UL), and the supported network protocols (IPv4, IPv6, etc). The submodel also provides information about the network topology and includes a list of all RAN and CN nodes that comprise the deployed 5G network. For each RAN and CN node, the submodel indicates the type of node (gNB, UPF, PCF, SMF, etc.), its IP address, its connections with other nodes, its location, the virtual machine hosting the node (if this is the case), and the computing and memory resources allocated at the virtual machine. In the case of RAN nodes, the submodel also includes information about the maximum transmission power, the spectrum band, and the receiver sensitivity. The submodel also includes information about the links between nodes as well as the links' capacities.

- *VirtualNetworks* submodel: 5G allows the implementation of virtual or logical networks to support different QoS (Quality of Service) levels. One option is network slicing, which exploits the virtualization and softwarization of networks to create different logical networks or slices over a common network infrastructure. Network slicing is considered one of the main 5G technologies to support industrial services with diverse and stringent requirements using a single 5G network. Network slices (NS) are configured to support specific QoS profiles. A 5G QoS profile defines a set of QoS parameters and characteristics that determines the treatment of the data traffic in the 5G network. 5G can also implement Virtual Local Area Networks (VLAN) to manage separately different traffic flows. VLANs operate at the data link layer and are logical Ethernet networks that are created over the same physical infrastructure by assigning ports on network switches to specific VLANs. In a VLAN, traffic is separated by including the VLAN ID in each frame.

  The *VirtualNetworks* submodel contains the list of NSs and VLANs created. For each NS, the submodel indicates the Single Network Slice Selection Assistance Information (S-NSSAI) as well as the configurable attributes and their values. The S-NSSAI identifies the NS and the Slice/Service type (SST). The SST defines the expected NS behaviour in terms of features and services. The attributes that characterize a NS include, among others, availability, area of service, isolation level, UE density, and UL throughput per NS. The submodel also identifies the computing resources of the virtual machines where the NS is executed. For VLANs, the submodel indicates the IP address, VLAN ID, priority, VLAN tag (identifies the VLAN in a data frame), and allocated computing resources to execute the VLAN.

The *VirtualNetworks* submodel has an active part to implement the *NetworkSliceReconfiguration* and *VLANReconfiguration* operations. These operations should emulate the reconfiguration process of a VLAN or network slice respectively, based on, for example, new network requirements.

- *Connectivity* submodel: The *Connectivity* submodel provides information about the UEs that are currently connected to the 5G network. In particular, it includes a list of attached UEs identified by their Permanent Equipment Identifier (PEI) and Generic Public Subscription Identifier (GPSI). The submodel also includes a list of the established PDU (Packet Data Unit) sessions. A PDU session is a logical connection within the 5G system between a UE and the user plane function (UPF) that provides access to a Data Network (DN). The submodel includes per PDU session the list of established QoS Flows. Each QoS flow is identified by a unique identifier called QoS Flow ID (QFI), and by a QoS profile that includes the QoS parameters that characterize the packet flow. The QoS parameters include the 5G QoS Identifier (5QI), Allocation and Retention Priority (ARP), Reflective QoS Attribute (RQA), Guaranteed and Maximum Flow Bit Rate (GFBR and MFBR, respectively), Notification control, Maximum Packet Loss Rate, and Aggregated Bit Rate. The 5QI is a scalar value that identifies a series of QoS characteristics. These characteristics include, among others, the resource type, priority level, packet delay budget or PDB (i.e. the latency requirement), packet error rate (PER), averaging window and maximum data burst volume. The submodel includes the QoS profile requested in the PDU session establishment procedure, and the QoS profile guaranteed by the network after the QoS mapping process. The requested and guaranteed QoS profiles might not match if the network does not have sufficient resources.

The *Connectivity* submodel lists four possible operations. The *EstablishConnection* operation should emulate the establishment procedure of new PDU sessions. It can be called, for example, when a new UE is attached to the network and needs to establish a new PDU session. The *QosMapping* operation should implement the QoS mapping process carried out in 5G when a new QoS profile is requested. The process considers the current state of the network to derive the guaranteed QoS profile. The *QosMapping* operation could also be executed to (re-) negotiate the specific QoS parameters of a new or already established connection if the UE QoS requirements change. The *Connectivity* submodel also includes the possibility to implement a *SetRANConfiguration* operation. This operation establishes the value of different RAN parameters (at both the 5G NW and 5G UE AASs) that are important to support specific QoS profiles, including among others, the use of a specific or a set of Modulation Coding Schemes (MCSs), the maximum number of retransmissions or the 5G numerology. Finally, the *Connectivity* submodel includes the *ModifyConnection* operation to emulate the PDU session modification procedure in 5G. This procedure is executed to modify the configuration, in particular the QoS profile, of an already established connection or PDU session. The procedure can be initiated by the network or following a request from a UE.

- *QosPerformance* submodel: The *QosPerformance* submodel provides information about the performance experienced for a specific packet transmission, a QoS Flow or a PDU session. The performance can be expressed in terms of: service availability (percentage of time that the service is delivered according with the guaranteed QoS profile), reliability (percentage of packets successfully delivered within the latency requirement or PDB), Packet Error Rate (PER), service bit rate, Block Error Rate (BLER), data throughput and latency. The submodel also indicates the update time that defines the periodicity with which the performance is calculated. The *QosPerformance* submodel also provides information about the performance achieved in each virtual or logical network (network slice or VLAN), in particular, the experienced PER, BLER, reliability, latency and throughput. This information is important to identify any malfunction or incorrect network configuration. The performance of the virtual or logical networks is also computed periodically based on the update time parameter configured in the submodel.

  The submodel provides a list of operations to estimate the performance experienced for a specific packet transmission (*PerformanceOfAPacketTx*), a QoS Flow (*PerformanceOfAQoSFlow*) or a PDU session (*PerformanceOfAPDUSession*). With these operations, it is also possible to provide information about the performance experienced by one or a group of UEs during a given period. Finally, the *QosPerformance* submodel contains a list of subscriptions and a list of notified events. An external application or AAS can subscribe to receive information periodically, on demand, or on an event-basis about the QoS performance of 5G connections or PDU sessions, leveraging the 5G exposure capabilities introduced with NWADF, NEF and SEAL. In addition, the submodel includes the *SubscriptionManagement* operation to manage subscriptions requested by UEs or vertical applications.

- *TSNCapabilities* submodel: 5G has been designed with the necessary enablers and functionalities to integrate with Time Sensitive Networking (TSN) networks and jointly support Time Sensitive Communications (TSC). In integrated 5G and TSN networks, the 5G network acts as a TSN bridge, also referred to as the 5GS Logical Bridge. In this context, it is important that the 5G NW AAS includes a *TSNCapabilities* submodel like any other AAS of a TSN bridge to identify the 5G network as a TSN-capable device. This submodel includes information specific for a 5GS bridge like the 5GS Bridge delay and the propagation delay per port, as well as configuration parameters including the IP, ports, VLAN-ID and VLAN priority. The submodel also includes a list of the active streams that correspond to different TSN flows. For each TSN flow, the model provides information about the parameters included in the TSCAI or TSC Assistance Information (survival time, packet arrival time and periodicity). We should note that the TSCAI parameters are provided by the TSN network so that the 5G system can adequately support the TSN traffic.

- *Location* submodel: 5G New Radio (5G NR) can significantly improve the positioning accuracy compared to previous mobile generations, which opens the door for the use of 5G positioning in vertical applications (including Industry 4.0). The 5G NW AAS includes a *Location* submodel to register the position of all UEs connected to the network. Each UE is identified using its PEI, and the submodel provides the position using Cartesian coordinates, along with speed and acceleration information. The submodel also includes the Location Service QoS Class (LCS QoS Class) that indicates the positioning accuracy and response time to the positioning request tailored to the specific requirements of each application/UE. The submodel also includes a list with subscriptions to different location data and events that can be provided on-demand, periodically or an event-triggered basis, along with a list of the notified events. For example, a UE can subscribe to periodic reports of its location, or an industrial application can subscribe to notifications when one or a group of UEs enter a given area. The submodel also includes the possibility to implement a *SubscriptionManagement* operation to manage the subscriptions to location events requested by UEs and vertical applications.

- *QosPrediction* submodel: With NWDAF, a 5G network can provide not only analytics but also the capacity to predict the QoS of 5G connections. Such predictions will be crucial for a proactive management of the 5G network that can anticipate the QoS performance and/or connectivity needs of the production processes it should support, e.g. based on the mobility of UEs, their traffic demand or the network status. The current implementation of the 5G NW digital twin corresponds to reactive AAS. However, it includes a *QosPrediction* submodel to facilitate its future evolution to a pro-active AAS.

  The submodel includes a list of subscriptions requesting either network QoS predictions or QoS predictions of specific UE connections. For each subscription, the submodel includes the QoS parameters to predict, the period of time for the prediction, and if applicable, information about the area or path of interest where the predictions are needed. Industrial applications can subscribe to receive QoS prediction events on-demand, periodically or an event-triggered basis. The submodel also includes a list of the notified QoS prediction events. The submodel includes a *SubscriptionManagement* operation so that a UE or external application can request a subscription to a QoS prediction event, specifying whether the subscription is for an on-demand, periodic or event-triggered notification.

2.8.5.2 Internal Architecture of the 5G UE Digital Twin

The 5G UE digital twin enables the effective management and monitoring of a 5G UE that is part of a 5G-capable industrial device. By providing a comprehensive digital representation of the UE, the 5G UE digital twin facilitates the seamless integration of 5G into production networks. The 5G UE digital twin is defined following a functional approach like in the case of the 5G NW digital twin. Figure 17 shows the proposed 5G UE digital twin with its submodels that are presented next.

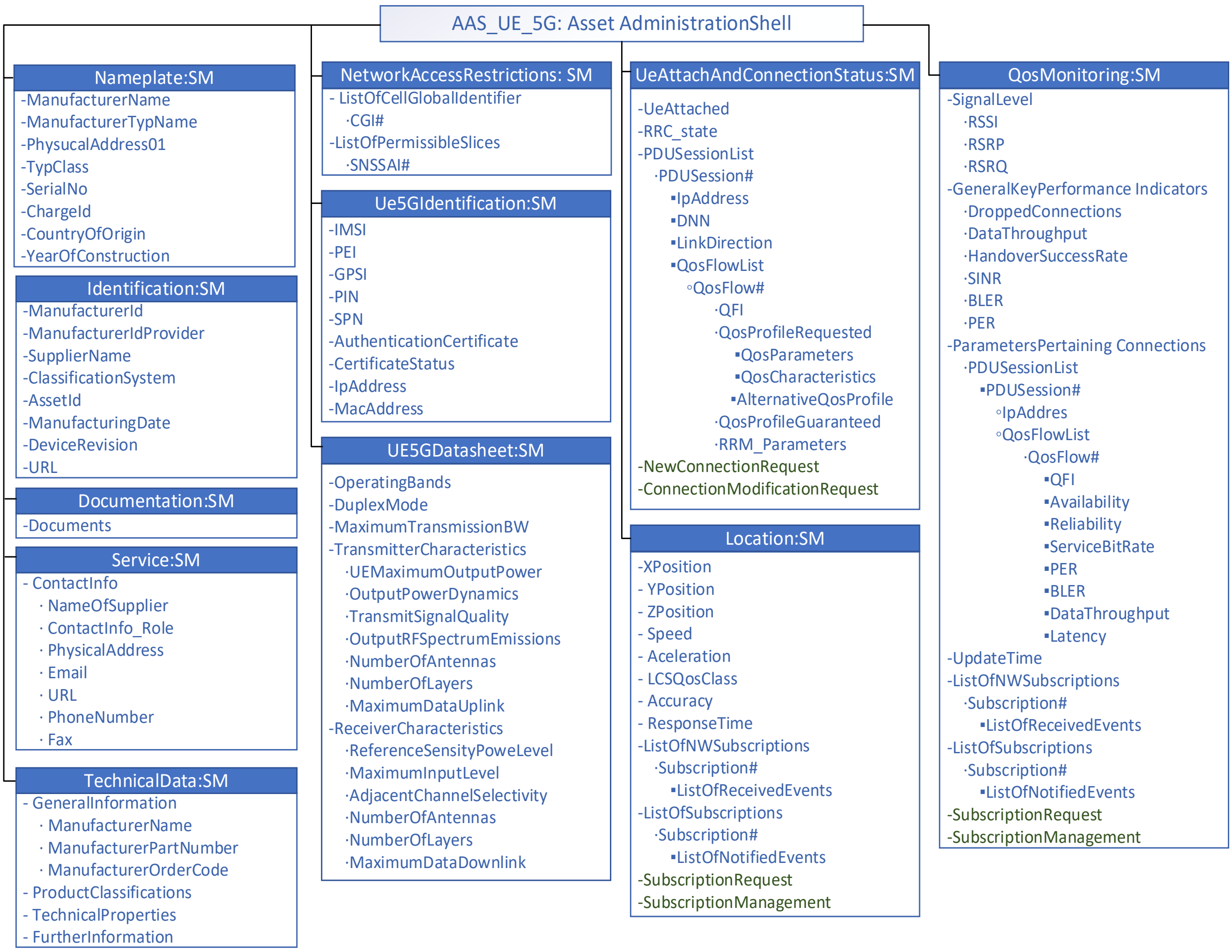


Figure 17: Submodels of the 5G UE digital twin or AAS

- *Ue5GIdentification* submodel: This submodel contains different UE identifiers, including the IMSI (International Mobile Subscriber Identity), the PEI (Permanent

Equipment Identifier), and the GPSI (Generic Public Subscription Identifier). The IMSI is a unique (and not public) identification code for each UE in a mobile network. The PEI is an identifier of the terminal equipment used by the UE. The GPSI is a public identifier used both inside and outside of the 3GPP system, for example, our public mobile phone numbers or mobile subscriber ISDN number (MSISDN). The submodel also includes the UE's PIN (Personal Identification Number) code, the ICCID (Integrated Circuit Card ID) that is a unique global serial number of the SIM (Subscriber Identity Module) card used by the UE, and the Service Provider Name (SPN) that identifies the serving mobile operator. Finally, the submodel includes the authentication certificate (and its status) used to authenticate a UE in a 5G network, as well as the IP and MAC addresses of the UE device.

- *NetworkAccessRestrictions* submodel: This submodel contains information related to the physical and logical access restrictions of the UE to the 5G network. The submodel stores the list of Cell Global Identifiers (CGIs) that identifies the 5G cells that the UE can connect to. It also has a list of the network slices (identified with their S-NSSAIs) that the UE can connect to. This list is determined by the 5G network together with the guaranteed QoS profile. It is important that the slices to which the UE can connect are capable of supporting the guaranteed QoS profiles for the UE.

- *UE5GDatasheet* submodel: This submodel includes technical characteristics of the 5G UE, including the operating bands and duplex mode it supports and parameters of the communications channel such as the maximum transmission bandwidth that the UE can support. The submodel also includes radio transmission and reception characteristics of the UE, including the maximum output power, output power dynamics, output RF spectrum emissions, reference sensitivity power level, adjacent channel selectivity, and number of transmission and reception antennas and layers.

- *UeAttachAndConnectionStatus* submodel: This submodel includes the UE connection status. In particular, it indicates if the UE is attached or not to the network, and the RRC (Radio Resource Control) state that indicates whether the UE is active, inactive, or idle. The submodel also includes a list of the active PDU sessions and QoS flows of the UE. For each PDU session and QoS flow, it also includes information about the QoS profiles requested and guaranteed by the 5G network. The submodel also includes the value of different Radio Resource Management (RRM) parameters necessary to satisfy each guaranteed QoS profile. These parameters are the MCS and Channel Quality Indicator (CQI) tables, the target BLER, the scheduling type and policy, the maximum number of retransmissions for Hybrid Automatic Repeat Request (HARQ), the number of repetitions (k), and the maximum transmission power for power control. We should note that the information related to the PDU sessions established by the UE is also contained for all connected UEs in the Connectivity submodel of the 5G NW AAS. However, we believe that it must be also accessible through the 5G UE AAS so that industrial applications implemented in 5G-capable devices can access this data.

The submodel includes the possibility to implement two operations. The *NewConnectionRequest* operation can be used by a UE to request the establishment of a new PDU session. The request includes the QoS profile requested by the UE. The submodel also includes the option to implement a *ConnectionModificationRequest* operation. The UE can use this operation to request modifying an established connection or PDU session, particularly its QoS profile guaranteed.

- *QosMonitoring* submodel: This submodel contains information about the performance experienced by the UE. It includes information about the signal quality received by the UE, including the RSSI (Received Signal Strength Indicator), RSRP (Reference Signal Received Power), and the Reference Signal Received Quality (RSRQ). The submodel contains information related to the performance experienced over all the PDU sessions established. In particular, it contains the average data throughput, the percentage of dropped connections, the handover success rate, the Signal Interference Noise Ratio (SINR), and the average BLER and PER. The submodel also contains information about the performance experienced per PDU session and QoS flow (service availability, reliability, PER, service bit rate, BLER, data throughput and latency). The submodel also indicates the update time that defines how frequently the data in the submodel is updated. We should note that the performance information for all PDU sessions and QoS flows per UE is also accessible through the *QosPerformance* submodel of the 5G NW AAS.

  Following the exposure capabilities introduced in 5G, the submodel contains a list of requested subscriptions to QoS performance events requested to the 5G NW AAS, and a list of received events notified by the 5G NW AAS. This allows the possibility to continuously monitor if the QoS requirements of the UE are met, and to manage and optimize the connections (and if necessary, the network configuration) accordingly. The submodel also contains a list of subscriptions to QoS performance events requested to the 5G UE AAS by industrial applications implemented in the industrial device-side. For each subscription, the submodel contains a list of the events notified to the industrial application. The industrial applications might subscribe to QoS performance events in the UE to, for example, adapt the operation mode of the application to changes in the QoS experienced. The *QosMonitoring* submodel includes two operations. The SubscriptionRequest operation provides the option to request to the 5G NW AAS a new subscription to events related to changes in QoS performance of a PDU session or QoS Flow, on-demand, periodically or an event-triggered basis. For instance, it could be used to implement a monitoring function that provides a notification if the experienced latency exceeds a threshold or if a minimum service bit rate is not guaranteed. The *QoSMonitoring* submodel also includes the SubscriptionManagement operation to manage the subscription to QoS performance events requested by industrial applications implemented in the device.

- *Location* submodel: 5G networks increase the positioning or location accuracy and this can be leveraged by industrial applications. To this end, it is important that the location information generated by the network is transmitted to the UEs and is made accessible to 5G-capable industrial devices. In this context, a *Location* submodel has been defined in the 5G UE AAS that contains information about the current UE location and the required location service quality (indicated by the LCS QoS Class, accuracy and response time). It also provides a list of subscriptions to location information provided by the 5G network and a list of received events from the 5G NW AAS. In addition, the *Location* submodel also contains a list of subscriptions to location events requested by industrial applications implemented in the device. These subscriptions allow industrial applications to consider the location on their operation. For each subscription, the submodel includes a list of notified events. The submodel also contains a list of subscriptions requested.

  This submodel includes a SubscriptionRequest operation so that a UE can request a new subscription to a location event to the 5G NW AAS (for example, changes in the device connection or periodic location reports). It also includes a SubscriptionManagement operation to manage the subscriptions to location events requested by the industrial applications.

2.8.6 API

The designed 5G UE and NW digital twins are able to interface with other AASs, digital twins and industrial applications using OPC UA. They both correspond to type 2 or reactive AASs following the classification from Platform Industrie 4.0[5]. Type 2 re-active AASs utilize standardized interfaces (REST or OPC UA) to interact using client/server or publisher/subscriber models.

The 5G UE and NW digital twins incorporate the necessary Application Programming Interfaces (API) to enable the 5G digital model to interface with other digital models or AAS

[5] *Plattform Industrie 4.0, Details of the Asset Administration Shell - Part I, Specification, v3.0RC02. May 30, 2022.*

and industrial applications via OPC UA and exchange information seamlessly. This API uses OPC UA; OPC UA is one of the technologies recommended by Platform Industrie 4.0 to implement AAS interfaces. The API is implemented following a client-server communication mode where the 5G digital twin and industrial applications act as clients and send requests to the OPC UA server.Table 2Table 2 reports the set of operations implemented in the developed API for the 5G digital twin to interface with a OPC UA server. For example, the PUTSubmodelElement operation updates the submodel that is indicated as input when the operation is called.

Table 2 OPC UA API operations implemented in the 5G digital twin.

Method	Description
GetAssetAdministrationShell	Returns the entire AAS structure
DeleteAssetAdministrationShell	Deletes an AAS
GETAsset	Returns the asset associated to the AAS
GETSubmodel	Returns an entire submodel
GETSubmodelElement	Returns a submodel element
PUTSubmodelElement	Updates a submodel element
POSTSubmodelElement	Creates a new submodel element
DELETESubmodelElement	Deletes a submodel element

2.8.7 Implementation Technology

The 5G digital twin or AAS has been developed following the specifications provided by Platform Industry 4.0 for the development of AASs. The first version of the developed 5G digital twin was implemented using the AASX Package Explorer software. This software is developed by Platform Industry 4.0 and employs a graphical interface for creating, editing, and viewing AAS in adherence to the standardized specifications of Platform Industry 4.0. It primarily focuses on creating passive AAS with the correct structure in a static manner. The 5G digital twin created using AASX in aasx format is moved to Python using the open-source library Basyx Python SDK to provide it with active operations.

2.8.8 Comments

This component provides the general description of a 5G digital twin or AAS and describes the different submodels and how the data should be organized to make it accessible to other digital twins or applications. It is important to highlight that the 5G digital twin or AAS needs to be specifically configured based on the requirements of the current deployment scenario.

2.9 Realtime Digital Twin Model (RDTM)

2.9.1 Description

The Realtime Digital Twin Model is a comprehensive digital illustration of a real machine tool. It contains all the key aspects and components of a real-world machine tool, including the numeric controller, operator station, and various other subsystems. In comparison to a physical machine tool, the Realtime Digital Twin Model utilizes software and a screen to simulate the fabrication of a virtual part, rather than actually manufacturing a physical part.

This component requires a seamless connection between the IT (information technology) and OT (operational technology) architectures within the manufacturing environment. On the IT side, the part design is created using a CAD (computer-aided design) program. The work preparation process is then handled by a postprocessor, which generates the necessary numerical control code that can be directly interpreted and executed by the CNC (computer numerical control) controller.

The integration of the Realtime Digital Twin Model into the manufacturing workflow offers several key benefits. It allows for virtual testing and optimization of the part design and manufacturing process before committing to physical production. This can help identify and resolve potential issues early on, reducing costly errors and delays on the shop floor. Additionally, the real time digital twin model provides a realistic simulation environment for operator training, enabling personnel to become proficient with the machine tool controls and procedures without disrupting actual production.

2.9.2 Input

Post processor

The post-processor converts the generic NC (Numerical Control) code generated by NX CAM into the specific syntax and format required by the target CNC controller, such as Sinumerik. This ensures that the NC code can be correctly interpreted and executed by the machine tool's CNC controller.

Virtual machine

This component uses a simulation to demonstrate a virtual fabrication of the virtual part. To give the operator a more realistic understanding of the process, the real machine can be created as a digital twin and be integrated into the simulation.

2.9.3 Output

The computerized numerical control system runs the NC code, the main output is the control of the Realtime Digital Twin Model to perform the desired manufacturing operations. The CNC controller translates the NC code instructions into precise movements of the machine tool's axes (e.g., X, Y, Z, A, B, C). This includes controlling the speed, acceleration, and deceleration of the axes to execute the programmed tool paths. Moreover, The CNC controller continuously monitors the machine tool's performance, including axis positions, spindle speed, and other relevant parameters. It provides feedback and error messages to the operator, allowing for real-time monitoring and troubleshooting of the machining process.

2.9.4 Information Flow

Running the Realtime Digital Twin Model (RDTM)

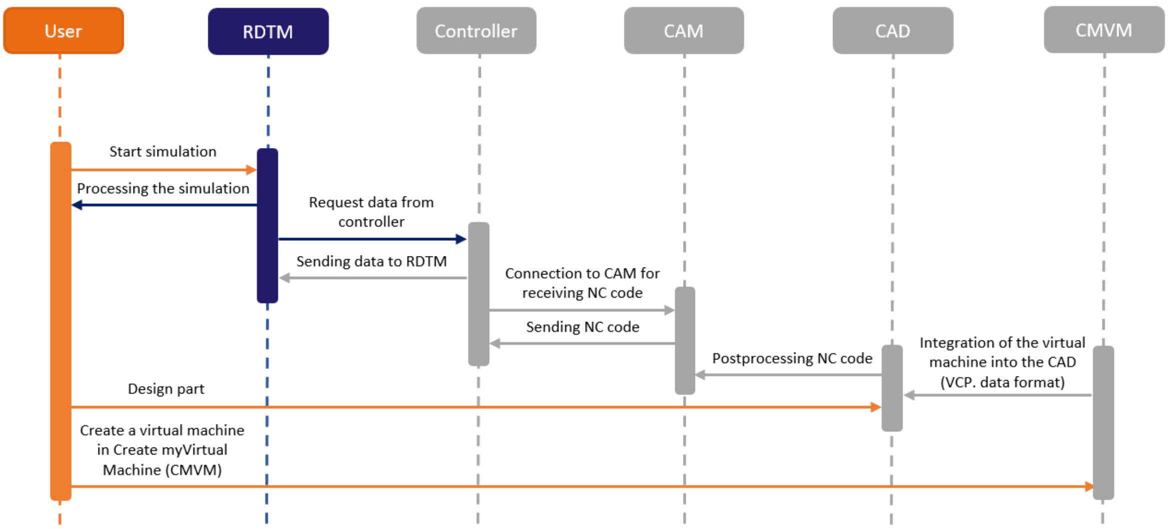


Figure 18: Running the Realtime Digital Twin Model

The user must initiate the simulation of the Real-time Digital Twin Model. After launching the simulation software, which is connected to the CNC controller, the NC code is executed. In the case of the Real-time Digital Twin Model, the controller integrated into the model is connected to a manufacturing station. The purpose of the manufacturing station is to create a virtual part using a CAD program. By employing Computer-Aided Manufacturing

(CAM) software that includes a post-processor, a numerical control (NC) code of the virtual part can be generated.

As long as the CNC controller is connected to the CAD/CAM software, the NC code can be transmitted from the IT level to the shop floor where the Real-time Digital Twin Model is situated. While the controller receives the NC code, it can commence transmitting the data to the Real-time Digital Twin Model.

2.9.5 Internal Architecture

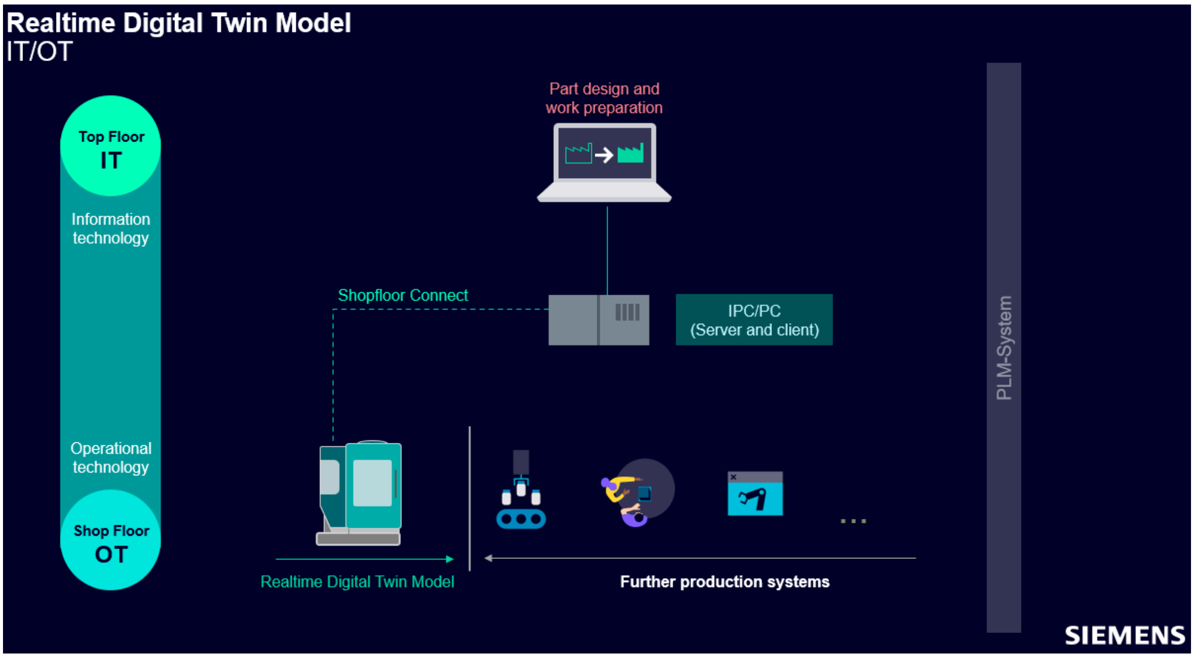


Figure 19: Internal architecture IT – OT

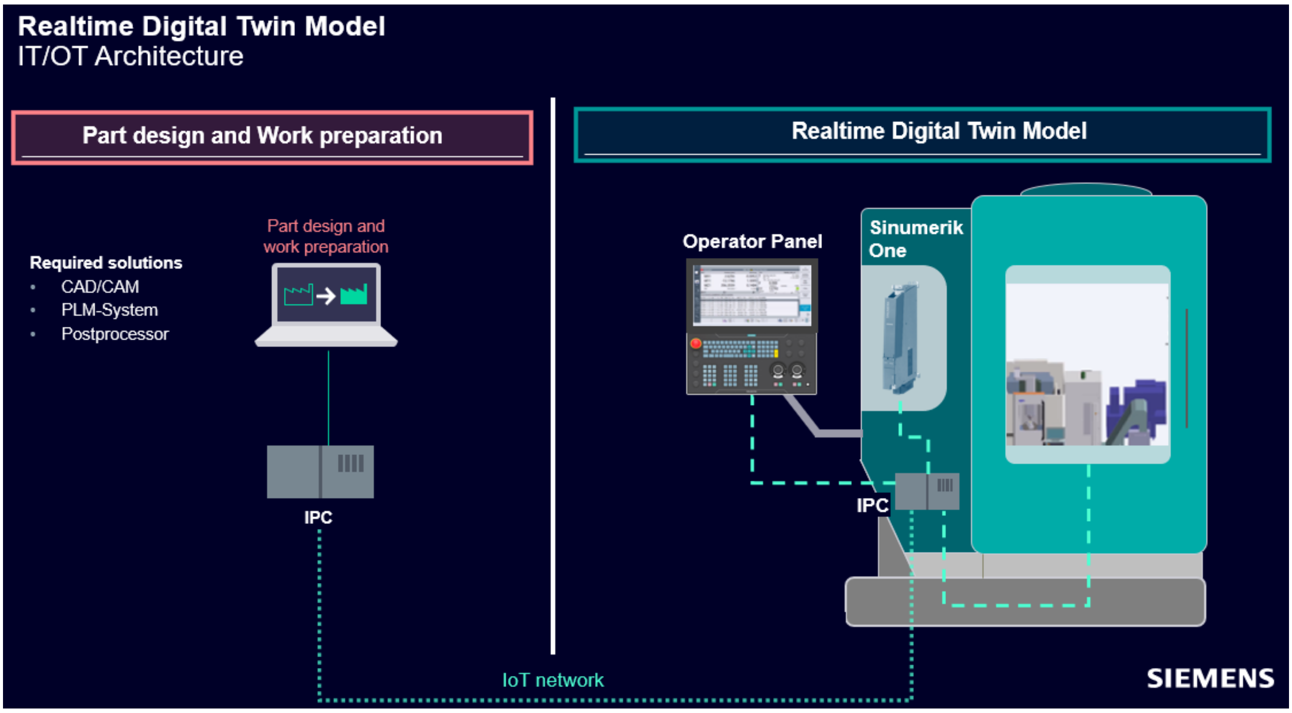


Figure 20: Internal network architecture

2.9.6 API

Open architecture:

The Sinumerik ONE controller offers an open architecture that allows integration with various software and hardware components. This open API enables users to develop custom applications and integrate the controller with third-party systems. The open architecture supports standard interfaces like OPC UA, PROFINET, and Ethernet/IP for seamless communication.

2.9.7 Implementation Technology (according to figure 3)

Sinumerik One

Sinumerik One is a CNC system for machine tool applications. This flexible, modular platform combines powerful real-time control, an intuitive touchscreen interface, and comprehensive software tools to help manufacturers boost productivity, quality, and flexibility across a wide range of applications, from milling and turning to additive manufacturing.

Operator Panel

An operator panel provides an interface for machine operators. With its large touchscreen display, the panel enables efficient control and monitoring of the CNC system. Operators can easily access critical production data, program machines, and make real-time adjustments to optimize performance and quality.

IPC

Industrial PCs (IPC) are rugged, specialized computers designed for use in harsh industrial environments. They are built to withstand vibrations, temperature extremes, dust, and other challenging conditions that standard commercial PCs cannot reliably operate in. IPCs are commonly used in manufacturing, process control, automation, transportation, and other industrial applications.

2.9.8 Comments

This component is part of TEF (SIPBB) setup and the experiment “Scalable industrial IoT Solution”.

2.10 ALIDA

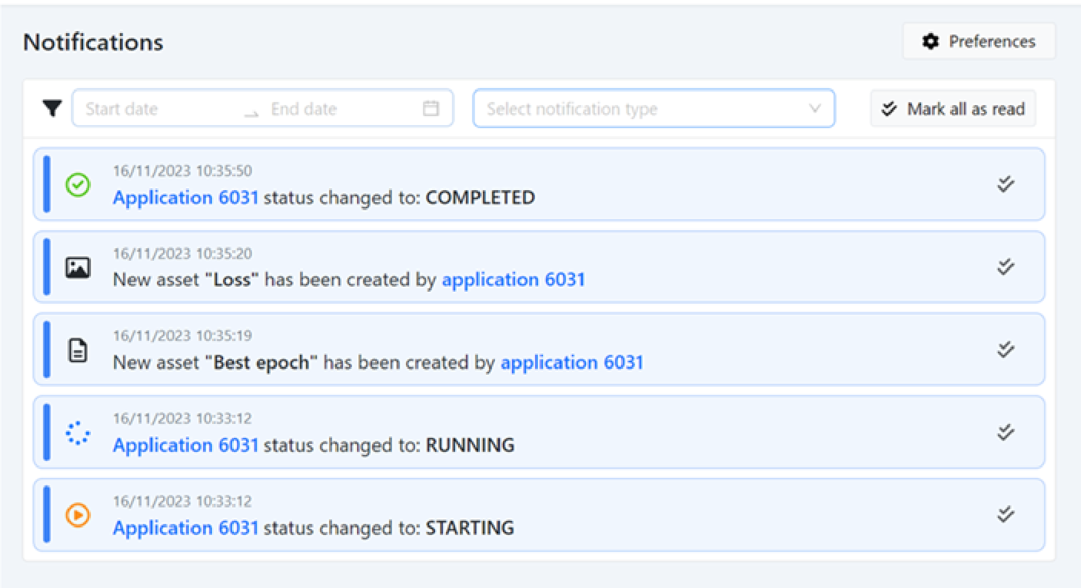
During the period covered by this report, ENG's ALIDA has undergone several updates aimed at increasing its robustness and performance to provide an enhanced user experience. The following sections outline them.

2.10.1 ALIDA Platform core

In this release, the integration of the Flower framework, central to the implementation of Federated Machine Learning in ALIDA, has been further extended to support secure TLS connections between ALIDA BDA Services implementing FML Aggregators and Participants. Consequently, a new sample template has been published on the ALIDA Template Repository and circulated to interested partners.

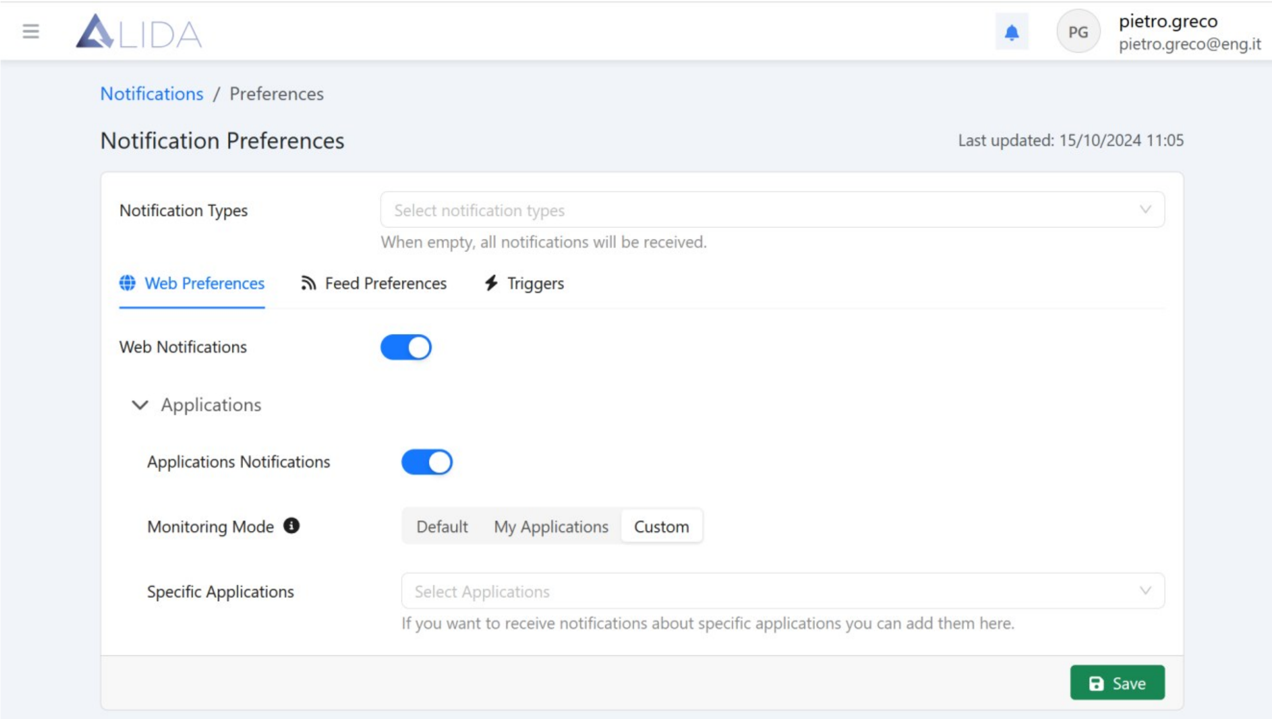
Another new feature of ALIDA is the Kafka-based Notification Manager. This allows data scientists to remain informed about the status of BDA Applications and resulting assets through notifications which can be transmitted through various channels.

The first channel is a dedicated panel in the UI, which allows users to view notifications and filter them by (i) a date range and (ii) type of notification.

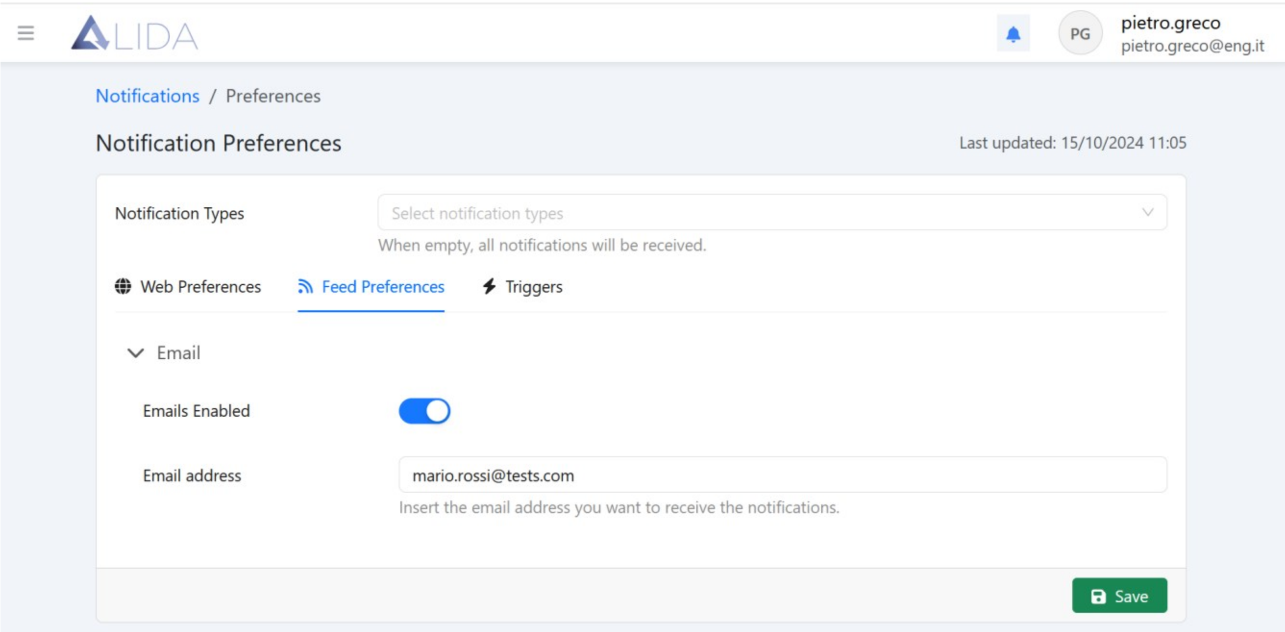


Users can configure UI notifications through the *Web Preferences* tab on the *Notification Preferences* page. There, among the other things, they can opt for one of the Monitoring Modes available:

- Default - Get notifications from applications you executed
- My Applications - Get notifications about the applications you created
- Custom - Get just the notifications from applications you specified



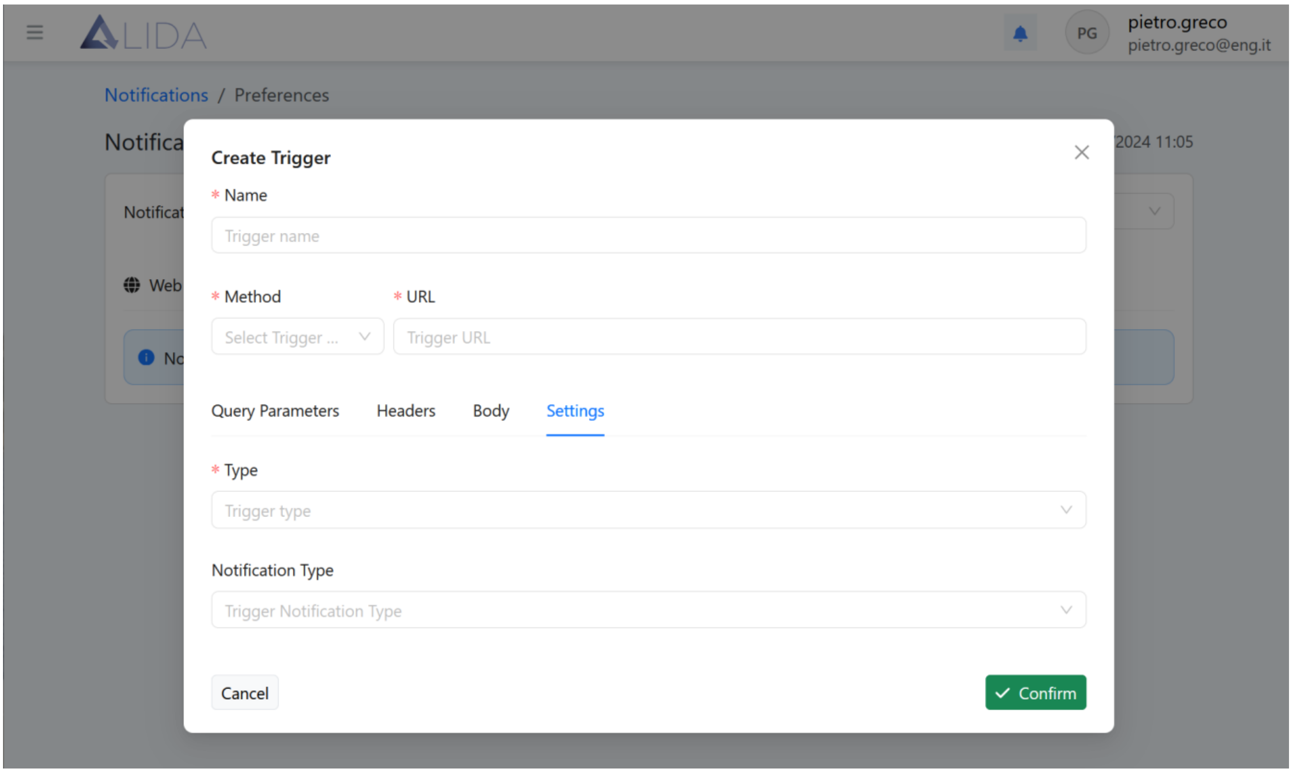
From the *Feed Preferences* tab, users can instead enable *email notifications.*



From the last tab, *Triggers*, users can instruct ALIDA to make a REST call to a specified endpoint. For each call, users can set:

- HTTP method
- Endpoint URL
- Query parameters
- Headers
- Payload body

and determine which kind of notifications are to be sent.



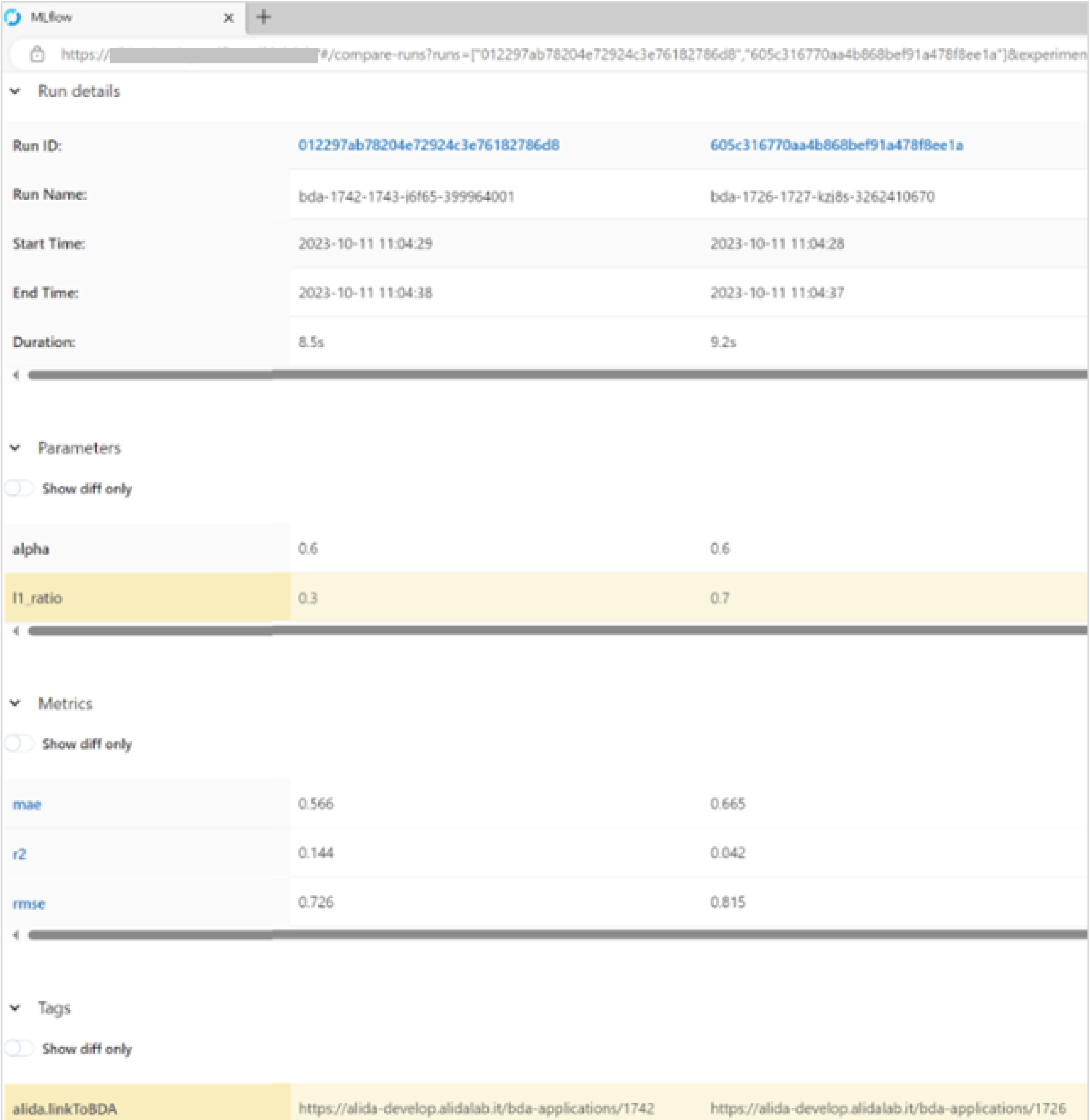
Another significant update to ALIDA concerns its *BDA Services serving* capabilities. Specifically, it is now possible to expose these services and control who has access to them through roles and groups defined on KeyCloak, and therefore based on user permissions.

Finally, the core of ALIDA has also seen several bug fixes which contribute to an even more reliable and effective user experience.

2.10.2 ALIDA Auxiliary Services

In addition to updates to the core of ALIDA, the platform also presents the possibility of activating auxiliary services such as MLFlow and Seldon.

MLFlow - a widely-used open-source platform for MLOps – is employed by ALIDA to provide model performance tracking capabilities. An MLFlow tracking server can be activated on the ALIDA cluster and the BDA Services instructed (through dedicated code) to transmit information about a running model behavior to it. Model metrics are then presented on a user interface.



Another auxiliary service, Seldon, can be activated to enable the serving of models. In this regard, work on the integration of the new version of the service - version 2 – into ALIDA has continued.

2.10.3 Integration with project's SSO

As already presented in previous WP3 deliverables, an instance of READY toolkit's KeyCloak (hereinafter KC) hosted by INTRASOFT is offered to the instances of other toolkit components to support the implementation of pilots and TEFs use cases and experiments. Together with the eIDAS node, it factually represents the project's identity and access management system.

Wanting to provide a unified registration and Single Sign-on (SSO) experience to the users of ALIDA, an integration between ALIDA's KeyCloak and the project's one has been set up by leveraging a mechanism called Identity Brokering. In such a scenario, the ALIDA's KC has been configured as an Identity Broker which demands the actual user authentication to the project's KC, configured to act as an Identity Provider. Doing so, users can register on the project's KC to create an SSO account and then use it to access ALIDA, as they would do with other platforms of the RE4DY toolkit who rely on INTRASOFT's instance of KeyCloak.

The resulting registration and login flow is now as follows:

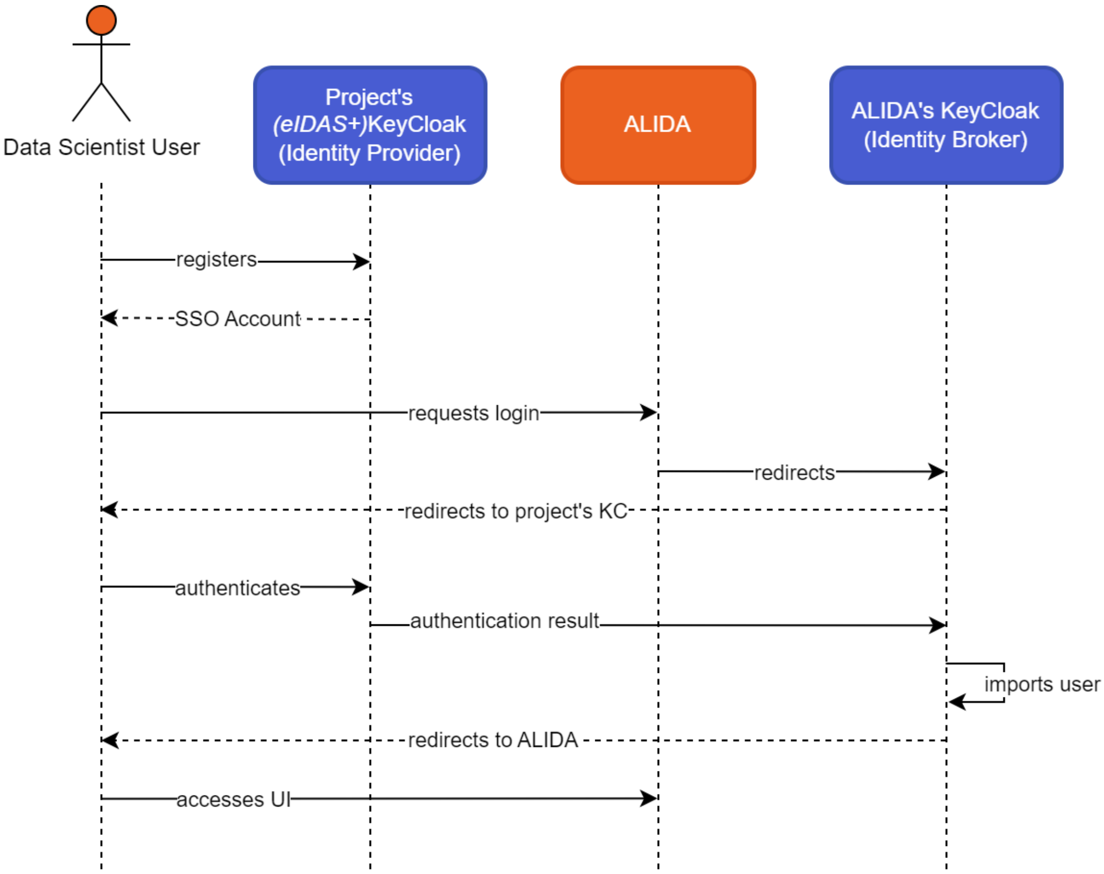


Figure 21: Sequence diagram illustrating the overall registration and login flow introduced by the latest updates

Once the unauthenticated users reach the ALIDA login page, they are presented with the possibility to login through the external KeyCloak (in this case the project's instance):

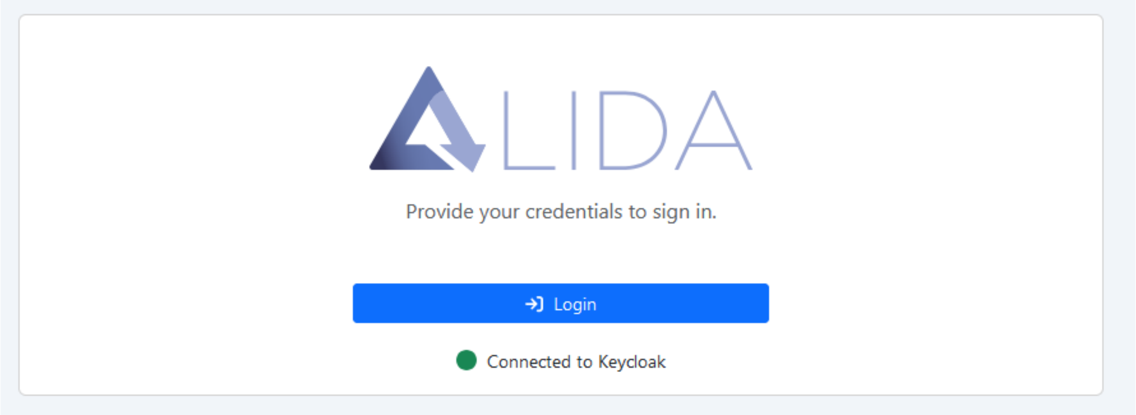


Figure 22: ALIDA default login page

The users are then redirected to the project's KeyCloak login page, where they can enter their SSO account credentials. In the Figure 23 notice the project's KeyCloak URL:

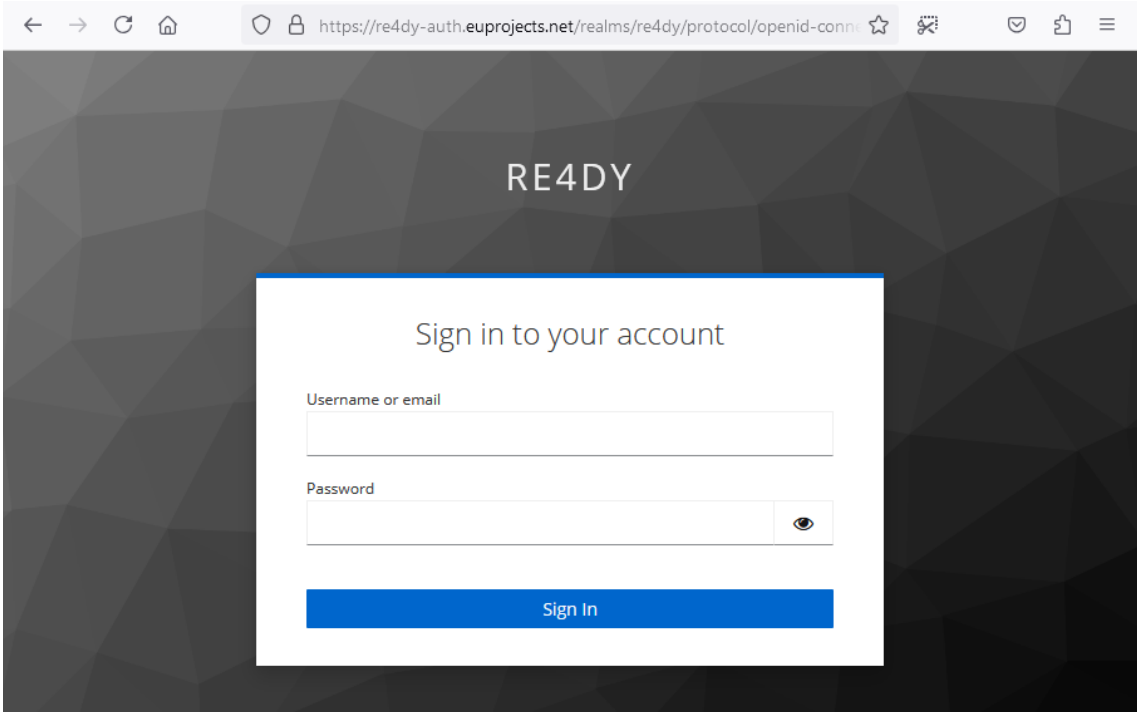


Figure 23: Project's KeyCloak login page asking for SSO credentials

The project's KeyCloak has also been configured to require an additional factor of authentication, a One-time Code:

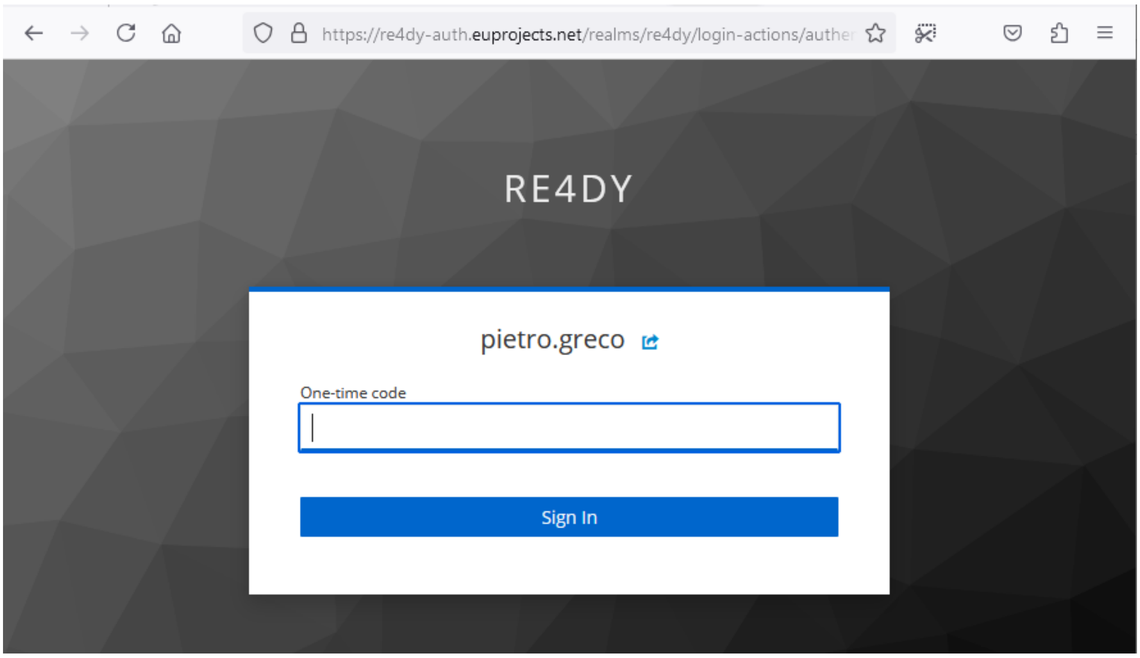
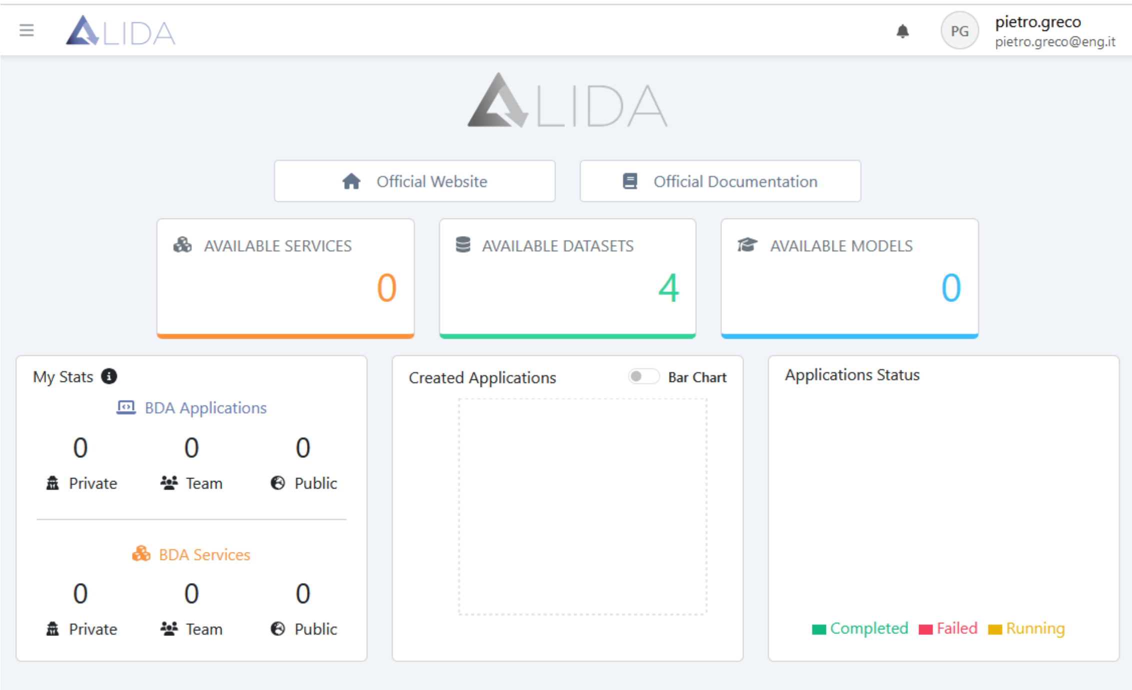
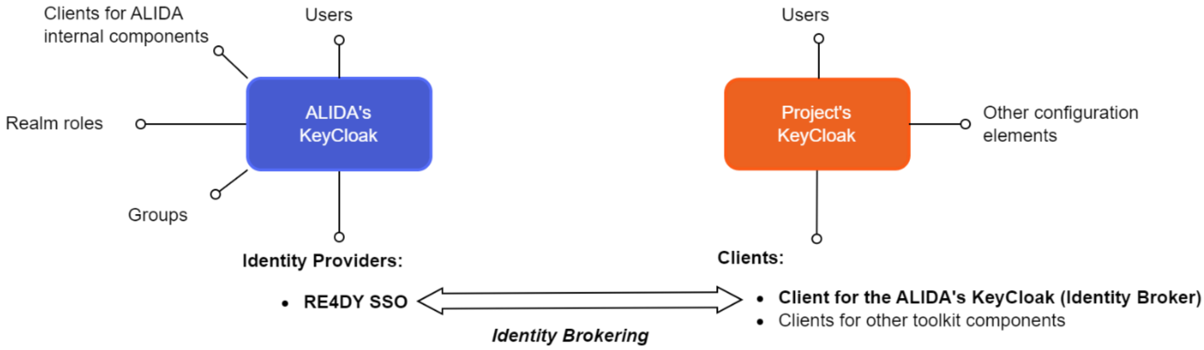


Figure 24: Project's KeyCloak requesting login One-time code

Upon successful authentication, the users are automatically redirected to the ALIDA home page for authenticated users:



From a technical standpoint, the use of Identity Brokering allows for a decoupled integration between ALIDA's KC – which has its own peculiar internal configuration (clients, groups, roles, etc.) – with the project's one. This decoupling not only simplifies the initial setup but also subsequent maintenance as it will be possible to update the ALIDA's KC configuration without affecting that of the project's instance.



2.10.4 Infrastructure-level Optimizations

Along with updates to the application-side of ALIDA, also the infrastructure supporting it has been revised. Among the most important changes, the use of K3S in lieu of K8S stands out. K3S is a lightweight alternative to K8S optimized for environment with limited resources and simpler use cases. K3S comes with a set of essential components and offers a simpler maintenance experience. All this while retaining full compatibility with the Kubernetes API.

Another important update to the ALIDA infrastructure lies in the use of GitLab CI and Argo CD for the continuous integration and delivery of ALIDA internal components. This allows for faster, safer and more robust development, testing and deployment phases; all elements that contribute to increased product quality. In this release, CI/CD is supported by a new set of end-to-end test cases automated with Selenium.

2.10.5 Comments

None

2.11 CERTH Sovereign Data Transformation Service

2.11.1 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[6], a recognized

[6] https://nifi.apache.org/

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[7] and provides all the necessary capabilities for setting rules regarding data access and sharing.

2.11.2 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[8] 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.

[7] https://internationaldataspaces.org/publications/ids-ram/
[8] 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.

2.11.3 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.

2.11.4 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.

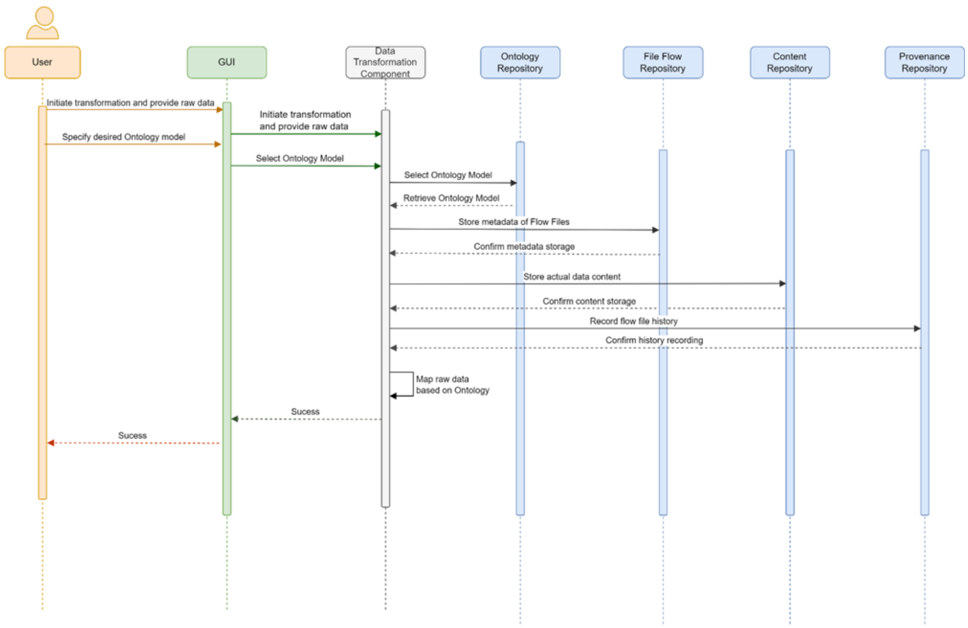


Figure 25: Data flow regarding ontological mapping of raw data

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.

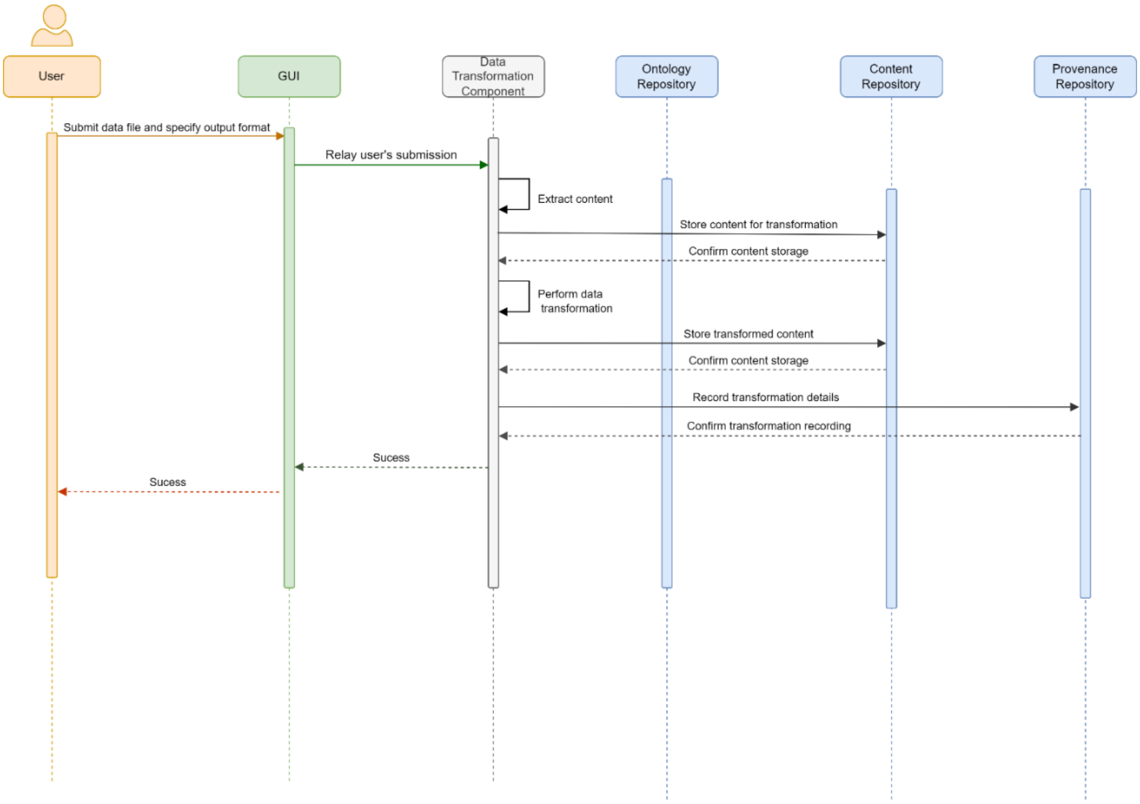


Figure 26: Data flow regarding data transformation example

2.11.5 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[9]: They are enabling trusted and sovereign communication between two parties that are involved in a data transformation service. A data consumer that needs a data source with a specific data format will

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

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

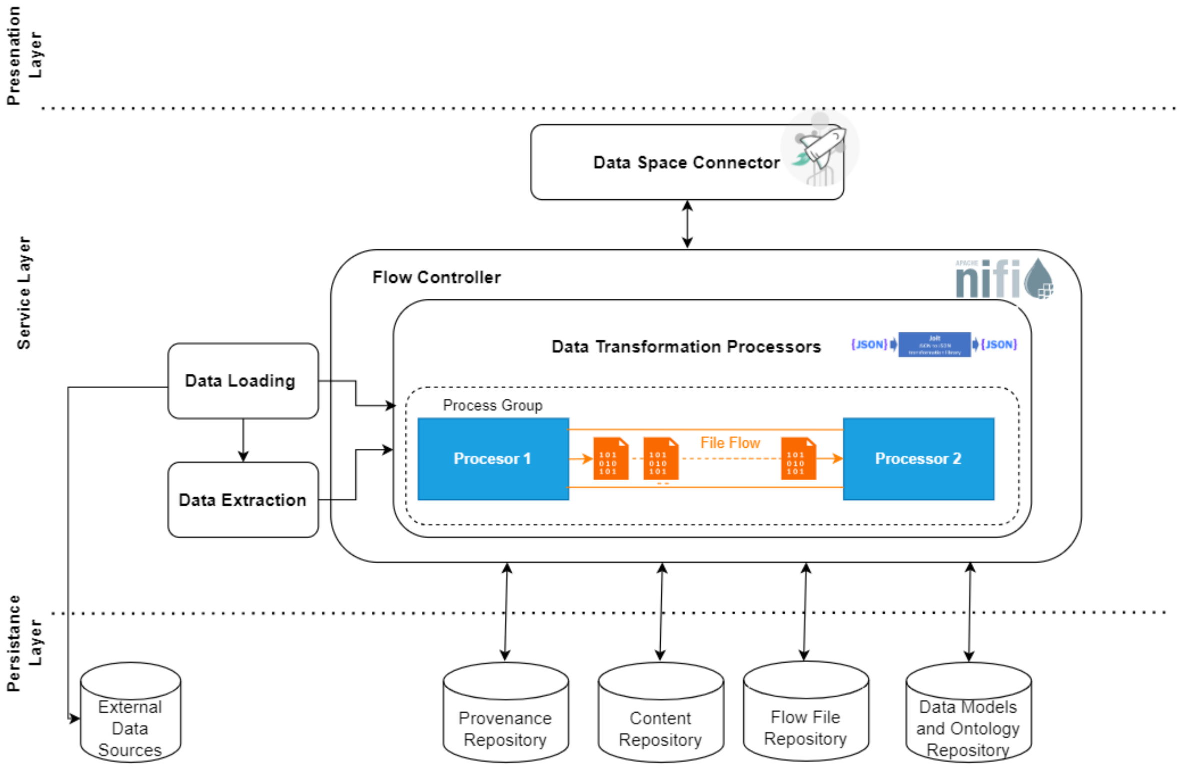


Figure 27: Architecture of RE4DY Data Transformation Services

2.11.6 API

Sent Data for Transformation

Using GET request (works for .json and .scv files):

Using POST request (works only for .json files):

Name	Sent Data (raw JSON)
URL	http://re4dy-project.iti.gr:8000
Method type	POST
Parameters	do not need to parse parameters
Body	The raw .json data
Description	Sent raw .json files with the data

Sent multiple files:

Name	Sent Data (multiple .csv files)
URL	http://re4dy-project.iti.gr:8000
Method type	GET
Parameters	Key = allcsv , Value = env if the file contains environment data or Value = energy if the file contains energy data
Body (form data)	Key = File , Value = <<file_name1>>.csv, <<file.name2>>.csv, ………
Description	Sent .scv files with the data

2.11.7 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,

Name	Sent Data (one file)
URL	http://re4dy-project.iti.gr:8000
Method type	GET
Parameters	<<file_name>>.csv or <<file.name>>.json
Description	Sent one .scv or .json file with the data

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. Regarding Data Space Connectors, EDC and Sovity implementations were used.

2.11.8 Comments

None

2.12 CERTH XAI and Active Learning Platform for Defect Detection

2.12.1 Description

The platform offers AI-driven defect detection by analyzing images of manufacturing assets, specifically tailored for the hard metal industry. The defect detection and localization platform is enhanced with AI explainability and human-AI collaborative features. Designed using a micro-service architecture, the platform is adaptable and extensible, catering to a diverse set of users. This includes data scientists who develop AI models and maintainers who monitor conditions with these pre-trained models. At its heart, the platform employs advanced Machine Learning and Deep Learning techniques, ensuring high precision in both defect recognition and localization.

The XAI and Active Learning Core Engine is a central component of the XAI and Active Learning platform, ensuring that key functionalities related to reasoning are seamlessly integrated and executed. It consists of three components briefly described below:

1. Reasoning/Recommender Module:
    - Purpose: This module stands as the primary component responsible for offering the inference and reasoning services of the platform.
    - Technical Approach: To cater to a diverse set of use-cases and ensure robustness in its inference capabilities, this module incorporates a multitude of techniques. Notably, it employs rule-based and graph-based reasoning methodologies. Additionally, fuzzy logic is integrated, enhancing the flexibility and adaptability of the reasoning processes. This ensures that the module can cater to both well-defined and ambiguous data scenarios, thereby broadening its applicability.
2. Data Management Module:
    - Purpose: At its core, this module's responsibility is to orchestrate the collection and subsequent management of data.
    - Data Sources: Data ingested by this module can stem from a variety of origins. One significant source is the simulated data generated by the Digital Twin. This serves to provide a virtual representation of the data scenarios. Additionally, this module integrates historical data and real-time data garnered from other WP3 components, ensuring a holistic data perspective.

3. Model Management Module:

   - Purpose: This module is dedicated to the meticulous management of AI models. Given the dynamic nature of artificial intelligence and the continuous evolution of models, a dedicated module for this purpose ensures that the most optimized and updated models are always in operation.

   - Functionality: Beyond mere storage, this module facilitates version control, monitoring, and updates for AI models, ensuring that they remain relevant and accurate over time.

4. XAI Module:

   - Purpose: This module is dedicated to interpret AI results and decisions.

   - Functionality: Visual representation (graphs etc.) regarding XAI is provided. For example graphs explaining the importance of various features during training phase.

2.12.2 Input

Main inputs for the XAI and Active Learning Core Engine are:

- Manufacturing Asset Images that the system analyzes. Given the specificity for the hard metal industry, these images might reveal intricate details, patterns, and potential anomalies or defects.
- Model Parameters and Configurations that guide how the image analysis and defect detection models should operate. For example, training configurations or specific thresholds for defect identification.
- Feedback Data for Active Learning collected during defects identification and validation (i.e., either confirmed or corrected defects). This feedback can be looped back into the system to refine and improve the AI models continually.
- Operational and Historical Data: Information about the manufacturing processes, historical defect rates, types of defects commonly encountered, and other related data can be valuable inputs. This data can provide context, aiding in the interpretation of results and in making informed recommendations.

2.12.3 Output

Main outputs of the XAI and Active Learning Core Engine are:

- Defect Detection Reports/Graphs about the type, size, location, and severity of each defect identified in the analyzed images.

- Explanations and Justification: For each defect detected, the system offers a human-readable explanation using Explainable AI (XAI) techniques, supplemented by visual aids where appropriate.
- Recommendations: Based on the identified defects, the engine generates actionable insights or recommendations, suggesting maintenance, repair, or additional inspections, and may provide predictive insights about potential future issues.
- Active Learning Feedback Loops: Leveraging active learning, the engine flags areas of uncertainty in its analyses and seeks human verification, while also reflecting improvements made from previous feedback.
- Data Summaries: Users are presented with summaries that provide overviews of the collected data, displaying trends in defect types and frequencies, as well as other relevant data usage statistics.
- Model Management Summaries: An overview of all AI models deployed, detailing their versions, update histories, and training data.
- Alerts and Notifications: Users are kept informed through real-time alerts and notifications about critical defects, anomalies, system health, and data collection status.

2.12.4 Information Flow

AI-Driven Defect Detection

The use case "AI-Driven Defect Detection" outlines the process through which a User (Data Scientist, Maintainer) interacts with the components of the "XAI and Active Learning Platform" (AI System) to analyze manufacturing asset images for potential defects. The procedure encompasses the provision of necessary input data (External Sources), processing of images using pre-trained AI models, generation of detailed defect detection reports, and the retrieval of these results by the User. Optionally, the User can contribute feedback for active learning to continually refine the AI models, enhancing the accuracy of defect detection over time. Through this structured interaction, the User can initiate a defect detection process, review the findings, and optionally participate in a feedback loop to improve future defect detection accuracy, thereby leveraging AI capabilities to maintain high standards of quality control in manufacturing assets.

Primary Actor: User (Data Scientist, Maintainer)

Secondary Actors: External Data Sources, AI System

Preconditions:

- The XAI and Active Learning Platform (AI System) is properly configured and operational.

- Relevant AI models for defect detection are available and properly trained.
- Manufacturing asset images along with any necessary model parameters and configurations are accessible.

Main flow:

- Input Data Provision - The User or External Data Sources provide manufacturing asset images, model parameters, and configurations to the AI System.
- Image Processing - The AI System processes the provided images using pre-trained machine learning and deep learning models to identify potential defects.
- Defect Detection Report Generation - Upon analyzing the images, the AI System generates defect detection reports. These reports detail the type, size, location, and severity of each detected defect.
- Results Retrieval - The User accesses the defect detection results, reports, and any associated data visualizations through the Graphical User Interface (GUI) or the REST API.

Optional steps:

- Active Learning Feedback - The User can provide feedback data for active learning, which is utilized to refine and improve the AI models continually, enhancing the accuracy of defect detection over time.

Exception paths:

- The User accesses the defect detection results, reports, and any associated data visualizations through the Graphical User Interface (GUI) (or a REST API).

Post-conditions:

- Defect detection reports are generated and accessible to the User.
- Active learning feedback loop may be initiated to enhance future defect detection accuracy.

Trigger:

The User initiates the defect detection process by providing the necessary input data or by instructing the system to utilize the data from External Data Sources.

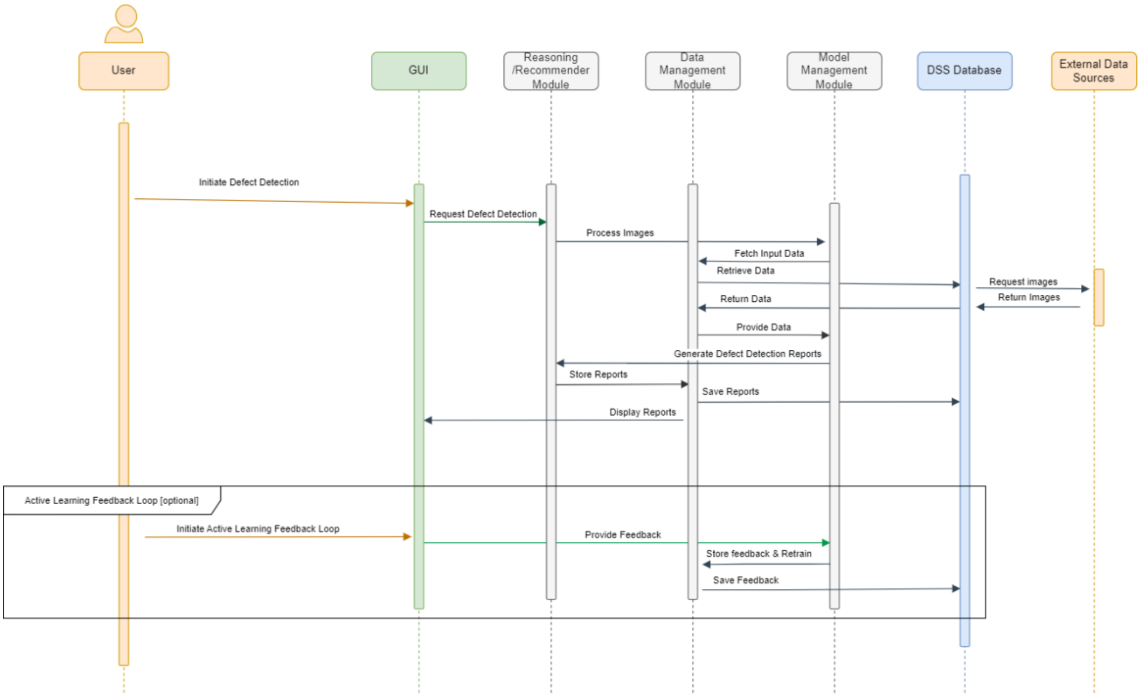


Figure 28: Data Flow of AI Driven Defect Detection

AI Explanation Retrieval

The "AI Explanation Request" use case facilitates a deeper understanding for the User (Data Scientist, Maintainer) by providing insights into the AI decisions and results, particularly pertaining to the identified defects in analyzed images, through the XAI (Explainable AI) Module of the system.

Primary Actor: User (Data Scientist, Maintainer)

Secondary Actors: AI System

Preconditions:

- AI System is properly configured and operational.
- Relevant AI models for defect explanation are available and properly trained.
- Images have been analyzed, and defects have already been identified by the AI System.
- The User has access to the system through the Graphical User Interface (GUI) or the REST API.

Main Flow:

- The User initiates a request for an explanation of the AI decisions and results through the GUI (or a REST API).

- The AI System processes the request through the XAI (Explainable AI) Module to generate explanations and visual representations (e.g., feature importance graphs) regarding the AI decisions.
- The explanations and visual representations are made accessible to the User through the GUI (or a REST API).

Post-conditions:

The User gains insights into the AI decisions through the explanations and visual representations provided by the XAI Module.

Trigger:

The User initiates the process by requesting an AI explanation.

Exception Paths:

In the event of any anomalies or issues during the AI explanation generation, an error notification is generated by the AI System and relayed to the User or System Administrator for resolution.

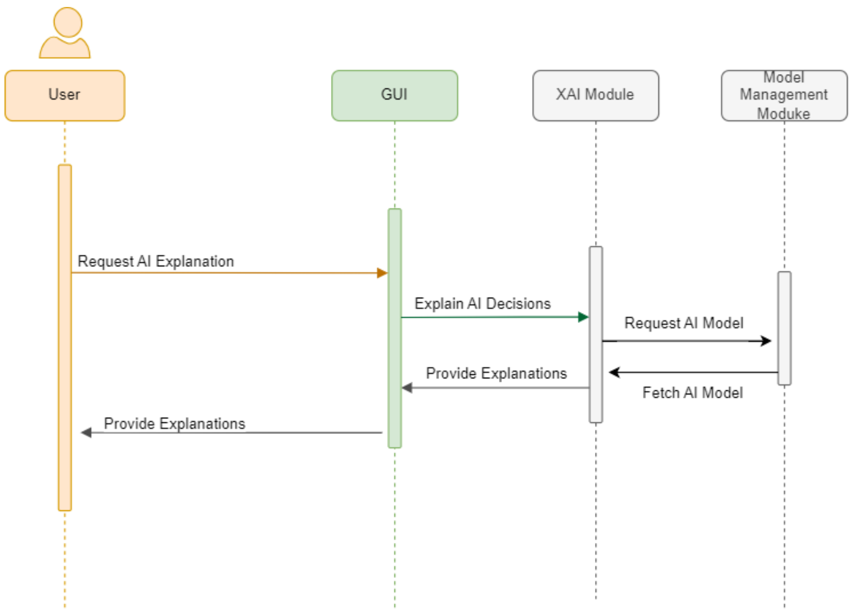


Figure 29: Data Flow of XAI

AI Model Management

The use case "Model Management", delineates the interaction between a User (Data Scientist, Maintainer) and the AI System for effectively managing the available AI models dedicated to defect detection. Through the Model Management Module, accessed via the

Graphical User Interface (GUI), the User engages in various model management tasks such as version control, monitoring, and updating AI models based on evolving requirements, feedback, or new data. These actions are facilitated by the AI System to ensure proper storage, tracking, and updating of the models as per the User's actions, enabling a structured approach to managing the AI models to ensure their optimum performance and relevancy over time. Through this defined flow, any encountered anomalies or issues trigger error notifications, ensuring the User or System Administrator is informed for prompt resolution, contributing to the effective management and utilization of AI models for defect detection.

Primary Actor: User (Data Scientist, Maintainer, System Administrator)

Secondary Actor: AI System

Preconditions:

- The AI System is properly configured and operational.
- Relevant AI models for defect detection are available and properly trained.
- The User has access to the system through the Graphical User Interface (GUI).

Main Flow:

- The User accesses the Model Management Module through the GUI to view and manage the available AI models.
- The User can perform various model management tasks including:
    - Version control to track and manage different versions of AI models.
    - Monitoring to oversee the performance and usage of AI models.
    - Updating AI models based on evolving requirements, feedback, or new data.
- The AI System facilitates these model management tasks, ensuring the models are properly stored, tracked, and updated as per the User's actions.

Post-conditions:

AI models are managed effectively through the Model Management Module, with their versions controlled, performance monitored, and updates properly executed.

Trigger:

The User initiates the process by accessing the Model Management Module to manage the AI models.

Exception Paths:

In the event of any anomalies or issues during the model management phases, an error notification is generated by the AI System and relayed to the User or System Administrator for resolution.

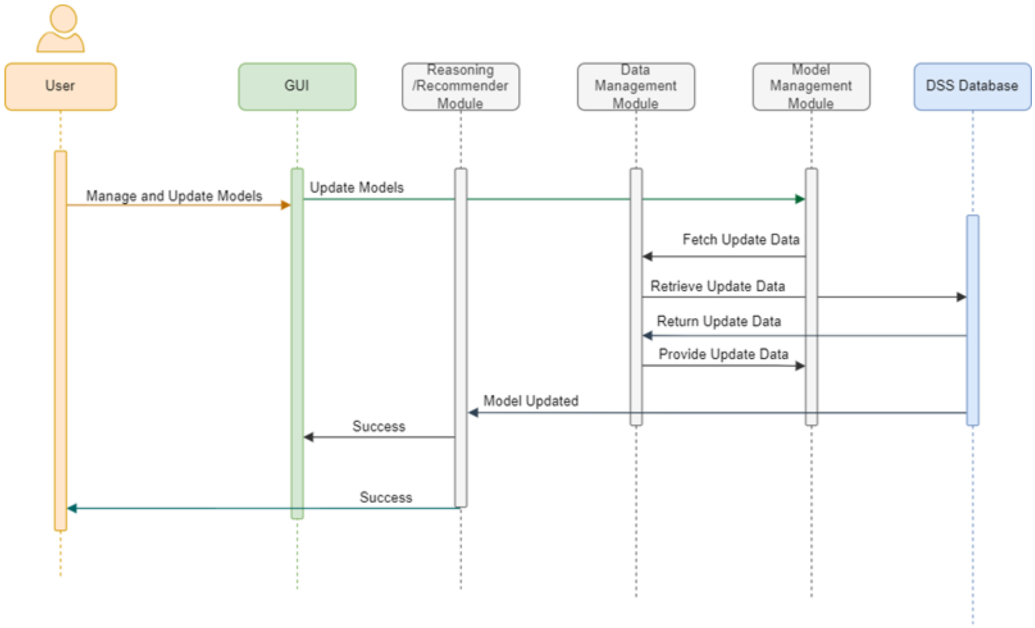


Figure 30: Data Flow for AI Models Management

2.12.5 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 XAI and Active Learning Platform for Defect Detection. This layered design also augments the platform's adaptability, ensuring that modifications or extensions in one layer have minimal ripple effects on the others. This is particularly advantageous in the microservices architecture employed by the platform, facilitating modular development and continuous enhancements.

Presentation Layer:

- REST API: Facilitates structured communication and data exchange between the client and server-side, generally employing JSON for data formatting.
- Graphical User Interface (GUI): Provides a user-friendly visual interface, enabling users to interact with the system and navigate through its functionalities effortlessly.

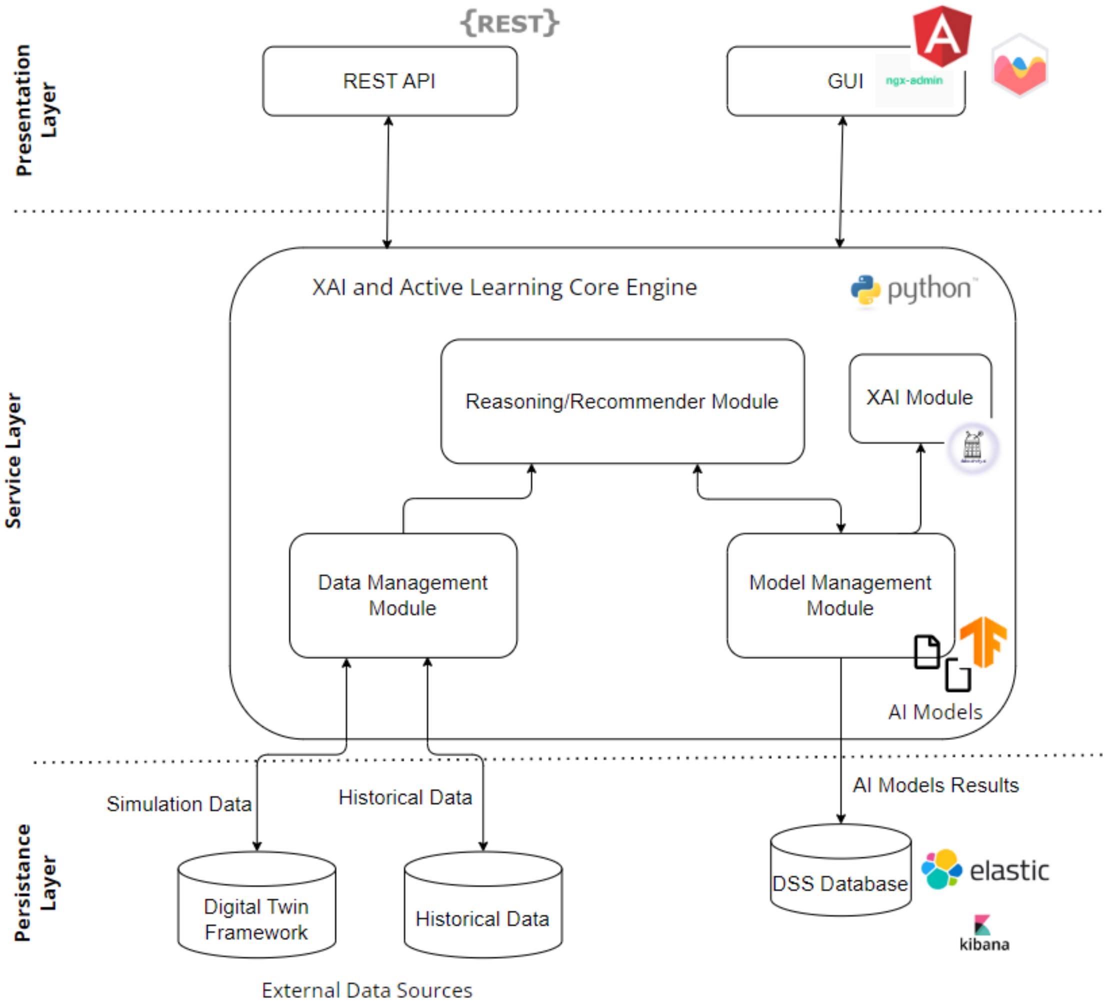
Service Layer:

- Reasoning/Recommender Module: Administers the inference, recommendation, and reasoning functionalities of the platform.
- Data Management Module: Orchestrates the collection, management, and integration of data from various sources, ensuring a well-rounded data perspective for the platform's analysis and decision-making processes.
- Model Management Module: Dedicates itself to the meticulous management of AI models, ensuring they are always optimized, updated, and operationally relevant.
- Explainable AI (XAI) Module: Concentrates on interpreting the AI's decisions and results, delivering visual representations and explanations that aid users in comprehending the AI's operations and outcomes.

Persistence Layer:

- DSS Database: Serves as the data storage hub where all pertinent data for the platform is securely stored and managed, ensuring data persistence and consistent availability over time.
- Historical Database: Houses the external data sources, preserving historical data, which can be leveraged for trend analysis, predictive maintenance, and other data-driven insights.
- Digital Twin Database (optional): Accommodates Simulation Data, providing a virtual representation of the manufacturing assets and processes. This database is crucial for testing, simulation, and analysis, enabling a better understanding and troubleshooting of real-world manufacturing scenarios.

The external data housed in the Historical Database and Digital Twin Database are crucial for the holistic data analysis and decision-making processes carried out by the platform. By segregating the Simulation Data and Historical Data, the architecture ensures organized, efficient, and accurate data management and retrieval, which in turn, supports the platform's AI-driven defect detection and reasoning capabilities. This addition emphasizes the storage of external data in dedicated databases within the Persistence Layer, explaining the significance of such segregation for the platform's operations.



2.12.6 API

Below we provide details in form of a table about the available REST endpoints to consume-load data:

Name	Explore models
URL	https:// re4dy-project.iti.gr/analytics/explore_models/<pilot_name>
Method type	GET
Body	-

Description	It returns to user a list with the names of the models and their description

Name	Explainability
URL	https:// re4dy-project.iti.gr/analytics/explainability /<pilot_name>
Method type	POST

Body	{
	 "index": index_id,
	 "method: method_id,
	 "model": model_id,
	 "startDate":time.
	 "endDate":time.
	 }

Description	Based on the data index, a time range, a model and a specific x-ai method the endpoint returns the result of the XAI algorithm.

Name	Explainability-Heatmap
URL	https:// re4dy-project.iti.gr/analytics/explainability_heatmap /<pilot_name>
Method type	POST
Body	{ "index": index_id, "method: method_id, "model": model_id, "is_heatmap": true or false}

Description	Based on the data index, a model, a specific x-ai method the endpoint and whether we want to return a heatmap the endpoint returns the heatmap of the most valuable pixels to the user.

2.12.7 Implementation Technology

The component was developed using Python. TensorFlow and Keras libraries were used alongside architectures like ResNet, Deep CNN, XG-Boost, Random Forest, Yolov3, Yolov5 regarding Machine Learning. It supports HTTP communication and JSON format for input/output. Regarding XAI, DALEX package is used. The front-end is developed on Angular TypeScript and ngx-admin framework. Libraries such as Chart.js were used.

Some of the implementation decisions:

- Deployment: The platform will be offered as a Docker Container. The only prerequisite for deployment is having Docker installed, facilitating a smooth experience to run the containerized image. The platform for inference capabilities can be deployed even in an edge device. The full platform with training/re-training needs a machine with at least 6GB GPU, 32GB RAM.
- Data Storage and Management: To address the storage needs, an Elastic Search Instance will be set up. It will be deployed locally within a Docker container. The choice of Elastic Search will be driven by its capabilities to efficiently store, search, and analyze vast data volumes in near real-time. In tandem, a Kibana Instance will be employed to visualize the content of Elastic Search indexes, offering a graphical perspective.

2.12.8 Comments

None

2.13 Federated Predictive Maintenance (FPdM)

2.13.1 Description

The primary objective of the Federated Predictive Maintenance (FPdM) component is to deliver predictive maintenance services within a trusted federated framework. The core functionality introduced is the capacity to run complex machine learning algorithms without requiring the transfer of valuable data to centralized repositories. Instead, each model is trained locally on its available data, and the final model is constructed by aggregating the parameters from each individual sub model, ensuring that valuable data never leave their source.

The FPdM, as a whole component, it is currently targeted on the operations of CNC milling machines and introduces the following functionalities:

1. The development and implementation of machine learning models focused on predicting tool wear and estimating remaining tool lifetime.
2. The integration of these models into a federated learning framework that prioritizes data privacy.

2.13.2 Input

The component requires the following input data:

1. CSV Files: These files contain operational data collected from milling operations. Each CSV file should include sampled data throughout the entire milling process, covering parameters such as spindle coordinates, spindle speed, vibrations, loads, and more. Every line should contain the values of the provided parameters along with the specific timestamp. An example of a supported file structure is below:

Machining_Data_File_Structure.csv					
Time	*Feature1*	*Feature2*	*Feature3*	…	*FeatureN*
2024-11-11T11:11:11.1100000Z	*0.1*	*0.525*	*True*	…	*0.1231*

2. JSON Files: These files contain static data specific to each milling job, such as the job name, program name, and other metadata. For training purposes, it is essential that these files include metrics related to tool wear (e.g., Vb). Every JSON file should also specify details about the cutting strategy (axial/radial infeed depth), the material of the workpiece, the type of tool, and the unique tool identifier. An example of a supported file structure is below:

Milling_Job.json

```
{
        "id": "127fbaa4-e565-4d70-8388-c2457d652d77",

        "job_name": "Example Job Name",

        "machineId": "Example Machine ID",

        "edgeDeviceId": "Edge Device ID",

        "programName": "Executed Program Name",
```

"startTimeUtc": "2024-11-11T11:00:00.00Z",

"endTimeUtc": "2024-11-11T12:00:00.00Z",

"material": "Workpiece Material Name",

"tool type": "Spindle Tool Type",

"axialInfeedDepth": 4,

"toolWear": 0.1,

…

}

For each milling job, there may be multiple CSV files containing data from various operations within the job, along with a single JSON file that consolidates the job-specific information.

2.13.3 Output

The component's output includes predictions regarding the tool wear and the estimated remaining life of the utilized tool upon the completion of a milling job Figure 31.

2.13.4 Information Flow

The information flow of the FPdM component is depicted in the following.

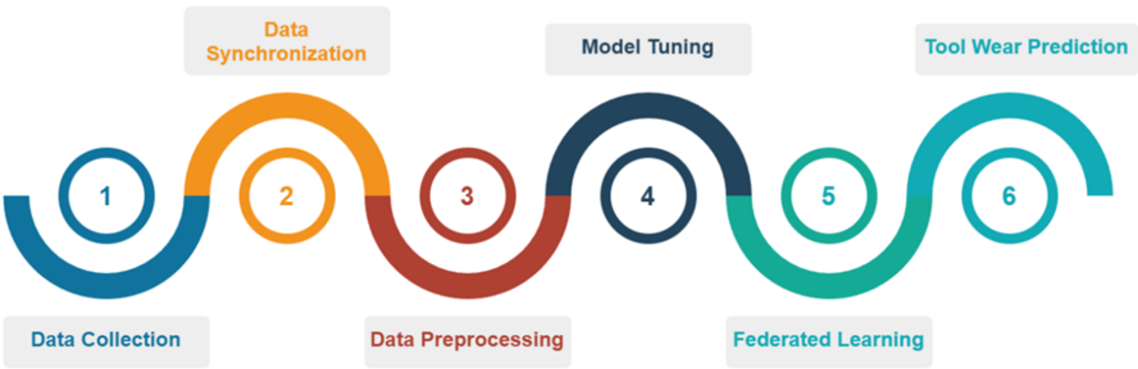


Figure 31: Information flow of the FPdM component

Each node of the flow is described below:

1. Data Collection: The process begins with gathering both dynamic sensor data and static information associated with multiple milling jobs. This includes

operational metrics such as spindle speed, vibrations, and tool usage, along with static, job-specific details.

2. Data Synchronization: The collected data is synchronized and aligned using the FPdM code, which combines CSV and JSON files for each corresponding job. This step ensures accurate association of operational data with the respective milling jobs.
3. Data Preprocessing: The FPdM system processes the synchronized data by extracting pertinent features, preparing it for use with machine learning models. This step ensures the data is structured and optimized for subsequent analysis by advanced AI models.
4. Initial Model Tuning: An initial machine learning model is developed, followed by hyperparameter tuning. This can be performed on a separate node using synthetic data to preserve privacy or on a single edge device utilizing its locally available data.
5. Federated Model Deployment: This critical phase involves deploying the initial model across all participating edge devices within a federated learning framework. Each edge device (client) trains the model locally, and the aggregated model parameters are updated without transferring raw data between devices.
6. Tool Wear Predictions: The final model parameters are distributed to every participating edge device. These devices use the federated model to predict tool wear upon completing their respective milling jobs.

2.13.5 Internal Architecture

A schematic representation of the component architecture is shown in the below.

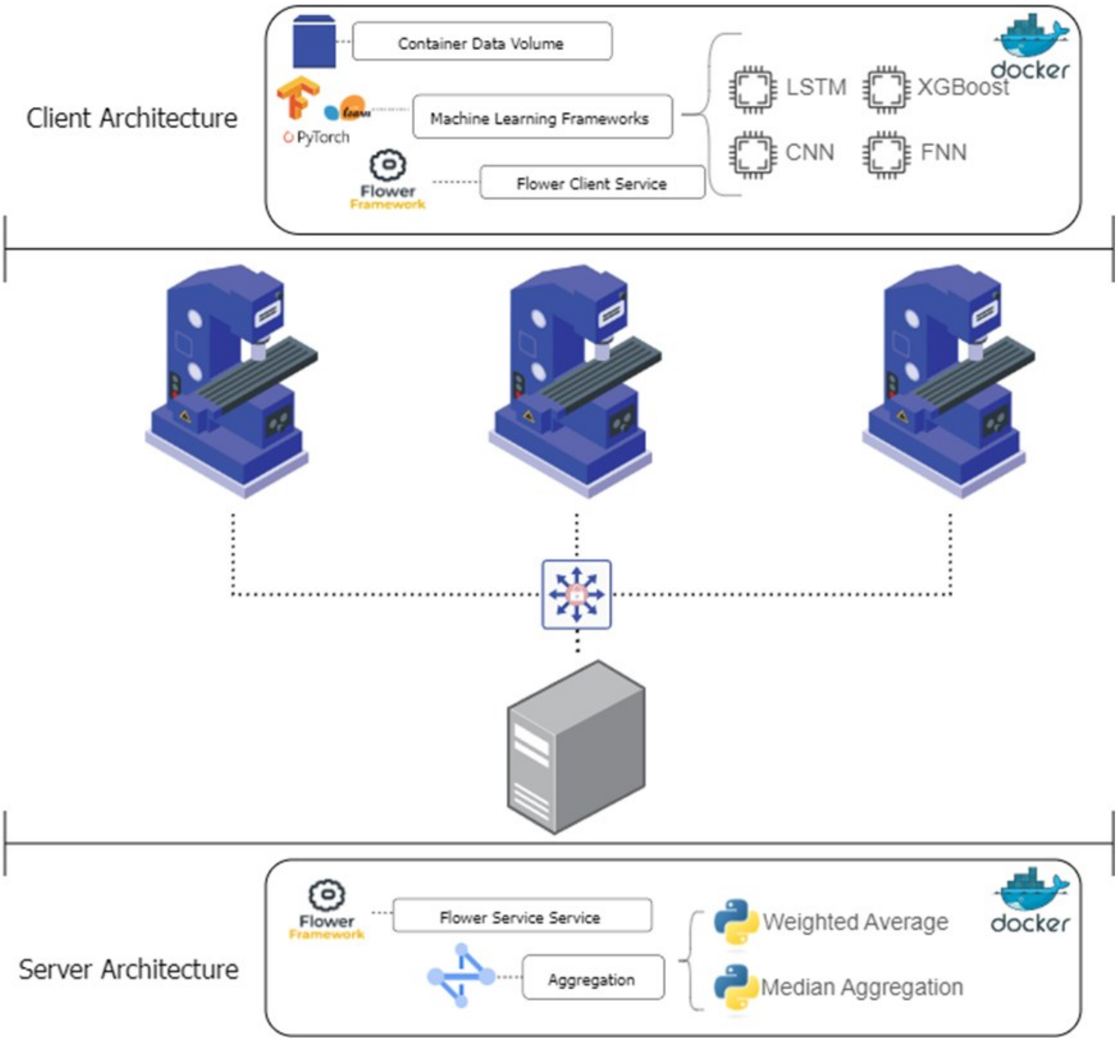


Figure 32: Architecture of the FPdM component

The overall architecture consists of two main subcomponents: the federated client and the federated server. Each subcomponent is built using specific Docker images, as detailed below.

- Client Architecture
  - File Storage: Each federated client must include a file storage module to store files containing both operational and static milling data.
  - Machine Learning Frameworks: TensorFlow or other machine learning frameworks such as PyTorch or Scikit-learn can be used to train and deploy models directly on the client. The client architecture is designed to support multiple machine learning models with minimal configuration changes. During the training phase, each participating client utilizes data from its local file storage to train the model and update its weights and hyperparameters. Once the specified number of training rounds is completed, the final model is prepared to predict tool wear for future milling operations on the client.

- Flower Client Service: A Flower client instance runs on every edge device. This service manages metadata communication between the edge device and the server during the training. After each training round, the Flower client service sends the client-specific metadata to the server and receives updates related to the model parameters.

- Server Architecture
  - Flower Server Service: A Flower server instance runs on a centralized host that has internal communication with every participating client. This instance is responsible for initializing the federated learning process with the specified configurations and parameters. During the training process, the server collects model metadata from each client after every round and aggregates them according to a chosen strategy. The aggregated parameters are then sent back to the clients until the specified number of training rounds is completed. The aggregation method is configurable and can be easily adapted to suit the requirements of each machine learning task.

2.13.6 API

The information required for the model training as well as for the predictions, within every client's environment, is fetched with the APIs designed under the scope of the component.

2.13.7 Implementation Technology

The component is developed using Python programming language. This decision was driven by Python's robust support for various machine learning algorithms and its full compatibility with Flower, the federated learning framework utilized in this component. Python's flexibility and extensive libraries make it an ideal choice for this application. In addition to Python, Docker is utilized for the containerization of both client and server instances, ensuring seamless deployment and scalability across the edge devices on the CNC machines and the centralized server host environment.

2.13.8 Comments

None.

2.14 Analysis Center

2.14.1 Description

The primary goal of the Analysis Center component is to serve as a comprehensive suite integrating data analysis functionalities and AI-driven models. This suite is designed to offer algorithms capable of detecting or identifying quality defects, tailored to the available data. A key advantage of this component is its ability to significantly enhance predictive quality, enabling smoother machine operations, minimizing unexpected downtime, and delivering valuable operational insights. While the primary focus of the Analysis Center is the examination of EDM machines, its functionalities can be extended to other types of machinery with minimal adjustments.

2.14.2 Input

The component requires input files from EDM machines in the following format:

- CSV Files: Each file should represent data from a single EDM machine. The data should include features related to the sparks generated by the EDM machine during its operation, as these are critical for building and training the models. These spark-related features are expected to provide valuable insights for defect detection and operational analysis. The supported format of the CSV files is shown in the example below, but the component could be easily modified to support other formats as well:
    - *timestamp, feature, datatype, value, site, machine_id*

2.14.3 Output

The component is designed to perform data analysis tasks, generating actionable insights based on the provided data. These insights can be shared with other system components, which are responsible for visualizing the information and presenting it to the user in an intuitive manner. For example, models designed under the Analysis Center could be integrated with the ALIDA platform as BDA services.

2.14.4 Information Flow

The generic information flow of the Analysis Center component is displayed in Figure 33.

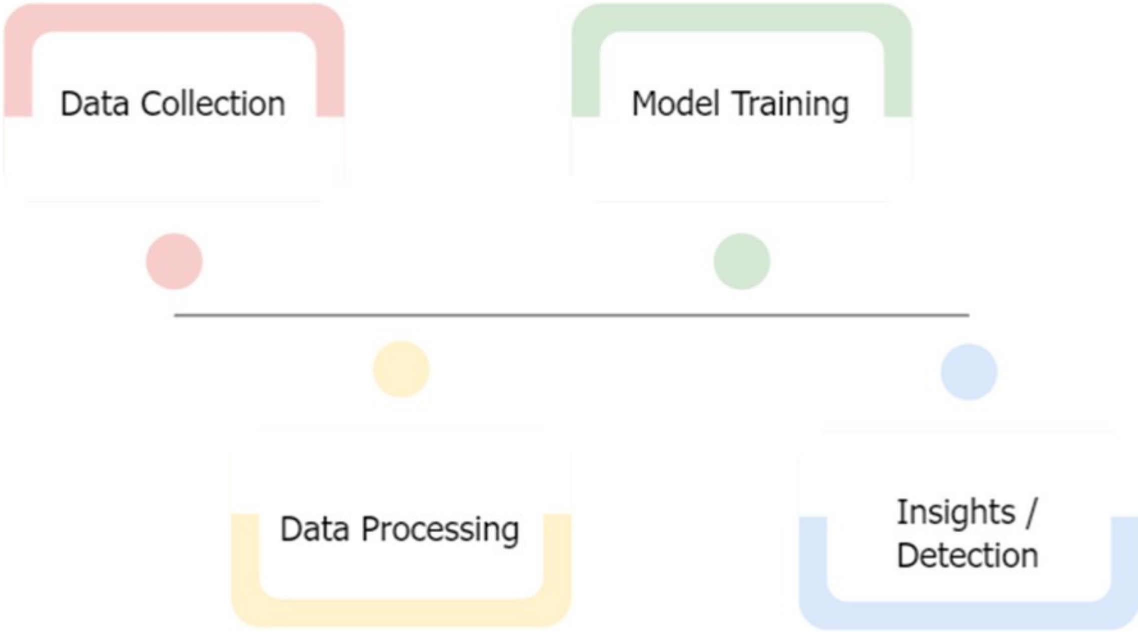


Figure 33: Analysis Center Information Flow

The distinct steps of the information flow are outlined as follows:

1. Data Collection: In this initial step, all the necessary data required by the models of the Analysis Center are gathered and transferred to the component for further processing.
2. Data Processing: This critical step involves preparing the collected data by performing transformations and preprocessing tasks, ensuring that the data is in a suitable format for use by the models.
3. Model Training: This step focuses on training the models using the processed data. The goal is to optimize the models to meet the predefined objectives, such as accurately detecting anomalies or identifying faults.
4. Insights/Detection: In the final step, the trained model is applied to the provided data to generate insights. These insights are aligned with the defined purposes, such as anomaly detection, fault identification, or other analysis tasks.

2.14.5 Internal Architecture

The architecture of the Analysis Center is depicted in Figure 34:

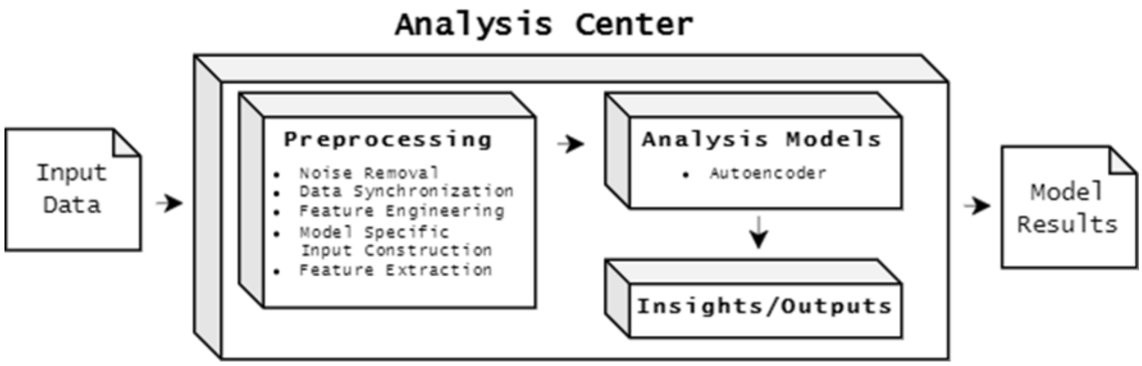


Figure 34: Analysis Center Architecture

The component is comprising three primary subcomponents, as detailed below:

1. Preprocessing: This subcomponent is responsible for transforming unstructured input data into a structured format suitable for use by the models. In addition to formatting, it handles crucial tasks such as data cleansing, noise removal, feature engineering, and any model-specific modifications required to optimize performance.
2. Analysis Models: This subcomponent encompasses the suite of models developed within the Analysis Center. These models are designed to analyze the processed data and address specific analytical objectives, such as defect detection or anomaly identification.
3. Insights/Outputs: This subcomponent takes the outputs generated by the analysis models and converts them into actionable, problem-specific insights. As the final output of the Analysis Center, these insights can be seamlessly integrated with other services or communicated directly to end users, enhancing decision-making processes.

2.14.6 API

The input data for the component, as well as the outputs generated by the models, should be exchanged with other components. This exchange could either be implemented via component dedicated APIs or by directly integrating the Analysis Center to the required component.

2.14.7 Implementation Technology

The component is developed entirely in Python, chosen for its extensive ecosystem of data analytics libraries, which significantly facilitate the development of algorithms within the Analysis Center. Additionally, Python's versatility and robust support for integration make it an ideal choice for seamlessly connecting this component with other system components.

3 Mapping with RE4DY Architecture and Digital Journey

The RE4DY toolkit helps manufacturing enterprises assess their current level of digital maturity, set goals for transformation, and create a roadmap for implementing Industry 4.0 capabilities. These include better data management, real-time communication across processes, and predictive analytics to enhance operational efficiency.

The toolkit includes several advanced features aimed at enhancing manufacturing processes and integrating digital capabilities, which can be breakdown into tools categories brought by RE4DY partners (see Figure 35):

- Static and Dynamic Executable Cognitive Digital Twin (xCDT) Commissioning Tools:

  These tools are used to create and manage cognitive digital twins, which represent real-world processes or systems in a digital form. They can operate in both static (predictive modelling) and dynamic (real-time adaptation) modes, enabling a virtual environment that mirrors the physical world. This supports testing, simulation, and optimization of processes before implementation on the factory floor.

- Trusted Industrial AI Federated Data Continuous Delivery Tools (Data Marketplace):

  These tools facilitate a federated approach to managing industrial data, allowing enterprises to share data securely across a distributed network. The continuous delivery aspect ensures that AI models and data are kept up-to-date, improving real-time insights and decision-making. The data marketplace enables organizations to share and access valuable datasets while maintaining data privacy and compliance standards.

- Knowledge Quality Tools, Reusable Models, Common Semantics, and Vocabularies 4.0:

  This category of tools focuses on standardizing and enhancing the quality of knowledge and data models used in Industry 4.0 systems. It emphasizes the use of reusable models, consistent terminologies, and shared vocabularies, facilitating interoperability between different digital tools and platforms. This standardization ensures that data and processes are understandable and consistent across the organization.

- Distributed Data Fabric FAIRNESS, Quality Sovereignty, Security, and Compliance Tools:

  These tools ensure that the distributed data across the enterprise adheres to FAIR principles (Findable, Accessible, Interoperable, Reusable). They also emphasize data sovereignty—ensuring that data remains under the control of its owner—and compliance with regulations like GDPR. Security measures are implemented to protect sensitive information across the data lifecycle.

- Optimized Distributed Data Edge-Fog-Cloud Computation and Networking:

  This category involves the strategic distribution of computational tasks across edge, fog, and cloud environments. It optimizes data processing by determining which tasks are best handled close to the data source (edge), at an intermediate layer (fog), or in a centralized data center (cloud). This allows for faster data processing, reduced latency, and better scalability, serving to the needs of smart factories.

Figure 35: RE4DY Digital 4.0 Fabric Management Toolkit (© RE4DY Consortium)

In terms of alignment with WP2 Reference Architecture, the following diagram (Figure 36) illustrates how its building blocks are mapped with toolkit components as well as with the five Digital Continuity domains (Digital Twin Continuity, Industrial AI/ML Continuity, Digital Thread Continuity and Computing Continuity) (Figure 37).

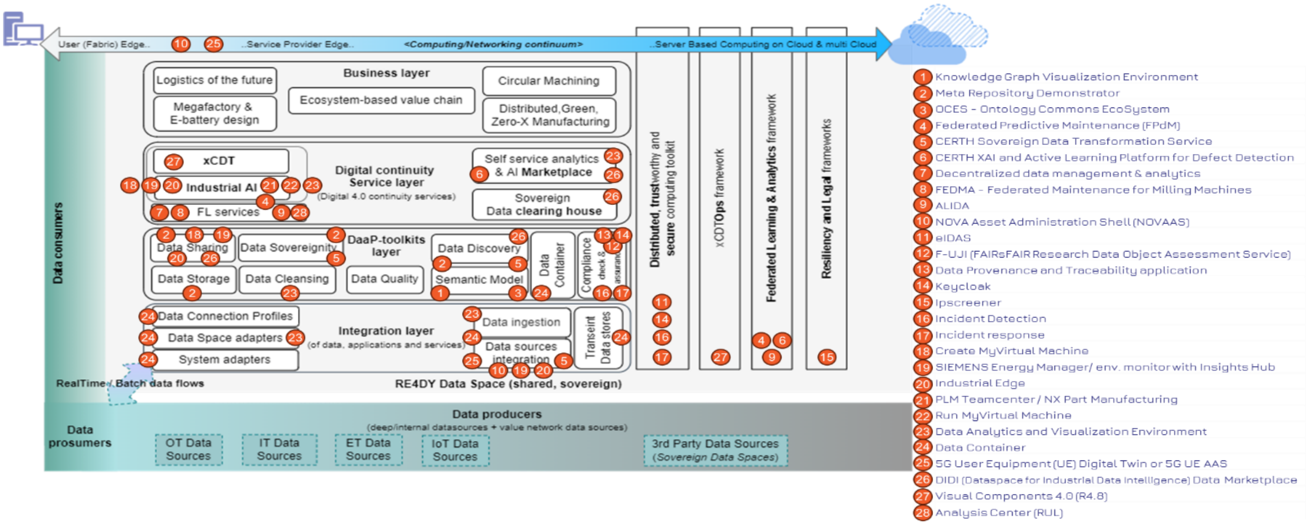


Figure 36: Mapping Toolkit components with Reference Architecture building blocks (© RE4DY Consortium)

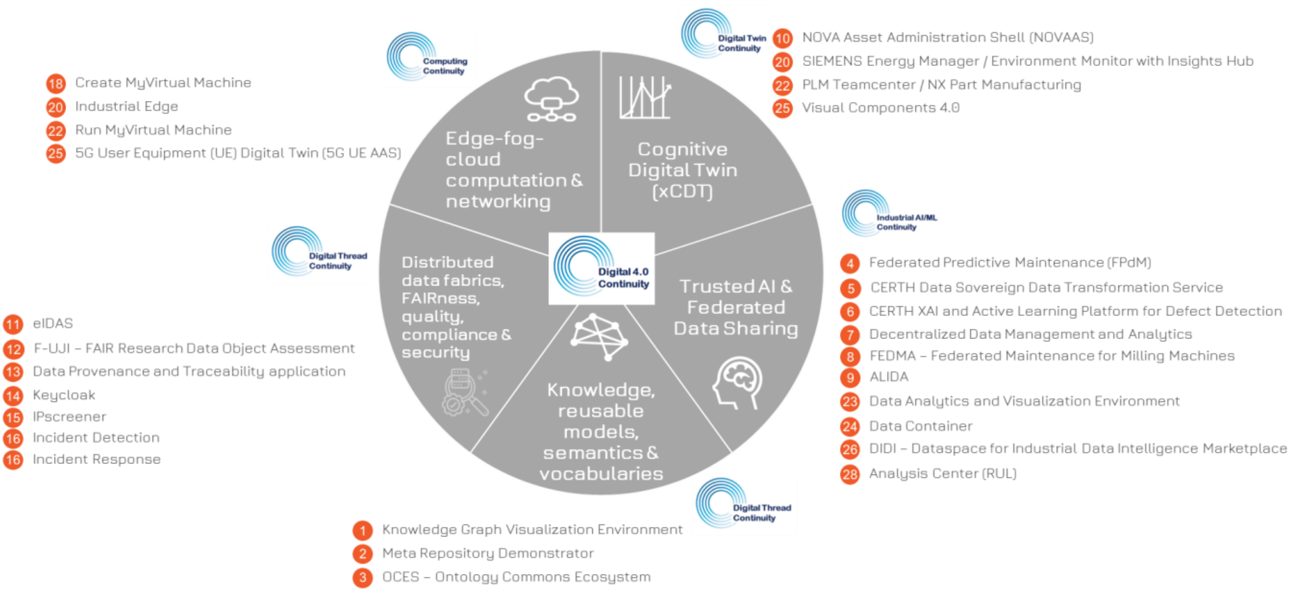


Figure 37: Toolkit components with Digital Continuity Domains (© RE4DY Consortium)

Besides, RE4DY toolkit also aligns closely with the various stages of the data journey in Industry 4.0, from data collection to advanced collaborative and sustainable manufacturing systems. Here's how each category fits into this data-driven process (see Figure 38):

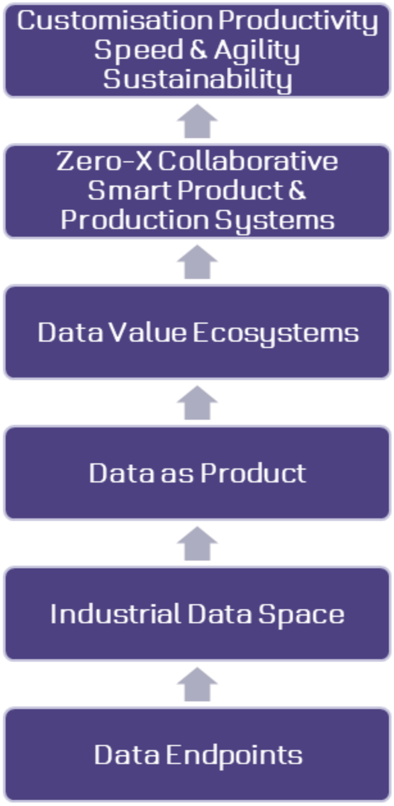


Figure 38: Data journey (© RE4DY Consortium)

- Data Endpoints: This is the initial stage where data is generated and collected from sensors, machines, and other industrial equipment. The Static and Dynamic Executable Cognitive Digital Twin (xCDT) Commissioning Tools play a crucial role here. They create virtual representations of these physical assets, allowing data from endpoints to be continuously monitored and analysed, enabling real-time simulations and predictive maintenance.
- Industrial Data Space: The Trusted Industrial AI Federated Data Continuous Delivery Tools (Data Marketplace) facilitate the transition into the industrial data space. This space enables secure data exchange and collaboration between different entities, while federated learning models ensure that AI is developed and deployed across multiple sites without sharing sensitive raw data, maintaining privacy and data integrity.
- Data as Product: As data becomes a key asset, it is treated as a product that holds intrinsic value for business processes. Knowledge Quality Tools, Reusable Models, and Common Semantics and Vocabularies 4.0 ensure that this data is standardized, accurate, and meaningful. These tools help refine data so that it can be directly used for analytics, AI training, and operational improvements, turning raw data into valuable digital assets.
- Data Value Ecosystems: This stage focuses on creating networks where data flows across different stakeholders, enhancing overall value. Distributed Data

Fabric FAIRNESS, Quality Sovereignty, Security, and Compliance Tools ensure that data shared within these ecosystems remains secure, accessible, and interoperable. They help maintain the sovereignty and integrity of data while it moves through various stakeholders in the ecosystem, facilitating collaboration and trust between entities.

- Zero-X Collaborative Smart Product & Production Systems: At this point, data-driven insights enable the development of smart, interconnected production systems that adapt and optimize themselves. The Optimized Distributed Data Edge-Fog-Cloud Computation and Networking plays a vital role here. It allows seamless computation across the edge (close to machines), fog (intermediate network layers), and cloud, providing the real-time responsiveness needed for such collaborative systems. This allows production lines to be more adaptive, flexible, and efficient, enhancing the interaction between different production components.
- Customization, Productivity, Speed & Agility, Sustainability: The entire data journey culminates in the ability to deliver customized products and solutions quickly and sustainably. The integration of all these tools—cognitive twins, AI, data marketplaces, and optimized computing—supports rapid decision-making and process adaptation. This flexibility allows manufacturers to meet customer-specific requirements while optimizing resource use, reducing waste, and ensuring sustainable production practices. It aligns with the goals of Industry 4.0, focusing on agility and sustainability in production systems.

With the aim of demonstrating and showcasing scenarios where the toolkit components could bring value in the different steps of the Digital Journey, a set of eight initial experiments – involving one or more toolkit components, in combination with other systems and datasets) were already initially described in deliverable D3.2 (and whose updates and results are presented in the following sections of this deliverable).

Figure 39 shows the mapping of the participation of the toolkit components in the different experiments and steps in the Digital Journey are covered by which experiments.

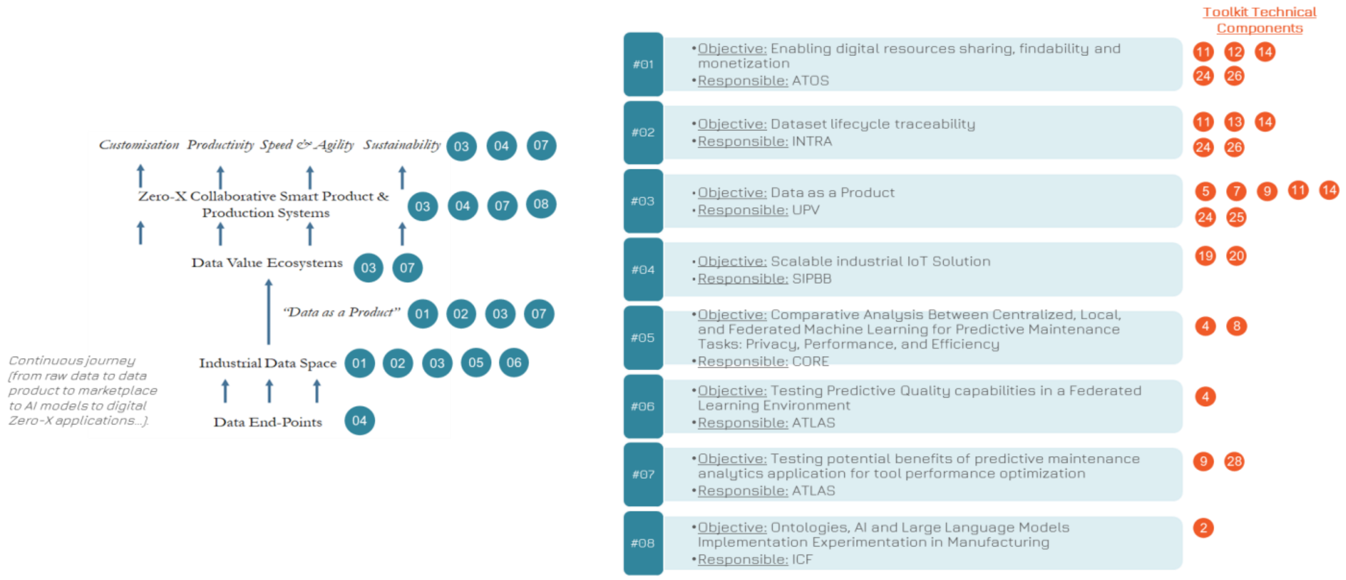


Figure 39: Participation of toolkit components in the experiments and what steps of the Digital Journey are covered by the experiments.

4 Experiment Plan and Evaluation

This section provides a list of experimentation plans used to assess the applicability and functionality of a tool or integrated set of tools on the TEFs and pilot sites included in the RE4DY project. Each plan describes the subject and aim of the experiment and introduces a number of scenarios which enumerate the steps to be followed in order to test each tool leveraged in the respective scenario. Furthermore, every scenario outlines the components which participate in the scenario, the datasets or AI models employed for its implementation and a list of proposed added values. In addition, it declares a number of KPIs used to examine the functionality of the related components and finally, the partners who are involved in the scenario.

4.1 Experiment 01: Enabling digital resources sharing, findability and monetization.

In the manufacturing sector, supply chain collaboration is crucial for the efficient and reliable production of goods. The aim of this experiment is to test and demonstrate the feasibility of enabling collaboration among multiple manufacturing organizations within a supply chain to share digital resources seamlessly and securely by using a digital data marketplace based on the dataspace concepts and technologies.

To that end, this experiment will consider the following scenarios:

- Resource Publication in the data marketplace
- Discoverability and Access (including FAIR Access Verification)
- Contract Signing and Payment

4.1.1 Scenario 01: Resource Publication in the data marketplace

In this scenario a set of digital resources have be published in the data marketplace using self-descriptions, which will include detailed metadata, specifying the data type, source, price, and usage restrictions among others. The type of digital resources considered for publication in the marketplace are datasets generated by the Swiss Smart Factory (SSF) manufacturing partner in RE4DY.

Two instances of the Sovity Connector were deployed in Atos infrastructure (Casasola - https://connector-ready-webui.casasola.ari-energy.eu, and Baikal - https://connector-ready-webui.baikal.ari-energy.eu).

In this case, the Casasola Sovity instance acts as proxy provider for the datasets hosted and offered by the Swiss Smart Factory (SFF) by means of the APIs of the SIEMENS Insights Hub component.

As shown in Figure 40 and Figure 41, the registration of the assets (i.e., datasets) can be done either using the integrated UI or by means of the available API.

In addition to that, both Sovity connector instances were configured and integrated with the Keycloak security service provided by Intrasoft, guaranteeing thus the possibility to interact in a secure manner with other WP3 components that are also using it.

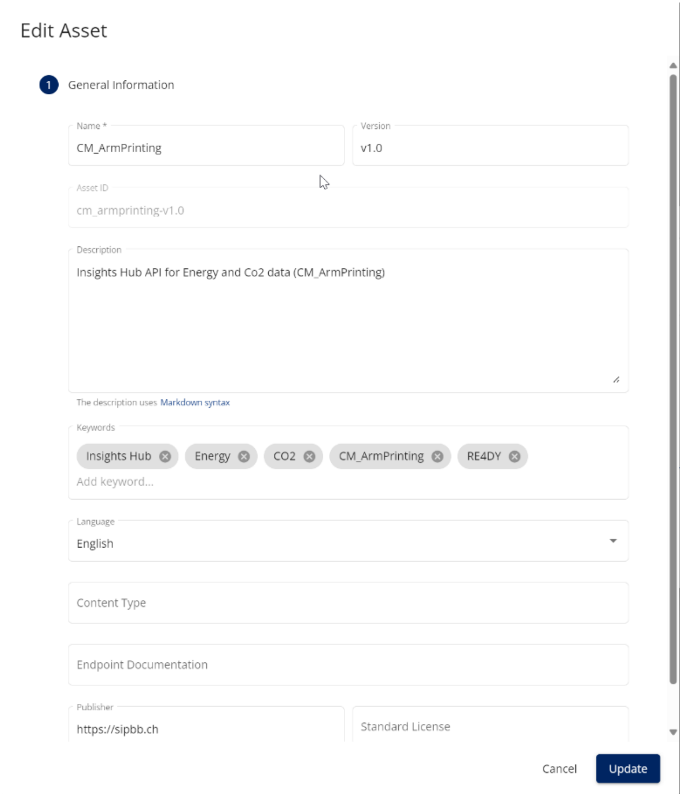


Figure 40: Sovity Connector - Asset registration via UI

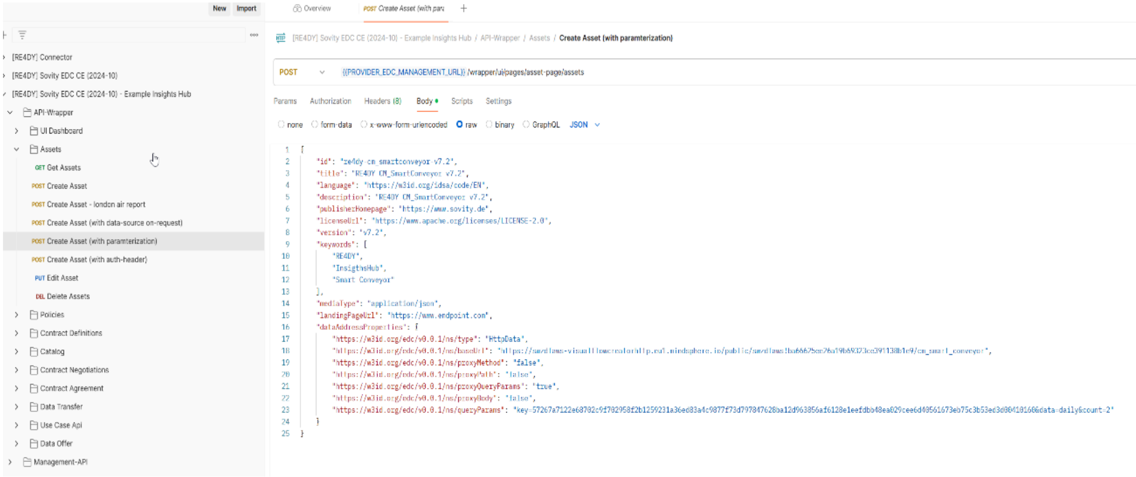


Figure 41: Sovity Connector - Asset registration via API

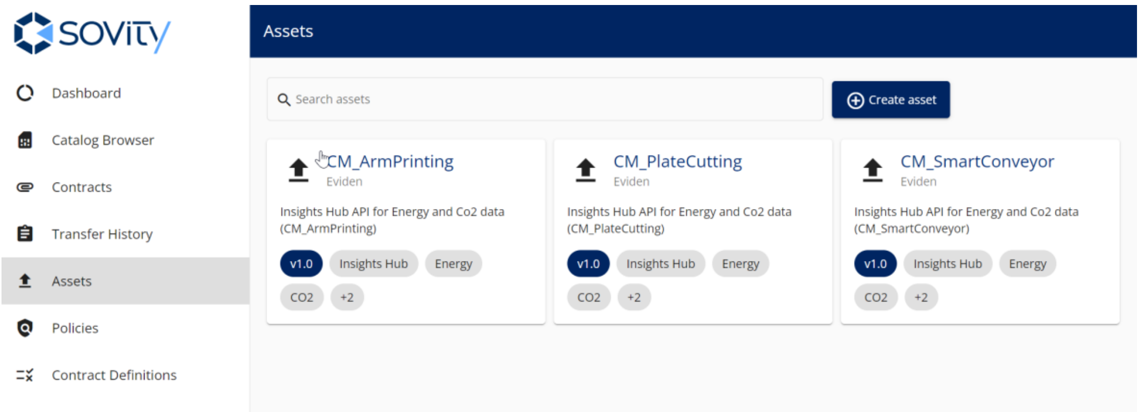


Figure 42: Assets registered in Casasola Sovity Connector instance

Components participation	• Sovity Connector (note: initially, Atos' 'Dataspace for Industrial Data Intelligence (DIDI)' based on several GAIA-X software components was considered, however, due to some limitations in its current implementation it was decided to switch to the Sovity Connector implementation) • eIDAS/Keycloak • Insights Hub • Data as Product Containers (DC)
Datasets/AI Models	Energy and Co2 data: ArmPrinting, PlateCutting and SmartConveyor
Proposed Added Value	• Interoperability: Standardized formats improve interoperability among manufacturing organizations, streamlining operations.

	• FAIR Compliance: The implementation of FAIR principles ensures that digital resources are well-structured and readily discoverable. • Data Governance and Compliance: DIDI data marketplace help enforce data governance policies and regulatory compliance by setting access controls, contracts, and usage terms for resources. This mitigates legal and compliance risks.	
Quantified KPIs	Number of datasets published: at least 2 datasets will be published free of cost and 2 others with an associated cost	3 datasets free of cost (the current implementation of Sovity connector doesn't allow to specify the cost in the UI form)
	Number of AI models published: at least 2 AI models will be published	0 (no AI models were available for publication)
Partners Participation	ATOS, INTRA, UPV	

4.1.2 Scenario 02: Discoverability and Access (including FAIR Access Verification)

In this scenario it has been tested that it is possible to easily search for digital resources using the Sovity Connector.

Users can access resources based on permissions and contracts defined within the Sovity Connector for each of the resources. Access can be free for open resources, while premium resources may require contractual agreements and potentially involve payment transactions (see scenario 03).

In some cases, access to the data will be done via specific connectors (e.g., Eclipse Data Connector (EDC) in compliance with IDSA/GAIA-X) whereas in others a link or API will be available for direct retrieval once the access has been authorized.

In this case, for the demonstration purposes of the scenario, we are using both Sovity instances. The Baikal instance is used as consumer wanting to find (and consume) some assets provided via the Casasola instance (acting as provider). As shown in Figure 43 and Figure 44, the discoverability of the assets can be done either by using the UI of the Sovity Connector or via its APIs.

In order to do so, the consumer (Baikal instance) needs url pointing to the provider (Casasola)'s catalogue: https://connector-ready.casasola.ari-energy.eu/api/dsp

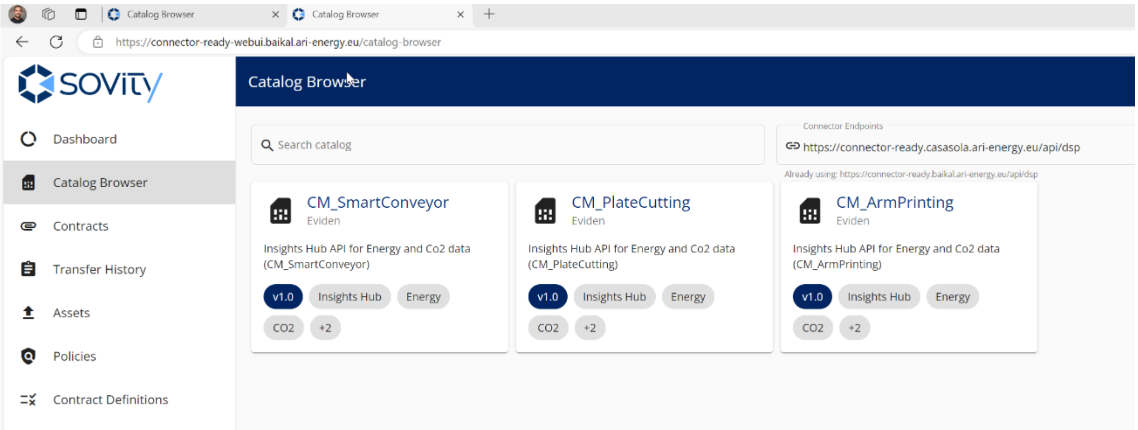


Figure 43: Baikal Sovity Connector (consumer) retrieving the list of assets (contracts) available in Casasola Sovity Connector via the UI

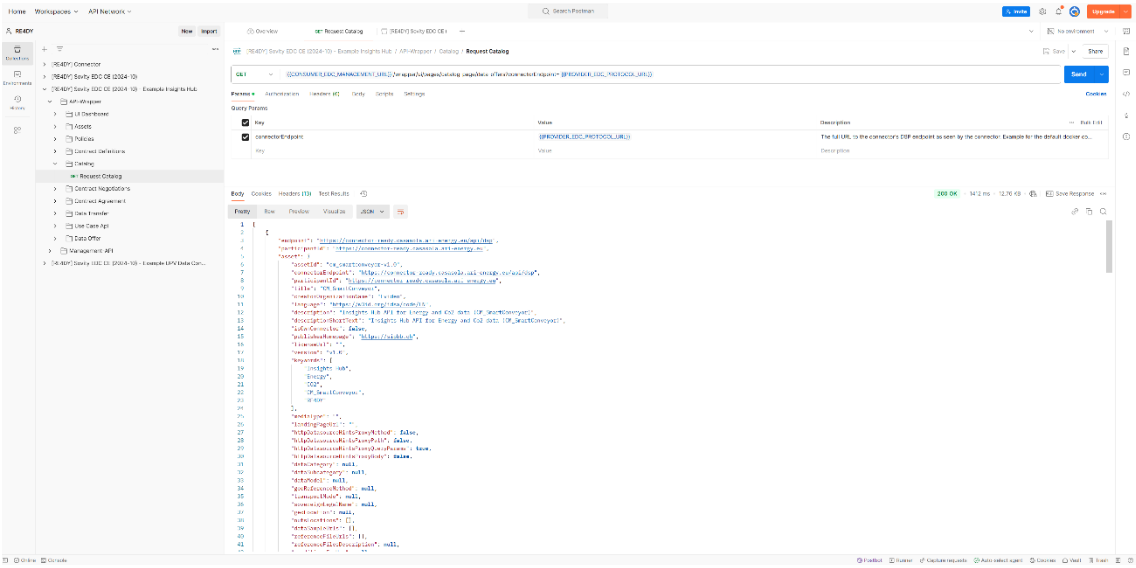


Figure 44: Baikal Sovity Connector (consumer) retrieving the list of assets (contracts) available in Casasola Sovity Connector via the API

Once the consumer has retrieved the list of assets, it can check the conditions of each of the contracts (ownership, duration, policies, etc.) in order to determine if it might be of interest acquiring it. Figure 45 shows in a graphical manner the information contained in one of the assets contracts.

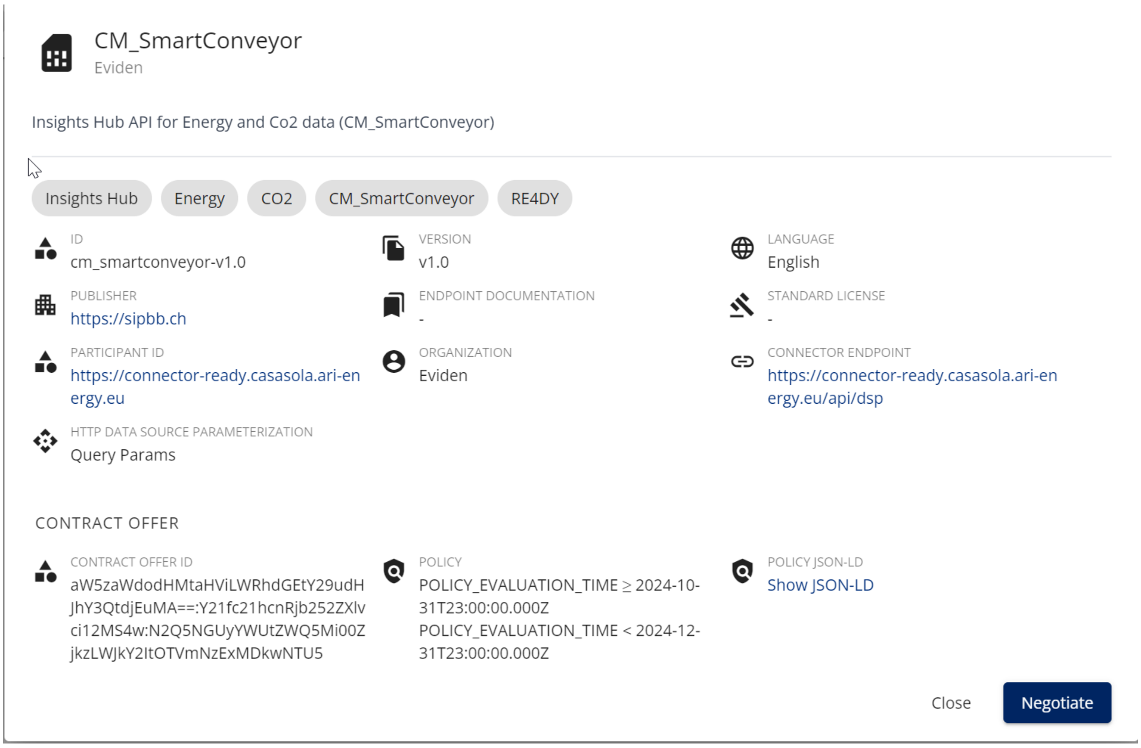


Figure 45: Detailed information of the contract for accessing to the CM_SmartConveyor asset provided by Casasola Sovity Connector instance.

Components participation	• Sovity Connector • eIDAS/Keycloak
Datasets/AI Models	Energy and Co2 data: ArmPrinting, PlateCutting and SmartConveyor
Proposed Added Value	• Improved Accessibility: Sovity Connector enables resources to be made accessible to a wider audience through standardized interfaces and authentication mechanisms. By ensuring that the right users have access to the right data, it promotes data accessibility, while also enforcing data security and privacy, which aligns with the FAIR principle of Accessibility. • Efficiency: The Sovity Connector facilitates the swift exchange of production data and models, enabling manufacturing organizations to respond quickly to supply chain demands. • Interoperability: Standardized formats improve interoperability among manufacturing organizations, streamlining operations. • Quality and Consistency: Standardized data formats and clear self-descriptions ensure data quality and consistency, reducing errors in data interpretation. • Cost Reduction: By improving resource discovery and access, Dataspaces can reduce costs associated with redundant data acquisition, processing, and

	management. This is particularly valuable in sectors where data efficiency is essential, such as manufacturing. • Automation and Scalability: Sovity Connector supports automated processes for resource discovery, retrieval, and management. Automation helps streamline operations and enables scalability, making it easier to handle an ever-growing volume of digital resources.	
Quantified KPIs	• FAIRNESS of the datasets discovered: assess the degree of fairness of each of the datasets discovered in the data marketplace.	None. This is covered in Experiment #02
	• Data access methods: at least one dataset will be accessed using the Data as Product Container API; at least one dataset will be accessed using the Eclipse Data Connector (EDC); at least one dataset/model will be accessed using a common/proprietary API (as defined by the data owner).	2 (Data is discovered using EDC protocol and consumed using the providers' proprietary API)
Partners Participation	ATOS, INTRA	

4.1.3 Scenario 03: Contract Signing and Payment

In this scenario it was tested and demonstrated the specific use case when ait is required signing a contract and adhering to the terms of use before accessing sensitive data or manufacturing AI models made available by the data provider through the Sovity Connector.

Initially, it was planned to implement and test three two monetization schemas: pay-per-use (that is, pay once and use it), a subscription model and free access to the assets. Given the current implementation status of the open-source community version of the Sovity Connector it was only possible to test the last option, postponing for a later stage the testing of the other monetization schemas.

Starting from the previous scenario, where the user has found an asset of interest (in this case, the CM_PlatteCuting one), he/she signs the contract (Figure 46 and Figure 47). Once it is signed, the user can decide to initiate the download of the asset. To that end, a data sink endpoint where to push the data to must be provided (i.e., in this test we provided an ephemeral url from the free service 'webhook.site').

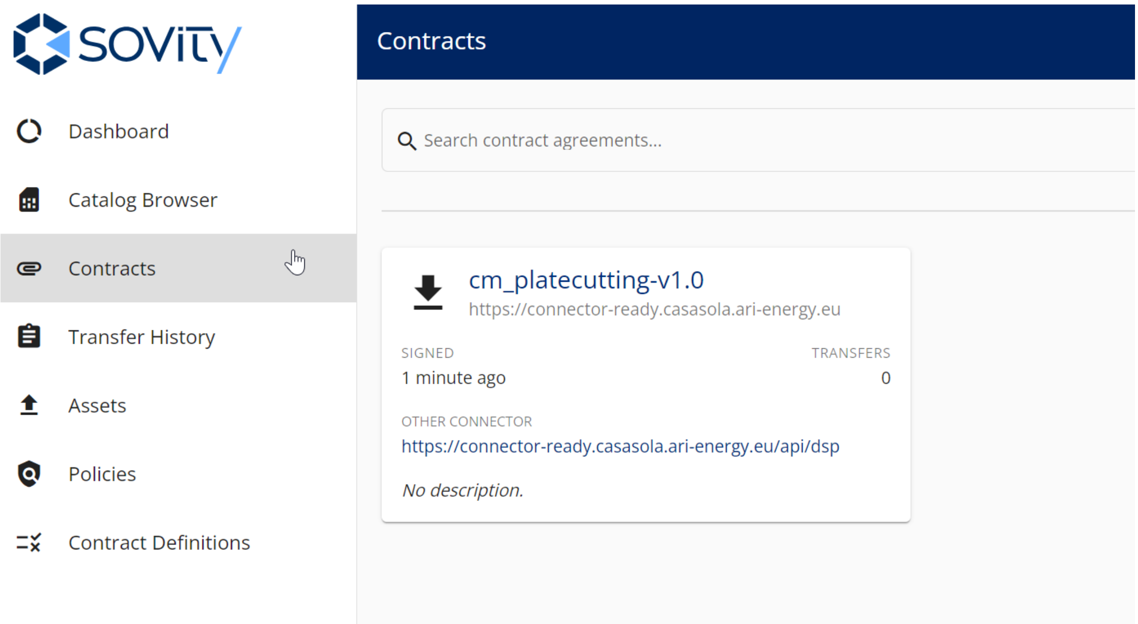


Figure 46: Contract signed by the consumer (Baikal), accepting the conditions for accessing the CM_PlateCutting machine data.

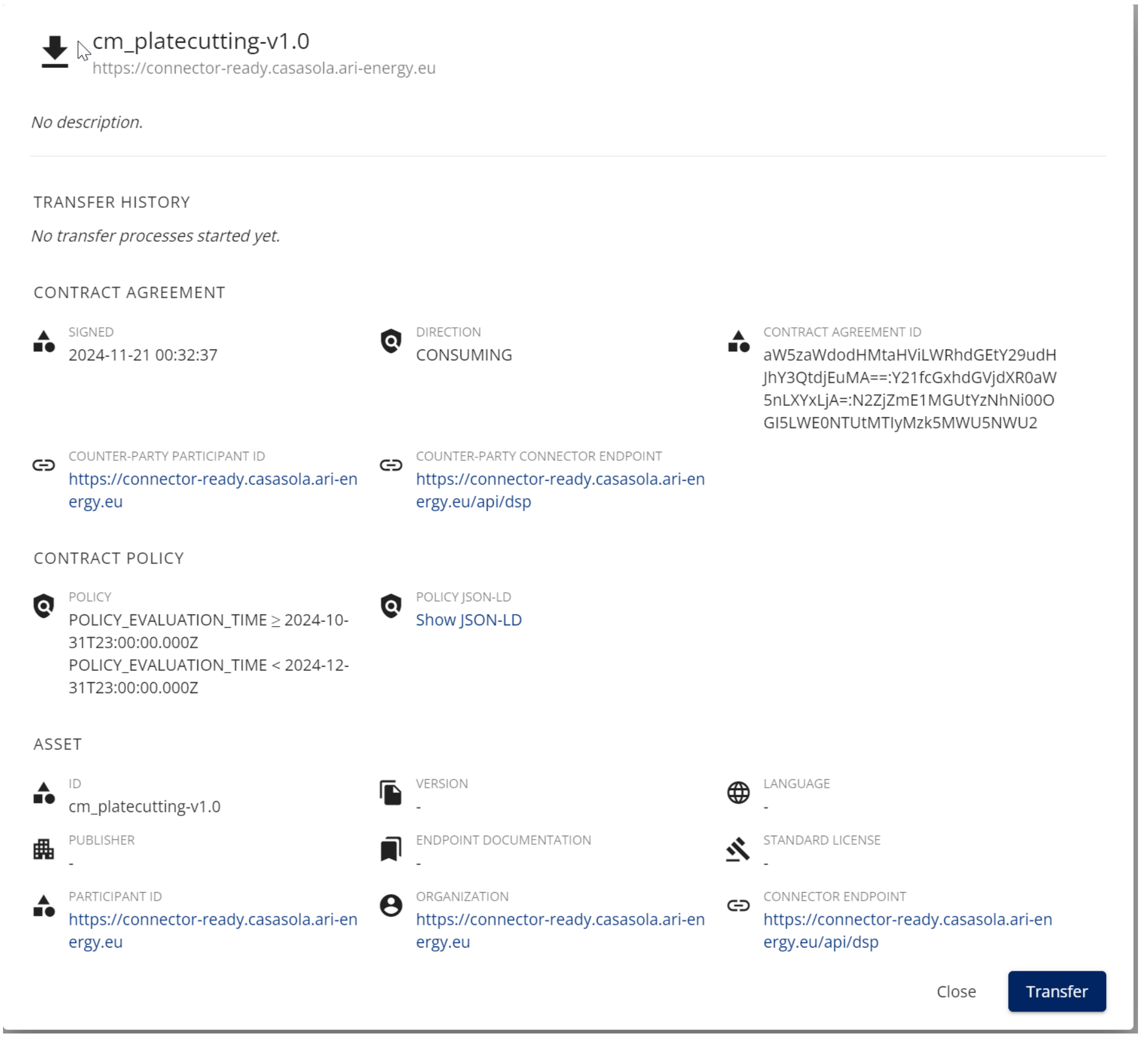


Figure 47: Detailed information about the contract for the CM_PlateCutting machine dataset

The transfer form also allows to provide additional query parameters (as shown in Figure 48) that will be appended to the request in the provider's side, which will further filter the data to be retrieved (in our case, we specified that we wanted to access to the current measurements from the plate cutting machine with the parameter 'data=current').

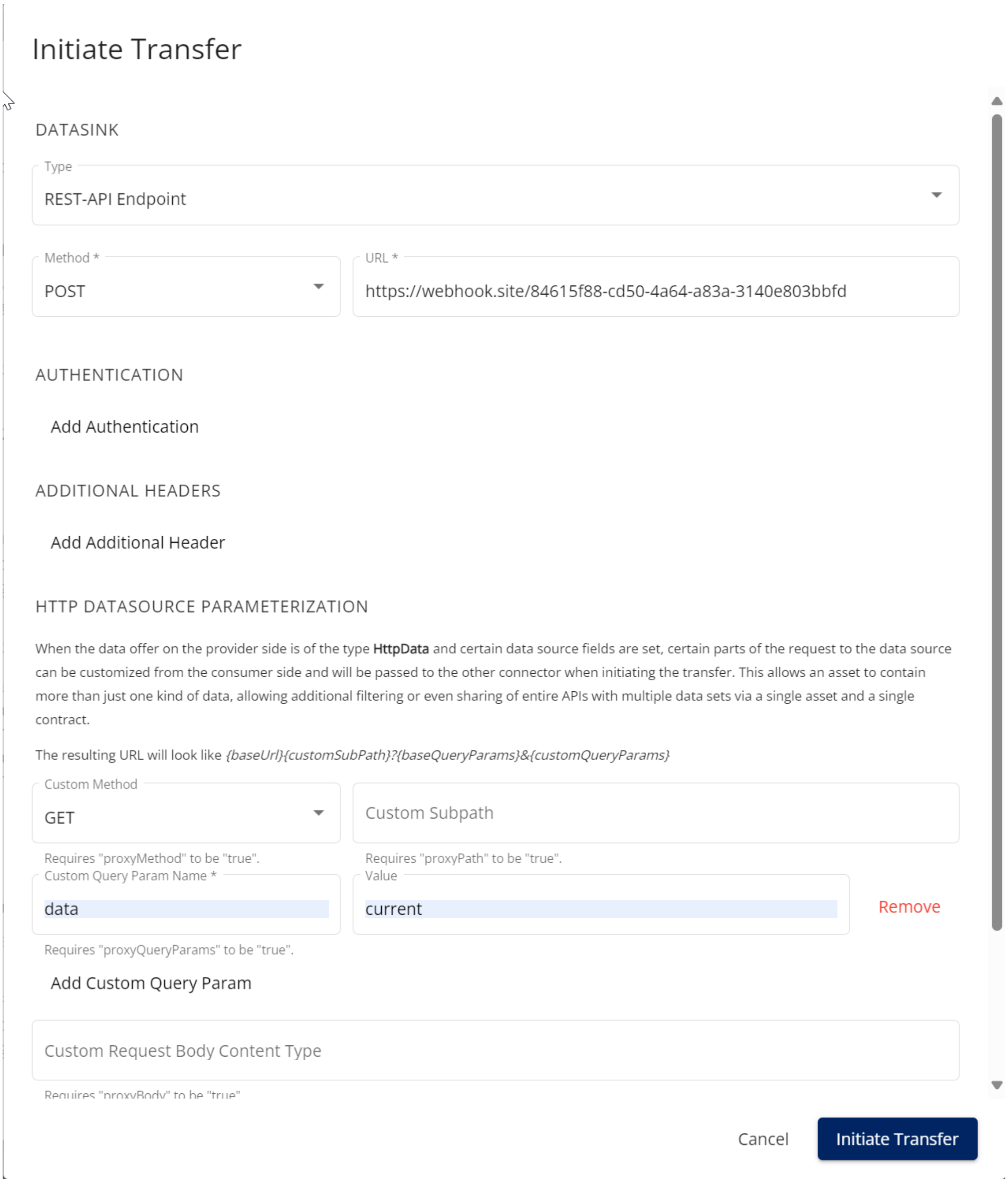


Figure 48: Initiate Transfer form, where additional parameters can be configured in order to make the request to the data provider

Finally, as shown in Figure 49, after a few seconds, the resulting dataset is made available in the https://webhook.site page.

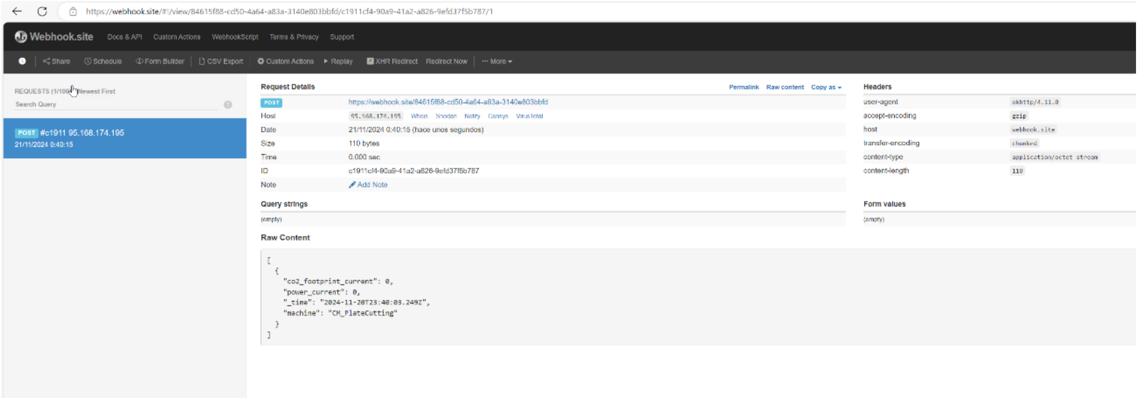


Figure 49: Reception of the CM_PlatteCutting machine data.

<table>
<tr><td>Components participation</td><td colspan="2">Sovity Connector
eIDAS/Keycloak
Insights Hub</td></tr>
<tr><td>Datasets/AI Models</td><td colspan="2">Energy and Co2 data: ArmPrinting, PlateCutting and SmartConveyor</td></tr>
<tr><td>Proposed Added Value</td><td colspan="2">Security: Strong authentication, authorization, and encryption mechanisms protect sensitive data and manufacturing models.
Monetization: The system allows for the monetization of premium resources, which can generate revenue for the resource owners.
Contract Management: Digital supply chain contracts are executed within the platform, guaranteeing that agreements are honoured, and that goods and services are delivered as specified.</td></tr>
<tr><td>Quantified KPIs</td><td>Number of monetization schemas: at least 2 datasets with two different payment schemas will be available, requiring signing a contract.</td><td>All 3 available datasets require accepting and signing a contract (with cost 0 euro) in order be able to access to the data.</td></tr>
<tr><td>Partners Participation</td><td colspan="2">ATOS, INTRA</td></tr>
</table>

4.2 Experiment 02: Dataset lifecycle traceability

Data provenance and traceability involves tracking and tracing the lifecycle of a piece of data. It constitutes a valuable tool in multiple sectors as it ensures transparency, accountability and aids in demonstrating regulatory compliance. This experiment aims to exhibit how blockchain-based asset traceability can prove beneficial in a data marketplace where transparency and accountability are of essential importance, by using a blockchain to securely record a dataset's lifecycle events such as creation, purchase and consumption.

With this objective in mind, this experiment encompasses the following scenarios:

- Dataset Lifecycle Record
- Dataset Provenance and Traceability Report

To support both scenarios a Data Provenance and Traceability application API is implemented providing endpoints used to record important information and dataset events. With Keycloak as our Identity and Access Management (IAM) system and eIDAS for individual identification, only registered and thus authenticated users can access these API endpoints.

Create Dataset

It is possible to create a reference to a dataset in the Data Provenance and Traceability Application by calling the /createDataset endpoint like so:

```
curl --location 'https://{data_provenance_url}/createDataset'\
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'datasetId=dataset1' \
--data-urlencode 'owner=leo' \
--data-urlencode 'listOfRelatedDatasets=["dataset2", "dataset3"]'
```

Where:

- **dataset1** is the dataset ID.
- **leo** is the dataset owner.
- **dataset2** and **dataset3** are datasets related to dataset1.

Change dataset owner

Once a dataset has been created, it is possible to change its owner by calling the /changeOwner endpoint like so:

```
curl --location 'https://{data_provenance_url}/changeOwner' \
```

```
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'datasetId=dataset1' \
--data-urlencode 'newOwner=bob'
```

Where:

- **dataset1** is the dataset ID.
- **bob** is the new dataset owner.

Record dataset access

Dataset access through the related DC should be recorded using the Data Provenance and Traceability application by calling the following endpoint like so:

```
curl --location 'https://{data_provenance_url}/recordAccess' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'datasetId=dataset1' \
--data-urlencode 'description=Which tool accessed the dataset'
```

Where:

- **dataset1** is the dataset ID.
- **description** is an arbitrary string that usually mentions the name of the tool that accessed this dataset.

This endpoint will return a traceability event ID such as **8a23c559e40c489eb86da3031550614e70091e2ca51ab3ea55c954a4eeff7f97**, which, if needed, may be used later to retrieve information regarding this event.

4.2.1 Scenario 01: Dataset Lifecycle Record

This scenario demonstrates the ability to record asset traceability events on-chain by leveraging blockchain-enabled smart contracts which allow secure and trusted algorithm execution.

Through the use of smart contracts, it is possible to develop functions which store a reference to an asset and events related to its lifecycle in an append-only decentralized ledger. For this experiment, the type of asset considered is a dataset published using a Sovity Connector instance. Hyperledger Fabric constitutes the blockchain of choice. An event will be sent to the Data Provenance and Traceability application whenever a dataset is published or purchased via the Sovity Connector. Furthermore, a consumption event will be sent to the Data provenance and traceability application whenever the dataset is accessed through the related Data Container. The aforementioned events will be stored on-chain using a smart contract deployed in a Hyperledger Fabric channel. Keycloak and eIDAS will be utilized for access control and in particular to manage rights to smart contract functions which implement traceability.

Components participation	• eIDAS • Keycloak • Data Provenance and Traceability app • Sovity connector • Data Container
Datasets/AI Models	2 Datasets: • RE4DY_Env_Data • RE4DY_Machine_Data
Proposed Added Value	• Trust: Due to the immutable and transparent nature of the blockchain, it is possible to ensure that stored records are tamper-proof and that members are able to access chain data from the first to the latest block and independently verify it. • Accountability: Transactions are signed with the private key of the transaction sender and transparently recorded within an append-only and immutable ledger, which ensures accountability of actions within the chain. • Security: The immutability and secure execution of smart contracts eliminates the need for centralized, trusted intermediaries and removes the possibility of tampering and manipulation of results.
Quantified KPIs	Number of datasets recorded: 2 datasets references (RE4DY_Env_Data & RE4DY_Machine_Data). Number of traceability events recorded: 26 events recorded for the first dataset RE4DY_Env_Data & 41 events recorded for the second dataset RE4DY_Machine_Data.
Partners Participation	INTRA, ATOS, UPV

4.2.2 Scenario 02: Dataset Provenance and Traceability Report

This scenario will serve as a test ground for the access and transparency of on-chain dataset references and their traceability events. A dashboard will be developed to present the data available on-chain in a user-friendly format. The data will be retrieved through interfaces implemented for Hyperledger Fabric and presented in a report which will enumerate in order the events related to the dataset, such as creation, purchase and access, from the point of its creation up to the present. Through the report, it will also be possible to trace the exchange of ownership of the related dataset. Keycloak and eIDAS will be integrated with Hyperledger Fabric for access control and in particular for the purpose of managing access rights to the Fabric channel where the related smart contract is deployed.

Below are the two generated reports for `RE4DY_Env_Data` and `RE4DY_Machine_Data`. They illustrate, in a chronological chain of events, when the dataset was created, each access occurrence (including user details and timestamps), and any changes in ownership.

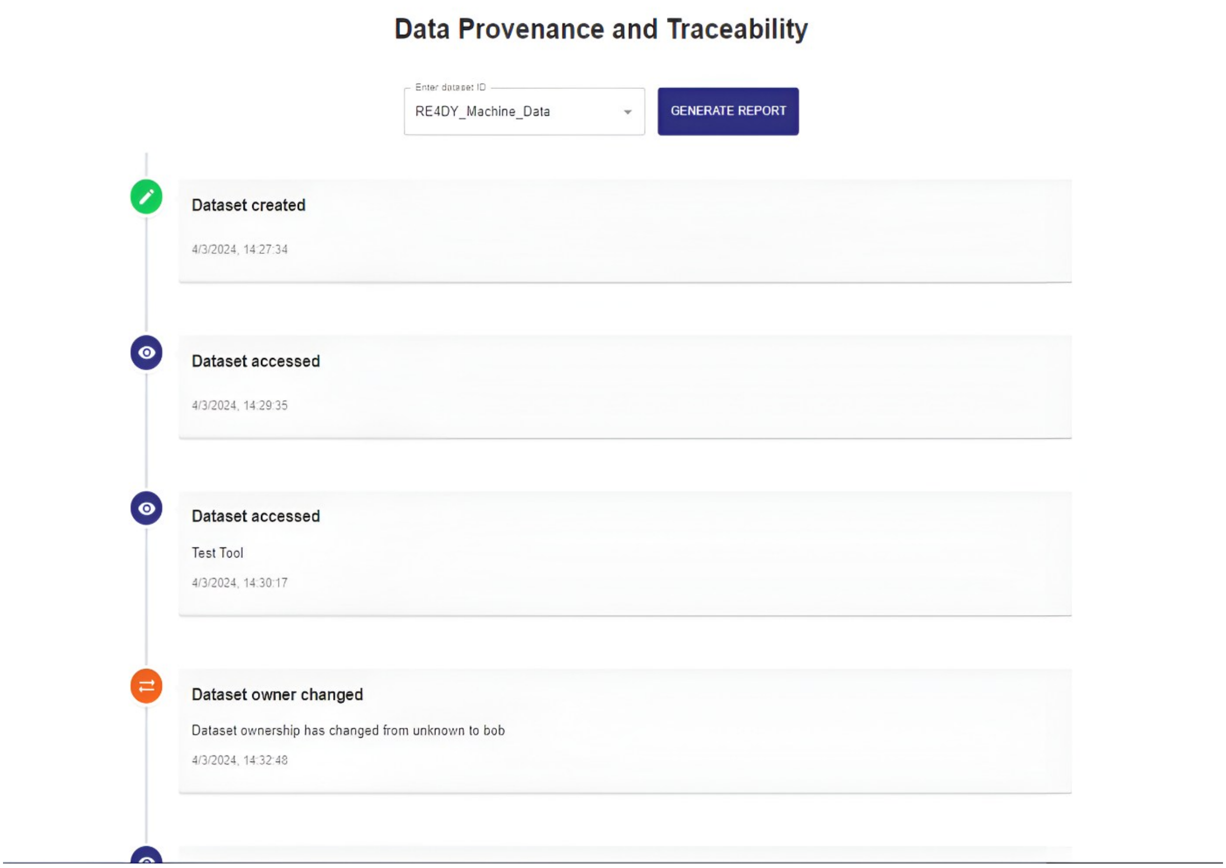


Figure 50: Dataset Provenance and Traceability Report for RE4DY_Machine_Data

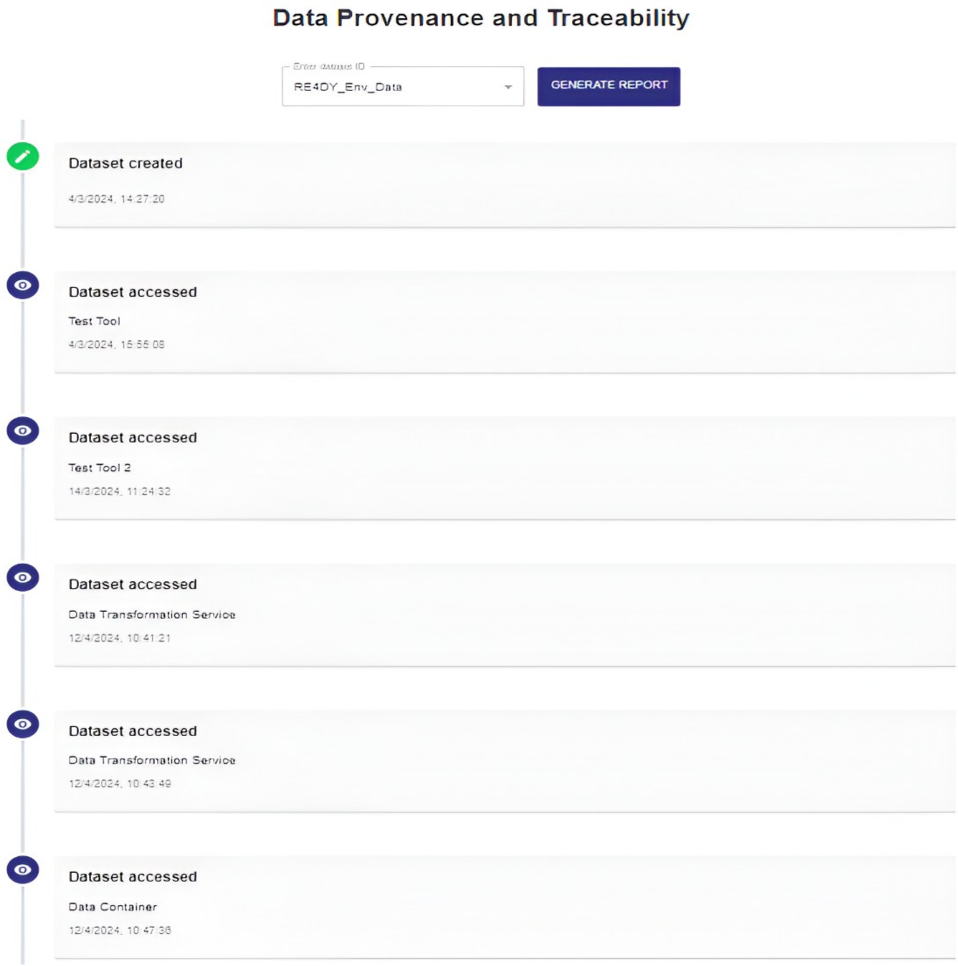


Figure 51: Dataset Provenance and Traceability Report for RE4DY_Env_Data

Components participation	• eIDAS • Keycloak • Data Provenance and Traceability application
Datasets/AI Models	2 Datasets: • RE4DY_Env_Data • RE4DY_Machine_Data
Proposed Added Value	Transparency: The code defining a smart contract is publicly available and accessible for any party to verify. After its deployment it becomes immutable and secure execution within the blockchain ensures that the rules and conditions defined in it are carried out as intended. Availability: Through its decentralized architecture, blockchain may achieve high availability. Nodes are

	continuously operational and due to redundancy, the network remains accessible regardless of occasional node failure.
Quantified KPIs	Number of reports generated: 2 reports, one per referenced dataset (RE4DY_Env_Data & RE4DY_Machine_Data).
Partners Participation	INTRA

4.3 Experiment 03: Data as a Product

Data as a Product methodology not only ensures data availability but also emphasizes its quality and usability, making it a valuable asset for informed decision-making and driving operational excellence. By treating data as a product, this approach fosters a culture of data-driven insights and continuous improvement, transforming historical data into a strategic resource that empowers the organization to adapt, optimize, and innovate in an ever-evolving landscape. This experiment aims to establish an open and transparent data ecosystem, where data and services can be provided, aggregated, and exchanged within a context of confidence and reliability.

With that objective in mind, this experiment will examine the subsequent scenarios:

- Historical Data as a Product
- DaaP and Data Spaces
- DaaP and decentralized Machine Learning
- DaaP and AAS / Digital Twin

4.3.1 Scenario 01: Historical Data as a Product

When historical data is well-prepared and standardized, it becomes a valuable resource for analysis and benchmarking, allowing the pilot to identify trends, patterns, and opportunities for improvement more effectively. This scenario is in charge of providing access to the pilot's historical data following the Data as a Product approach. It provides an easy-to-use environment for sharing, collecting, and pre-processing data that simplifies the entire workflow. This simplicity not only saves time and therefore reduces costs, but also enhances data quality and consistency.

The Data Container provides a unified API to access the data from the different connected sources. To enable the DaaP workflow, the Data Container has been integrated with the Data Transformation Services and with the security components of the RE4DY toolkit (Figure 52).

When new data is requested, the Data Container sends a request to the corresponding Data Transformation Service, which communicates with the data source and translates the data into a data model that can be understood by the client application. The Data Container collected this data, applied the proper filtering rules and transformed it into the output format requested by the client.

Regarding the security, the integration of Keycloak and eIDAS enables authentication and authorization. Moreover, the Incident Detection tool monitors the endpoints and infrastructure and offers a comprehensive view of security-related events and potential threats, while the Incident Response tool performs automated responses to mitigate the threats. To integrate the Incident Detection and Response components, agents were deployed in the infrastructure running the Data Container and Keycloak. These agents collected all the relevant information from the logs and sent it to the Incident Detection tool, which analysed this data to generate security alerts based on rules. These alerts where sent to the Incident Response, which determined the risk level and the proper action. The implemented actions include sending a notification via Telegram or MS Teams, blocking/unblocking the attacker's IP address, complete host contention/uncontention, and Keycloack ad-hoc actions (block user, update user, delete user), among others.

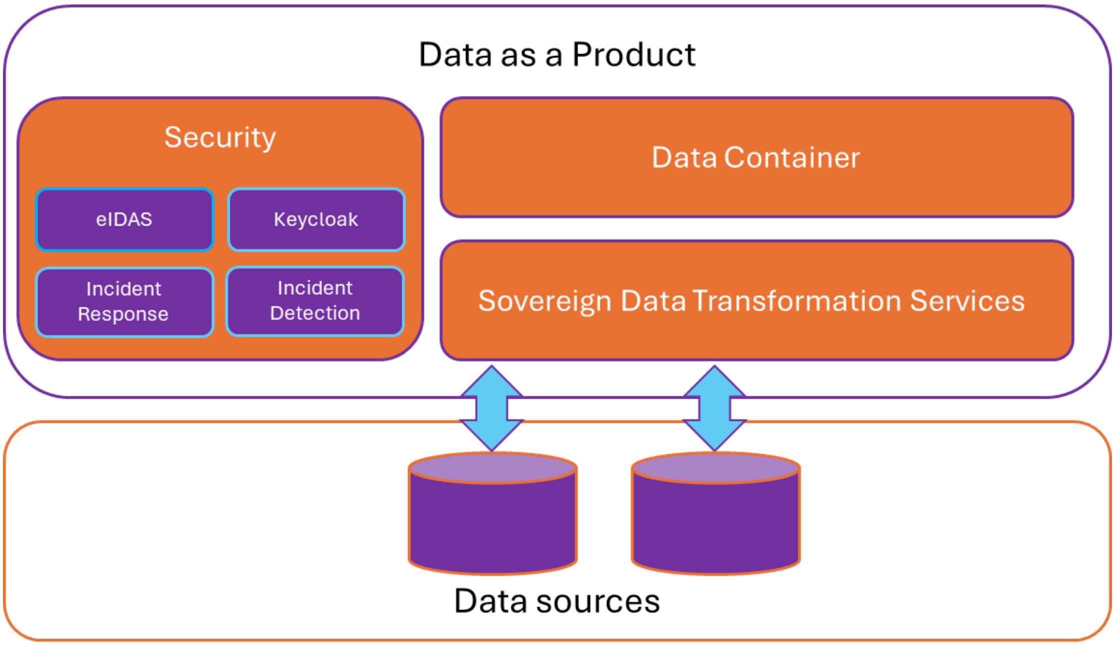


Figure 52: Integration of the Historical Data as a Product scenario

Components participation	• Data as Product Container (DC) • Data Connection Profile (DCP)

	• Sovereign Data Transformation services • Keycloack • eIDAS • Incident Detection • Incident Response
Datasets/AI Models	Energy and environmental data from SSF TEF demonstrators, provided by Siemens Energy Manager with Insights Hub.
Proposed Added Value	• Trustworthy and truthful: The data should accurately portray the organization's actual state to be of value. To achieve this, various methods will be employed, including data purification and evaluating data quality. Additionally, incorporating data source and data history details within the metadata will further enhance trustworthiness. • Interoperability: Ensuring the existence and application of standards and harmonization rules is essential to facilitate the sharing and correlation of data across diverse domains. Standardization empowers data consumers to meaningfully correlate and merge data from disparate domains. • Security: For the secure sharing of data, it is imperative to guarantee the preservation of data confidentiality, availability, and integrity. This involves the essential implementation of data access control; wherein access control policies are thoughtfully established to match the precise requirements of each individual data product. • Data quality: Accurate and consistent data builds trust and confidence, ensuring that collaborating entities are working with a common, reliable foundation.
Quantified KPIs	• Number of data sources shared: 2. • Number of DCPs delivered: 2.
Partners Participation	UPV, CERTH, INTRA, S21SEC

4.3.2 Scenario 02: DaaP and Data Spaces

This scenario will demonstrate the use of the Data Containers in a Data Space to enable trusted and secure data sharing among different stakeholders in order to create data-driven solutions. The use of IDS-certified technologies will allow the different parties to remain in control of the data that they decide to share, while use of DC will guarantee that the data being shared meets the quality criteria of the consumers.

The second scenario leverages the first as an evolution of its concepts (Figure 53). For this experiment, the Sovity connector was selected to provide the IDS connector functionality. The Keycloak instance from the previous scenario was integrated with the IDS connectors deployed in this scenario to act as the DAPS component of the data space. An IDS connector was added on top of the Data Container to provide access to the data consumers in the dataspace (through their own instances of the Sovity connector). In the Data Container's side, a new component (named DC client) was added in order to facilitate the communication with the IDS connector. In addition, the communication between the Data Transformation Services and the data sources was also mediated by instances of the Sovity connector.

Moreover, the Data Provenance and Traceability application was integrated in this scenario to track and record all the accesses to the data. In particular, the Data Provider registers the datasets with the Data Provenance and Traceability application. The Data Transformation Services and the Data Container send a notification to the Data Provenance and Traceability application whenever the dataset is requested from the component.

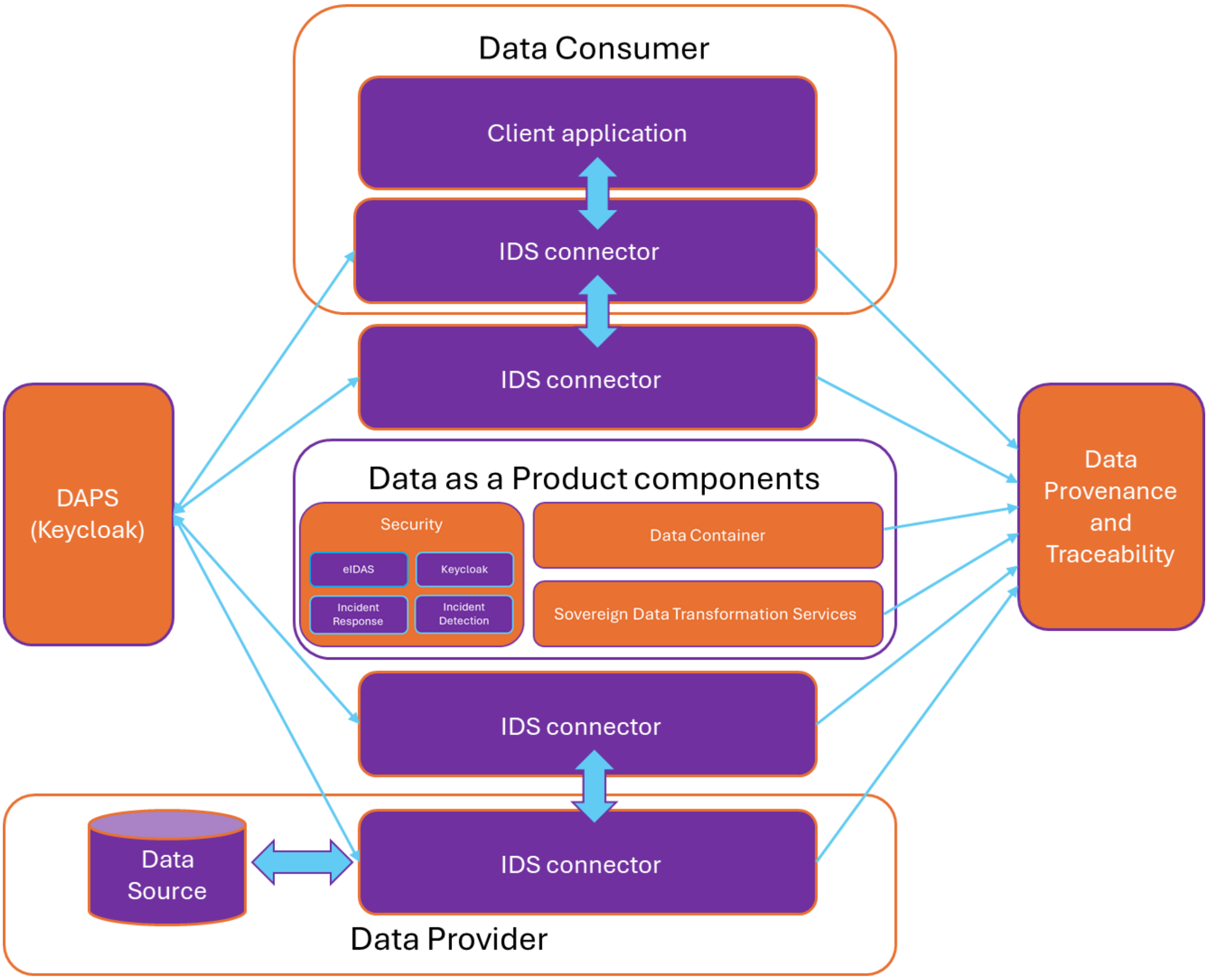


Figure 53: Integration of the DaaP and Data Spaces scenario

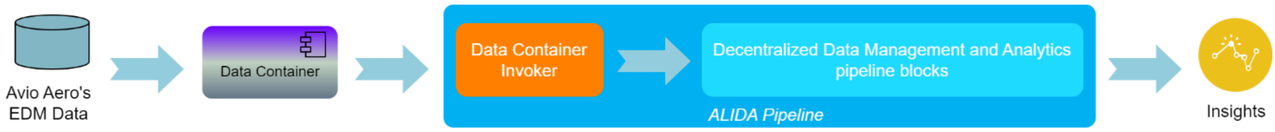
Components participation	• Data as Product Container (DC) • Data Connection Profile (DCP) • Sovereign Data Transformation services • IDS connector • Data Provenance and Traceability application • Keycloak • eIDAS
Datasets/AI Models	Energy and environmental data from SSF TEF demonstrators, provided by Siemens Energy Manager with Insights Hub.
Proposed Added Value	• Discovery: Enables the capability to access and utilize information independently, without requiring direct involvement from the data supplier. Additionally, metadata assumes a significant role in rendering data

	comprehensible, while also contributing to the formulation of data quality standards and policies for data usage. • Data Sovereignty: Maintaining control over data help organizations to mitigate risks of data breaches and unauthorized access. It also allows them to uphold the integrity and quality of their data, protecting them against the potential data degradation when shared or manipulated data among multiple parties. • Trusted data sharing: Encourages collaboration between organizations. When entities have confidence in the integrity and security of shared data, they are more willing to work together, leading to synergistic efforts and enhanced outcomes. • Interoperability: Transforms isolated data repositories into interconnected ecosystems, where information flows freely, supporting collaborative research, innovation, and informed decision-making.
Quantified KPIs	• Number of data assets shared using IDS connectors: 2.
Partners Participation	UPV, CERTH, INTRA

4.3.3 Scenario 03: DaaP and decentralized Machine Learning

In this scenario, Machine Learning is combined with Data Management under the DaaP paradigm. The Data Containers act as safeguards for data quality, enable access to historical information, and facilitate data preprocessing to enhance the performance of federated machine learning algorithms. By employing Data Containers, a data quality assurance system is established, increasing the accuracy and robustness of the algorithms, thereby contributing to more effective machine learning models.

The scenario (see picture blow) is based on and extends the Avio Aero's predictive quality business case described in detail in D5.2:



EDM (*Electric Discharge Machining*) data coming from Avio Aero's shopfloor are collected by the Data Container, pre-processed by the embedded Data Transformation services and offered to the client services such as the *ALIDA Platform*. ALIDA interacts with the DC by means of a so-called *Data Container Invoke,* which is a custom ALIDA *Big Data Application (BDA) Service* capable of interacting with the DC and its API to retrieve the desired datasets and pass them to the remaining blocks of the pipeline. By implementing the *decentralized data management and analytics models,* these blocks perform the actual federated model training by using the retrieved datasets. At the end of the process, the resulting trained model generates insights into the quality of the production process.

The following paragraph describes the Data Container Invoker - central to the integration between ALIDA pipeline and Data Containers – more in detail.

Data Container Invoker: integrating ALIDA pipelines and Data Containers

The *Data Container Invoker (DCI)* is a custom reusable ALIDA BDA Service designed to simplify the way Data Scientists who use ALIDA leverage the Data Container (DC) to access the necessary datasets.

Here is a detailed interaction flow between the DCI and DC:

1. The DCI requests a dataset to the DC via a REST call. In the request payload, the DCI also provides a callback notificationUrl and query parameters.
2. The DC retrieves data from the Data Storage and transforms them through its embedded Data Transformation services.
3. The DC stores the resulting Dataset on an SFTP Repository (deployed beside the DC).
4. The DC notifies DCI at DCI's notificationURL about the Dataset being ready for consumption after the Data Trasformation services have completed.
5. The DCI retrieves the Dataset through the DC's /get?id=xyz endpoint.
6. The DCI writes the Dataset into a *resources* folder on the local filesystem.
7. The DCI exits and the FML Participant Core starts.
8. The FML Participant Core reads the Dataset from the *resources* folder.

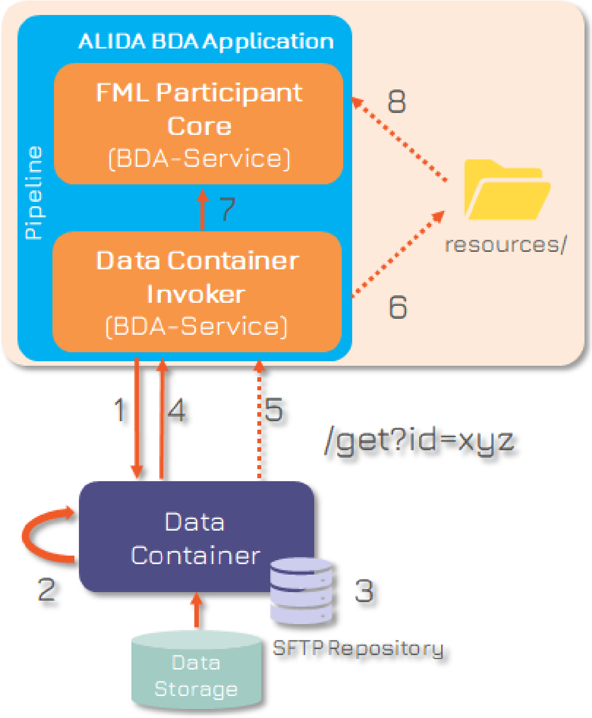


Figure 54: ALIDA Pipe line/Data Container Interaction Workflow

Decentralised Machine Learning

The overall workflow dedicated to executing the federated training of AI models in this scenario is the following. From an architectural point of view, there will be two FML clients and one FML server. The two clients will have access to two separate datasets, each one related to one EDM. Each FML client will train an AI model on the local data with the goal of being able to learn the underlying data structure. In this phase one option will be to adopt lightweight ML models such as Isolation Forest or Artificial Neural Networks models such as Autoencoders. These AI models are often used in the literature in the context of anomaly detection and predictive maintenance. The overall goal of the federated learning process is to let the local AI models (one for each client) learn not only the patterns and behavior contained in the local data but also to integrate those embedded in the other clients. To this end, the server will act as an aggregation and point of synchronization. This means in practice to make clients' models available among them, including both paradigms like traditional federated Learning or federated ensamble learning. The learning task will be tackled as an unsupervised learning problem, thus relieving the need for resorting to an often expensive and sometimes tricky labeling process. The target of the training will be to identify if new data patterns are or might be connected to an anomalous behavior of the EDM. The task in this context is particularly challenging due to the specifics of the underlying physical process, i.e., the data and the potential failures to happen at different time scales. Therefore, the prediction results will be considered as "indications" that a human operator can use to get more insights from the process. Prelimiinary analisys has been performed on one dataset provided by AVIO Aero and collected on one of their sites (i.e. Bielsko). The dataset contains data from two EDM machines, i.e., Bielsko1 and Bielsko2. To test the methodology, we split the data from two machines in train and test set with proportion 70-30, focusing on a subset of signals (['ARCKILL', 'ARCVOLTAGE', 'MACHININGSPEED', 'SEQUENCETIME']). We injected artificially in the two test sets 30% of anomalous data by applying to the features values a perturbation of ±3*std. We train two Isolation Forest models, one per EDM data on the train set assuming a contamination rate of 0.05. and we test the prediction of the ensemble on the concatenation of the two test sets. The results show 94.3% of accuracy with the following confusion matrix:

	Normal	Anomalies
Normal	91.88%	8.12%
Anomalies	0%	100%

Our aim was to validate our methodology applied to data coming from the Pilot. Further analysis and other learning models and techniques will be applied and possibly adapted to the needs of the Pilot during the activity of WP5.

Components participation	• Data as a Product Containers (DC) • Data Connection Profile (DCP) • Decentralised data management & analytics • ALIDA • Sovereign Data Transformation services
Datasets/AI Models	Dataset: AVIO Dataset collected in Bielsko during September. Models: Isolation Forest
Proposed Added Value	• Interoperability based on the use of standardized vocabularies and metadata, as well as the use of homogenised data formats and APIs to access the data. • Data quality: Accurate and consistent data builds trust and confidence, ensuring that collaborating entities are working with a common, reliable foundation. • Efficacy: Ability to train ML models in a collaborative fashion while preserving data privacy. • Efficiency: Ability to reach similar performance of centralised ML in terms of accuracy, without moving data from where they are generated. • Data Sovereignty: Maintaining control over data helps organizations to mitigate risks of data breaches and unauthorized access. It also allows them to

	uphold the integrity and quality of their data, protecting them against the potential data degradation when shared or manipulated data among multiple parties. • Reusability: enhanced through an extensible catalog of BDA services for ingestion, preparation and ML/DL. • Scalability: supported and enhanced by a micro-service based cloud-native platform. • Easier sourcing data, building models and pipelines integrated through a platform for BDA applications such as, but not limited to, FML participants and aggregator services.
Quantified KPIs	• Number of models trained: 1 • Accuracy: 94,3% • Number of FML participant and aggregator BDA services deployed: 2 • Number of DCPs delivered: 1.
Partners Participation	UPV, CNR, CERTH, ENG

4.3.4 Scenario 04: DaaP and AAS / Digital Twin

In this scenario, the digital twin or Asset Administration Shell (AAS) of a 5G network exposes information about the 5G network to external nodes or application. The 5G system digital twin is made of the AAS of the 5G UE and the AAS of the 5G network (NW). The 5G UE is the endpoint of a 5G link and a functional part of a 5G-capable industrial device. The 5G NW AAS models the most relevant functions of the 5G Radio and Core networks. The 5G NW and UE AASs follow a functional design where information is structured and organized by network functions or operations. The 5G UE and NW AASs could be used to connect to the physical assets in the context of a digital twin, or they could be used as well to connect to a digital model of a production plant to integrate 5G connectivity.

In this experiment, we consider an industrial pressing plant, divided in three main areas: a warehouse where the steel sheets are stored before production, a production area with three parallel press lines, and a shipping warehouse where the final product is stored, covered by a 5G network. In this scenario, a remote application located in the cloud accesses data generated in the industrial plant to take decisions and sends commands to the industrial devices in the plant. The data generated in the industrial plant is transmitted through the 5G network and Internet towards the remote application in the cloud node, and commands are transmitted by the remote application through Internet and the 5G network towards the industrial devices (see Figure 55). In this context, Data Containers will be used to facilitate the exchange of data and interoperability between different components and guarantee the data quality. The objective of this experiment is to analyse the impact of the latency experienced in the data transmission to/from the remote application on the plant operation and analyse how the use of the Data Containers enhances interoperability among different components.

To carry out this experiment, we use the digital models or digital twins of the industrial plant and the 5G network, respectively. The digital model of the industrial production plant is implemented using Visual Components and the 5G digital model is implemented using the 5G AAS implemented using BaSyx (the 5G AAS is described in deliverable D3.2). The integration of the digital twin of the industrial plant and the digital twin of the 5G network provides a valuable tool to evaluate how the performance of the communications can affect the operation and productivity of industrial processes. The digital model of the industrial plant and the digital model of the 5G network are connected using an OPC UA communication interface, which connects each 5G-capable industrial device in the plant with their respective UE in the 5G AAS. The OPC UA interface uses the client-server communication mode where the industrial and 5G digital twins act as clients and send requests to the OPC UA server. Visual Components provides a communication interface that enables direct communication between the industrial digital models and the OPC UA server. For the 5G digital model, the necessary Application Programming Interface (API) has been implemented so that the 5G AAS could interface with OPC UA and exchange information with the industrial digital model through the OPC UA server. The remote application deployed on the Cloud is integrated with the 5G AAS also using the OPC UA interface. The remote application also acts as a client. We use Data Containers to facilitate the exchange of data between the 5G AAS and the remote application. This Data Container obtains the 5G Digital Twin data from the OPC UA server using OPC UA subscriptions and makes it available for the remote application through its subscription API.

We demonstrate the performance and benefits of the integration of the digital twin of industrial models and 5G with external applications integrating also Data Containers to guarantee interoperability, security, traceability and data quality using an AGV-based use case. The use case focused on the data from AGVs that transport the materials from the warehouse to the production lines. These AGVs exchange their positions over 5G periodically to avoid collisions. These messages need to be transmitted within a time limit (considering a maximum time deadline or maximum latency) to guarantee the adequate operation of the AGVs. The 5G NW AAS estimates using the latency model presented in

[10],[11] the latency experienced by each packet in the transmission through the 5G network (considering the radio and core latency). The packet will also experience a transmission latency due to the transmission of the packets from/to the 5G network to/from the remote node through Internet. When the time expended from the last packet received by an AGV surpasses a threshold (due to high packet's latencies), the AGV pauses its movement to avoid collisions until the communication is restored with enough QoS.

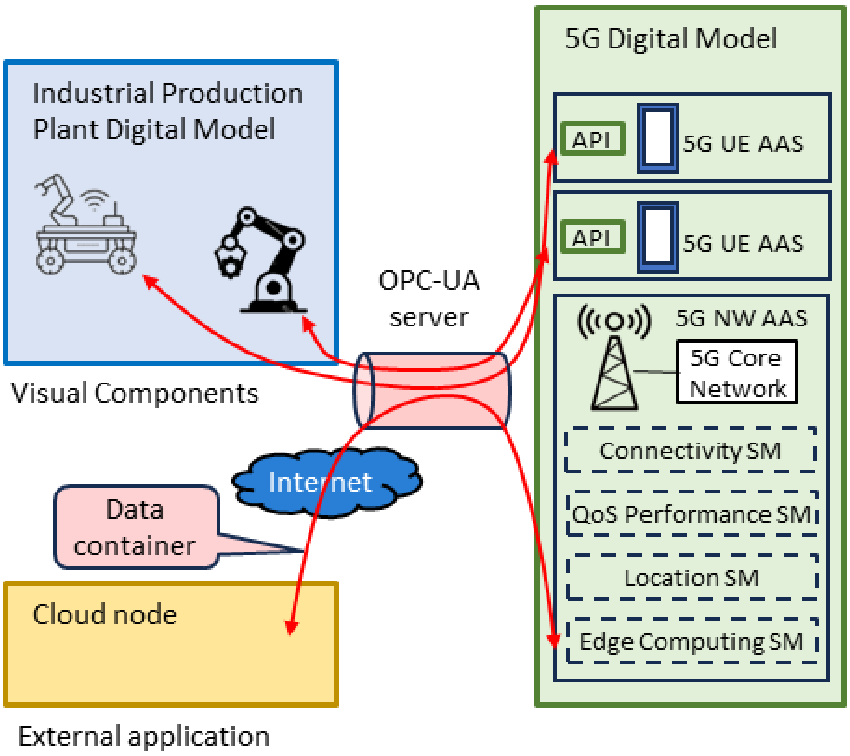


Figure 55: Integration of the DaaP and AAS/Digital Twin scenario

[10] *B. Coll-Perales, M.C. Lucas-Estañ, T. Shimizu, J. Gozalvez, T. Higuchi, S. Avedisov, O. Altintas, M. Sepulcre, "End-to-End V2X Latency Modeling and Analysis in 5G Networks", IEEE Transactions on Vehicular Technology, vol 72, no. 4, pp. 5094 - 5109, April 2023. DOI: 10.1109/TVT.2022.3224614*

[11] *M.C. Lucas-Estañ, B. Coll-Perales, T. Shimizu, J. Gozalvez, T. Higuchi, S. Avedisov, O. Altintas, M. Sepulcre, "An Analytical Latency Model and Evaluation of the Capacity of 5G NR to Support V2X Services using V2N2V Communications", IEEE Transactions on Vehicular Technology, vol. 72, issue 2, pp. 2293-2306, Feb. 2023.*

Components participation	• Data as Product Containers (DC) • Data Connection Profile (DCP) • 5G UE AAS and 5G NW AAS • OPC -UA server • Visual Components tool
Datasets/AI Models	Simulation data from Visual Components tool.
Proposed Added Value	• Interoperability guaranteeing interconnected ecosystems with data exchange among them, as well as the use of homogenised data formats and APIs to access the data. • Discovery: The AAS and digital twins will expose relevant data to other applications and the use of the Data Containers will facilitate the access and the use of the information without knowing the technical characteristics of the corresponding assets.
Quantified KPIs	• Number of DCPs delivered: 1. • Number of components connected using Data Containers: Successful integration of 3 components: the digital model of an industrial plant (using Visual Components software), the digital model of a 5G network (5G AAS), and an external application using the Data Container to ensure standardized data formats and APIs.
Partners Participation	UPV, UMH, VIS

4.4 Experiment 04: Scalable industrial IoT Solution

In this experiment, SIEMENS' Insights Hub platform is connected and used within the TEF of the Swiss Smart Factory. Several demonstrators of the Test – and Demo platform of the Swiss Smart Factory are connected to the Insights Hub platform. From the demonstrators, data regarding the machine status, environmental data and energy related data will be provided. The aim of this experiment is to monitor and analyze this gathered data in the Insights Hub platform.

With that objective in mind, this experiment will examine the subsequent scenarios:

- Asset monitoring dashboard(s)
- Energy Management of the demo factory

The experiment was successfully implemented at the SSF facilities, where it generated valuable historical datasets on environmental conditions and energy management. This data enabled real-time monitoring of various assets through the creation of dynamic, interactive dashboards, providing comprehensive insights into machine status and energy consumption. Additionally, the data served as a key input for the WP3 demo, designed to showcase the capabilities of WP3 tools in managing and analyzing industrial data.

One significant outcome of the experiment was the development of an API to automatically collect and process data from the demonstrators. This API ensures that future iterations of the demo can be fully automated, streamlining the data collection process and enhancing scalability for further experiments.

Overall, the experiment demonstrated the effectiveness of connecting industrial assets to a central platform for improved operational efficiency and data-driven optimization.

4.4.1 Scenario 01: Asset monitoring dashboard

The aim of the first Scenario is mainly to enable the monitoring of the environment (temperature, humidity, vibration and atmospheric pressure) in the TEF Swiss Smart Factory. By using the gathered data of the environment, optimization of the factory and improved business decisions can be made. Based on this, numerous dashboards and alerts will be implemented.

Components participation	• Insights Hub of SIEMENS • TEF Swiss Smart Factory
Datasets/AI Models	• Dataset with environmental data of the SSF TEF demonstrators of the drone factory
Proposed Added Value	• Enabling monitoring of environment in the factory • Creating alerts based on specified threshold values • Creating reports of measured data • Enabling optimization of the factory based on created reports. • Enable improved operational and business decisions with data-driven insights.
KPIs	• Number of Dashboards implemented: 10

	• Number of monitored stations: 7 connected stations with environmental data monitoring and dashboards • API access: 3 stations accessible through API for connection to other tools
Partners Participation	SIEMENS, SIPBB

4.4.2 Scenario 02: Energy Management of the demo factory

In the second scenario, an energy management of the TEF Swiss Smart Factory will be executed. The connected assets of the Swiss Smart Factory are providing specific energy related data which will be managed and monitored in the energy manager app of the Insight Hub platform. This will enable several added values like the identification of anomalies and the provision of a high transparency about the energy consumption.

Components participation	• Insights Hub of SIEMENS • TEF Swiss Smart Factory
Datasets/AI Models	• Energy related dataset of the SSF TEF demonstrators of the drone factory
Proposed Added Value	• Enabling power monitoring of the factory • Transparency about energy consumption over the factory • Enabling identification of anomalies • Optimization of the energy consumption • Batch and material related energy analyzation
KPIs	• Number of monitored stations: 7 connected stations • Emissions measured: Energy emission data are measured for each connected asset in kgCO2. • Number of reports generated: At least 1 report per connected asset per month is created based on the energy consumed and emissions generated • API access: 3 stations accessible through API for connection to other tools
Partners Participation	SIEMENS, SIPBB

4.5 Experiment 05: Comparative Analysis Between Centralized, Local, and Federated Machine Learning for Predictive Maintenance Tasks: Privacy, Performance, and Efficiency.

This experiment aims to compare centralized, local, and federated computing methods for AI-driven wear prediction, Remaining Useful Life (RUL) estimation, and anomaly detection of milling tools during and after operation. The goal is to provide valuable insights into the strengths and limitations of each approach, highlighting their respective advantages and trade-offs for predictive maintenance tasks. To achieve this, signals data, including vibrations, forces, and temperatures, will be utilized, along with essential insights into machining strategies such as axial infeed depth (ap), radial infeed depth (ae), tool geometry, and workpiece material. These factors will provide a comprehensive foundation for comparing the effectiveness of the aforementioned approaches.

This experiment will consider three scenarios:

- Use of local computing method: The AI models will be implemented on each edge device using localized data.
- Use of centralized computing method: In this scenario, a centralized AI model will be developed from data collected from all edge devices.
- Use of federated learning method: AI models will only aggregate parameters from all edge devices without transferring raw data.

4.5.1 Scenario 01: Local computing for AI applications implemented for each edge device.

In the local computing approach, each CNC machine (edge device) locally trains an AI model using only the data it generates. This method ensures that sensitive data remains on the device, preserving privacy but potentially sacrificing predictive accuracy due to limited data volume and diversity.

Components participation	• CORE: Docker Image Creation; Develop and package the AI model using Python and containerize it with Docker to ensure portability and consistency across devices.

	• CORE: Docker Image Sharing; Distribute the Docker image to stakeholders for local deployment and training on edge devices.
Datasets/AI Models	MODELS: AI model for Tool wear, RUL Predictions & Anomaly Detection. DATASETS: 1. Sensor Data and metadata for Milling Operations from My rConnect. 2. Labels from FRAISA
Proposed Added Value	• Privacy-First Approach: Data remains on the edge device, ensuring compliance with data privacy regulations. • Efficient Deployment: Dockerized model allows for standardized, portable, and consistent deployments across all devices.
Quantified KPIs	• Data Privacy: High (data stays on device) • Model Performance: *Wear Prediction Accuracy: 72%* *F1-Score: 0.70* *F1 Score: A metric that combines precision and recall to provide a balanced assessment of model performance, especially useful for imbalanced datasets. • Training Time: 5 mins
Partners Participation	CORE: Software for data manipulation and machine learning algorithm implementation. Docker Image creation and sharing with stakeholders. GF and FRAISA: Collaborating Stakeholders, providing required data and GF assists with deployment options

4.5.2 Scenario 02: Centralized Machine Learning Implementation

In the centralized approach, data from all CNC machines is collected and processed on a central server, where a unified AI model is trained using the combined dataset. This setup leverages data volume and diversity, which can improve model performance, but poses privacy concerns and requires substantial data transfer resources.

Components participation	• CORE: Centralized Model Training; Aggregate and process all collected data and train the AI model on a central server.

Datasets/AI Models	MODELS: AI model for Tool wear, RUL Predictions & Anomaly Detection. DATASETS: 1. Sensor Data and metadata for Milling Operations from My rConnect. 2. Labels from FRAISA
Proposed Added Value	Enhanced Performance: Training on a diverse, aggregated dataset enhances model robustness and generalizability. Scalable Resources: Centralized training can leverage scalable cloud or server resources, enabling complex model architectures and faster training.
Quantified KPIs	• Data Privacy: Low (centralized raw data) • Model Performance: *Wear Prediction Accuracy: 88%* *F1-Score: 0.85* • Training Time: 20 mins
Partners Participation	CORE: Manages data aggregation and centralized model training. GF and FRAISA: Collaborating Stakeholders, providing required data.

4.5.3 Scenario 03: Federated learning for machine learning applications by using flower framework.

The federated learning approach distributes model training across edge devices, using the Flower framework to coordinate the training process. Each device trains a local model on its data and shares only the model weights with the central aggregator, which constructs a global model without accessing raw data. This method balances privacy with predictive accuracy, benefiting from collaborative insights across devices. The process is as follows:

- Model Initialization: Each edge device receives an initial global model to begin training.
- Local Model Training: Each device trains the model locally using its data.
- Model Updates: Devices send their trained weights to the central server without exposing raw data.
- Aggregation Server: The central server aggregates weights from all devices, creating an updated global model.

- Model Redistribution: The updated global model is redistributed to edge devices, allowing for iterative improvement.

Components participation	• CORE: Federated Learning Framework Development; Utilize the Flower framework to enable federated learning, ensuring that only model weights, not raw data, are shared with the central server.
Datasets/AI Models	MODELS: AI model for Tool wear, RUL Predictions & Anomaly Detection. DATASETS: 1. Sensor Data and metadata for Milling Operations from My rConnect. 2. Labels from FRAISA
Proposed Added Value	• Data Privacy Compliance: Federated learning allows collaborative model improvement without transferring sensitive data, aligning with privacy regulations. • Improved Model Generalization: Federated learning benefits from diverse data across devices, offering better model performance compared to local-only training.
Quantified KPIs	• Data Privacy: High (only weights shared) • Model Performance: *Wear Prediction Accuracy: 86%* *F1-Score: 0.83* • Training Time: 10 mins
Partners Participation	CORE: Responsible for federated learning setup, including the Flower framework deployment and model aggregation GF and FRAISA: Collaborating Stakeholders, providing required data and GF assists with deployment options.

This analysis serves as an initial test, demonstrating promising results for each approach in wear prediction and remaining life estimation of milling tools. The federated learning test was conducted using a proof-of-concept setup, representing two client devices and a central server, where the available data was distributed between the clients.

In the final phase, more detailed results and analysis will be presented once Docker images and the federated learning framework have been fully deployed on actual edge devices. This deployment will enable a comprehensive evaluation of each approach's performance, privacy compliance, and efficiency in real-world operational settings. Additionally, the final analysis will incorporate a larger dataset with a wider variety of wear levels, further validating each model's robustness and adaptability to diverse wear patterns.

4.6 Experiment 06: Testing Predictive Quality capabilities by developing AI algorithms and use of analytics in a Federated Learning Framework

Near real-time predictive quality can bring great benefits in several manufacturing processes. The aim of this experiment is to explore the aspects of Predictive Quality involving EDM machines, and to facilitate the smoother implementation of the production process, by gaining insights into machines' operation and reduce unexpected interruptions. The experiment involves the development of AI tools that will first identify and detect potential anomalies. Interpretation of detected failures and correlation with real events will facilitate the strategic prevention of actions that lead to degraded quality.

Firstly, a connection with the data exchange ecosystem will be implemented for data acquisition and data analysis. This experimentation process will provide the first results and preliminary training of the model will take place. The results will be evaluated and validated during this phase. Afterwards, the Analysis Center, consisting of the trained model, will be integrated into the Federated Framework provided by ALIDA, to ensure safe and trusted data transfer. Connectivity and communication with ALIDA will be tested as well.

To that end, the following testing scenarios were chosen:

- Performance and Model Accuracy
- Deployment to ALIDA Federated Learning Framework and Robustness Evaluation

4.6.1 Scenario 01: Performance and Model Accuracy

In this first scenario, an analytical core that consists of trained models will be developed. The first step will consist of a preliminary exploratory analysis, where the data will be filtered, and different analysis techniques will be tested. Once the models are developed and trained, the evaluation stage will begin.

The trained models will be evaluated based of their ability to identify any occurred Quality defects. For both training and evaluation phase it will be necessary to obtain labelled data (either simulated or actual) with existing defects.

Components participation	ATLANTIS: Analysis Center AVIO AERO: EDM Machines
Datasets/AI Models	EDM data with artificially injected outliers and ML Models with defect Identification capabilities
Proposed Added Value	• Optimal Model Quality This stage is essential in the development of the solution since accurate models are the basis of an effective tool that can provide early detection of anomalies or compromised quality. The final optimized model will reduce unexpected quality defects by interpretation of production data. • Increased Trust Successful execution of this scenario contributes to increased trust in the tool's ability to deliver accurate models within a FL environment. Testing of the model's accuracy ensures that the complete solution can be useful to the end user by providing accurate identification results, while maintaining data security and interoperability.
Quantified KPIs	All artificially injected outliers are identified from the model. Anomaly Detection Rate (for artificial outliers): 95%
Partners Participation	ATLANTIS ENGINEERING SA, AVIO AERO

4.6.2 Scenario 02: Deployment to ALIDA Federated Learning Framework and Robustness evaluation

Under the scope of this second scenario, ALIDA platform will also be utilized. ALIDA is a platform for the composition, deployment and execution of Big Data Analytics (BDA) applications that also supports Federated Learning applications.

The Analysis Center that is developed and tested in the previous scenario, will be deployed in the ALIDA Framework. Connectivity of the Analysis Center with the framework will be assessed, and its performance as a complete solution will be evaluated in terms of identifying and subsequently preventing potential errors throughout the production process.

Components participation	ATLANTIS: Analysis Center ENGINEERING: Development of ALIDA
Datasets/AI Models	Datasets: EDM machine data AI models: AI algorithms providing fault/defect identification capabilities for the AVIO AERO's Shopfloor
Proposed Added Value	• *Increased awareness of the Production Line* Identification of known defects is expected to provide the end user with a holistic view of the production line and potential areas of investigation that will lead to more agile manufacturing processes. • *Optimization of product quality & automation of the quality inspection* The developed solution is expected to handle and exploit the available raw data efficiently, in order to provide the end users with useful insights about the Quality process. Visualization and reporting functionalities of the complete solution will encourage proactive actions that will benefit the overall product quality, by triggering preventive actions. • *Trusted Data Exchange* Placing the developed solution into a Federated Learning Framework offers a more trusted data exchange approach, thus encouraging data sharing in general. The Federated Learning approach ensures that only metadata is used for model training in order to increase security of the exchanged information. • *Federated Learning – Data sharing* The Analysis Center component is planned to be a part of the ALIDA Federated Learning Framework, where via trusted data sharing mechanisms the gained knowledge and be transferred, applied, and enriched with information originating from other sources, several machines, etc. As a result, the foreseen added value will be an optimized training for the model, which will be able to provide more accurate results.
Quantified KPIs	1. Successful integration with ALIDA and efficient data retrieval

	2. User feedback regarding user experience, notifications effectiveness and usability 3. Performance comparison between centralized and federated approach
Partners Participation	• AVIO AERO: Data providers, evaluation of results • ATLANTIS: Technology Provider of the Analysis Center • ENG: Provider of the ALIDA Federated Framework

4.7 Experiment 07: Testing potential benefits of predictive maintenance analytics application for tool performance optimization in a Federated Learning Framework

The primary objective of this experiment is to evaluate the efficiency of predictive analytics and the accuracy of Remaining Useful Life (RUL) estimation in optimizing tool performance. RUL estimation for milling components will rely on metrics derived from tool wear analysis. The degree of tool wear will be determined through surface analysis and by processing data such as time, position, acceleration, and vibration. Additionally, the execution of the aforementioned analysis will be performed within a federated learning framework.

Initially, the experiment will establish a connection with the data exchange ecosystem to facilitate data transfer and offline analysis. During this preliminary phase, foundational results will be produced, enabling early model training. These initial results will undergo thorough examination and analysis to inform subsequent steps..

In the later stages, the Analysis Core, comprising the trained RUL model, will be integrated into the FPdM (Federated Predictive Maintenance) framework. This custom framework, developed by ATLANTIS, is designed to develop and train machine learning models within secure and reliable data transfer procedures. Testing the connectivity and communication capabilities of the FPdM framework will be a critical aspect of these phases. To that end, the following scenarios were chosen:

- Federated Learning Framework Access Control
- Performance and Model Accuracy

- Robustness of the FPdM Platform

4.7.1 Scenario 01: Federated Learning Framework Access Control

The goal of this scenario is to validate the access control mechanisms of the FPdM framework, ensuring that only authorized entities can interact with specific models. This is critical for maintaining data privacy and platform security. Key objectives include establishing secure connections with data providers, ensuring user-friendly connectivity experiences, evaluating authentication protocols, and maintaining consistent access stability. Additionally, this scenario will assess the platform's ability to transmit model updates efficiently and ensure that there are not data leakages between the participants of the federation.

Components participation	FPdM Component
Datasets/AI Models	Labelled milling data from the GF/Fraisa Ecosystem
Proposed Added Value	• Trusted Data Exchange Placing the RUL solution into a Federated Learning Framework offers a more trusted data exchange approach, thus encouraging data sharing in general. By implementing a Federated Learning Framework only metadata is exchanged, thereby enhancing the security of the transmitted information.
Quantified KPIs	Data Leakages: Zero. Verified that train data remain on the federated clients and the federated server can only access model metadata.
Partners Participation	ATLANTIS ENGINEERING SA, GF MACHINE SOLUTIONS

4.7.2 Scenario 02: Performance and Model Accuracy

The Federated Learning Platform will incorporate supervised deep learning models, that will be trained on the data of interest. This scenario focuses on assessing the efficiency of these models and their ability to make accurate predictions across decentralized nodes, focused on the prediction of milling tool wear and Remaining Useful Life.

As an initial step, a preliminary exploratory analysis will be conducted to gain a thorough understanding of the available data and to plan subsequent actions by activating a range of algorithms. To enhance model robustness and address the limitations of the provided datasets—such as a small sample size or inconsistencies in the data—feature engineering will be employed to expand the feature set for training. This process will leverage both domain knowledge and advanced techniques to enrich the dataset with meaningful features. Additionally, domain expertise will guide the application of filtering and noise removal strategies to improve data quality by eliminating irrelevant or noisy information.

The trained models will be evaluated on their ability to estimate the Remaining Useful Life (RUL) of milling equipment. Since the lifecycle of a tool is inherently variable and cannot be expressed as a fixed value, the RUL will be determined by calculating the difference between the predicted tool wear and a predefined wear threshold established by domain experts. For this experiment, the threshold for the tool wear metric (*vb*) is set at 0.2. *Vb* values approaching 0.2 indicate severe tool wear, signifying the end of the tool's useful life.

To support both the training and evaluation phases, labelled historical data will be required. These datasets—whether simulated or real—should ideally reflect the complete deterioration process up to the end of the tool's lifecycle. Given the scarcity of such labelled data, clustering techniques will be employed to generate labels for unlabeled data points. Clustering will group tools and milling jobs with similar characteristics, and appropriate tool wear labels or values will then be assigned based on selected strategies. The aforementioned steps are depicted in the following Figure 56.

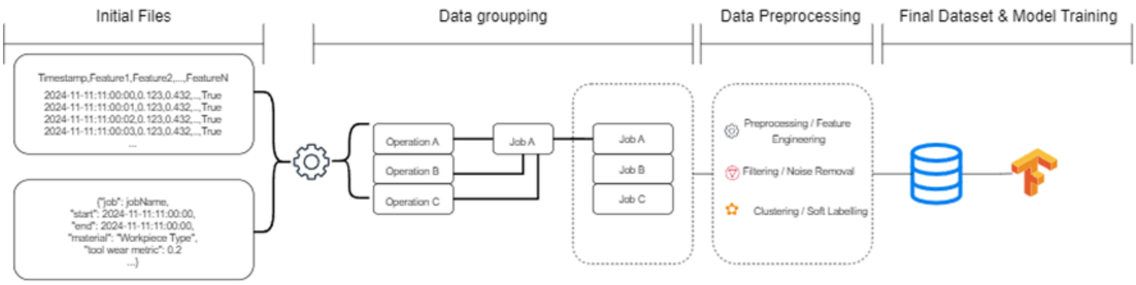


Figure 56: Stages of scenario 02

Components participation	Analysis Models FPdM (Federated Predictive Maintenance)
Datasets/AI Models	Analysis Models for RUL Milling data from the GF/Fraisa Ecosystem
Proposed Added Value	• Dataset Construction Build the final dataset using files provided by the GF/Fraisa Ecosystem. This dataset will combine pre-processed data, newly extracted features, and soft labels derived from clustering techniques, ensuring a comprehensive foundation for analysis. • Develop a RUL estimation model The estimation of Remaining Useful Life (RUL) typically requires extensive labelled datasets that cover complete lifecycles—a resource that is often scarce. Therefore, developing a robust model capable of accurately estimating the RUL of machinery components, even with limited labelled data, is of critical importance. • Ensure Optimal Model Quality Evaluate and refine the model to guarantee its efficiency and effectiveness. This stage focuses on fine-tuning to achieve high performance and reliability in predictions. • Increased Trust Successfully executing this scenario will enhance confidence in the model's capabilities, building trust in its reliability and utility for real-world applications.
Quantified KPIs	Relative Mean Squared Error of wear metric prediction = 0.076 Mean Absolute Error of wear metric prediction = 0.063
Partners Participation	ATLANTIS ENGINEERING SA, GF MACHINE SOLUTIONS, FRAISA

4.7.3 Scenario 03: Robustness of the FPdM Component

In this scenario, the RUL capabilities of the complete FPdM solution will be tested. The goal of this experiment is to reach a level of desired proactivity and optimization of the milling processes, by estimating the foreseen remaining life of crucial machine parts to avoid unexpected failures and interruptions across the production line.

In this scenario the developed model will be deployed in the custom FPdM platform providing a cohesive and complete result. The Flower Framework[12] will be used for that purpose.

Components participation	ATLANTIS: FPdM Component for predictive maintenance applications called GF/Fraisa milling machines
Datasets/AI Models	Datasets: Milling data from the GF/Fraisa Ecosystem. To simulate a federated scenario, the datasets were split and the clients were trained upon specific data batches and not on the whole dataset. AI models: Develop a model for Remaining Useful Life estimation of the involved machinery equipment
Proposed Added Value	• Proactive machine maintenance The development and application of RUL algorithm will provide accurate insights about the foreseen breakage point of machinery equipment involved in the milling process and therefore, encourage and assist proactive maintenance actions. This will help reduce downtime by preventing unexpected machine failures and provide a holistic view of the machinery's status. • Optimize waste management Once the foreseen breakage point/failure or defect has been identified, the end user can program the replacement of the involved equipment beforehand. In this way, the obsolete but not end-of-life part can be handled differently, in a more sustainable way (e.g. reuse, remanufacture, recycle, etc). The tool can support decisions about the post-use tools handling and timely replacement to improve cost management. • Federated Learning – Data sharing The RUL component is planned to be a part of the FPdM Federated Learning Framework, where via trusted data sharing mechanisms the gained knowledge will be transferred, applied, and enriched with

12 Flower: A Friendly Federated Learning Framework

<table>
<tr><td></td><td>information originating from other sources, several machines, etc. As a result, the foreseen added value will be an optimized training for the model, which will be able to provide more accurate results. The Federated Learning approach ensures efficient and secure data transfer and processing.</td></tr>
<tr><td>Quantified KPIs</td><td>Three aspects of the experiment need to be monitored:
1. Users' experience by using the FPdM – interfaces, security requirements, etc.
2. Federated approach's performance by comparing the calculated machine learning evaluation metrics between the centralized and the federated implementation. The evaluation metrics are aggregated using a weighted average technique across the participating clients.
<table>
<tr><th>Federation Round</th><th>Mean Absolute Error</th><th>Root Mean Squared Error</th></tr>
<tr><td>1</td><td>0.075</td><td>0.102</td></tr>
<tr><td>2</td><td>0.064</td><td>0.095</td></tr>
<tr><td>3</td><td>0.076</td><td>0.099</td></tr>
<tr><td>4</td><td>0.060</td><td>0.085</td></tr>
<tr><td>5</td><td>0.064</td><td>0.088</td></tr>
<tr><td>6</td><td>0.052</td><td>0.080</td></tr>
<tr><td>7</td><td>0.050</td><td>0.0788</td></tr>
</table>
3. Framework's scalability by evaluating the ability of the federated framework to support multiple client instances. Tested that the FPdM component can successfully support up to 4 federated learning clients.</td></tr>
<tr><td>Partners Participation</td><td>• GF/FRAISA: Data providers
• ATLANTIS: Technology Provider, (FPdM and AI algorithms)</td></tr>
</table>

4.8 Experiment 08: Ontologies, AI and Large Language Models Implementation Experimentation in Manufacturing

The use of ontology-based and Knowledge Graph-based solutions and solutions implementing AI and LLMs (Large Language Models) are rare and challenging to develop and implement in manufacturing industries. At the same time, these tools can significantly contribute to securing supply chains and internal logistics efficiency and security and allow manufacturing organisations to increase their resilience and innovative capacity.

The experiments described below aim to test and demonstrate several approaches for implementing supply chain/logistical/legal ontologies, Knowledge Graphs, AI- and LLM-based approaches for analysis and information gathering combined with new and innovative visualisation approaches. The different tools and approaches are expected to interact with each other during the experimentation.

To that end, this experiment will consider the following scenarios:

- Develop and document experiences from implementing an internal ontology system for a manufacturing process internal logistical system and develop a functional Knowledge Graph for analysis and visualisation.
- Develop and implement new, innovative visualisation approaches capable of handling multiple ontology interactions simultaneously – integration of legal & innovation ontologies into supply chain ontologies as an example.
- Evaluate, develop experiments, and document experiences and recommendations covering innovative use of LLMs when developing search mechanisms & eventual market systems for replacement products during manufacturing supply chain failures.
- Demonstrate interaction with external specialised API-based services covering patents and potential innovation markets, supported by ontology systems for innovations.

4.8.1 Scenario 01: The logistics Knowledge Graph – Development of manufacturing internal logistics ontology – IOF/UiO

In this scenario, existing supply chain ontologies will be evaluated and used as examples when creating an internal logistics system ontology. The resulting ontology will be used to create a Knowledge Graph repository and then used to test visualisation and analytical functionalities. The process will result in recommendations and documented experiences.

- An initial version of an internal logistics ontology system
- A Knowledge Graph and potential interaction with the visualisation tools developed in scenario 2 (visualisation tool).
- Documentation of experience and recommendations for future development and implementations.

Components participation	• Supply Chain Reference Ontology (SCRO) examples by the IOF Industrial Ontologies Foundry • Knowledge Graph system • RE4DY ontology visualisation environment
Datasets/AI Models	Shopfloor components (equipment and workstations) and relationships / dependencies - possibly subsection of production steps as example of where bottlenecks occur so that a KG can be evaluated
Proposed Added Value	• Improved understanding, security and analytical capabilities: Integration of ontologies and Knowledge Graphs increases the accuracy of analysis of big data instances and improves the quality of decisions.
Quantified KPIs	• Interim KPIs: draft version of the VW Knowledge Graph (early 2024) • Final KPIs (2025) One developed internal logistical ontology combined with documented experiences and recommendations.
Partners Participation	Lead: IOF/UiO; Partners: VW, Uninova, KU Leuven, Chalmers, ICF, SSF

4.8.2 Scenario 02: Legal Ontology of IP Rights – KU Leuven

In this scenario, a novel legal ontology of IP rights will be developed and deployed to assist in delineating and tracking legal rights that subsist in assets across the digital value chain. The use of such an ontology will contribute to efficient asset management by improving transparency with regard to legal rights and obligations. This in turn will engender trust in data sharing and create the preconditions for actors to contribute more of their background IP to the data sharing ecosystem.

The legal ontology of IP Rights is being developed using the Protégé open source software and will be made accessible via the RE4DY online repository.

Components participation	• RE4DY ontology visualisation system
Datasets/AI Models	• Novel Ontology
Proposed Added Value	• Improved understanding, traceability, and legal certainty wrt. digital asset rights. • Increased data sharing as a result of greater trust and transparency in the data sharing ecosystem.
Quantified KPIs	• Interim KPIs (draft version (early 2024)): One successful application of the legal ontology for the purposes of tracking IP across assets. • Final KPIs (2025): One developed legal ontology of IP Rights combined with a minimum of two successful applications of the same ontology.
Partners Participation	Lead: KU Leuven; Partners: IOF, Chalmers, ICF, SSF

4.8.3 Scenario 03: Dynamic interface – new and innovative ontology visualisation approaches – ICF

This scenario will further develop and implement innovative solutions developed in the first round of RE4DY dynamic interfaces that focused on the management of circularity, reference documentation and asset management in industrial value chains. These visualisation approaches allow for integration of ontology data from several ontologies in parallel and allow interaction with external systems managing third party data (such as dynamic and up-to-the-minute mapping of the patenting and business landscape with WIPO and EPO) that result in robust decision-making systems.

Examples of such in parallel managed ontology systems can be legal and IP management ontologies and also ontologies for resilient production.

Components participation	• RE4DY ontology visualisation system • Legal Ontology of IP Rights • Resilience Ontology (initial draft) • IP Screener API • Use of IP Screener for AI-assisted decision support
Datasets/AI Models	Ontology examples & management systems

Proposed Added Value	• Efficiency: More efficient management of complex, heterogeneous data and asset management in industrial settings. • Interoperability: Standardized formats covering heterogeneous complex data improve interoperability among manufacturing organisations, streamlining operations. • Quality and Consistency: The risks of missing vital data interconnections are minimised using Knowledge Graphs and ontology visualisation approaches to knowledge management in dynamic interface environments. • Cost/Risk Reduction: Lowering costs and risks resulting from mistakes by supporting unified standards for data checks and secure data transfer of complex, heterogeneous data sets.
Quantified KPIs	• Interim version: First integration of components including “at least *two* ontologies referenced”; at least *one* interactive environment prototyped • Final version: Successful implementation of several ontologies: implementation of several supporting ontologies in a manufacturing Knowledge Graph such as ontologies for logistics, legal assets and sustainability. • Demonstration of logistical flows visualisation: utilising the above data sets to present product and data flows along an internal logistics system. • Documented evaluations and recommendations: Documentation on the development process and recommendations for future implementation of Knowledge Graphs on complex, heterogeneous data structures and flows.
Partners Participation	Lead: ICF; Partners: IOF, UiO, KU Leuven, Chalmers, SSF, VW, Uninova

4.8.4 Scenario 04: Chalmers

Supply chains for complex and advanced products have been proven during COVID-19 to be very vulnerable to disruptions and cancellations, often disturbing complex and important production chains within essential industries. The search and procurement of replacement products are often complicated and risky, going beyond the challenges stemming from integrating the replacement products effectively into the production flows. Innovative use of LLMs in combination with specialised transformation processes of product data can eventually offer more effective and accurate search mechanisms to find replacement products. Scenario 4 aims to evaluate and test eventual approaches for creating innovative product data transformers and related LLM approaches to support effective replacement product market systems.

Components participation	• Research on the use of LLMs and AI mitigating supply chain challenges (Chalmers) • Use of AI / LLM in the context of IP Screener
Datasets/AI Models	Internal and external time series
Proposed Added Value	• More effective and accurate search for replacement products. • Search mechanisms can be adapted over time with terms covering sustainability, resilience and circularity. • Support the design and development of products and production processes to be more resilient.
Quantified KPIs	• Interim version: Determine the first iteration of a model and test on procurement scenario • Final version: Improved performance model and test on procurement scenario
Partners Participation	Lead: Chalmers; Partners: ICF, SSF, IOF, UiO, KU Leuven, VW, Uninova, AvioAero

4.8.5 Scenario 05: Asset management – modularised access to IPScreener patent and business landscape systems – ICF

Optimal access and use of internal and external innovations have become increasingly important for companies and organisations delivering complex and advanced products. Being able to quickly find out both competing products, solutions and ideas and actors with which one can collaborate with and license from will become a major factor for success. Another success factor is the ability to locate gaps in the current innovation and patent spaces, gaps that can form the foundation for new products and collaborative patterns. Scenario 4 aims at developing, test and implement first version of APIs and modular components that can demonstrate innovative new approaches for IP screening interoperability for end-users.

Components participation	• IPScreener APIs and tools • Interoperability with IP management ontologies • Interoperability with Legal ontologies • Solutions and approaches for developing custom tools
Datasets/AI Models	WIPO, EPO and worldwide patenting databases (normalized) accessed via NLP query and API
Proposed Added Value	• Optimised search with KG for knowledge-based interaction and narrowing down of options • Faster evaluation of the IP and patent business space within supply chains • Faster localisation of potential collaboration partners • Faster localisation of IP gaps in different markets and more effective exploitation of these gaps with new IP • AI-assisted workflow
Quantified KPIs	• Interim version: first iteration of AI-assisted decision support • Final version: automatic support to decision making workflows • Optimised search with KG for knowledge-based interaction and narrowing down of options
Partners Participation	Lead: ICF; Partners: IOF, UiO, KU Leuven, Chalmers, SSF, VW, Uninova, AvioAero

4.8.6 Scenario 06: Asset-centred & Integrating Pilot Platform – ICF

The pilot, developed by agile prototyping environments, aims to present an example of an integrated GUI system and examples of interconnectivity with different data and service sources, including ontology resource access. Amongst the different data sources are production data, ontology resources and knowledge/skills resources. Interconnectivity with external data sources such GAIA-X will be tested and evaluated.

Components participation	• Testing interoperability with existing and legacy systems • IPScreener APIs and tools • Interoperability with IP management ontologies • Interoperability with Legal ontologies • GUI tools & application server systems supporting web-based solutions.
Datasets/AI Models	Datasets from supply chains, access to web-based tools and ontologies, accessible across the Internet by URLs, JSON data exchange and if needed direct database access. Access to different external multimedia and other data and information.
Proposed Added Value	• Pilot of combined meta-data and portfolio manager for resources and services in production and supply chains. • Ability to dynamically add interoperability to future services and data.
Quantified KPIs	• Interim version: First pilot of user interface and core selection of information types/interoperability with external services and data sources. • Final version: Information types and flows managing ontologies – ontology management tools access and supporting resources. Interoperability with visualisation tools and processes.
Partners Participation	Lead: ICF; Partners: IOF, UiO, KU Leuven, Chalmers, SSF, VW, Uninova, AvioAero

5 Pan-European Resilient Data Space TEF

5.1 Pan-European Resilient Data Space TEF

To better illustrate the concept of a network among facilities centered around the sharing and use of Administration Asset Shells (AAS), we propose a demonstration leveraging the capabilities of the three TEF facilities involved in the RE4DY project. This demonstration is structured around a digital factory concept that simulates the product lifecycle of a drone circuit board from the SSF drone production line, with each TEF facility playing a critical role in different lifecycle stages. The circuit board will be represented by an AAS, which serves as the digital twin for the product, dynamically capturing and updating relevant data as the product moves through its lifecycle.

For the purpose of simplicity, we focus on the drone's circuit board as a subcomponent in this demonstration. However, this concept is scalable and can be extended to the entire product (the drone), offering a more comprehensive overview of product lifecycle management using AAS in a multi-facility network.

5.1.1 Overview of the Digital Factory and TEF Network:

The TEF network concept integrates the strengths of multiple facilities, demonstrating how collaboration across different expertise can optimize product management. In this demonstration, the lifecycle of the drone circuit board is broken into four key stages, each managed by a TEF facility. The AAS links the physical product with its digital representation, ensuring that updates occur seamlessly as the product moves through the different stages. This approach not only improves traceability but also facilitates real-time data sharing and decision-making across facilities.

Four Stages of the Circular Process:

Circuit Assembly at SSF:

- The lifecycle begins with the assembly of the circuit board at the SSF (Smart Factory Facility).
- At this stage, the circuit board is linked to its AAS, which contains critical information such as CAD files, technical specifications, and a unique identifier (nameplate).

- This linkage ensures that any changes to the physical product are tracked and updated in the AAS throughout the product's lifecycle.

Quality Control at Innovalia:

- The assembled circuit board is sent to Innovalia for detailed quality control.
- During this stage, the AAS is updated with quality control reports, including data on defects, testing results, and any adjustments required for the product.
- This ensures that the latest status of the product is reflected in the AAS and accessible to all stakeholders.

Disassembly at Polimi (if defective):

- If the quality control identifies defects, the circuit board is transferred to Polimi for disassembly.
- Here, the circuit board can be broken down into its components, and disassembly data (e.g., how components were dismantled, their condition) is added to the AAS.
- This stage is crucial in promoting circularity, as the components can be repurposed, reused, or recycled, with full transparency through the AAS.

Reintegration or Recycling at SSF:

- Once the disassembly is complete, the components are sent back to the SSF.
- At this point, the repurposed or recycled components are reintegrated into the production line, and the AAS is updated to reflect their new lifecycle status.
- This stage ensures the continuity of the circular process, enabling efficient recycling or repurposing of components while maintaining a complete digital history.

5.1.2 Administration Asset Shell (AAS) Description:

The AAS serves as the digital backbone of the lifecycle management process, providing a rich dataset for each circuit board throughout its various stages. The AAS for the circuit board contains the following key information:

- Technical files of the components: Detailed design files representing the structure and configuration of the circuit board.
- Technical specifications: Information on component properties, tolerances, and required performance metrics.
- Nameplate (Individual Identifier): A unique ID assigned to each circuit board for easy traceability throughout its lifecycle.

- Changelogs: A detailed history of modifications, updates, and lifecycle events, such as quality control results and disassembly information.

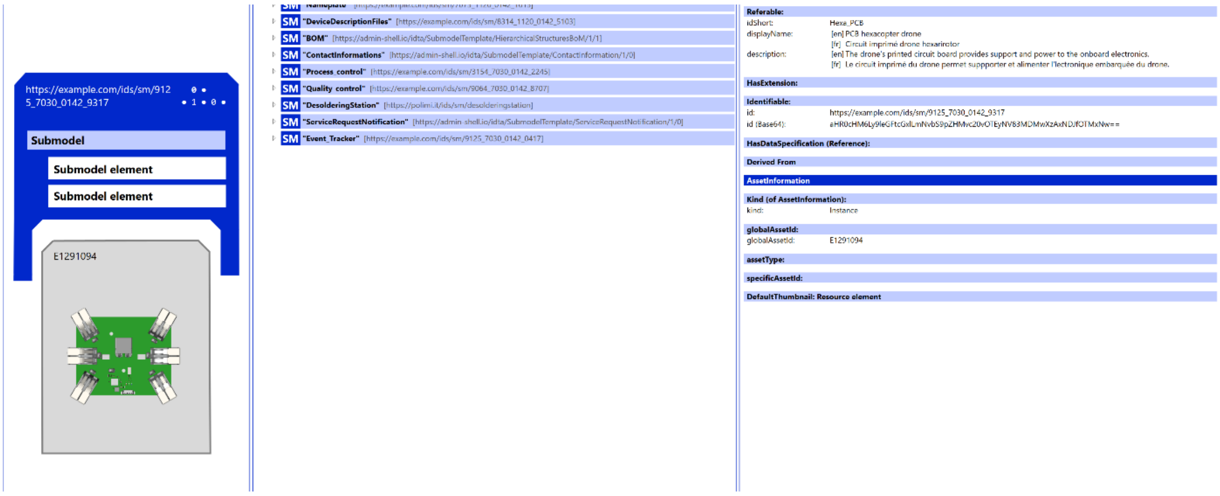


Figure 57: Planned structure for AAS

5.1.3 AAS Server:

The AAS is stored on a shared server accessible to all TEFs through data connectors and a shared data space, ensuring real-time updates and synchronization across the network. This server utilizes the AASX server as a base, leveraging Docker to facilitate deployment and integration. The AASX server acts as an access hub, allowing each facility to access and modify the latest version of the AAS and update it as the product moves through its lifecycle.

The shared server provides the following benefits:

- Real-time synchronization: Ensures all TEFs are working with the most up-to-date version of the AAS, minimizing errors and discrepancies.
- Lifecycle tracking: Facilitates seamless tracking of product changes, ensuring that every stakeholder has access to the latest data.
- Scalability: The server can handle multiple AAS instances, allowing for the management of various products and components simultaneously.

The AAS is stored on a shared server accessible to all TEFs, ensuring real-time updates and synchronization across the network. This server interacts with the RE4DY Data Space, a secure and scalable platform for data sharing and collaboration between the TEFs. The Data Space enables the seamless exchange of critical product information, including AAS data, quality control reports, and maintenance logs.

By the end of the project, we aim to further integrate the AAS server with the RE4DY Data Space and other tools from the toolkit, such as Data Containers. This will allow for a more seamless and efficient data exchange between the TEFs, enabling advanced analytics and optimization of the manufacturing processes.

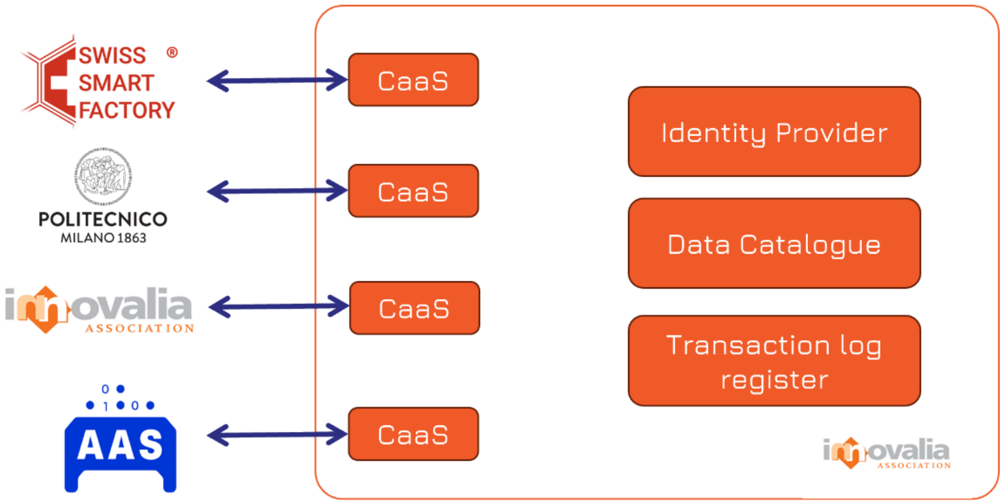


Figure 58: TEF Data space

5.1.4 Conclusion of overview:

This demonstration showcases how the integration of Administration Asset Shells within a network of TEF facilities can optimize lifecycle management for complex products. By tracking the lifecycle of a drone circuit board across multiple stages, from assembly and quality control to disassembly and reintegration, we highlight the potential for improved traceability, circularity, and collaboration in manufacturing processes. This approach not only enhances efficiency but also supports sustainable practices by promoting the reuse and recycling of components through real-time data sharing.

5.2 TEF Specific tasks:

5.2.1 SSF:

The SSF production line is designed for the manufacture of quadro- and hexacopter drones in batch size 1. The part that concerns this demonstration is around the assembly and soldering of the PCB component of the drone which is handled by the THT station.

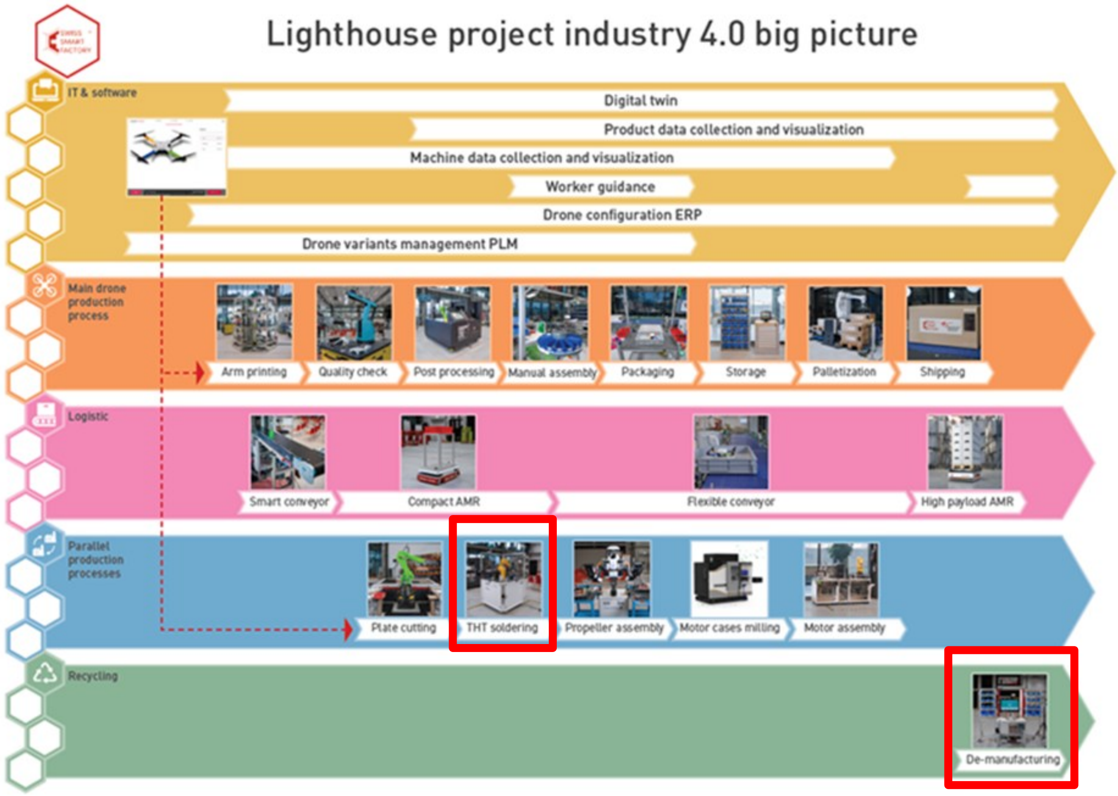


Figure 59: SSF Drone production line

The THT soldering station description:

Introduction:

Soldering THT components onto PCBs is typically done in two ways: fully automated machines for high volumes (over 10,000 units) or manual soldering for low volumes (under 10,000 units). Producing intermediate quantities (200 to 10,000 units) is currently not economically viable in Europe. While manual soldering suffers from a lack of experienced workers and higher production costs, fully automated soldering processes require significant financial investment in machinery and is only feasible with rigid production lines.

This station targets a gap in the current manufacturing capabilities related to Through-Hole Technology (THT) soldering processes. The automation through AI allows automated component picking, generation of part placing strategies and soldering instructions.

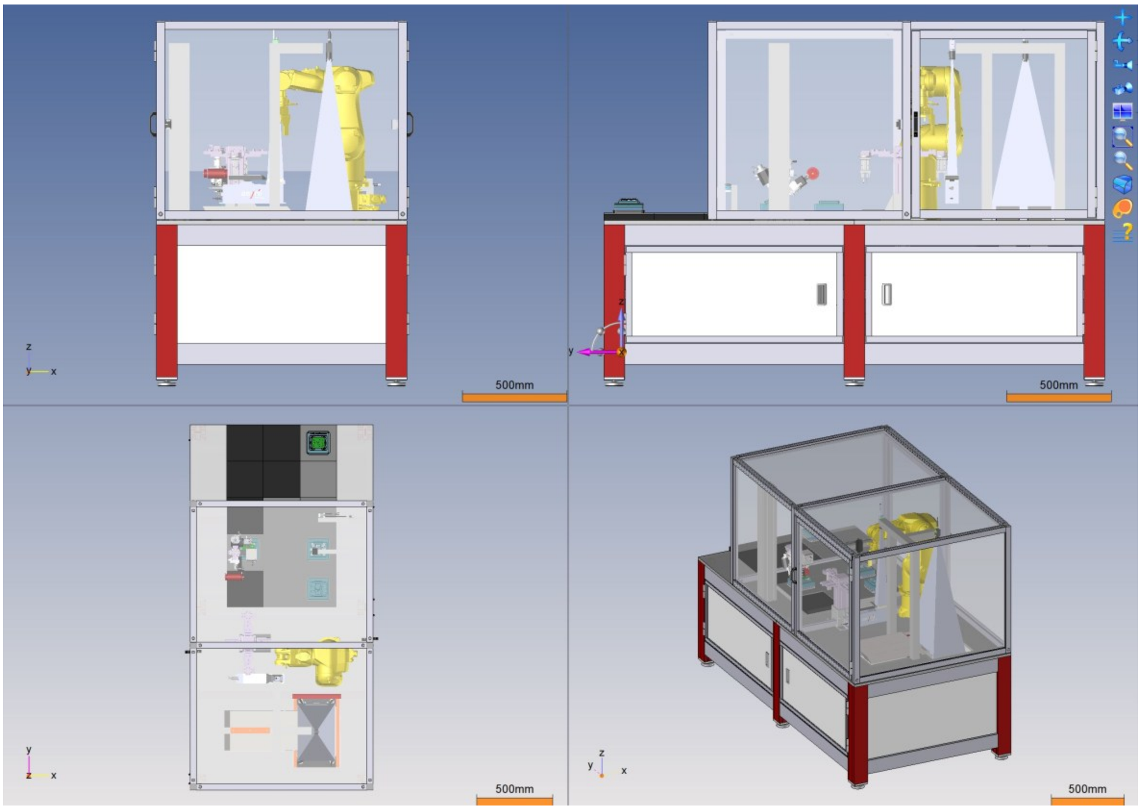


Figure 60: 3D Model of THT station

The automated THT soldering process can be divided into the following steps:

1. Feeding the components
2. Identifying the components for pickup and generating a gripping strategy
3. Gripping and placing the components
4. Verifying successful placement of the components
5. Flipping the PCB board to position the THT legs upward
6. Generating the soldering strategy
7. Executing the soldering process using the recommended parameters
8. Monitoring and verifying the success of the process

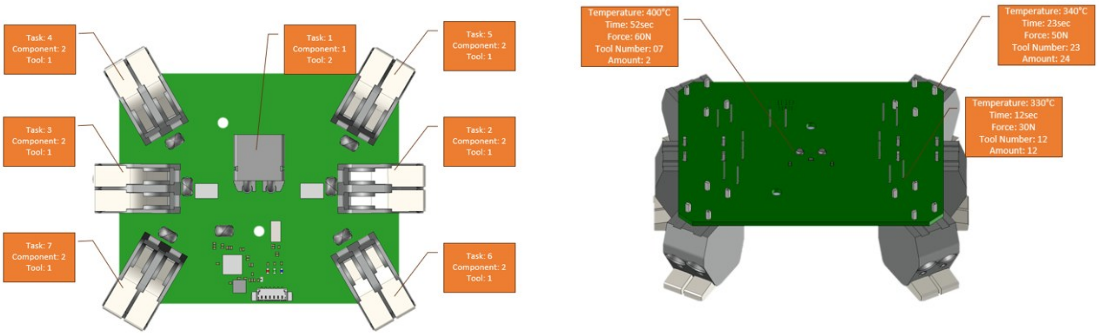


Figure 61: Placement of the components to be soldered and soldering parameters generation

Interest of the TEF network demo for the use case at SSF:

As we aim to automate the typically manual task of soldering components, in low-to-mid batch size production, the digital twin becomes essential in this scenario. By representing and storing the product's data—including process control, batch number, temperature profiles, and other production parameters—within the AAS, we ensure that all aspects of the process are captured and tracked in real-time, without the need for direct operator supervision.

We expect the following benefits from the implementation of the AAS in our process:

- The digital twin allows for continuous monitoring of the soldering process, capturing critical data points such as soldering temperatures, component placements, and cycle times. This real-time data can be compared with ideal process parameters, allowing for immediate adjustments or alarms if deviations are detected, thus reducing the likelihood of defects.
- With a complete history of each production cycle stored in the digital twin, the system can track patterns over time—such as recurring defects or inefficiencies in the soldering process. By using this data, the model can learn and refine the process, making intelligent adjustments to optimize performance and reduce errors.
- The digital twin provides a long-term view of the product's life cycle. It enables tracking of how well each PCB performs once deployed (the product life cycle),

gathering data on failures or issues over time. This feedback can be looped back into the manufacturing process, continuously improving production parameters.

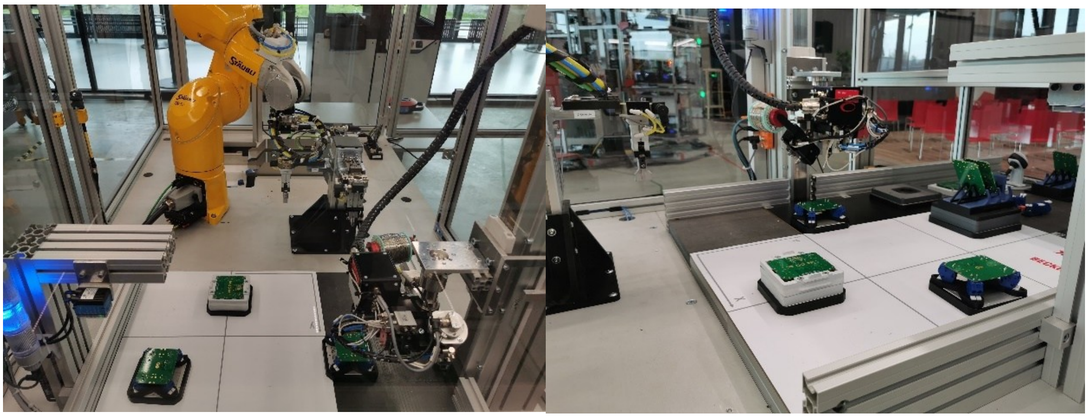


Figure 62: Soldering area of THT station

AAS - Product and process control information:

Since our facility is responsible for the initial production of the PCB, the main subsections of the AAS we plan to populate are the following:

1. Technical files of the part: Detailed design files of the circuit board.
2. Technical specifications: Information on component properties.
3. Nameplate: A unique ID for each circuit board and manufacturer information.
4. Process control data:
    a. Batch number – To track which production batch the process belongs to.
    b. Date – The date when the process was completed.
    c. Cycle time – The total time taken to complete the soldering process.
    d. Machine ID – Identifies which machine performed the process.
    e. Error/Warning Logs – Any errors, warnings, or interruptions during the soldering process.
    f. Component Serial Numbers – For specific traceability of critical components.
    g. X/Y Positioning Accuracy – Data on the precision of component placement.
    h. Energy Consumption – Amount of energy used by the machine during the process.

5.2.2 Polimi:

Within RE4DY project, POLIMI relies on its INDUSTRY4.0LAB infrastructure as its Testing and Experimentation Facility. Given the facilities of this laboratory and within the scope of this demonstration, a semi-automated set-up has been settled in order to perform the desoldering task of potential defects present on the drone printed circuit boards. In the proposed set up, the UR5e cobot equipped with a hot air nozzle has been exploited in order to desolder the defective points on the board while an operator takes control of the removal of the desoldered component from it.

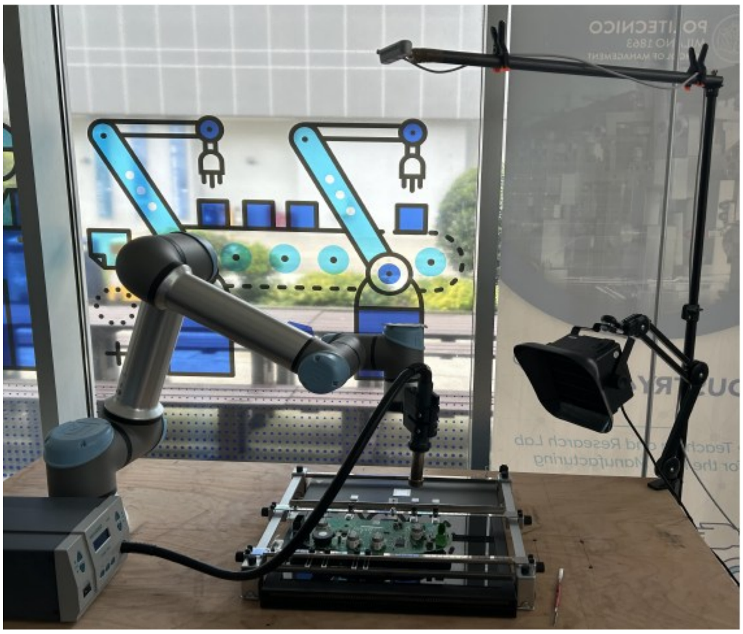


Figure 63: The Desoldering set-up

Given the opportunities offered by AAS technology and following the structure provided by the SSF as the manufacturer, a dedicated AAS package has been developed to represent the operational data of the desoldering station. The overall view of the AAS package is depicted in Figure 64 where 8 SubModelCollection have been inserted in order to satisfy the required information for execution of the desoldering task.

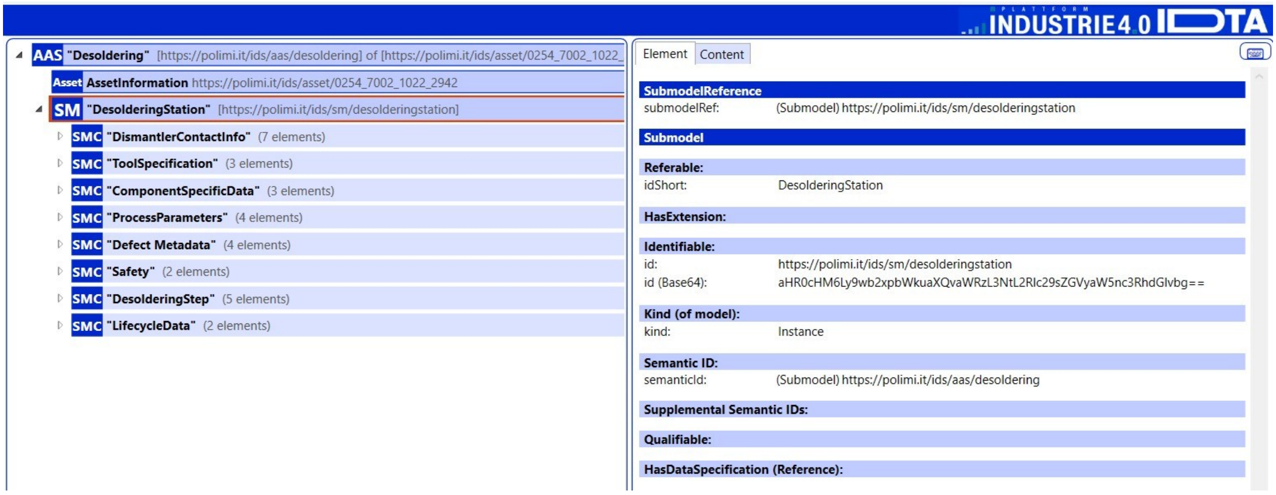


Figure 64: Overall view of desoldering AAS and its SubModel

Figure 65 indicates mainly the static data related to the dismantler, heating tool specification and component specific data. Under each SubModelCollection, several properties with ConceptDescription and SemanticIDs have been inserted to cover all the characteristics of the interest.

Figure 65: Dismantler, Tool and Component collections

Figure 66 represents the parameters and variables related to the desoldering process and defect while the safety SubModelCollection mentions general required safety measures for the desoldering task. The expected defects can be the defective soldering points while assembling the main power supply connectors (J4-J9) on the PCB drone which the coordinates of these points are included in the "Defect Metadata" SubModelCollection.

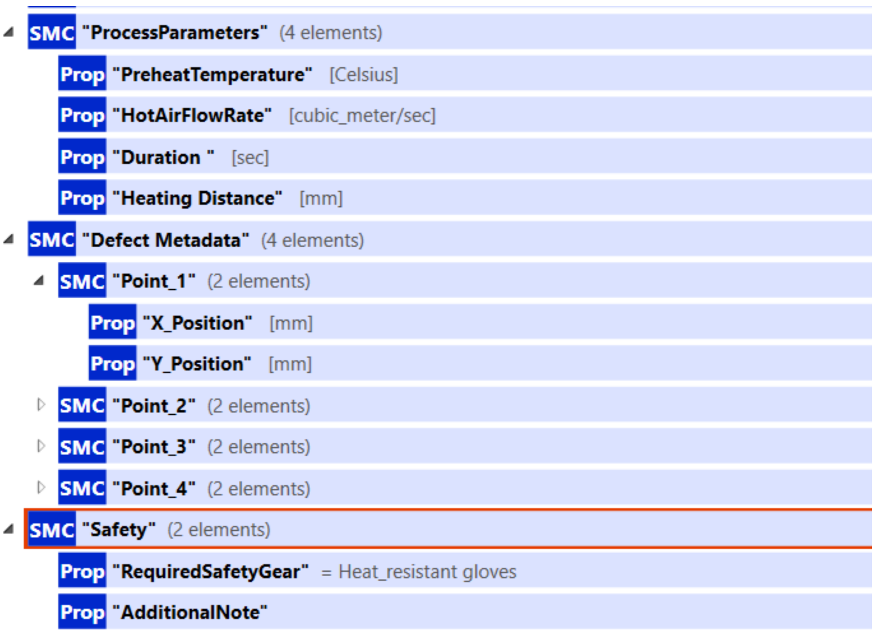


Figure 66: Process, Defect and Safety collections

Figure 67 depicts the Desoldering state and LifeCycle data of the defected component. Boolean properties have been identified to trace the workflow of the component through the desoldering steps while a field is dedicated to record the condition of the component after desoldering to potentially re-use it in the manufacturing line.

Figure 67: Desoldering step and LifeCycle collections

5.2.3 Innovalia:

Innovalia's role within the RE4DY TEF network is to provide advanced quality control and inspection services. By leveraging advanced inspection techniques and AI-powered analysis, Innovalia will ensure the quality and reliability of the produced components.

Key Contributions:

- Quality Control and Inspection: Implementing automated inspection systems to identify defects and anomalies in the components.

- Data Collection and Analysis: Collecting and analysing data from the inspection process to identify trends and improve future production.
- Data Sharing and Collaboration: Actively participating in the Data Space, sharing inspection data and collaborating with other TEFs to optimize the overall manufacturing process.

Quality Control Process

1. Initial Inspection and Measurement Planning:
   - CAD Analysis: A thorough analysis of the CAD model is conducted to identify critical dimensions, tolerances, and geometric features.
   - Measurement Plan Development: In collaboration with the customer, a detailed measurement plan is created, specifying the precise measurements to be taken and the appropriate measurement techniques.
2. Measurement Execution:
   - Part Setup: The component is carefully secured on the Coordinate Measuring Machine (CMM) using appropriate fixtures.
   - Program Execution: The pre-defined measurement program is loaded into the CMM, and the machine automatically performs the required measurements.
   - Data Acquisition: The CMM captures the measured data, generating a point cloud or a set of discrete points.
3. Data Processing and Analysis:
   - QIF File Generation: The measurement data is processed and exported in the Quality Information Framework (QIF) format. This standardized format allows for efficient data exchange and analysis.
   - Quality Report Generation: A detailed quality report is generated, summarizing the measurement results, comparing them to the specified tolerances, and indicating whether the component meets the quality requirements.

Interest of the TEF network demo for the use case at Innovalia:

By leveraging the TEF network and the AAS, Innovalia aims to enhance its quality control processes and contribute to the overall efficiency and sustainability of the manufacturing supply chain.

Key benefits of the TEF network for Innovalia:

- Enhanced Quality Control: By sharing and analysing data from multiple TEFs, Innovalia can identify common quality issues and implement preventive measures.
- Improved Process Optimization: The real-time data exchange and analysis capabilities of the Data Space enable Innovalia to optimize its inspection processes and reduce cycle times.

- Increased Traceability: The AAS provides a comprehensive digital record of each component, facilitating traceability and accountability throughout the supply chain.
- Enhanced Collaboration: The TEF network fosters collaboration between different stakeholders, enabling knowledge sharing and the development of innovative solutions.

AAS - Product and process control information:

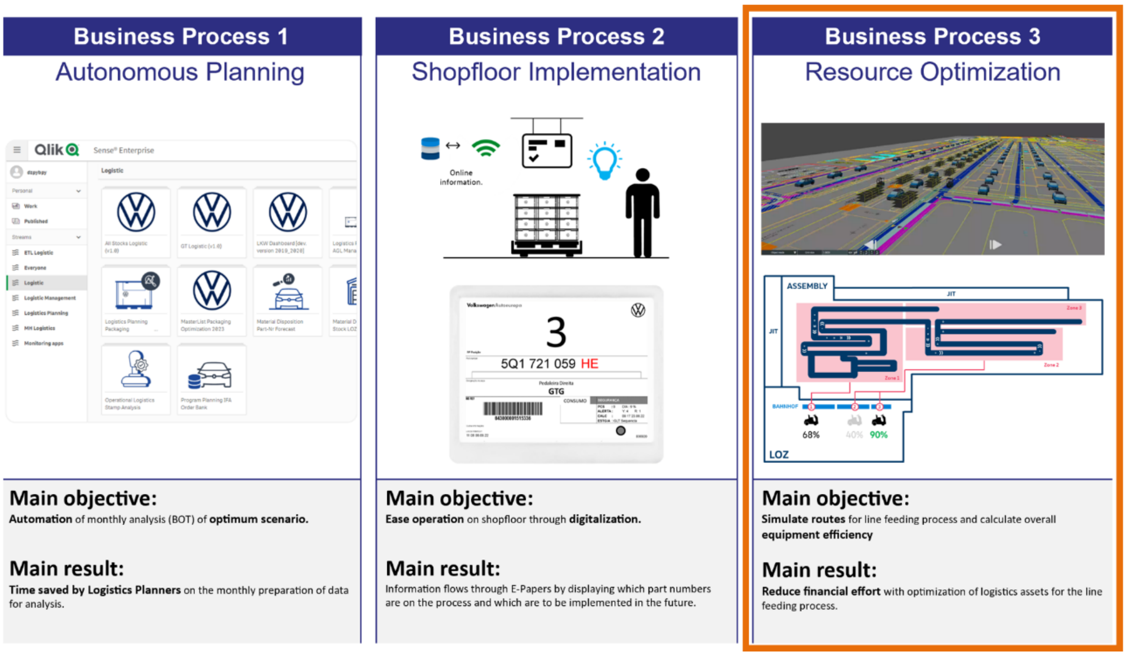
The AAS for the quality control process will contain the following key information:

1. Quality control data:
    a. Date – The date when the process was completed.
    b. Machine ID – Identifies which machine performed the process.
    c. Measurement Programme – The reference of the program that has been used to measure the part
    d. Measurement Results (QIF) – QIF File with the results of the process
    e. Results Report – PDF document with the Results Report.VW use case:

This use case is independent of the TEF network demonstration presented above. It was conceived and realized from the collaboration of various partners in the Extended task force as well as the VW pilot. Since it was related to TEF activities and that an event was conducted at the SSF TEF, it was decided to report the reports within this deliverable.

5.2.4 Experimentation Plan derived from VWAE Pilot:

These following sections outline the experimentation plan for the RE4DY Test and Experimentation Facility (TEF) at SSF, which is created based on the specifications and needs of the RE4DY-Pilot at VW. The objective of this experimentation was to set up a test case that replicates a specific business case of the VW-Pilot to analyze the optimization potential.



Main goals of the experiment:

1. Create a satisfying replica of the GT process (described in VWAE experiment plan) in the TEF at SSF.
2. Run the experiment in the TEF of SSF with the goal to analyse the asset optimization on the line feeding process:
    a. With randomized events that create deviations.
    b. Detect and alert when asset enters unexpected area of operation.
    c. Measure time of each cycle.
3. Evaluate the experiment regarding the asset optimization potential in the GT process.

Simulation:

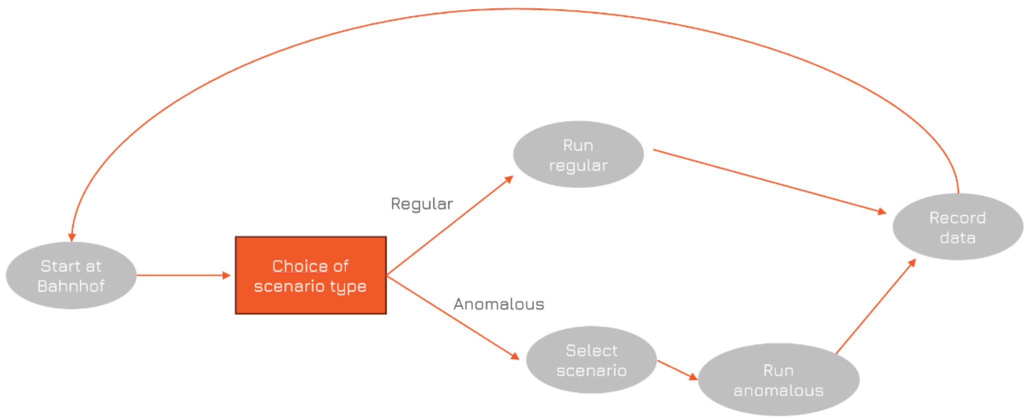
The AGV at SSF, simulates a human operated tugger train. The tugger train follows a scenario or mission which is composed of a certain route with stops and events.

At the end of each mission, the AGV returns to the Bahnhof. At this point, the simulation runs a randomization algorithm to determine the next mission, which can be either a "regular" or an "anomalous" mission. A regular mission has a probability p of occurring, while an anomalous event has a probability 1-p, which is lower than that of a regular operation. This probability is currently set arbitrarily for visibility during experiments but can be adjusted to reflect Volkswagen's actual data.

If a regular event is selected, the AGV follows the standard route. If an anomalous event is chosen, a second randomization algorithm selects from a set of 10 anomalous missions. These anomalies fall into three categories:

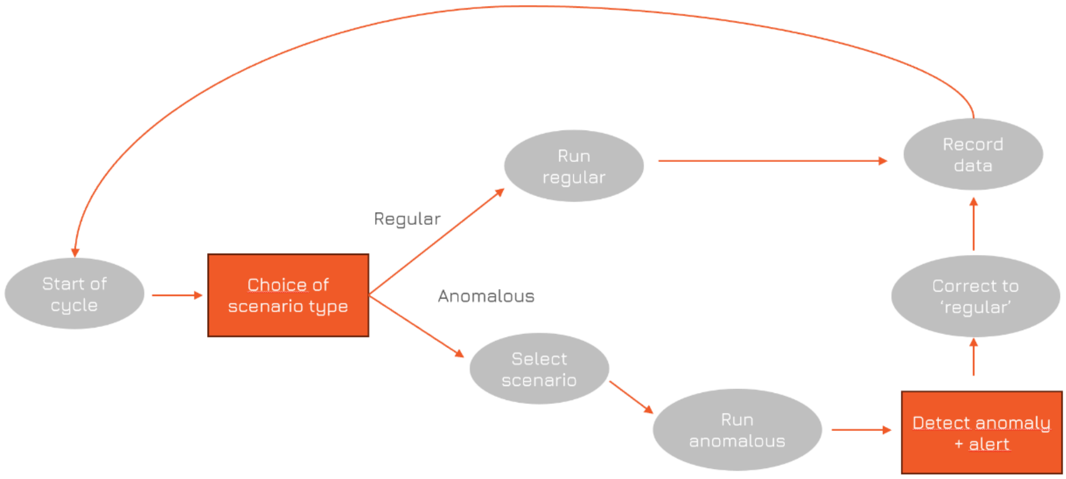
1. Unexpected Area of Operation: The AGV enters a "forbidden" area, simulating an operator moving into an area where the tugger train is not supposed to operate.
2. Unexpected Route Duration: The AGV stops at a station longer than expected, simulating an operator taking too long to unload parts or encountering an obstacle.
3. Unexpected Route: The AGV visits stations in the incorrect order, simulating an operator not following the proper schedule.

The basic cycle of the simulation can be seen in the graphic below:



The goal is to quantify the "loss" produced by these anomalies over a specific period. While our current simulation involves only one asset, this methodology can be applied to any number of assets. Quantifying this loss will help us measure the inefficiency of a particular logistic system and determine the associated costs.

The next step was to implement strategies to eliminate or reduce these anomalies (e.g., sounding an alert when the asset enters an unexpected area), measure the improvement in efficiency, and evaluate the logistics resource needs of the improved system.



Components:

AGV:

- The agv is exquiped with containers which can be filled with various parts at the Bahnhof
- It has a tag which is tracked by the RTLS system which locates it in the facilty.
- The AGV is connected to the facility network by wifi.
- The AGV has a graphical interface which can be accessed through an endpoint on the network. This is where one can design the map and the missions. The main functions are: Create destinations, moving actions, create preferered areas for routes (not a fixed route, AGV can adapt and avoid obstacles within this area), sound/light outputs and other.
- The AGV also an API which allows to send request from node-red (mostly GET and POST requests)

Figure 68 AGV with part holder and "Bahnhof"

RTLS:

- The RTLS system is only used to get the x, y postions and the zone name (area in the facility created in the SICK graphical interface (seen below).
- It is done through a http GET request in the API.
- The system is composed of "tags" which are fixed to the moving assets we track and "anchors" which are the receiving antennas located around the faciltiy.

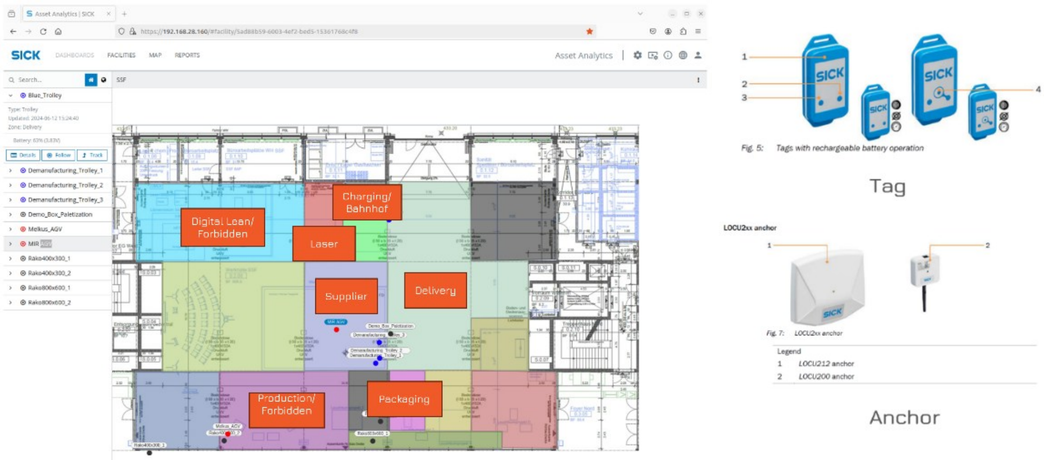


Figure 69: RTLS graphical interface and hardware

Node-red

- Functions are based on Javascript: it generates the requests to the APIs, selects the scenarios and records the data.
- Node-red does two main requests to the MIR:
  - It gets the queue of missions to verify if the previous mission is finished
  - If it is, it selects the following random mission and sends a POST request to launch the next mission
- Node-red requests through a GET the position of the AGV at a fixed interval from the Sick API. For demo purposes we also record the time spent in each zone and display it in a graphical dashboard.
- The recording of the data is currently done into a local CSV on the server which runs the node-red simulation. It pulls the x,y postion (of the AGV (the tag on it) from the SICK API, adds a timestamp from an internal timer, and associates the zone code to a zone name.
- Generated data:
  - X,Y: float
  - Timestamp, zone: string.
  - Request to APIs: JSON
  - Measurements which can be implemented: Time spent in a certain zone, route duration, occurrence rate of anomalies.

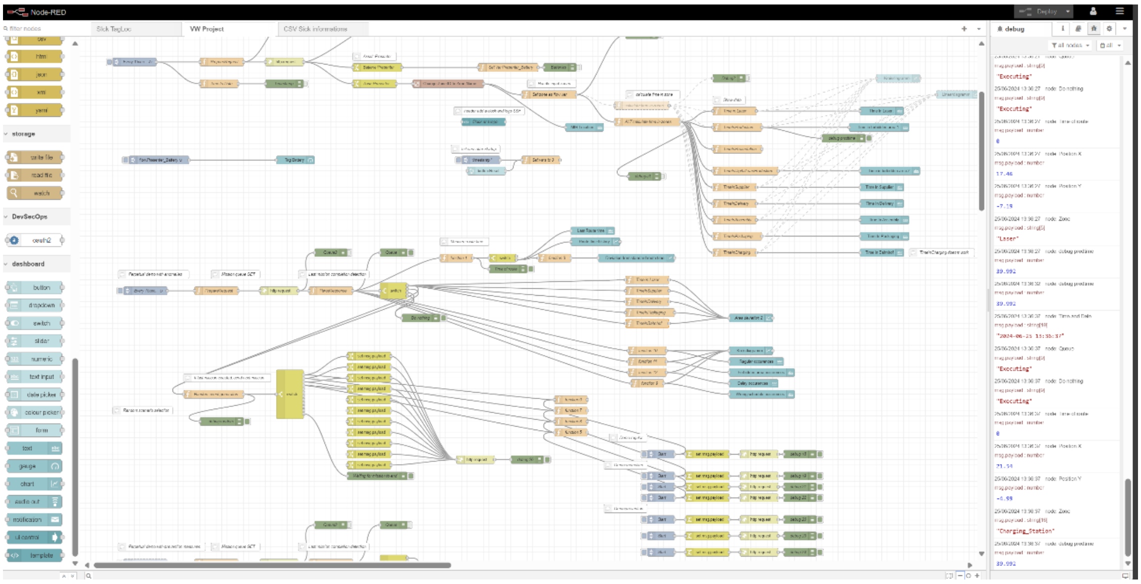


Figure 70: Node red flows

Results

As a reminder, the main anomalies we measured are the following:

1. Unexpected Area of Operation
2. Unexpected Route Duration.
3. Unexpected Route.

The simulation runs for uninterrupted, going through regular scenarios and anomoulous scenarios and at any time it is possible to see those deviations in the Dashboard. Not only does it allow to detect anomalies but to locate them to specific areas. Those results aggregated allow to quantify the inefficiency of the logistics system. In our case this should roughly correspond to the rate of anomalous scenarios we decide to inject.

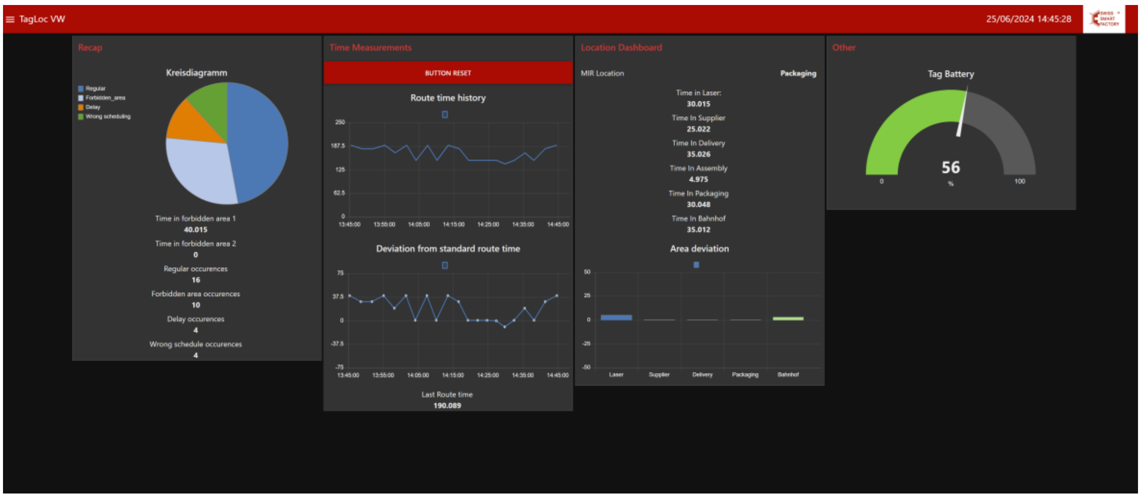


Figure 71: Line Feeding asset monitoring dashboard

The second phase involved implementing corrective measures and evaluating their impact on efficiency. Our initial approach was to develop a notification system that alerts the operator on-screen whenever a deviation from the planned route is detected. So far, we have only been able to simulate the operator's response. To obtain more realistic and reliable results, further testing with actual operators is necessary.

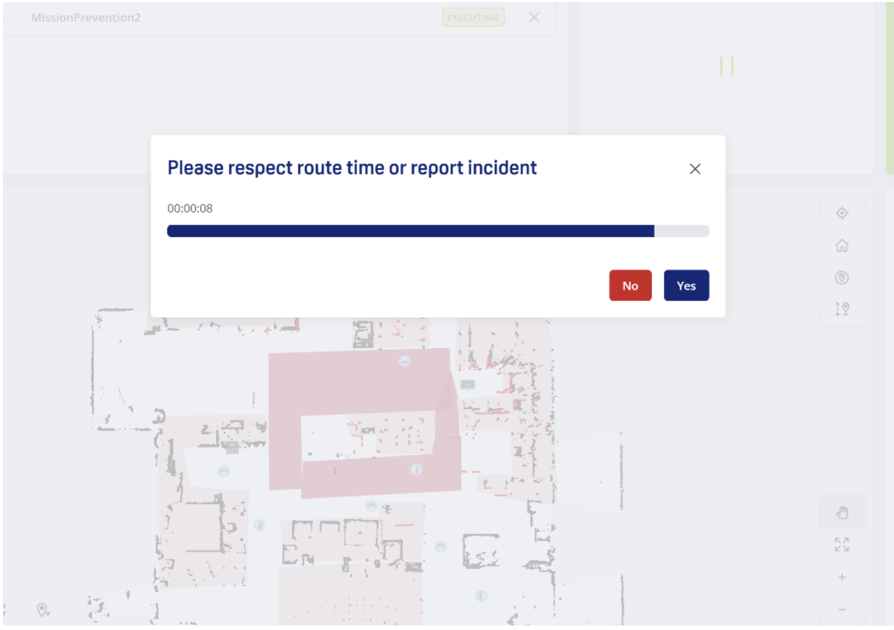


Figure 72: Corrective measures

5.3 Conclusion and next steps

In conclusion, we successfully developed an initial simulation of a line-feeding asset that incorporates unpredictable behaviour. This allowed us to measure the inefficiencies caused by these anomalies and represent the results both visually in a dashboard and as a dataset.

For the next steps, improvements could include increasing the number of assets in the simulation to evaluate the usefulness of aggregated information on a larger scale. Ultimately, implementing the simulation in a real-world scenario would enable us to test the effectiveness of the corrective measures we have designed.

6 Conclusion

Deliverable D3.3 builds upon the foundation established in D3.2, advancing the Digital Continuum 4.0 Toolkit by introducing upgrades to existing tools and integrating new ones, thereby broadening its functionality and capabilities. These enhancements reflect the iterative development process within WP3 and the ongoing effort to addressing challenges in interoperability, scalability, and industrial resilience.

The inclusion of detailed technical descriptions of the upgraded and newly introduced tools provides a comprehensive understanding of their functionalities, inputs, outputs, and integration points. This serves as a valuable resource for project partners, facilitating the seamless integration of these tools into the RE4DY Reference Architecture and their application across various industrial contexts.

The execution of the experimentation plans, which were initially outlined in D3.2, validate the integrated tools. This provided a qualification of the toolkit on the network of TEFs and pilot sites, as well as quantified KPIs as measurable evidence of their performance.

In conclusion, D3.3 represents a significant step forward in the qualification of the Digital Continuum 4.0 Toolkit, ensuring its readiness for large-scale trials under WP4 and WP5. By combining detailed technical documentation with robust experimentation frameworks, this deliverable provides critical insights into the performance, integration, and practical utility of the toolkit, further strengthening RE4DY's vision of enabling resilient and interoperable industrial ecosystems across Europe and ensuring its capacity to address complex challenges in industrial digitalization.

7 References

RE4DY D3.2 First generation digital continuum 4.0 open toolkit