



Flexible **R**emufacturing using AI and Adv**a**nced Robotics for Circular Val**u**e Chains in **E**U Industry

Grant agreement n°: 101138415

Call identifier: HORIZON-CL4-2023-TWIN-TRANSITION-01-04
Factory-level and value chain approaches for remanufacturing
(Made in Europe Partnership) (IA)

D4.1: Digital infrastructure for remanufacturing management – Initial prototype

Deliverable nature: Demonstration

Dissemination level: Public

Delivery date: 27/06/2025

Version: V2.0

Lead Author: INESC TEC

Contributors: INTRA, LMS, IIT, STAM

Reviewer: CEA, DTI



Funded by
the European Union

This project has received funding from the European Union's Horizon Europe Framework Programme under grant agreement n° 101138415.



Executive Summary

This deliverable D4.1 presents the initial prototype of the RENÉE digital infrastructure for remanufacturing management, consolidating the development and configuration of its core technological components. In particular, this document:

- Defines the architecture and interconnection of four main modules: the Process Scheduler and Orchestrator (PSO), the Digital Twin (DT), the Digital Vault (DV), and the Supplier Network Management Tool (SNMT).
- Describes the data exchange protocols and communication mechanisms that ensure interoperability, scalability, and secure information flow among modules.
- Introduces the RENÉE Integration Platform that provides user access and orchestration of modules via graphical user interfaces and presents the initial implementation of containerised deployment using technologies such as Docker and Kubernetes.

The RENÉE digital infrastructure contributes to the realisation of the project's four strategic pillars: circular value chains, robot skills and flexible production, digital infrastructure, and workforce upskilling & human factors. The infrastructure is based on a modular and service-oriented design, allowing easy integration, monitoring, and execution of remanufacturing scenarios through a centralised but flexible control environment.

Further information is provided in the following sections:

- Section 2 outlines the overall infrastructure architecture, module responsibilities, and communication flow.
- Section 3 provides details on process orchestration and scheduling functionalities.
- Sections 4 and 5 describe the Digital Twin and Digital Vault implementations.
- Section 6 presents the Supplier Network Management Tool.

This deliverable provides the first complete implementation of the RENÉE infrastructure and sets the foundation for future iterations and deployment in pilot case studies, where the system will be refined and validated under real operational conditions.

Disclaimer

This document reflects only the authors' view. Responsibility for the information and views expressed therein lies entirely with the authors. The European Commission is not responsible for any use that may be made of the information it contains.



Table of Contents

Executive Summary.....	2
Disclaimer	2
Table of Contents	3
Abbreviation List	5
List of Figures.....	7
1 Introduction.....	9
2 WP4 Architecture.....	9
2.1 Overall WP4 Architecture	9
2.2 WP4 Modules Interconnection and Data Exchange	10
2.2.1 Traceability data storage to SNMT	11
2.2.2 Registration of Products to SNMT	11
2.2.3 Evaluation Process	12
2.2.4 Product State Diagnosis and Disassembly Steps Generation	12
2.2.5 Execution Process	13
2.3 RENÉE Integration Platform	14
3 Remanufacturing Process Scheduling and Orchestration	15
3.1 General Overview of Module	15
3.2 Data Entry Modules	16
3.2.1 ERP Orders Translation	16
3.2.2 Video-based Process Plan Extraction.....	18
3.3 Optimization of Scheduling.....	19
3.3.1 Offline Optimization.....	19
3.3.2 Online Optimization.....	20
3.4 Orchestrator.....	23
4 Digital Twin	24
4.1 General Overview of Modules.....	24
4.2 Digital Twin for the Shopfloor.....	24
4.2.1 IOT Platform	25
4.2.2 Simulation Environment.....	26



4.2.3	User Interface	27
4.3	Digital Twin for the Remanufacturing Cell	29
4.3.1	Digital Environment	29
4.3.2	Robot Behaviour Generation	31
4.3.3	Flexible Resource Controller	32
5	Digital Vault for remanufactured product traceability	32
5.1	General Overview of Module	32
5.2	Client Modules and Data	32
5.3	AAS/Standardization	33
5.3.1	Process as an Asset	33
5.3.2	Product as an Asset	34
5.3.3	Resource as an Asset	36
5.3.4	MongoDB	37
6	Supplier Network Management Tool	37
6.1	General Overview of Module	37
6.2	Overview of the IDS Reference Model	38
6.3	RENÉE Data Space and Major Components	39
6.3.1	DataspaceApp4EDI Connector	40
6.3.2	Identity Provider	41
6.3.3	Metadata Broker	41
6.4	Traceability use case	42
6.5	Data Analytics & ML for Supply Chain Optimization	43
7	Conclusions and Future Activities	44
	References	45



Abbreviation List

Abbreviation	Definition
AAS	Asset Administration Shell
AI	Artificial Intelligence
AIP	AI Perception
CAD	Computer-Aided Design
CSV	Comma-Separated Values
DES	Discrete Event Simulation
DPP	Digital Product Passport
DPT	Disassembly Planning Tool
DV	Digital Vault
ERP	Enterprise Resource Planning
ET	Evaluation Tool
FRC	Flexible Resource Controller
GUI	Graphical User Interface
IDS	International Data Spaces
IDSA	International Data Spaces Association
IDS-RAM	International Data Spaces Reference Architecture
IoT	Internet of Things
ISA	International Society of Automation
IT	Information Technology
KPI	Key Performance Indicator
LLM	Large Language Model
ML	Machine Learning
PSD	Product State Diagnosis
PSO	Process Scheduler and Orchestrator
RC-DT	Remanufacturing Cell Digital Twin
ROS	Robot Operating System
S-DT	Shopfloor Digital Twin
SNMT	Supplier Network Management Tool



SRDF	Semantic Robot Description Format
TFR	Toolbox for Remanufacturing
Twin4X	Twin4eXcellence Framework
UI	User Interface
URDF	Unified Robot Description Format
VLM	Vision-Language Model
WP	Work Package



List of Figures

Figure 1: WP4 architecture	10
Figure 2: Store Traceability Data on the SNMT	11
Figure 3: Registration of Products or Components on the SNMT	11
Figure 4: Evaluation process	12
Figure 5: Product knowledge extraction	13
Figure 6: Execution process	13
Figure 7: RENÉE integration Platform.....	14
Figure 8: Integration platform UI - choice of modules	15
Figure 9: Process Execution Layers	16
Figure 10: ERP Translation UI	17
Figure 11: ERP data translation to AAS.....	17
Figure 12: Video Process Plan extracted data translation to AAS.....	18
Figure 13: Extraction from Video UI	19
Figure 14: Extraction from Video Results	19
Figure 15: Offline Planner UI	20
Figure 16: Online planner results	22
Figure 17: Orchestrator UI.....	24
Figure 18: Shopfloor Digital Twin Architecture	25
Figure 19: UI Mock-up of S-DT Monitoring.....	28
Figure 20: UI Mock-up of S-DT Simulation Interface	28
Figure 21: RC-DT Architecture	29
Figure 22: Workcell real-time visualisation in RVIZ	30
Figure 23: Workcell simulation in Isaac Sim.....	31
Figure 24: Path Planning generation	32
Figure 25: JobOrder AAS model.....	34
Figure 26: ProductType AAS model.....	35
Figure 27: Product AAS model.....	36
Figure 28: WorkCentre AAS model	36
Figure 29: Output of PSO tasks in the JobOrder submodule.....	37



Figure 30: The main Data Space concepts defined by the IDS Reference Architecture Model.....	39
Figure 31: RENÉE Data Space	40
Figure 32: Dataspace Connector Swagger Page.....	41
Figure 33: RENÉE Traceability System: major elements	42
Figure 34: DataSpaceApp4EDI Connector: identification of traced objects.	43



1 Introduction

This deliverable presents the initial prototype of the digital infrastructure developed to support remanufacturing management, focusing on the orchestration of processes, digital representation of operations, and the management of data and supplier networks. It addresses the configuration and demonstration of the modules that enable data collection, processing, exchange, and secure storage across the remanufacturing value chain, ensuring traceability and trusted access throughout the product lifecycle.

This document brings together both technical (with a high level of detail) and conceptual results from Work Package 4 (WP4), aiming to provide a comprehensive view of the current developments.

Specifically, the following modules are presented in this deliverable:

- **Process Scheduler and Orchestrator (PSO)**
- **Digital Twin** (at both shopfloor (**S-DT**) and remanufacturing cell (**RC-DT**) levels)
- **Digital Vault (DV)**
- **Supplier Network Management Tool (SNMT)**

The sections that follow provide a detailed overview of these modules, including their current status at project month M18 and their role within the broader digital infrastructure.

2 WP4 Architecture

2.1 Overall WP4 Architecture

As is shown in Figure 1 (illustrates a part of RENÉE reference architecture presented in D2.2 - Design of RENÉE solutions reference architecture, but focused on WP4) WP4 primarily deals with the control of the execution of the use cases of RENÉE, as well as the static and dynamic data management of the project. Specifically, the PSO module is responsible for the scheduling, orchestrating and monitoring of the execution, the two DT modules are responsible for handling the data in the shopfloor line and the cell, the DV module acts as the central database of the project, that contains the product and process data, and the SNMT module supports the management of supplier networks for remanufacturing.

While Figure 1 focuses on the internal architecture of WP4, it also visually shows its integration with other work packages. WP3 modules (shown in green and described in detail in D3.1) provide key inputs such as product data, KPIs, and guidelines to WP4 modules via the DV and other interfaces. Similarly, WP5 modules (depicted in blue and described in detail in D5.1) represent resource-level components that interact with WP4 through the RC-DT and DV to enable real-time execution and control.

These interconnections reflect the close collaboration between WP4 and both WP3 and WP5, ensuring that planning, execution, and feedback mechanisms are fully integrated. The relationship between WP4 and other work packages was initially introduced in Deliverable D2.2, which provides a system-wide architectural overview and integration logic across the RENÉE digital infrastructure. The current deliverable builds on that foundation, focusing specifically on the WP4 modules, their

connections and their implementation status at M18. Additional details on WP3 modules can be found in Deliverable D3.1, while WP5 modules are further described in Deliverable D5.1.

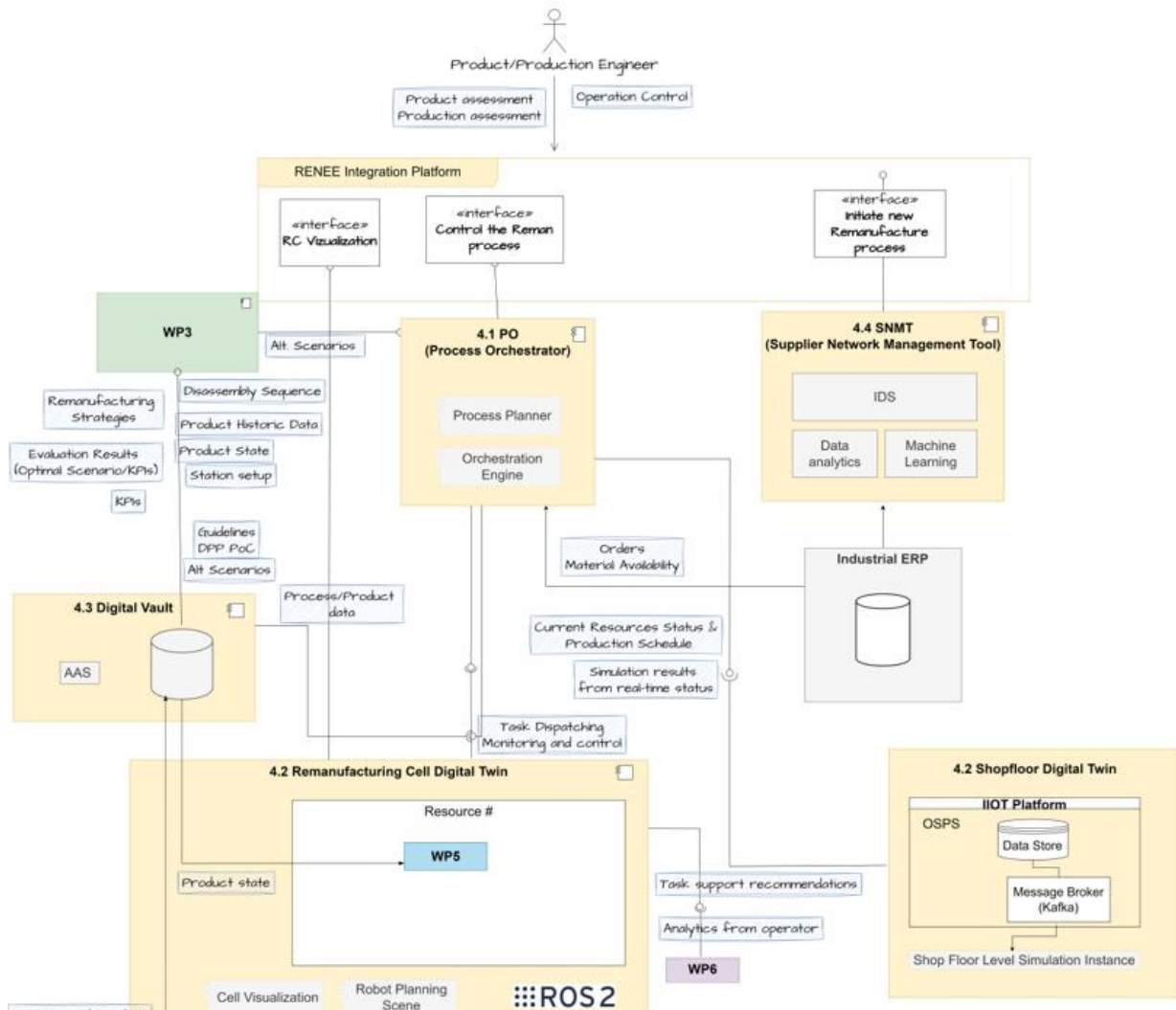


Figure 1: WP4 architecture

2.2 WP4 Modules Interconnection and Data Exchange

This section outlines the interactions and data exchange mechanisms among the key modules developed within WP4 and their integration with other system components. It presents how traceability data, product registration, evaluation results, and execution-related information flow through the architecture, ensuring seamless coordination between modules such as the Supplier Network Management Tool (SNMT), Process Scheduler and Orchestrator (PSO), Evaluation Tool (ET), Digital Vault (DV), and the Digital Twins (S-DT and RC-DT).

2.2.1 Traceability data storage to SNMT

The storage and retrieval of traceability data is described in Figure 2. There, the Production Engineer is simulating the Process Orchestrator, which will be integrated on the second version of the prototype.

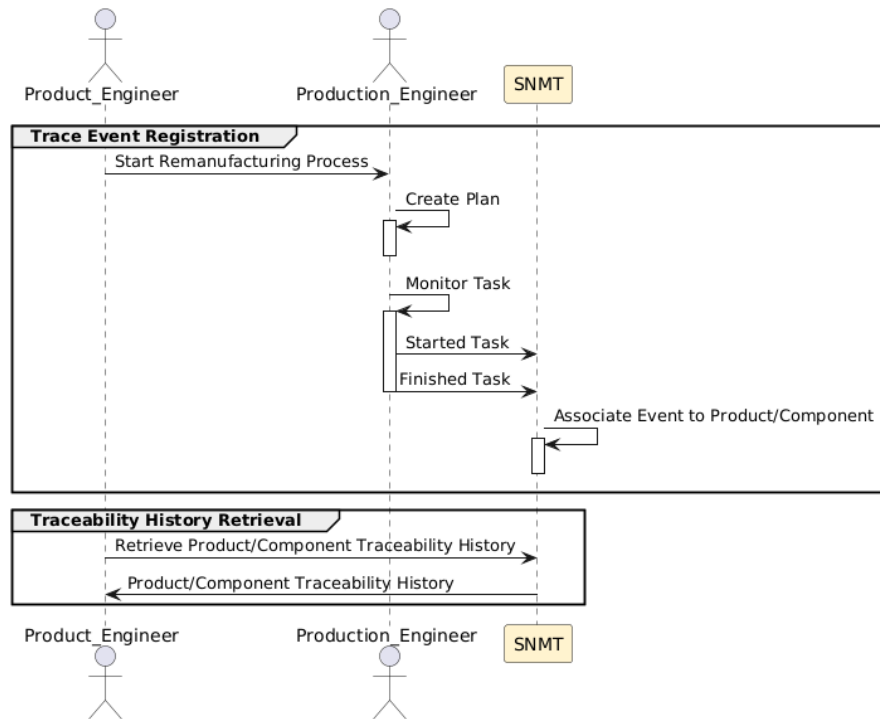


Figure 2: Store Traceability Data on the SNMT

2.2.2 Registration of Products to SNMT

Figure 3 describes the process to register products or components on the SNMT. In the future, an Enterprise Resource Planning (ERP) will replace the Product Engineer to automate this process.

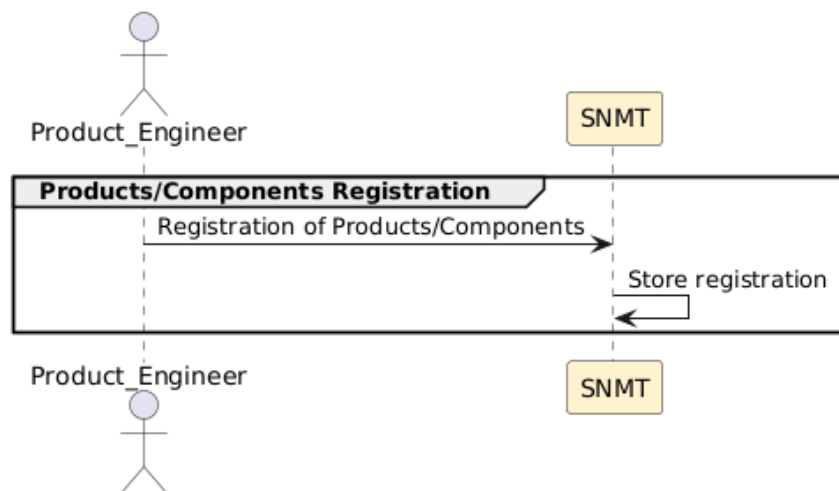


Figure 3: Registration of Products or Components on the SNMT

2.2.3 Evaluation Process

Figure 4 presents the communication between the modules for the evaluation process. While the ET module belongs to WP3, there are necessary interactions with WP4 modules. The Integration Platform handles all initial interactions between the production engineer and the modules, providing the parameters to ET. The ET, then, produces alternative resource utilization scenarios which the PSO receives to perform allocations and extract KPIs. This information is stored in the DV and extracted from the ET to calculate the optional configuration of resources. Results are stored and received from DV. TFR then performs simulations of alternative line configurations with the optimal workstation configuration from ET's results and extracts KPIs. Twin4X compares the KPIs with baseline KPIs given by the engineer. Through the Integration Platform, the results are presented to the production engineer.

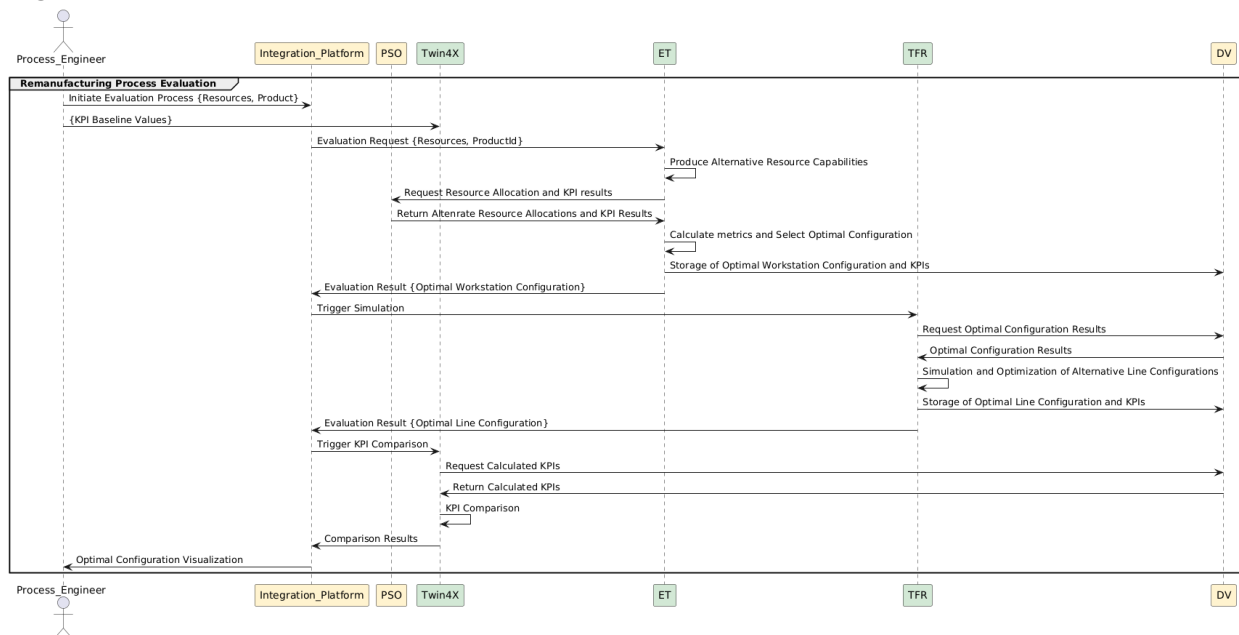


Figure 4: Evaluation process

2.2.4 Product State Diagnosis and Disassembly Steps Generation

DPT and PSD are responsible for the generation of disassembly steps and the diagnosis of the product state, and store their results to DV, from which the PSO can receive them to use as parameters for planning.

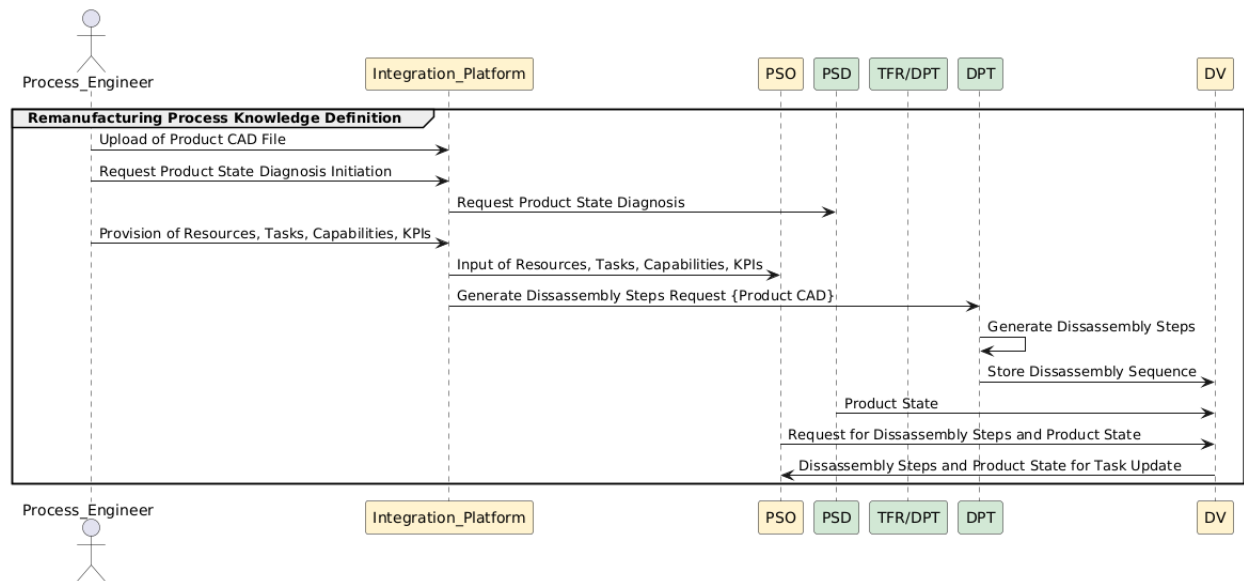


Figure 5: Product knowledge extraction

2.2.5 Execution Process

During the execution process, the PSO has the central role, as it is the module that generates the schedule of tasks with the feedback of the RC-DT and dispatches those tasks to the Flexible Resource Controller (FRC) interfaces that control the resources. S-DT provides simulations for the training of the reinforcement-learning part of the schedule optimization.

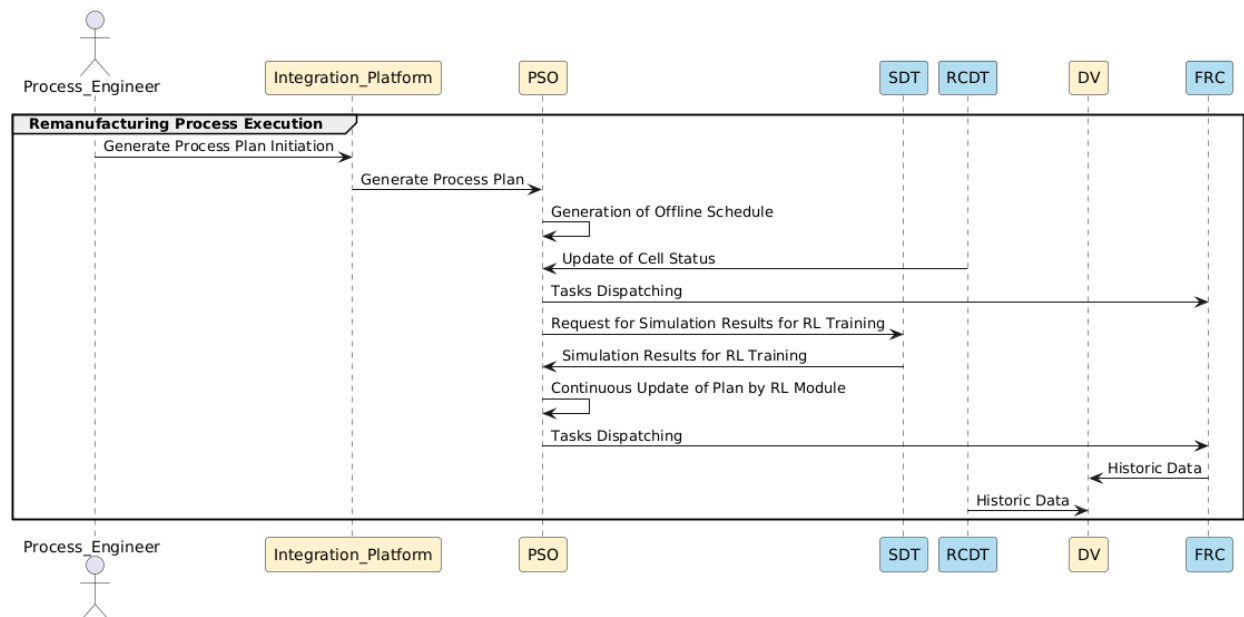


Figure 6: Execution process

2.3 RENÉE Integration Platform

The purpose of the RENÉE Integration Platform is to monitor and control the RENÉE solution before and during execution. In addition, the platform will be the means to connect and deploy the different modules in a confidential manner for the partners, who will not have to share private development details to the integrators and use case coaches. The concept can be found in Figure 7.

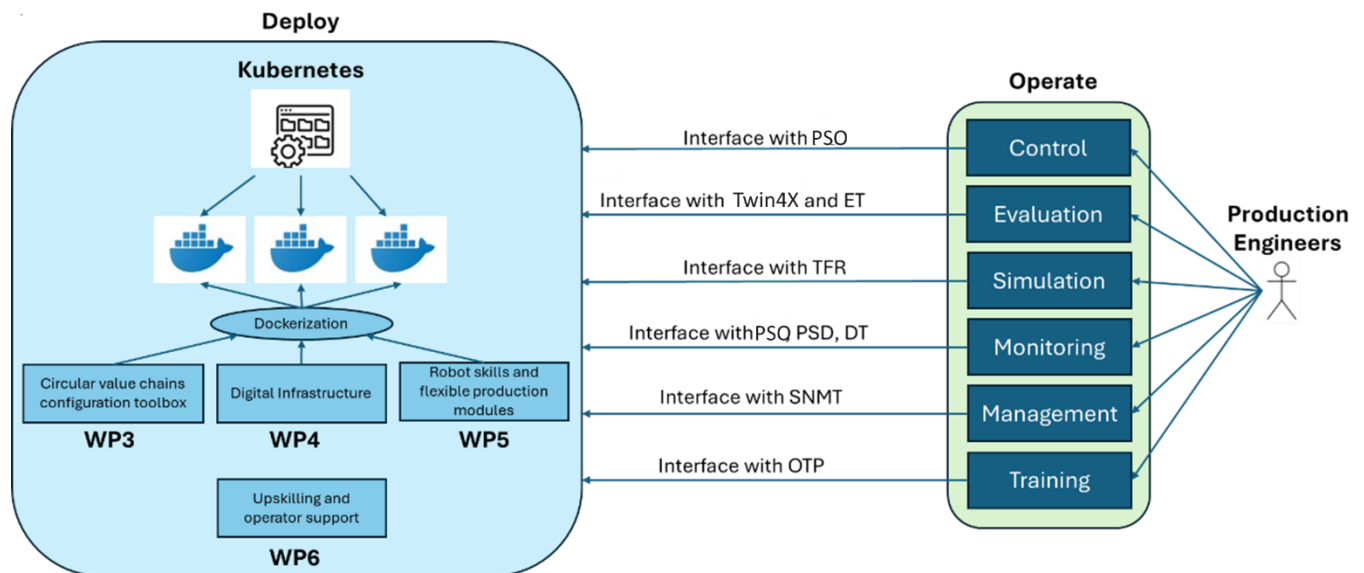


Figure 7: RENÉE integration Platform

The platform’s backend deploys the modules of the project in containers handled by Kubernetes (Burns et al., 2016). Its UI (Figure 8), which can be accessed with username and password from RENÉE members, will have any necessary interface with specific modules in tabs, which can be chosen and open the UI of the module (Figure 8). Through these interfaces, the production engineer will be able to utilize these interfaces to perform operations such as control, evaluation, etc., as they are offered by the RENÉE modules. The platform handles the communication between the different technologies of the modules and will pack everything in an instantly deployable executable, to be utilized in the RENÉE pilots. A prototype of the integration platform has been created, both the backend and frontend, with the integration of the PSO and PSD modules to begin with. Through the application’s GUI, the user can access the GUI and capabilities of the two modules..

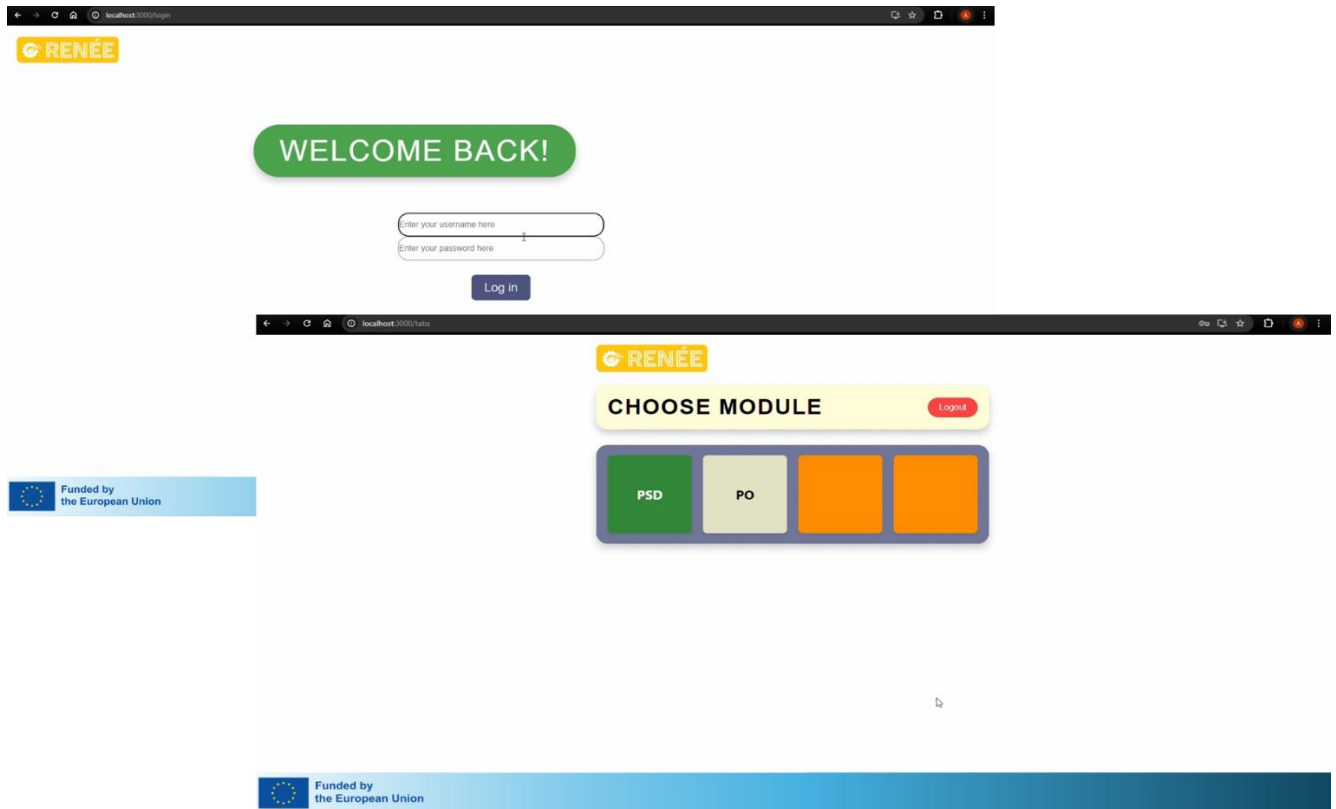


Figure 8: Integration platform UI - choice of modules

3 Remanufacturing Process Scheduling and Orchestration

3.1 General Overview of Module

The PSO module is responsible for the planning, orchestrating and monitoring of the tasks and actions of the RENÉE pilot cases. PSO has three main parts:

- the automatic translation of Enterprise Resource Planning (ERP) data to standardized Asset Administration Shell (AAS) models via Large-Language-Models (LLMs),
- the optimization of task schedule and allocation to resources for the pilot case execution, and
- the dispatching of the actions to the corresponding resources.

All functionalities are discussed in the following subsections. In Figure 9, the process execution layers are shown, in which all functionalities play a central role. The ERP translation is set as the first layer, as it extracts the needed data, like resources, tasks etc., to be fed to the next layers. The planning and orchestration capabilities communicate between them via a database, where the output of the planning is saved and retrieved by the orchestrator as input. The communication goes both ways, as the orchestrator can inform the planner if there is a need for replanning. The orchestrator layer connects with the RC-DT that acts as the Digital World Model in order to stay updated on the shopfloor and resources status and connects with the resource control layer via a

server/client interface using the Robot Operating System (ROS) framework and specifically ROS2. Each layer will be discussed in more detail in the subsections below.

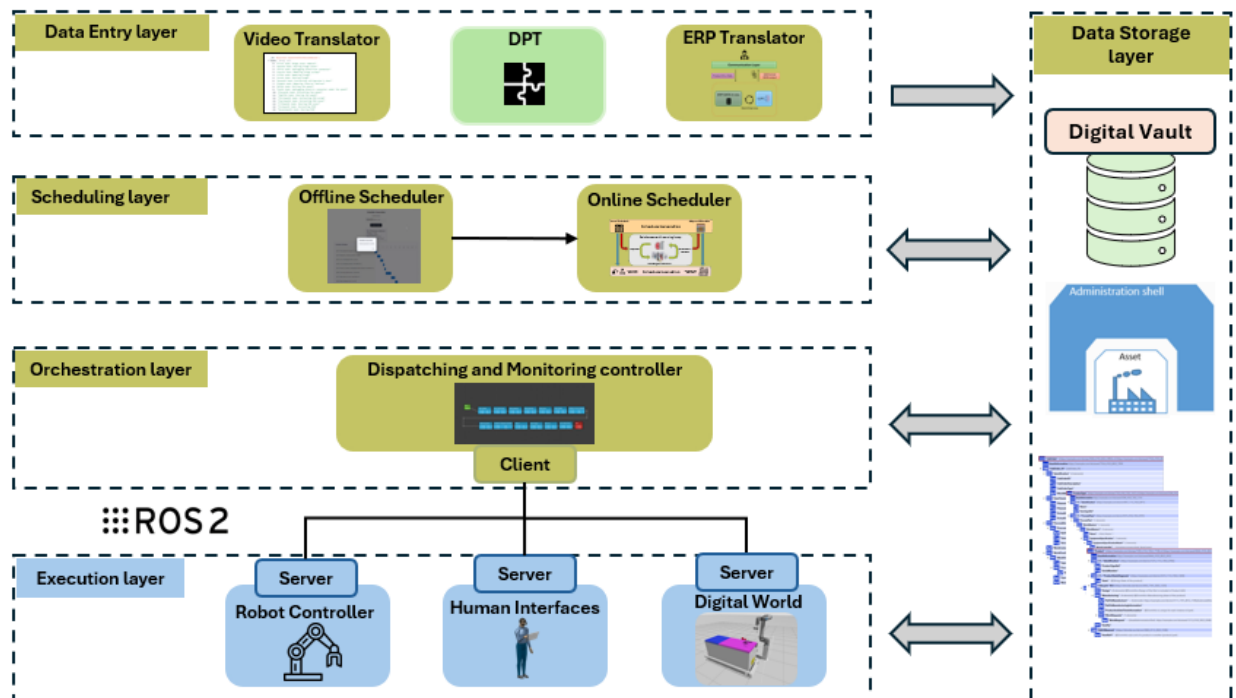


Figure 9: Process Execution Layers

3.2 Data Entry Modules

The Data Entry Modules enable the automated extraction, processing, and standardisation of production-related data from diverse input sources into structured formats suitable for further use by the system's core components. Specifically, they facilitate the transformation of production orders from ERP systems and process execution details from video recordings AAS-compliant structures. Leveraging advanced AI models, like LLMs and Vision-Language Models (VLMs), these modules ensure accurate data interpretation, enrichment, and integration into the Digital Vault, forming a critical foundation for subsequent planning and optimisation processes.

3.2.1 ERP Orders Translation

The input of the Task Scheduler is a set of production orders that need to be executed. These orders are generated normally by ERP tools and therefore this information needs to be extracted and processed to be used as input by the Scheduler. This data is automatically extracted from csv files and stored in the central database in a standardised format, as described in more detail later in this section, to be accessible by the Scheduler.

For the extraction, reasoning and structuring of this information to the appropriate format, LLMs are employed for their capacity to handle complex, large-scale datasets while complying with operator-defined instructions. Users interact with a graphical interface to select either specific product IDs or

date ranges, allowing targeted extraction from CSV files that record production orders over extended periods (Figure 10).

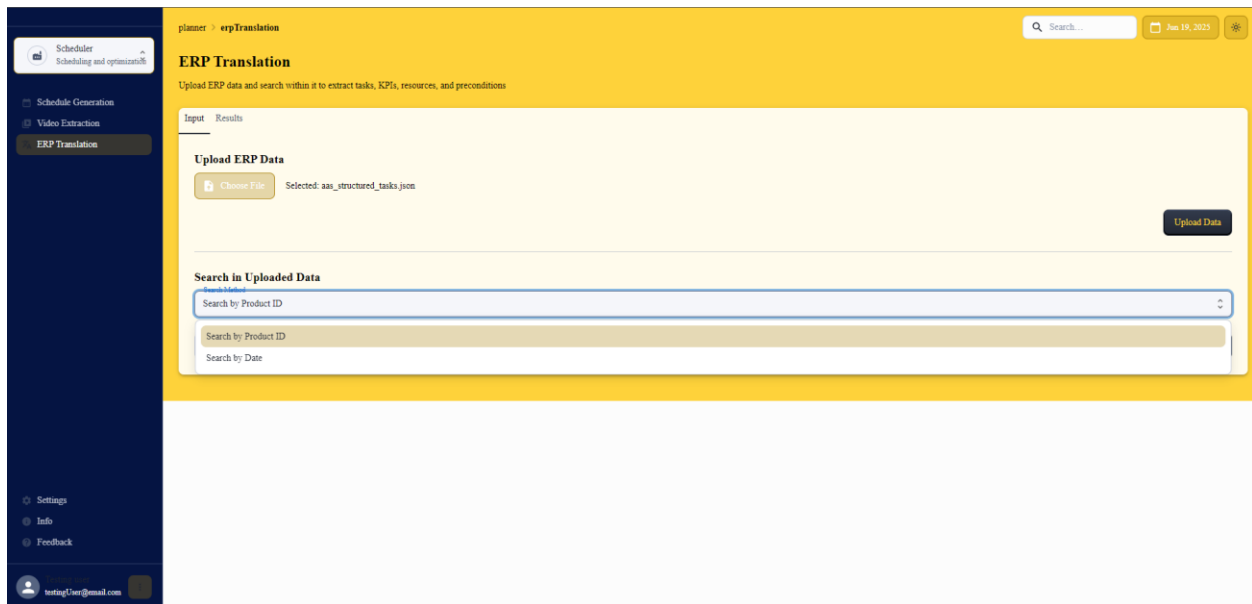


Figure 10: ERP Translation UI

Upon the user request, the LLM is prompted to extract structured process data, including task names, capabilities, durations, and inferred execution dependencies (preconditions). These preconditions are not present in the original dataset but are generated through the LLM’s reasoning capabilities.

The results are saved in a JSON file built following the AAS standard. The structure of the AAS model has followed the ISA-95 standard, an international standard developed by the International Society of Automation for integrating enterprise and control systems in manufacturing. The AAS format is not generated by the LLM but constructed hardcoded based on the model's findings. The output is stored in the DV and parsed by the PSO for optimization of resource allocation (Figure 11).

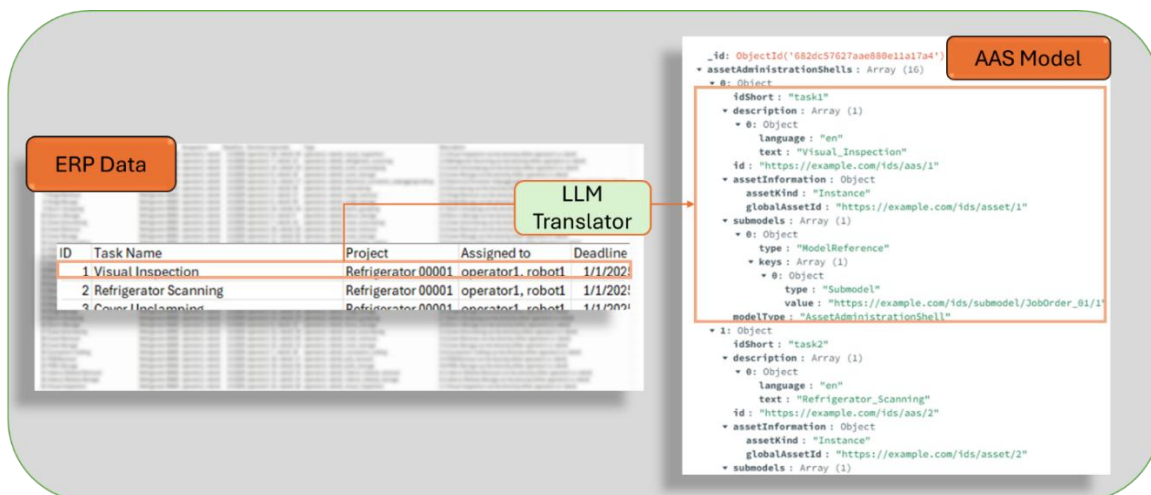


Figure 11: ERP data translation to AAS

3.2.2 Video-based Process Plan Extraction

Automating the information related to the process plan linked to a specific product (e.g. the disassembly of a refrigerator) is also performed through the analysis of visual-verbal input turned into structured task information. In this case, video recordings of the execution of the dedicated process serve as the input for identifying executed tasks and their sequence. A single video recording is used, capturing both the operator and the robot performing the complete set of disassembly tasks. Data extraction is achieved through the combination of a Vision-Language Model (VLM), an audio-parsing LLM, and a generic LLM for further data processing. For this purpose, the “Action Recognition with VLMs/LLMs” perception module, further elaborated under D5.1, is employed. From these recordings, the VLM provides the task titles per frame, for the LLM to parse the names and order to make the complete task list. The task list extracted is then structured accordingly to follow the appropriate AAS model (as described in Section 7.3) and stored in the DV (Figure 12). The AAS format is not generated by the LLM but constructed hardcoded based on the model's findings.

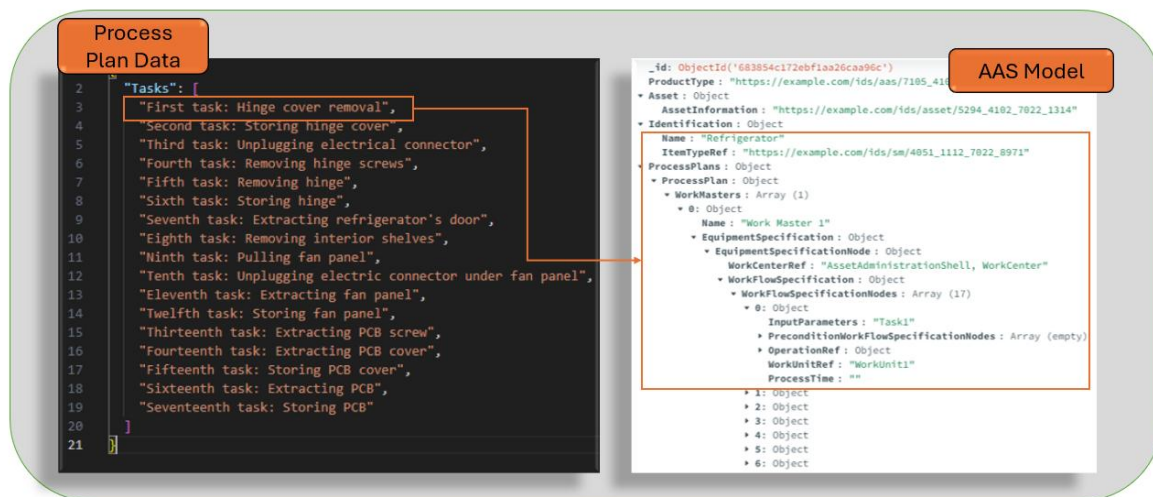


Figure 12: Video Process Plan extracted data translation to AAS

Similar to the ERP-based information extraction, the user provides the needed input (video) through the UI (Figure 13) and the extracted list of tasks identified in the video is presented to the user for validation (Figure 14) before being restructured and stored in the DV as described above).

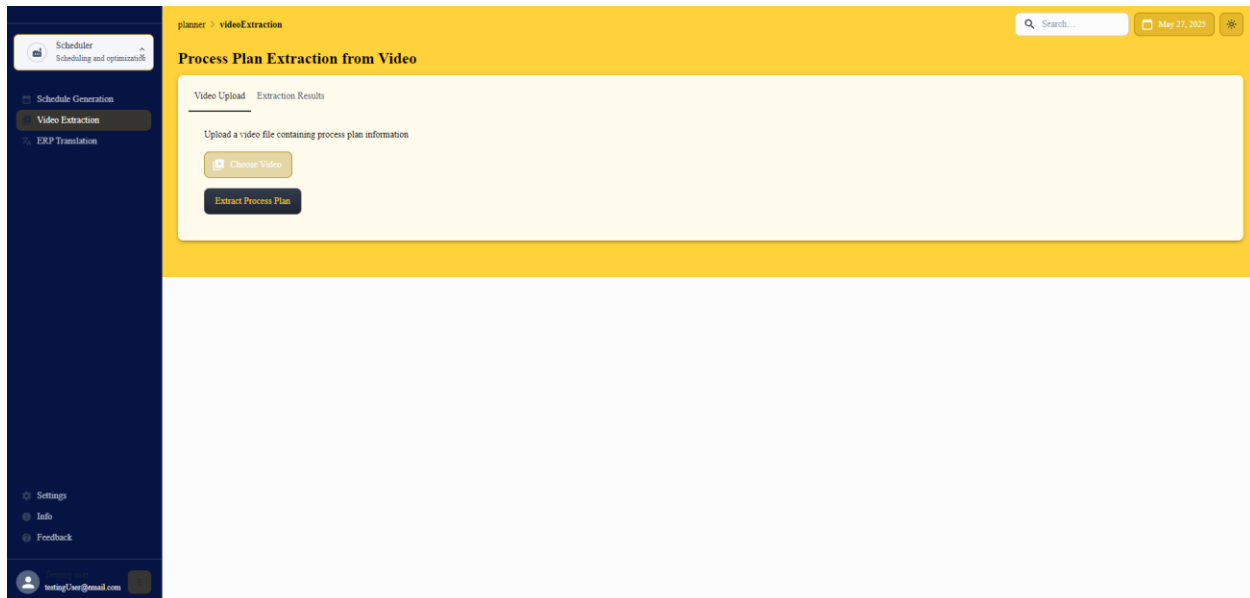


Figure 13: Extraction from Video UI

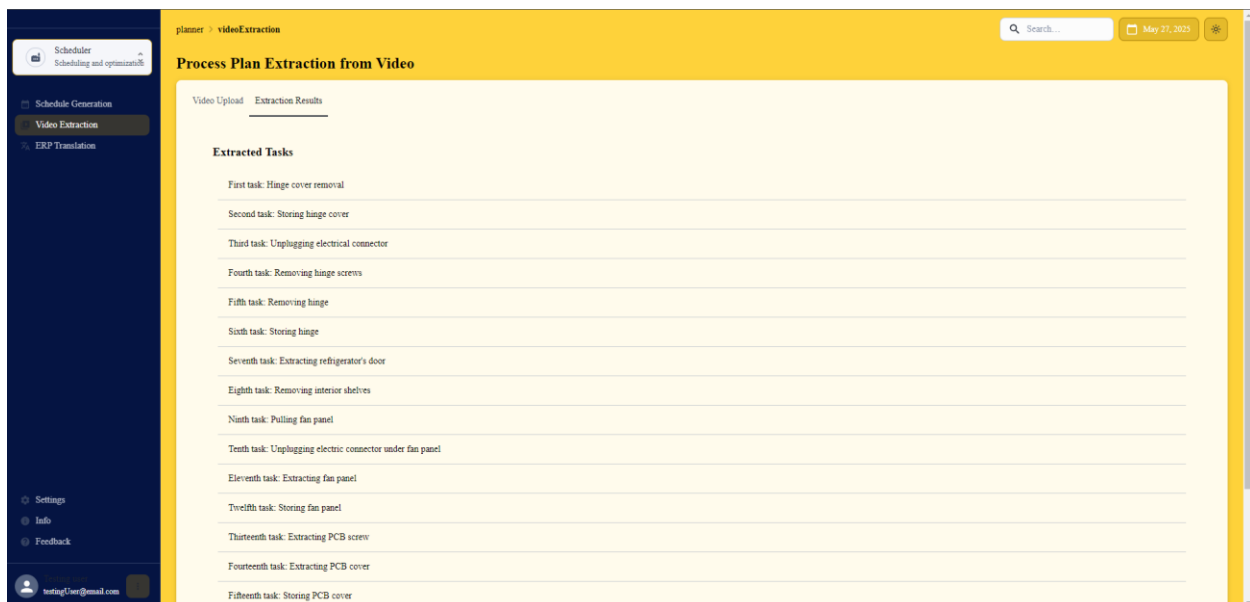


Figure 14: Extraction from Video Results

3.3 Optimization of Scheduling

3.3.1 Offline Optimization

The objective of the scheduling is the assignment of tasks to resources and specific time slots in such a way that the given metrics, such as duration, will be optimized. Stochastic methods are best suited for larger scale problems and handle variability naturally. While they do not necessarily find

the most optimal solution, they are robust and computationally scalable, providing adequate solutions within bounded time. Therefore, these methods are considered fitting for the re-manufacturing problem. As it regards stochastic methods, metaheuristics such as Genetic Algorithms (ForrestStephanie, 1996), Simulated Annealing (Bertsimas & Tsitsiklis, 1993), or the Equilibrium Optimizer (“Equilibrium Optimizer,” 2020) – which was the one chosen -, were considered to be the best choice for offline planning and allocation compared to other methods, such as heuristics and deep learning. Metaheuristics are ideal for complex, high-dimensional problems which may not have a precise model and don't require prior training, unlike heuristics which work for simpler setups, and deep learning. Deep learning is a powerful tool for adaptation and long-term dynamic planning but requires extensive training or time for the algorithm to learn with trial and error, which is unnecessary complex for offline planning. The offline plan is meant to give a head start to the online planning, which is executed by reinforcement learning techniques in order to reduce the learning time and search space, leading to faster convergence and better handling of constraints.

The offline optimization algorithm based on metaheuristics has been completed and has been tested in the Arcelik use case. The main GUI of the application has already been created as shown in Figure 15. For future developments, the choice for more KPIs that are tailored to the different use cases will be added.

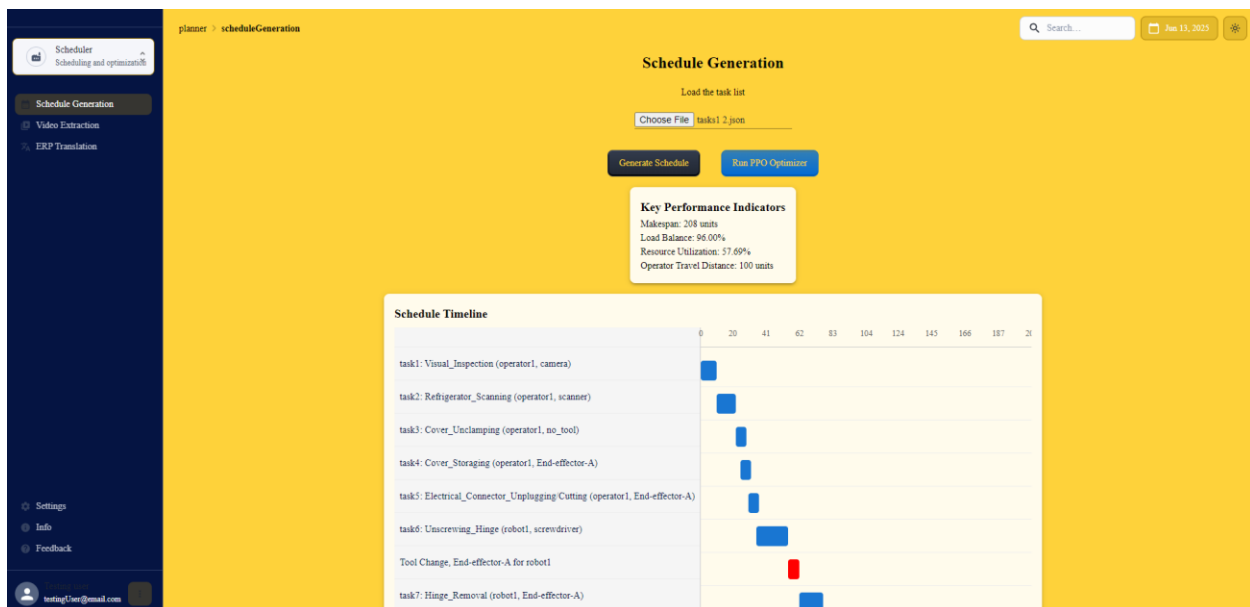


Figure 15: Offline Planner UI

3.3.2 Online Optimization

Online planning capabilities are necessary for case studies characterized by dynamic traits and uncertainty, and it also works on cases which are less dynamic, but effectiveness and utilization of resources must be optimal. Re-manufacturing by its nature has a measure of uncertainty due to the condition of the returned products that will be re-manufactured, which are in different states of quality degradation. The flaws of the returned product cannot always be extracted on arrival of the product. These flaws may be detected in inspections done during re-manufacturing processes such



as disassembly. Therefore, new tasks can be added to the process while executing other tasks, requiring the allocation of new time slots and resources. By performing dynamic scheduling, the planner can consider the new tasks and parameters in synergy with the rest of the remaining schedule and re-allocate all tasks in an optimal way during the entire execution eliminating interruptions of the process. Aside from this advantage, online planning promotes adaptability and robustness in any schedule, due to its ability to re-allocate tasks without interruptions when an event happens, such as the change of a resource's availability, which may happen due to an error. The planner will take into account the loss of capabilities in the schedule and create a new schedule with the remaining tasks without the utilization of the unavailable resource. If the resource enters the production again, the planner will recalculate the remaining tasks assignments accordingly and may even schedule correction actions to solve the error that caused the re-planning. Finally, while the offline stochastic plan presents a good initial plan, the online planner, trained by the results of the running execution, can improve the existing plan, combining the strength of domain knowledge and machine-learning adaptability for optimal results, leading to increase validation of the combined methods.

The first implementation of the online planner has also been created with reinforcement learning (RL). It has not yet been improved with extensive simulation training. The online planner receives the plan from the offline planner and improves upon it, running the plan in increments to check if there are improvements to be made while the policies are improved. It also checks if tool changes are needed between processes, and re-schedules, accordingly, minimizing the number of tool changes required. In case of new events, the online planner reschedules the remaining tasks, considering the event parameters.

The approach is based on the Proximal Policy Optimisation 2 (PPO2) algorithm (Antonin RAFFIN et al., 2025), implemented using OpenAI's Stable Baselines library. The agent is designed as a Multi-Layer Perceptron (MLP) neural network, that will be trained on simulated shop floor data, e.g. a simulation model from TFR in WP3.



Figure 16: Online planner results

3.3.2.1 System Structure

The system is structured in two phases:

- **Training Phase:** A baseline task-resource assignment solution is first generated using the Assembly Line Balancing (ALB)/Equilibrium Optimizer (EO). The RL-based planner verifies the optimization level of the baseline planner and adapts to deviations during the execution. The agent will be trained using simulated data from the TFR environment, where it interacts with the simulation by selecting actions (i.e., assigning tasks to resources), receiving feedback (rewards), and updating its policy to maximize cumulative performance. The training process is continuous and not triggered by deviations.
- **Execution Phase:** Once a stop criterion is met, either because of a predefined number of training episodes or when reaching a target reward, the trained agent can be deployed for both offline optimization and real-time decision-making. It dynamically adjusts task assignments based on the evolving state of the production system, including tool changes and workstation locations.

PPO2 is a policy-gradient method known for its stability, efficient convergence, and ease of implementation. The agent is modelled using a Multi-Layer Perceptron (MLP) neural network, which receives as input a 7-dimensional state vector for each task, including:

- Task timing information (start time, duration)
- Executor assignment and availability
- Precondition status and timing
- Tool change requirements
- Workstation information

The reward function is defined as a weighted combination of:

- Make span improvement compared to the initial EO schedule
- Constraint violation penalties
- Tool change efficiency
- Resource utilization
- More KPIs will be added according to each use case's needs

Negative rewards are applied when the agent selects infeasible actions (e.g., violating precedence, assigning unavailable resources, or inefficient tool changes), in which case the simulation is reset to maintain learning consistency.

This RL-based tool provides a flexible and adaptive planning mechanism, able to respond to disturbances and system variability in both offline and real-time scenarios. It complements the EO-based mathematical model by offering reactive intelligence, helping ensuring robustness in actual remanufacturing environments where process variability is the norm.

3.4 Orchestrator

The orchestrator constitutes an integral part of the real-time execution and monitoring of the re-manufacturing process. Once the planner has produced a schedule and assigned resources, the orchestrator is responsible for dispatching these tasks to the appropriate physical or digital resources on the shopfloor. It acts as a link between planning outputs and resources. The orchestrator handles the triggering and monitoring of actions across heterogeneous systems. Each task in the schedule is broken down into fundamental actions. The orchestrator controls the execution flow either via ROS 2 interfaces or custom interfaces that can either be integrated in the orchestrator core or implemented as plugins. These communications ensure that each action is delivered to the correct resource (robot, human workstation, sensor, inspection tool, etc.), and that feedback is received regarding its success, failure, or status updates. A key feature of the orchestrator is its hierarchical execution model, where multiple lower-level activities can be grouped in a single activity. This model allows flexible control, monitoring, and potential interruption or reconfiguration at different levels of granularity. In case of a failure or deviation detected during execution, the orchestrator can trigger corrective actions or signal the planner to initiate re-planning.

The orchestrator logs each action's lifecycle and associated metadata in a database. This supports traceability, KPIs extraction, and enables intelligent decision-making during runtime by providing feedback to the planner and RC-DT. From an integration perspective, the orchestrator exposes and utilizes interfaces in a service-based architecture, ensuring that components such as the resource controllers or the RC-DT can evolve independently. This modular approach enables scalable deployment across different re-manufacturing setups and simplifies adaptation to new resources.

As it regards the latest progress of the orchestrator development,, the initial UI has been created, and the transfer from the output of the planner in the database to the orchestrator has been tested. As part of the Arcelik use case, the mechanism for the dispatching of tasks to the resource controlling module (FRC) has been tested, as seen in Figure 17.

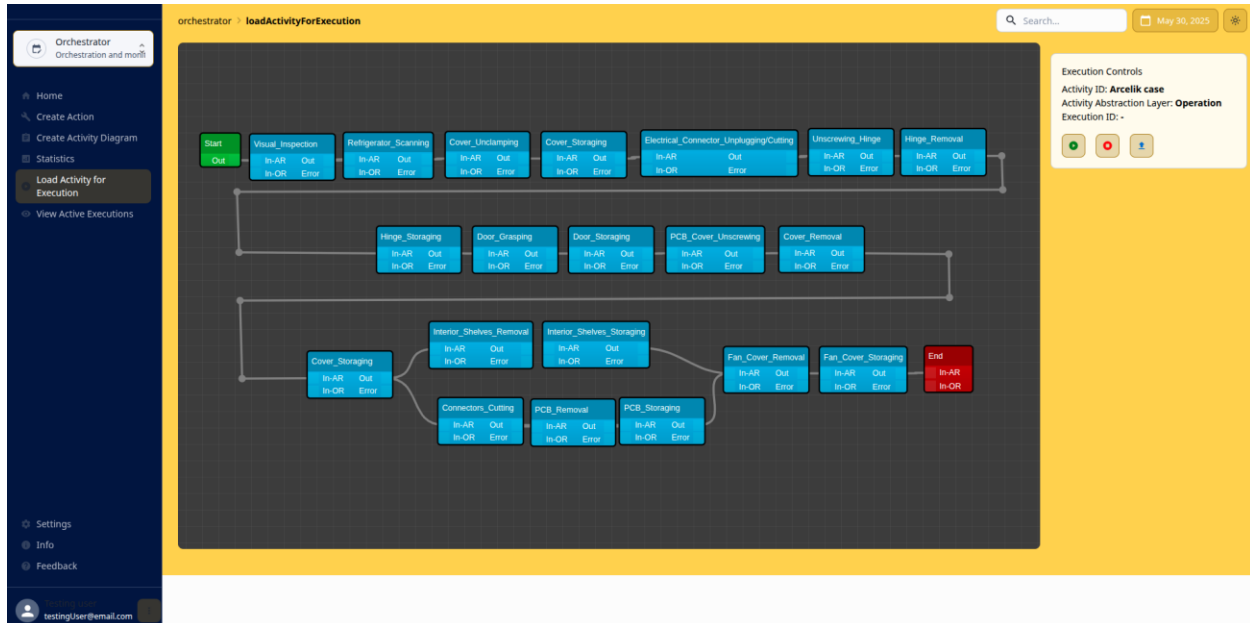


Figure 17: Orchestrator UI

4 Digital Twin

4.1 General Overview of Modules

This chapter presents two key Digital Twin (DT) modules that support the digital infrastructure of remanufacturing: the Shopfloor DT (S-DT) and the Remanufacturing Cell DT (RC-DT). Both aim to create a real-time, data-driven virtual representation of the physical production system, but at different levels.

The S-DT operates at a broader level, offering a global view of the production environment. It combines IoT data, simulation tools, and user interfaces to monitor operations, run performance simulations, and support data-driven decision-making.

In contrast, the RC-DT works at a more detailed level, controlling individual resources within a work cell. Built on the ROS2 framework, it enables real-time monitoring, robot path planning, and adaptive control based on current conditions.

Together, these modules offer a layered approach that connects high-level planning with detailed execution, enhancing the efficiency and flexibility of remanufacturing systems. The next sections describe each DT module in detail.

4.2 Digital Twin for the Shopfloor

The Digital Twin for the Shopfloor (S-DT) serves as a core component of the digital infrastructure for remanufacturing management, enabling real-time monitoring, simulation, and data-driven decision-making. Designed as an integrated system, it combines simulation technologies, IoT data integration

mechanisms, and advanced data storage solutions to deliver an accurate and dynamic virtual representation of the physical production environment.

This module aims to replicate and monitor the current state of the entire shopfloor by leveraging real-time data streams from various sources and continuously updating a simulation model that reflects ongoing operational conditions. The DT not only enhances visibility over remanufacturing processes but also supports predictive analytics and adaptive control, which are crucial for optimizing resource use, minimizing downtime, and reacting to dynamic changes in (re)production scenarios.

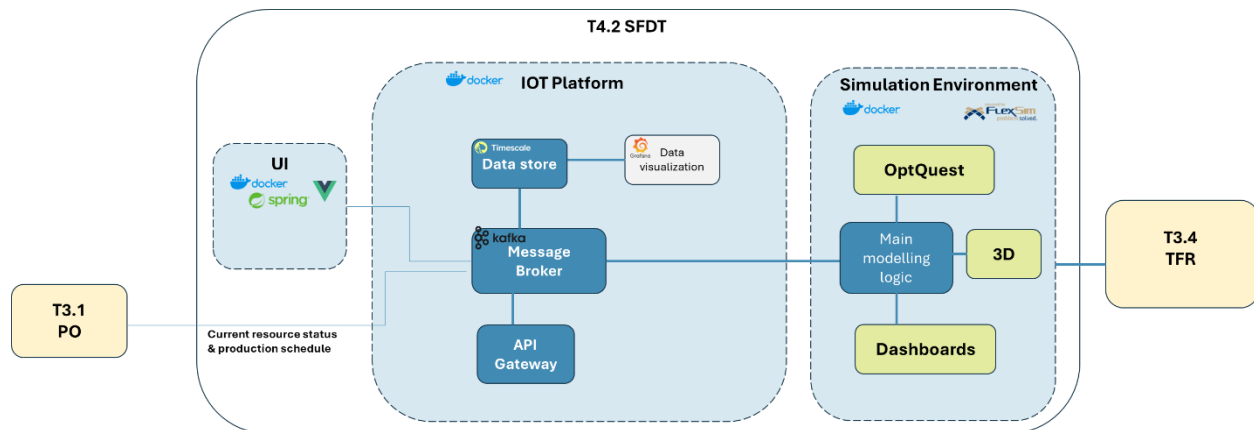


Figure 18: Shopfloor Digital Twin Architecture

The S-DT follows the architecture depicted in Figure 18 and is composed of the following key components:

- IOT Platform (IOT Broker, Time Series DB and APIs);
- Simulation Environment;
- User Interface

4.2.1 IOT Platform

The IoT Platform forms a foundational layer of the S-DT, enabling structured data handling, asynchronous communication, and historical data management across the digital infrastructure. Rather than interfacing directly with shopfloor equipment, the IoT Platform receives real-time data streams via the PSO, which acts as the dedicated bridge between physical assets and digital services.

The platform is designed using a microservices-based approach, where each service is developed, deployed, and operated independently within Docker containers. This architecture ensures modularity, scalability, and ease of maintenance, which are essential in complex and evolving manufacturing environments.

A central component is the Message Broker, which handles the routing of messages across the system. Real-time and event-based data originating from the shopfloor are collected by the PSO, published to the Message Broker, and then disseminated to the appropriate consumers, such as the simulation engine, analytics modules, or data storage components. The Message Broker,

implemented using technologies such as Apache Kafka, ensures high-performance, fault-tolerant communication, and decouples data producers from consumers.

To support system-wide interoperability, an API Gateway is used to expose a unified endpoint through which services and external modules can interact with the IoT Platform. This abstraction layer simplifies integration and enhances security and access control.

The platform also includes a historical Data Store, implemented with MongoDB, which archives all events and messages processed by the broker. This centralized repository supports both operational queries and advanced analytics. Stored data can be reused for performance monitoring, anomaly detection, or feeding higher-level decision-support tools such as the DV.

In addition, the platform offers a data visualization layer, with dashboards built using tools like Grafana. These dashboards provide stakeholders with intuitive, real-time views of key performance indicators, sensor readings, and process data, aiding in operational transparency and strategic decision-making.

In summary, the IoT Platform plays a critical role in the S-DT by receiving and managing real-time data from the PSO, orchestrating service interactions, storing and exposing historical information, and delivering actionable insights through visualization. Its design emphasizes interoperability, modularity, and scalability, supporting the broader objectives of a resilient and intelligent remanufacturing system.

4.2.2 Simulation Environment

The Simulation Environment is a core component of the S-DT, providing a dynamic, visual, and interactive representation of the remanufacturing environment. It is developed using FlexSim, a state-of-the-art simulation software that supports Discrete Event Simulation (DES) and rich 3D visualization capabilities. Within the digital infrastructure, the simulation model instance is provided by the Toolbox for Remanufacturing (TFR) module and is directly called by the S-DT to reflect real-time shopfloor conditions, visualize process flows, and support optimization and analysis activities. A more detailed description of the TFR (and other WP3 modules) can be found in Deliverable D3.1.

Key functionalities of the Simulation Model include:

- **Real-Time Visualization:** The simulation model serves as the visual environment of the S-DT. When fed with data from the IOT Platform (via the PSO), it reflects the real-time status of shopfloor operations and enables users to observe process dynamics visually as they unfold.
- **Scenario Analysis and Experimentation:** Users can run simulations with varying parameters and configurations to test alternative remanufacturing strategies, assess system performance under different conditions, and support decision-making at both strategic and operational levels.
- **Optimization Support:** The model is prepared to integrate with optimization and ML tools (namely Reinforcement Learning training models), allowing for the evaluation and



identification of optimal process configurations. This functionality enables decision-support for production scheduling, layout modifications, and resource allocation.

- **Interactive Dashboards:** Through FlexSim’s native tools and integration with external platforms (e.g., Grafana), multiple dashboards can be built on top of the simulation outputs. These dashboards enable users to monitor key performance indicators (KPIs), analyse bottlenecks, and extract insights from both historical and real-time system behaviour.
- **Integration with Real-Time Data:** FlexSim allows for seamless data exchange with external sources. In the context of the S-DT, the simulation model can be updated continuously based on real-time inputs from the IOT platform, ensuring that the virtual model remains synchronized with physical operations.

Overall, the Simulation Model enhances the functionality of the S-DT by bridging the physical and digital layers through an intuitive, immersive, and analytically powerful environment. It not only supports system monitoring and visualization but also acts as a platform for experimentation and optimization, contributing directly to the goal of achieving adaptable and efficient remanufacturing systems.

4.2.3 User Interface

The User Interface (UI) of the S-DT is designed to provide stakeholders with a responsive, user-friendly, and functionally rich environment for interacting with the underlying simulation and data infrastructure. The UI serves as the primary point of access for operators, engineers, and managers, enabling real-time monitoring, visualization, and interaction with the digital representation of the remanufacturing shopfloor.

The UI is being developed using a web-based architecture, with a Vue.js frontend and a Spring Boot backend, supported by a MySQL database. This technology stack ensures scalability, performance, and maintainability while allowing seamless integration with the other components of the digital infrastructure.

Key functionalities of the User Interface include:

- **Real-Time Monitoring:** Users can view live dashboards showing key indicators such as machine status, process times, and production flow, helping them monitor shopfloor performance (Figure 19).
- **Simulation Visualization:** The UI provides access to a 3D view of the simulation model, allowing users to see how the shopfloor is operating in real time or under different scenarios (Figure 20).
- **Data Access:** Historical data and simulation results are stored in a database and made accessible through the UI, supporting performance analysis and decision-making.

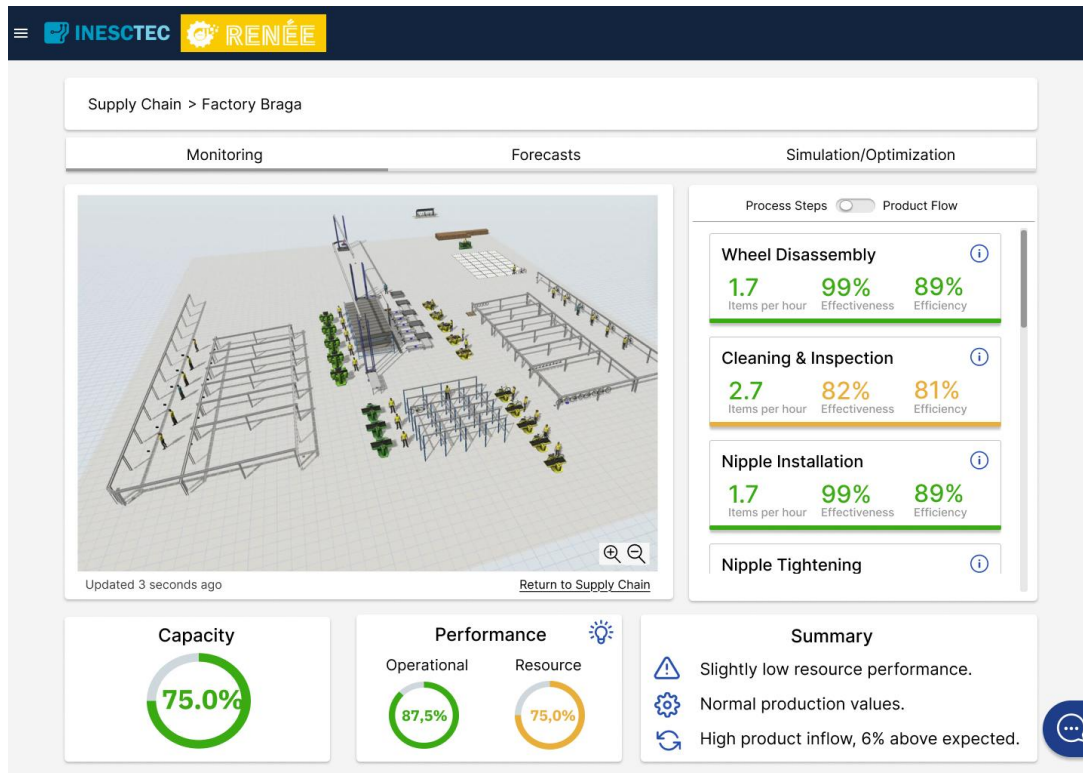


Figure 19: UI Mock-up of S-DT Monitoring

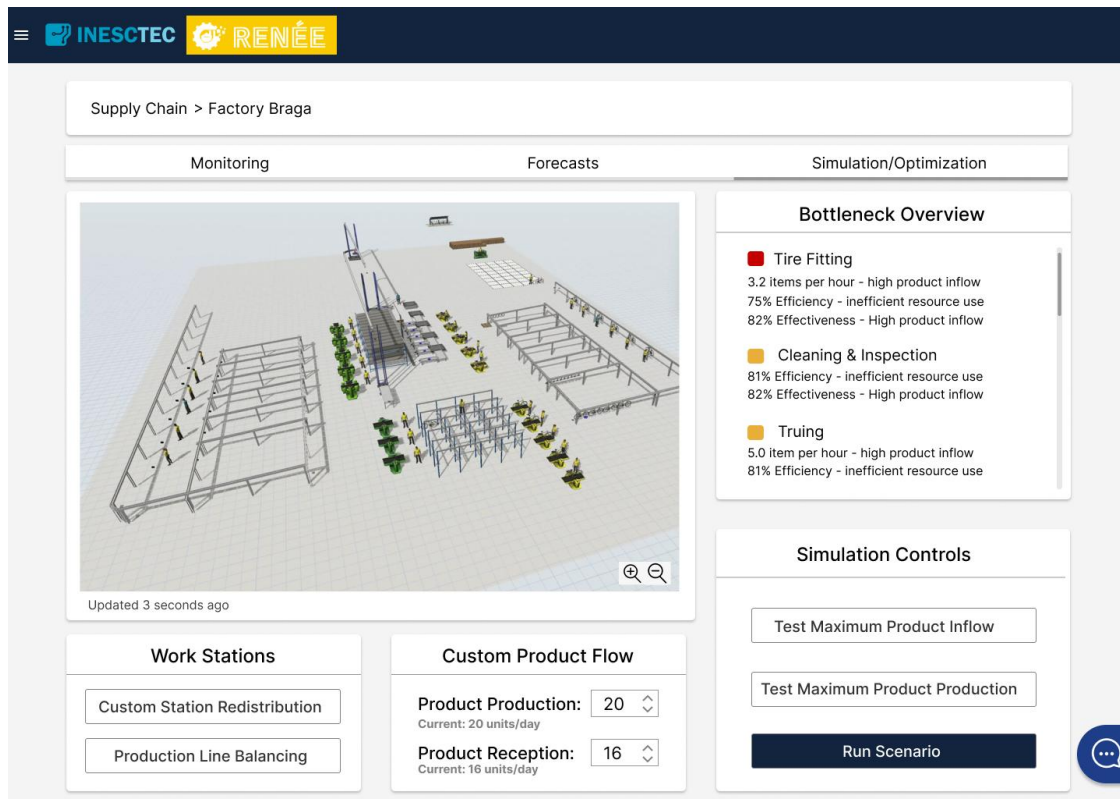


Figure 20: UI Mock-up of S-DT Simulation Interface

4.3 Digital Twin for the Remanufacturing Cell

The Digital Twin for the Remanufacturing cell (RC-DT) serves as a core component of the digital infrastructure for the work cell, enabling real-time monitoring, simulation, and control of the resources' behaviour and task execution. The RC-DT implementation is based on the ROS2 framework, utilising the pub/sub architecture of ROS for data exchange among the various components.

This module aims to replicate and monitor the current state of a remanufacturing cell by leveraging real-time data streams from various sources and continuously updating a simulation model that reflects ongoing operational conditions. The RC-DT not only enhances visibility over ongoing task execution of the operating resources but also realises online robot path planning and adaptive control, which are crucial for real-time resource control, considering the working environment status and reacting to unforeseen events. The RC-DT follows the architecture depicted in Figure 21 and is composed of the following key components:

- Digital Environment Representation for Visualisation and Simulation
- Robot Behaviour Generator
- Flexible Resource Controller

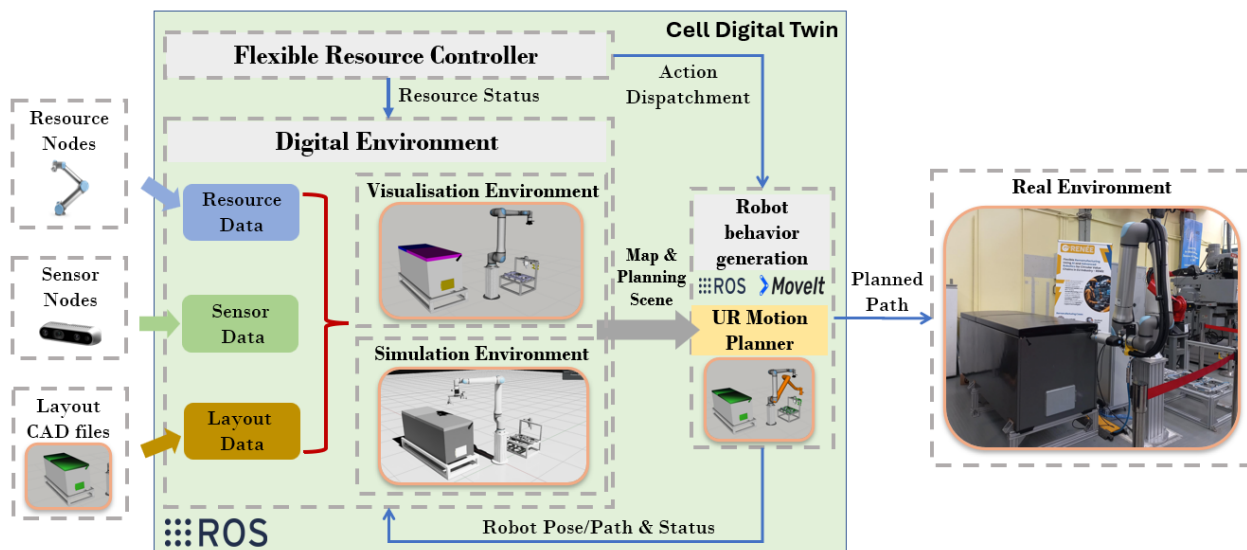


Figure 21: RC-DT Architecture

4.3.1 Digital Environment

The Digital Environment acts as the virtual representation of the physical work cell, integrating real-time data from the robot controllers and the various sensors as well as CAD files of the static equipment (refrigerator base, tool changing station, etc.). Moreover, information about the geometry, configuration and movement limitations of the robots is also provided in dedicated file formats (urdf, srdf). All this information allows to maintain an accurate and up-to-date model of the real environment. This environment is essential for providing both visual feedback of the current environment and the resources state and simulation capabilities that support robot path planning and adaptive behaviour.

The Digital Environment representation can be split into two distinct components, the real-time digital visualisation of the real environment and the simulated environment, as described in more detail in the following sub-sections.

4.3.1.1 Visualisation Environment

The Visualisation Environment is responsible for rendering the real-time status of the remanufacturing cell using the RViz tool within ROS2 (Figure 22). It visualises data streams including robot pose, sensor inputs, and layout configuration, helping operators and developers monitor the execution of tasks in real time and identify potential issues. This component provides a high-fidelity interface for interpreting the robot's state and trajectory within the digital model.

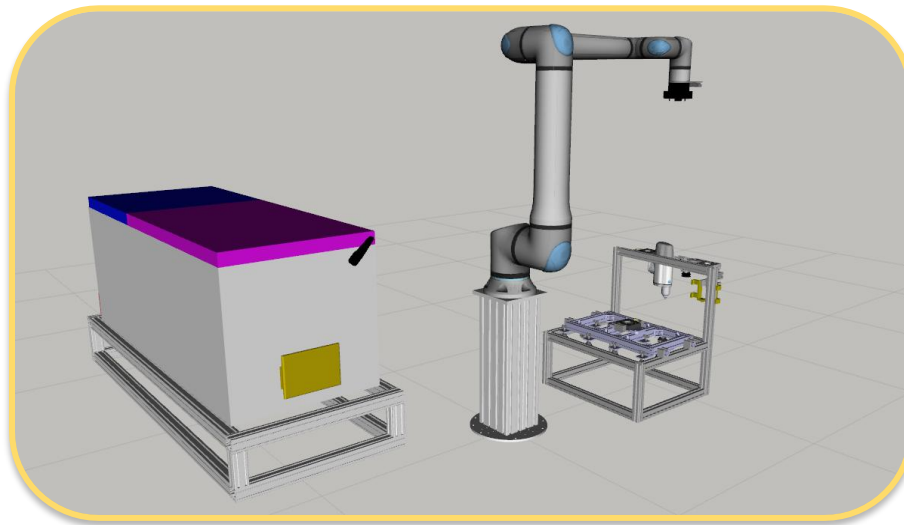


Figure 22: Workcell real-time visualisation in RVIZ

4.3.1.2 Simulation Environment

The Simulation Environment is a core element of the RC-DT, enabling virtual emulation of the robot's behaviour and interaction with its surroundings. This environment is implemented using Isaac Sim, an advanced robotics simulation platform developed by NVIDIA, which integrates seamlessly with ROS2¹. It allows for real-time, physics-based simulation of robotic systems and their interactions with dynamic or static objects in the remanufacturing cell (Figure 23).

By replicating the physical environment and robot kinematics within a virtual domain, the simulation environment provides a risk-free testbed for validating motion plans, verifying task feasibility, and assessing system performance under various operational scenarios. It supports scenario-driven testing enabling the system to adaptively refine robot behaviour before deployment in the physical environment.

The integration with ROS2 ensures that simulated robots and sensors behave as if they were operating in the real world, allowing developers to test the complete pipeline of perception, planning,

¹ <https://developer.nvidia.com/isaac/sim>

and control. Moreover, the feedback generated during simulation is fed back into the digital twin loop, supporting iterative improvement of the robot's task execution strategy.

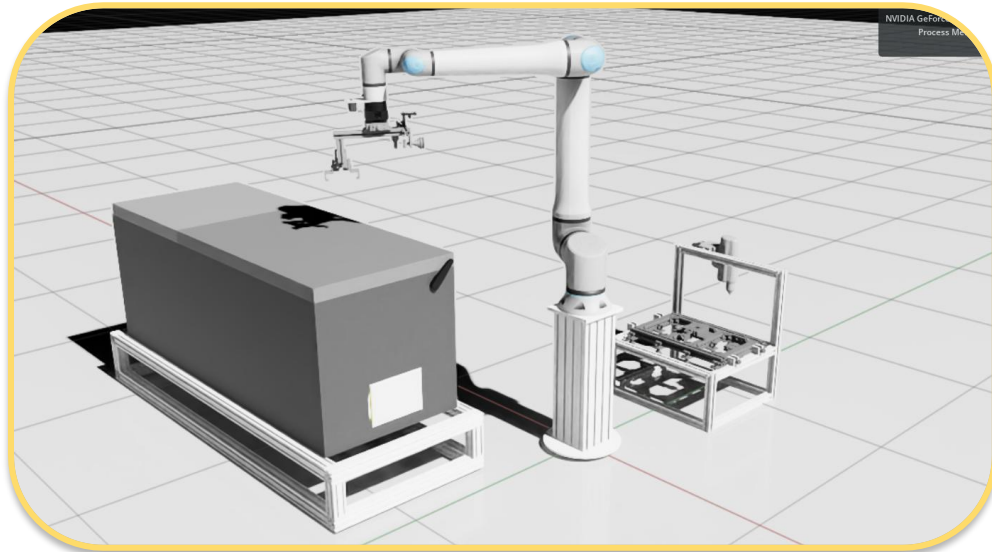


Figure 23: Workcell simulation in Isaac Sim

4.3.2 Robot Behaviour Generation

The Robot Behaviour Generation module is responsible for generating robot motion plans based on real-time contextual awareness provided by the digital environment. It acts as the core decision-making component that transforms action commands into executable trajectories for the robot.

This module is implemented using MoveIt, a widely adopted motion planning framework that integrates natively with ROS2. MoveIt constructs a continuously updated planning scene by combining live sensor data, robot state information, and environmental layout geometry. Based on this scene, it generates collision-free and kinematically feasible motion plans considering the kinematic and control requirements of the robot and potential process or working area-related constraints (joint angle, workspace, etc.).

The system ensures that all planned paths respect environmental constraints, avoid obstacles, and optimise specific kpis relative to the selected optimisation algorithm (like execution time or energy use). As low-level robot actions are dispatched by the Flexible Resource Controller, this module reacts by regenerating or adjusting motion plans in real-time to account for dynamic changes such as object displacement, unexpected obstacles, or resource availability.

Once a path is computed, the trajectory is published via ROS2 interfaces and executed by the robot controller in the physical or the simulated cell, while is also visualised via the RVIZ environment (Figure 24).

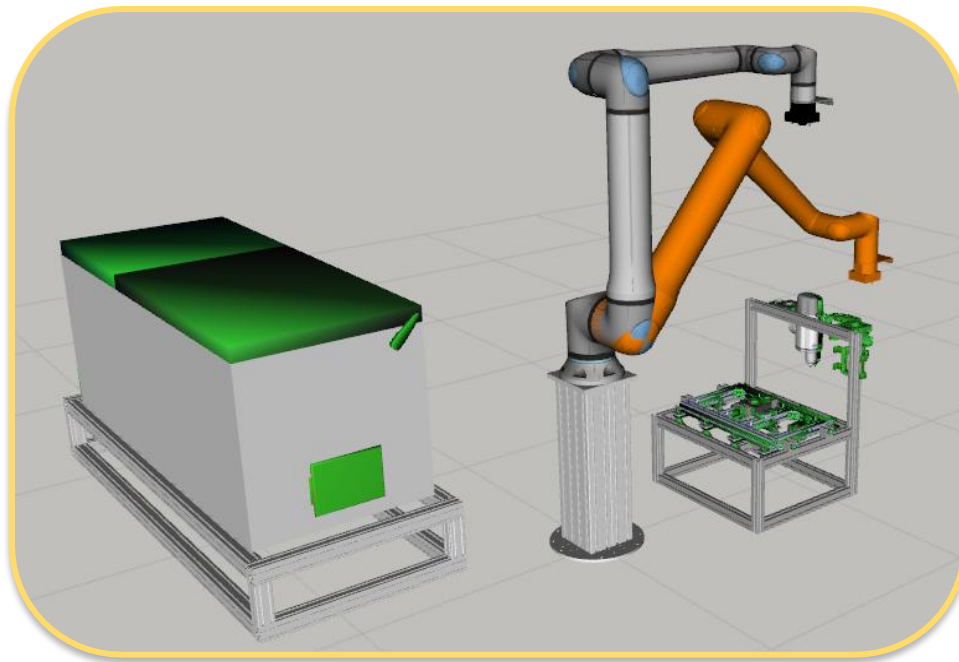


Figure 24: Path Planning generation

4.3.3 Flexible Resource Controller

The Flexible Resource Controller (FRC) is not part of the RC-DT but directly communicates with the RC-DT core components overseeing the dynamic allocation of the dedicated actions that each resource needs to perform based on the generated schedule and dispatching of tasks done by the PSO. More specifically, the FRC is responsible to dispatch low-level actions to the robots' motion planners and from there to their controllers (real or virtual), considering the real time status of the robots. Moreover, the FRC monitors the execution of the dispatched actions and provides feedback to the PSO in case of unforeseen events. More details about the FRC are presented in D5.1.

5 Digital Vault for remanufactured product traceability

5.1 General Overview of Module

The Digital Vault (DV) will support the deployment of the architecture defined in D2.2 by providing a secure database where any data related to traceability will be stored and updated. The data will be available for the relevant modules that need it and tamper-evident to ensure the DV's updates are trustworthy.

The DV will contain the data in a standardized format, to promote standardization in the project and in remanufacturing. The details are explained in the subsections below.

5.2 Client Modules and Data

The DV will connect with several modules, according to the architecture depicted in D2.2:



- From **PSD**, DV will receive mainly Product State data, in the form of a header file containing the ID of the product and information about its state.
- From **DPP**, DV will receive Product Historic data, such as ID, manufacturing date, component details, etc.
- From **DPT**, it will receive the disassembly sequence in STEP format.
- From **ET**, it will receive the Evaluation results, mainly in Json format.
- From **Twin4X**, it will receive KPIs.
- From **TFR**, it will receive workstation setups and alternative use case scenarios
- From **ET**, it will receive evaluation results and will send to **ET** the alternative scenarios it receives from **TFR**.
- From **RC-DT**, it will receive and Product and Process Historic data.
- From **ERP**, it will receive the ERP's output of tasks, resources and capabilities.
- To **PSO**, it will send the ERP's output of tasks, resources and capabilities
- To **AIP**, it will send Product State and Historic data.

Figure 1, Figure 4, Figure 5, and Figure 6 also show the interconnections of DV with the other modules. As it regards the data types, in whichever format the data is before the DV receives it, connectors developed by the Module owners with the support of INTRA will transform these formats to a standardized AAS implementation. Accordingly, when the data is received by the Modules from DV, similar connectors will transform it into Module-specific formats.

5.3 AAS/Standardization

The Asset Administration Shell (AAS) provides a structured way of representing assets digitally, whether they are systems, machines or products. Since it provides a semantic standardization, it facilitates interoperability of data and precise monitoring and controlling of production processes, by providing seamless communication between the modules of the project. AAS alone does not cover operational and organizational structures as comprehensively as standards, like ISA-95, cover. In order to be up to date with the latest standards, RENÉE will research the existing ones and will combine them with AAS to ensure standardization of data. During the first phase of the project early prototype AAS models related to the process-product-resource were created to facilitate the initial prototypes of the planning modules and the DV. These models have been based on ontology and semantics proposed by the ISA-95, but they do not fully comply with the standard as the examination of the existing standards is still ongoing.

5.3.1 Process as an Asset

As it regards the modelling of the process, an AAS model representing the JobOrders has been created. This model includes information related to the scheduling of each order (timing, resource assignment, preconditions), as well as information regarding its actual execution (Figure 25). The information regarding how this order will be carried out is related to the dedicated process plan and the resource assigned to execute this order. This information is predefined and stored in the database as part of the ProductType AAS, as described in the following sub-section.



Figure 25: JobOrder AAS model

5.3.2 Product as an Asset

The ProductType AAS model includes the information regarding a specific type of product. This information includes some identification-related information together with the related Process Plans, the recipe to manufacture a product using different parameters (resources, capabilities, etc.). This information is essential for the Process Planner in order to allocate the JobOrders to the appropriate resources matching the capabilities of the available resources with the required

capabilities of each Process Plan related to the ProductType included in the processed order. The initial structure of this model is visualised in Figure 26.

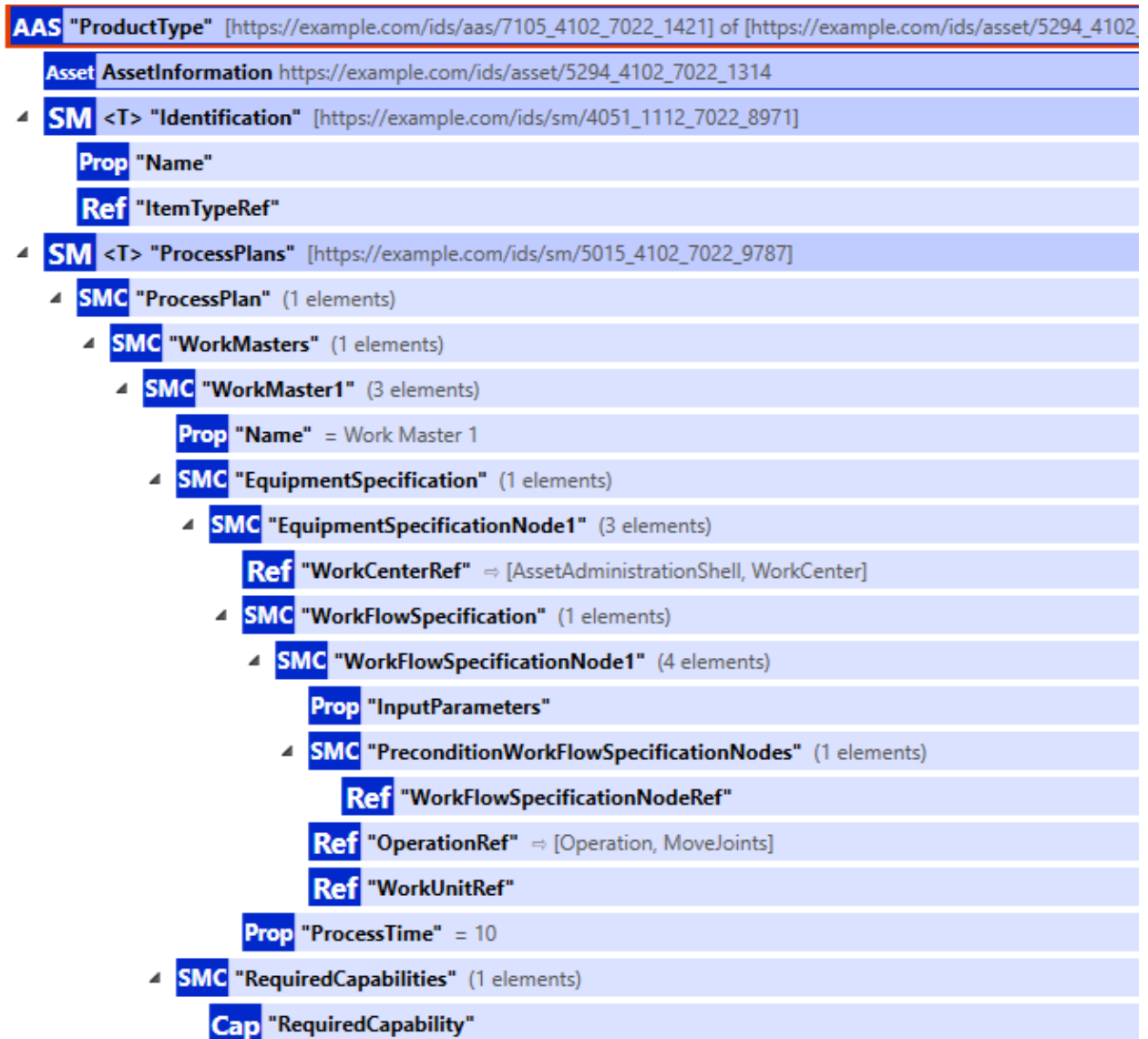


Figure 26: ProductType AAS model

Besides the ProductType, the Product itself, as an instance, has been modelled in an AAS format including historical information about the product related to each remanufacturing order. This model will be connected to the DPP at the next phase of the project as the methodology and modelling approach to be followed regarding the DPP is ongoing (more details on the DPP implementation can be found in D3.1). The initial version of the Product AAS model is presented in Figure 27.

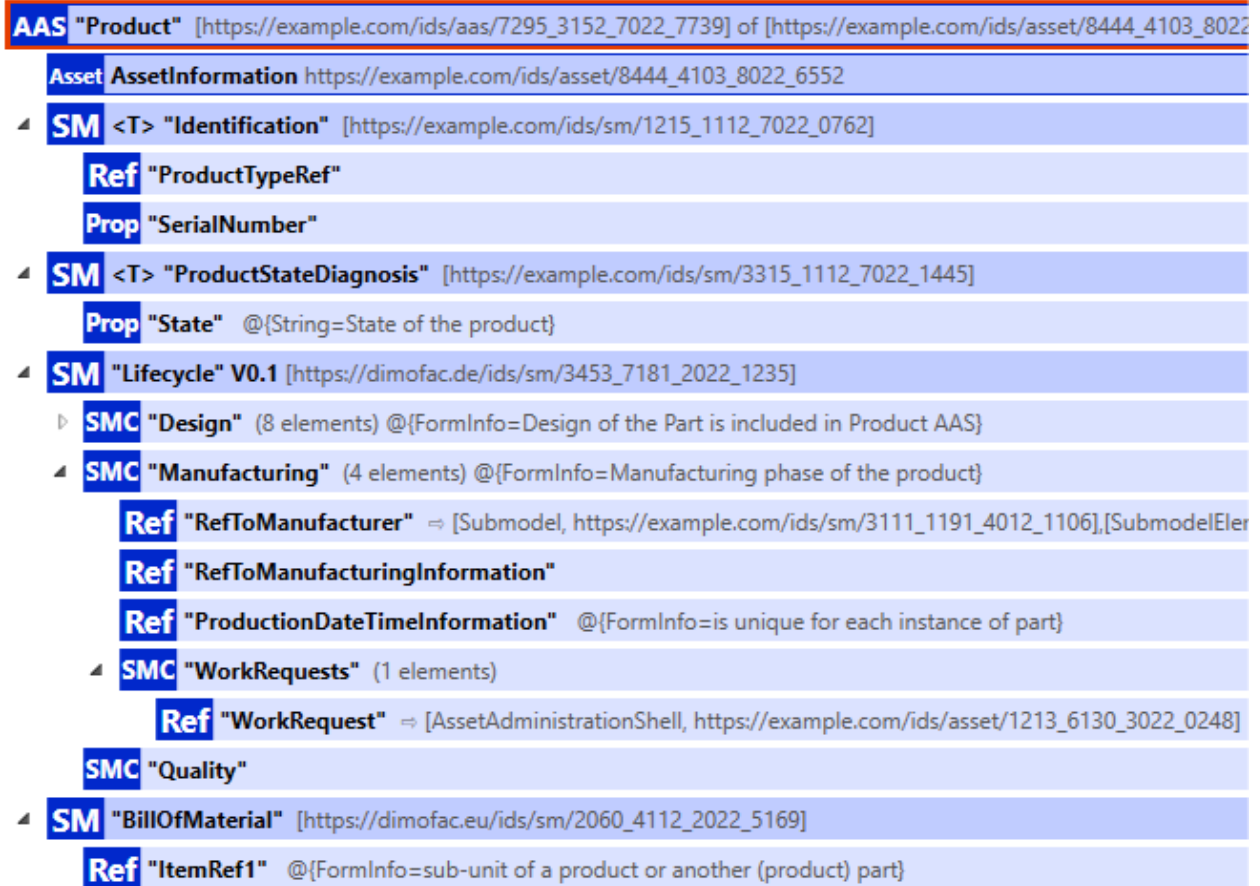


Figure 27: Product AAS model

5.3.3 Resource as an Asset

Finally, a dedicated AAS model has been implemented to basically represent the capabilities of each resource, which is essential for the planning procedure as explained above. The initial version of the Product AAS model is presented in Figure 28.



Figure 28: WorkCentre AAS model

5.3.4 MongoDB

The AAS models have been modelled in MongoDB (Chodorow, 2013). MongoDB is flexible, as it can store different fields, types and structures and export all in JSON format. It is designed for fast inserts and updates and works well with development stacks such as Python, Node.js, etc. Additionally, it is compatible with the AAS, and thus, it has been selected as the framework database for the DV. In Figure 29, the output of the PSO tasks in the JobOrder submodule is shown.

```
  _id: ObjectId('682dc57627aae880e11a17a4')
  ▼ assetAdministrationShells: Array (16)
    ▼ 0: Object
      idShort: "task1"
      ▼ description: Array (1)
        ▼ 0: Object
          language: "en"
          text: "Visual_Inspection"
          id: "https://example.com/ids/aas/1"
      ▼ assetInformation: Object
        assetKind: "Instance"
        globalAssetId: "https://example.com/ids/asset/1"
      ▼ submodels: Array (1)
        ▼ 0: Object
          type: "ModelReference"
          ▼ keys: Array (1)
            ▼ 0: Object
              type: "Submodel"
              value: "https://example.com/ids/submodel/JobOrder_01/1"
          modelType: "AssetAdministrationShell"
    ▼ 1: Object
      idShort: "task2"
      ▼ description: Array (1)
        ▼ 0: Object
          language: "en"
          text: "Refrigerator_Scanning"
          id: "https://example.com/ids/aas/2"
      ▼ assetInformation: Object
        ...
        ...
        ...
  modelType: "ProcessDurations"
  idShort: "ProcessDurations"
  modelType: "SubmodelElementCollection"
  ▼ value: Array (2)
    ▼ 0: Object
      idShort: "operator1"
      modelType: "Property"
      valueType: "xs:double"
      value: "14"
    ▼ 1: Object
      idShort: "robot1"
      modelType: "Property"
      valueType: "xs:double"
      value: "19"
  ▼ 3: Object
    category: "CONSTANT"
    idShort: "WorkFlowSpecification"
    ▼ description: Array (1)
      ▼ 0: Object
        language: "en"
        text: "WorkFlowSpecification"
    ▼ value: Array (1)
      ▼ 0: Object
        category: "CONSTANT"
        idShort: "WorkFlowSpecificationNode"
        ▼ description: Array (1)
          ▼ 0: Object
            language: "en"
```

Figure 29: Output of PSO tasks in the JobOrder submodule

6 Supplier Network Management Tool

6.1 General Overview of Module

This section presents an overview of the Supplier Network Management Tool (SNMT), a software implementation responsible for supporting the management of supplier networks for remanufacturing. This will be accomplished by creating a distributed data space ecosystem to allow supply chain partners to share information regarding their remanufacturing process.

This data-sharing ecosystem will be based on the International Data Spaces (IDS) model, with the primary goal of keeping data sovereignty (i.e., the data owner has absolute control over its own data), which will result in data not being devaluated. This ecosystem will tackle the traceability of components and products in the value chain.

Additionally, a data analytics tool will be developed, with machine learning (ML) support, to enhance traceability across manufacturing value chains while optimizing its operations.

6.2 Overview of the IDS Reference Model

The SNMT is built upon the modules of the IDS model. A key purpose of a data space is to enable orchestration of information exchange in a secure, trusted and sovereign manner.

The Data Space architecture chosen for the RENÉE project is the International Data Spaces Reference Architecture (IDS-RAM), which was developed by the International Data Spaces Association (IDSA) and has become a standard for building Data Spaces in Europe. It is an international standard to share data based on sovereign principles. Its main goals are secure and trusted data exchange in a business ecosystem and to guarantee data sovereignty, which the IDS-RAM document defines as "a natural person's or corporate entity's capability of being entirely self-determined with regard to its data.". Additionally, the IDS-RAM aims to be applied across company borders and existing supply chains, and, to this end, it must be easy to adopt, technology-independent, reusable, and comply with existing technologies (Otto et al., 2019).

To cope with the data sovereignty challenges, the IDS-RAM specifies usage control, which is an extension of traditional access control, defining granular rules of what must (not) happen to data. Thus, usage control is concerned with requirements that pertain to data processing (obligations) rather than data access (provisions) (Eitel et al., 2021).

The IDS-RAM aims to meet multiple strategic requirement (see Figure 30):

- **Trust** is the basis for a Data Space environment, accomplished by the use of certification. Each participant (e.g., organisation) is evaluated and then certificated, only being capable of accessing the business ecosystem after the certification is granted. The same should happen for software components that share or consume data in the ecosystem.
- **Security and Data Sovereignty** are fulfilled by the evaluation and certification of each technical component. According to the IDS core aspect of ensuring data sovereignty, a data owner attaches the usage restriction information to their data before transferring it to the Data Consumer. To use the data, the Data Consumer must accept the data owner's usage policy (Otto et al., 2019). This process forces the data consumer to adhere to the previously negotiated usage policies (e.g., the data consumer can only access the data once) before the data is exchanged. A usage policy is a primary reason for enforcing usage control. It is an organisation's rule and is composed of an obligation, permission or prohibition.
- **Ecosystem of Data** is built using decentralised data storage. This means that until the data is transferred to a trusted party, it stays with the owner. This approach requires a description of the data and data sources. Additionally, the IDS Metadata Broker also offers data search capabilities.
- **Standardised Interoperability** requirement refers to the IDS Connector, which is the main technical component of the IDS-RAM. It can be implemented in different variants and acquired from different vendors. However, they all can communicate (Otto et al., 2019).

- **Value Adding Apps** are data services injected as Data Apps into the IDS Connector to provide services on top of data exchange processes, e.g., services for data processing, data format alignment, and data exchange protocols ².
- **Data Markets** enable the creation of novel data-driven services leveraging data apps. Also, they foster new business models for these services by providing clearing and billing mechanisms, domain-specific broker solutions, and marketplaces.

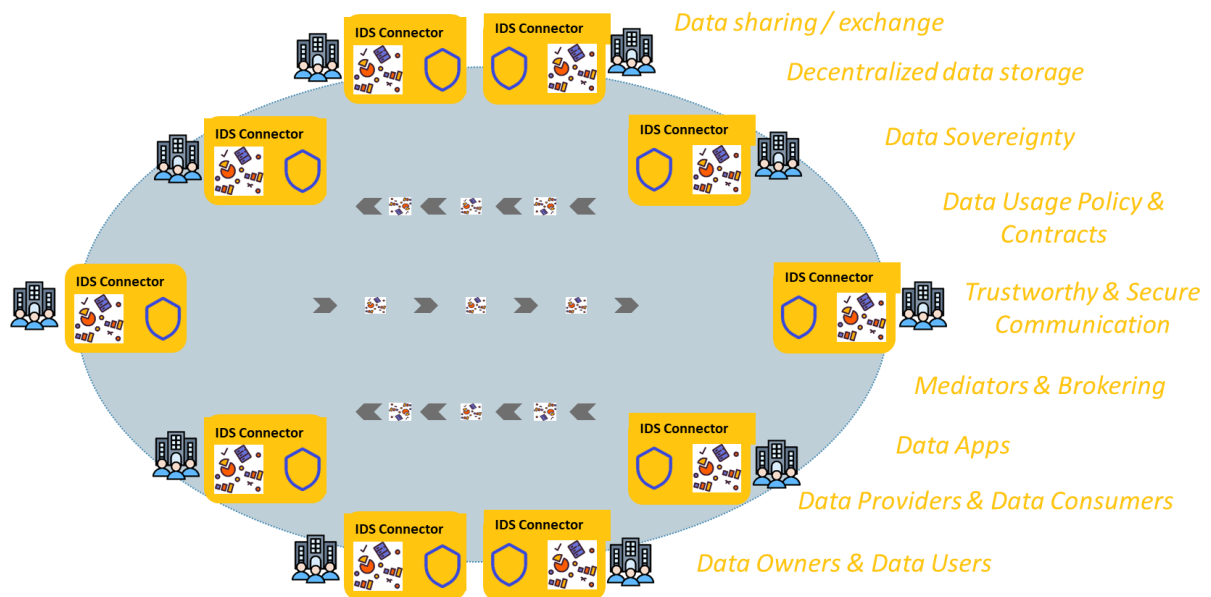


Figure 30: The main Data Space concepts defined by the IDS Reference Architecture Model

6.3 RENÉE Data Space and Major Components

The Data Space that has been created for RENÉE, with all its components, is shown in Figure 31 below. In the centre are the mediator components, responsible for enabling the information exchange and enforcing security and trust. Around them are the IDS Connectors, implemented by instances of the DataSpaceApp4EDI Connector, that manages data that comes/goes from/to external systems and human beings. The DataSpaceApp4EDI Connector, the IDS Identity Provider and the IDS Meta Data Broker modules are mandatory components. In the following sections, all RENÉE Data Space components are explained in detail.

² https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/Use-Case-Brochure_2018.pdf

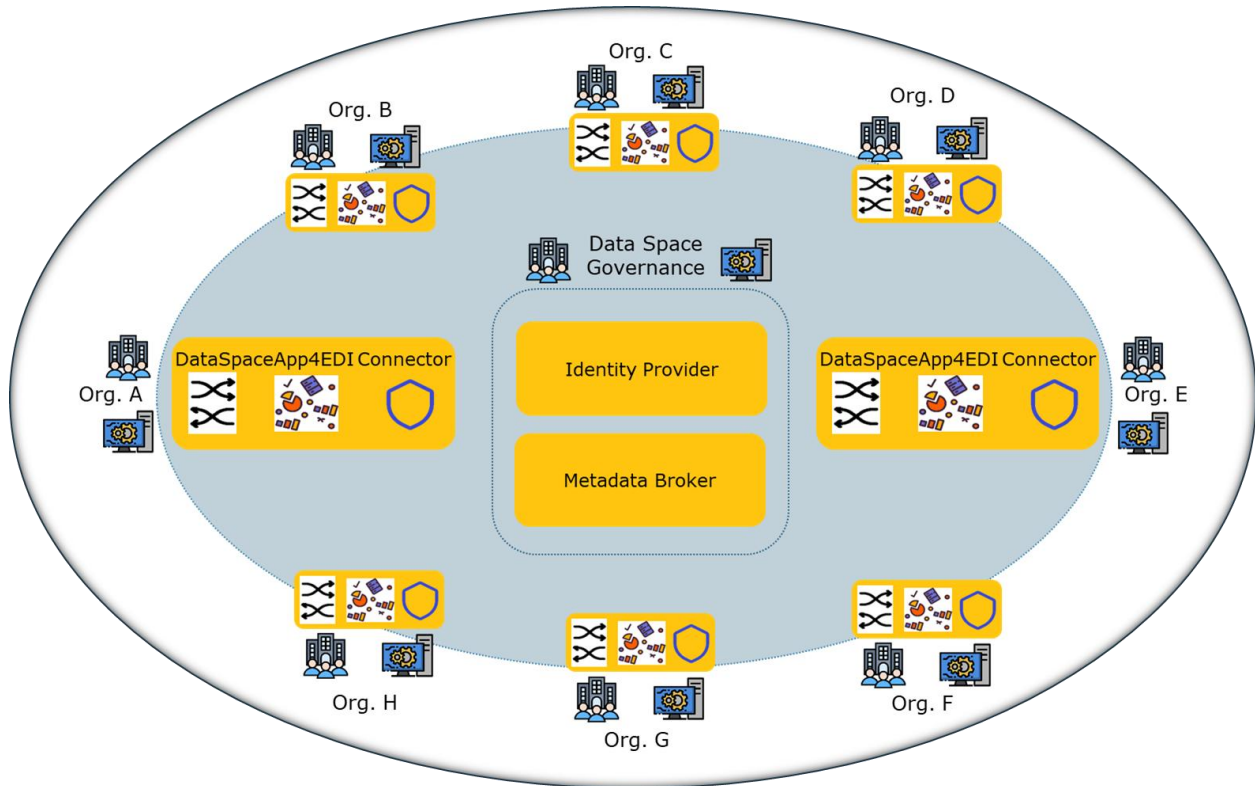


Figure 31: RENÉE Data Space

6.3.1 DataspaceApp4EDI Connector

The success of trusted data ecosystems requires IDS Connectors, as it is the main component for secure and trusted data exchange. Its primary function allows data owners and providers (Org. A ... Org. H in Figure 31) to exchange and share their data while ensuring data sovereignty. It is through an IDS Connector that an organization is able to share or to access data in Data Space ecosystem.

Additionally, the IDS Connector, implemented in the Data Space by the Dataspace4EDI Connector, is responsible for defining the usage restrictions for the data. Therefore, it allows the Data Provider to keep control of its data by defining restrictions such as how often each data resource can be shared, which Data User can access the data or how long the data can be available.

The Dataspace4EDI Connector is an implementation that was first developed by Fraunhofer ISST and later on adopted, adapted and evolved with new functionalities by INESC TEC. Simple communication interfaces are provided, a REST-based interface to external IT systems (shown on Figure 32 in the form of a Swagger page) and a user interface to be used by human beings.



Swagger Explore

DataspaceApp4EDI - renee - inesctec (SharedConnector) 1.17.35 OAS 3.1

/inesctec/api-docs

IDS App extending the IDS Dataspace Connector for supporting Electronic Data Interchange (EDI) and common data sharing scenarios.

Contact César Toscano

Apache License, Version 2.0, January 2004, <http://www.apache.org/licenses/>

Servers Authorize

- 1. EDI Artifacts - sender side
- 10. Traced Objects
- 2. EDI Artifacts - receiver side
- 3. Data Resources - provider side
- 4. Data Resources - consumer side
- 5. Data Resources - on Broker
- 6. Data Ecosystem
- 7. App Configuration
- 8. Connector Configuration
- 9. Keep-Alive / Status

Schemas

Figure 32: Dataspace Connector Swagger Page

6.3.2 Identity Provider

The Identity Provider is crucial since it avoids unauthorized access to data. It offers all the features associated with identity management and confirms the legitimacy of each participant (organization) and each IDS Connector (software). One of the Identity Provider's primary responsibilities is to create, preserve, and revoke Data Space identities for every Connector in a data space. In order for Connectors to communicate, their digital identify have to be confirmed by the Identity Provider. This mechanism prevents foreign, non-authorized Connector, to enter into the data space.

The software implementation used in RENÉE prototype uses a combination of open-source software, supplemented by developments of INESC TEC, in order to accomplish all the requirements of an IDS Identity Provider.

6.3.3 Metadata Broker

The Metadata Broker is responsible for describing all the data sources available in the data space and to know which IDS Connector provides the respective data. Its main goal is to promote new business partnerships by offering registry and discoverability functionalities. As this component is a mediator component managed by the Dataspace Governance entity, it is a trusted party and, therefore, it automatically creates trust between unknow participants (i.e., organizations).

6.4 Traceability use case

A specific application of the RENÉE Data Space enables the traceability of products and components in a value chain. Traceability is the ability to track the forward and backward movement, transformation, aggregation, or disaggregation of an asset (such as materials, products, or components) through specified stages of a remanufacturing supply chain. In this matter, it is essential to record the identifiers of inputs and outputs (in terms of components and/or products) and to establish a link between those articles and logistics unit(s). Generally, a record of who is doing what, when, where and why (the five Ws model) should be achieved.

A traceability system has been developed for RENÉE. As shown in Figure 33, all inbound and outbound activities, along with manufacturing activities carried out by production units throughout a value chain, are reported to a centralized traceability repository in the form of events that describe who is doing what, when, where and why. Collecting these events in the central repository enables the identification of the traced assets and their relationship in terms of previous and subsequent assets (components/products).

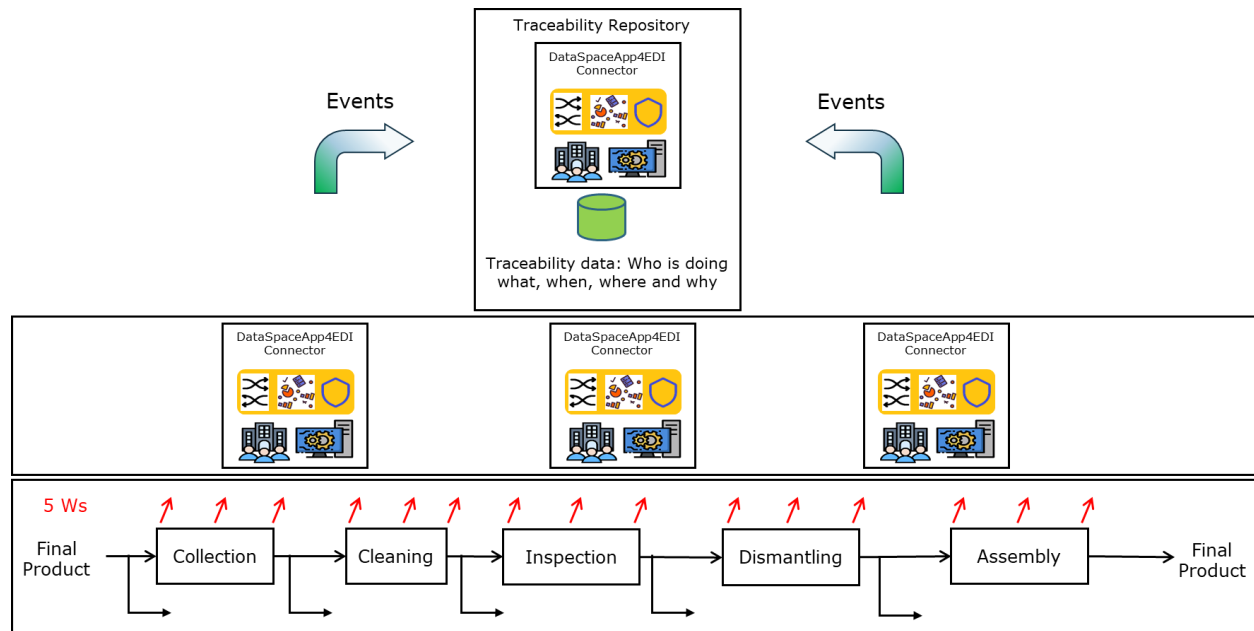


Figure 33: RENÉE Traceability System: major elements

A DataSpaceApp4EDI Connector is assigned to each production unit, being responsible for gathering information about the processed components and products and encapsulating this data in event messages sent to the central repository. On the central repository, another DataSpaceApp4EDI Connector receives events from all the production units, saving them in a relational database.

Specific data screens are provided by the DataSpaceApp4EDI Connector, allowing users in each production unit to describe the traceable assets and the related events and have access to the track history of any identified asset.

The screenshot shows the INESCTEC application interface. On the left is a dark sidebar with navigation options: Information, Traceability, New Traceability Object, Traced Objects (highlighted), IDS Scenario (Provider, Consumer, Broker), EDI Scenario (Sender, Receiver, Partnerships), and Other. The main content area has tabs for Information, New Traceability Object, and Traced Objects. A 'Reload' button and a message 'Any event performed on a Traced Object can be registered on this page by using the button in the 'Actions' column.' are visible. A 'Show Locked Objects' toggle is present. Below is a table with columns: Object ID, Product ID, Object Type, Instance Type, Quantity, Locked?, Registration Date Time, and Actions. The table contains six rows of data for various bicycle components.

Object ID	Product ID	Object Type	Instance Type	Quantity	Locked?	Registration Date Time	Actions
+ 0KN4ANVTRV...	1T7U4K	Saddle	Instance		No	2025-05-08@15:44:16	[Icons]
+ 0KN4BKNVCV...	IKGHTESEFG	Battery	Instance		No	2025-05-08@15:48:21	[Icons]
+ 0KN4BHJ30V...	7RECGTJ	Handlebar	Instance		No	2025-05-08@15:48:03	[Icons]
+ 0KN4C68H4V...	TGY37DG	Brakes	Instance		No	2025-05-08@15:50:53	[Icons]
+ 0KN4BJ2WVX...	7RECGTJ	Wheel	Instance		No	2025-05-08@15:48:11	[Icons]
+ 0KN4BGFQ4V...	45hrDI	Wheel	Instance		No	2025-05-08@15:47:54	[Icons]

Figure 34: DataSpaceApp4EDI Connector: identification of traced objects.

The data model for describing trace events consists of three primary entities: traceable objects, digital representations of physical assets; trace events, which represent the processing operations involving these assets; and traceability reports. Additionally, four types of events have been identified:

- **Object Events** represent operations that process one or more objects, where the identifiers of the objects remain unchanged.
- **Aggregation Events** represent operations involving the processing of one or more objects that are either physically combined or separated. The identifier of the parent object is distinct from the identifiers of the child objects.
- **Transformation Events** represent operations that fully or partially consume input objects and produce new output objects. The identifiers of the output and input objects are separate.
- **Transaction Events** represent cases where one or more objects become associated or disassociated with one or more identified trade transactions.

6.5 Data Analytics & ML for Supply Chain Optimization

This contribution initiates an investigation into supply chain data analytics aimed at enhancing traceability across manufacturing value chains, with a particular focus on remanufacturing processes. Recognising the importance of commercial confidentiality in cross-organisational collaboration, RENÉE’s approach prioritises data sovereignty from the outset. In the current phase, we are conducting a series of structured interviews with industrial partners to explore the availability, granularity, and format of traceability-related data within their operations. These dialogues are helping to map existing data infrastructures, identify commonalities across organisations, and surface early opportunities for alignment.

Although the overall framework is still under development, preliminary insights suggest that AI techniques—such as temporal pattern recognition, anomaly detection, and performance prediction—could support more informed decision-making across supplier networks. In order to



enable such applications without compromising sensitive information, federated learning emerges as a promising direction. This approach would allow machine learning models to be trained locally on proprietary datasets, sharing only model parameters rather than raw data. Future work will focus on formalising these findings into a structured methodology for secure, scalable, and interoperable supply chain analytics.

7 Conclusions and Future Activities

In this deliverable, the initial prototype of the RENÉE digital infrastructure has been defined and demonstrated. The prototype integrates essential modules that enable smart, traceable, and flexible remanufacturing management, including the Process Scheduler and Orchestrator (PSO), the dual-layer Digital Twin modules (S-DT and RC-DT), the Digital Vault (DV), and the Supplier Network Management Tool (SNMT). The communication mechanisms and data exchange processes among these modules were defined within a modular and scalable architecture, supported by Kubernetes-managed deployment.

The interconnectivity and functionality of these modules align with the four pillars of the RENÉE architecture: circular value chains, robot skills and flexible production, digital infrastructure, and workforce upskilling and human factors. The current prototype represents a robust technological basis for further integration and testing.

Future work will focus on the implementation of the final prototype, building upon the functional foundation set by the initial system. This next phase will involve the full deployment of the digital infrastructure in selected case studies, with emphasis on extending the orchestration capabilities, refining the digital twins with increased real-time responsiveness, and completing the standardization and integration of AAS models for products, processes, and resources. This extended deployment will validate the RENÉE infrastructure under real operational conditions and strengthen its potential for broader adoption in circular manufacturing ecosystems.



References

- Antonin RAFFIN, Quentin Gallouédec, Noah Dormann, Adam Gleave, Anssi, Alex Pasquali, Juan Rocamonde, M. Ernestus, Patrick Helm, Corentin, Quinn, Thomas Simonini, Tobias Rohrer, Sidney Tio, Mark Towers, Rohan Tangri, Tom Dörr, Wilson, Steven H. Wang, ... Grégoire Passault. (2025). *DLR-RM/stable-baselines3: V2.6.0: New ``LogEveryNTimesteps`` callback and ``has_attr`` method, refactored hyperparameter optimization* (Version v2.6.0) [Computer software]. Zenodo. <https://doi.org/10.5281/ZENODO.8123988>
- Bertsimas, D., & Tsitsiklis, J. (1993). Simulated Annealing. *Statistical Science*, 8(1), 10–15. <https://doi.org/10.1214/ss/1177011077>
- Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade. *Queue*, 14(1), 70–93. <https://doi.org/10.1145/2898442.2898444>
- Chodorow, K. (2013). *MongoDB: The definitive guide* (Second edition). O'Reilly.
- Eitel, A., Jung, C., Brandstädter, R., Hosseinzadeh, A., Bader, S., Kühnle, Ch., Birnstill, P., Brost, G., Gall, Bruckner, F., Weißenberg, N., & Korth, B. (2021). *Usage Control in the International Data Spaces (Position Paper)*. <https://doi.org/10.5281/ZENODO.5268926>
- Equilibrium optimizer: A novel optimization algorithm. (2020). *Knowledge-Based Systems*, 191, 105190. <https://doi.org/10.1016/j.knosys.2019.105190>
- ForrestStephanie. (1996). Genetic algorithms. *ACM Computing Surveys (CSUR)*. <https://doi.org/10.1145/234313.234350>
- Otto, B., Lohmann, S., Auer, S., Brost, G., Cirullies, J., Eitel, A., & et al. (2019). *Reference Architecture Model, IDSA*.