Technologies for Manufacturing as a Service Ecosystems

Deliverable 4.1

Tech4MaaSEs integrated platform v1

WP4: Integration and pilot deployments

Editor:	Sotiria Antaranian
Lead beneficiary:	ATC
Version:	1.0
Status:	Final
Delivery date:	30/06/2025
Dissemination level:	PU

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101138517.

Deliverable Factsheet

Grant Agreement No.	101138517
Project Acronym	Tec4MaaSEs
Project Title	Technologies for Manufacturing as a Service Ecosystems
Start date	01/01/2024
Duration	36 months

Deliverable Name	D4.1 Tech4MaaSEs integrated platform v1
Related WP	WP4 Integration and pilot deployments
Due Date	30/06/2025

Author	Sotiria Antaranian (ATC)
Contributor(s)	Giorgos Triantafyllou, Panagiotis Stergioulas Bolis, Alkiviadis Aznavouridis (ATC), Ljiljana Stojanovic (IOSB), Patricia Casla, Álvaro García, Ignacio Lázaro, Gonzalo Gil (TEK), George Zois, Yiannis Mourtos, Ioannis Avgerinos (AUEB), Yuanwei Qu, Martin G. Skjæveland (UiO), Kyriakos Bitsis, Konstantinos Kaparis, Dimitris Thomaidis, Andreas Georgiou, Ioannis Konstantaras, Angelo Sifaleras, Georgios Tsaples (UoM), Eleni Gkinou, Eleni Kartsonaki, Dimitris Verlekis (MAG), Mert Can Özcan (SMO), Marc Haller (IOSB)
Reviewer(s)	Andreas Georgiou (UoM), Metin Turkay (SmartOpt)
Approved by	All partners

Document History

Version	Date	Author(s)	Organisation	Description
0.1	11/02/2025	Sotiria Antaranian	ATC	ToC
0.2	21/05/2025	Ljiljana Stojanovic	IOSB	Input on Digital Twins
0.3	21/05/2025	Sotiria Antaranian	ATC	Initial input
0.4	26/05/2025	Patricia Casla, Álvaro García, Ignacio Lázaro, Gonzalo Gil	Tekniker	Partner contribution
0.5	16/06/2025	Kyriakos Bitsis, Konstantinos Kaparis, Dimitris Thomaidis, Andreas Georgiou, George Zois, Yiannis Mourtos, Ioannis Avgerinos, Marc Haller	UoM AUEB IOSB	Partner contribution
0.6	18/06/2025	Mert Can Özcan, Yuanwei Qu, Martin G. Skjæveland	SMO UiO	Partner contribution
0.7	23/06/2025	Eleni Gkinou, Eleni Kartsonaki, Dimitris Verlekis	MAG	Input on Supply Chain Digital Twins, Balanced Scorecard and AI Model Governance
0.7	24/06/2025	Sotiria Antaranian, Giorgos Triantafyllou, Panagiotis Stergioulas Bolis, Alkiviadis Aznavouridis	ATC	Preparation for peer review
0.8	25/06/2025	Andreas Georgiou, Metin Turkay	UoM, SMO	Peer review
0.9	26/06/2025	Sotiria Antaranian	ATC	Integration of comments from internal review
0.10	30/06/2025	Armend Duzha	MAG	Quality Check
1.0	30/06/2025	Sotiria Antaranian	ATC	Final version ready for submission

Executive Summary

This document describes the closed alpha release of the Tec4MaaSEs integration prototype, v0.5. As a result, this refers to the first working version of the prototype that offers an initial set of the expected functionalities, and acts as the testbed for the Tec4MaaSEs stakeholders to experience the initial set of tools, to assess their performance and also the concept and functionalities conveyed by the project.

The deliverable provides an overview of the Design principles of Software & System and presents the integration of multiple components. Then it offers the project's current functionalities in the form of a user guide, exposing every available feature so far. The scope of this document is to act as an appendix to the current version of the Tec4MaaSEs initial integrated release.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Acronym	Description
T4M	Tec4MaaSEs
MaaS	Manufacturing as a Service
vCPU	Virtual Central Processing Unit
GB	Gigabyte
RAM	Random Access Memory
IDE	Integrated Development Environment
UI	User Interface
CI/CD	Continuous Integration / Continuous Deployment
VN	Value Network
VNs	Value Networks
FA³ST	Fraunhofer Advanced Asset Administration Shell Tools
API	Application Programming Interface
AAS	Asset Administration Shell
IMF	Information Model Framework
BOP	Bill of Processes
TDC	Tekniker Dataspace Connector
DT	Digital Twin
OPC UA	Open Platform Communications Unified Architecture
MQTT	Message Queuing Telemetry Transport
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
JAR	Java Archive
LLM	Large Language Model
RDF	Resource Description Framework
OWL	Web Ontology Language
XSL	Extensible Stylesheet Language
GraphML	Graph Markup Language
REST API	Representational State Transfer Application Programming Interface
2D / 3D CAD	2-Dimensional / 3-Dimensional Computer-Aided Design
ARIMA	AutoRegressive Integrated Moving Average
SES	Simple Exponential Smoothing

Acronym	Description
KPI	Key Performance Indicator
ETL	Extract, Transform, Load (data processing pipeline)
SMT	Submodel Template
CSV	Comma-Separated Values
PDF	Portable Document Format
URL	Uniform Resource Locator
AASX	AAS Package Exchange Format
SM	Submodel
US	User Story
UC	Use Case

1 Introduction

1.1 Purpose and Scope

This deliverable aims to provide an overview of the closed alpha release of the Tec4MaaSEs system, developed as part of the Tec4MaaSEs platform for collaborations in a common environment among consumers and providers of Manufacturing as a Service ecosystems and related Value Networks. The document details the current functionality of the dashboard and presents a user guide to facilitate its adoption and use.

The Work Package 4 (Integration and Pilot deployments) is responsible for the implementation and integration of Tec4MaaSEs components, ensuring compliance with the requirements, architecture and technical specifications defined in Work Package 2 (Reference framework, specifications and core enablers) and to integrate the DT Models and tools developed in Work Package 3 (DT Modelling, Operation and Governance for resilient value networks). The design of the Tec4MaaSEs platform is driven by these specifications to support basic user operations (user/organization management, system notifications) together with more advanced tools such as Dataspace Connectors, Search & Match, Optimization, and Forecasting. These are complemented by more advanced services such as Bill of Process Generator, Predictive/Proactive Analytics, and Supply Chain Management.

Following the development of individual tools such as the ones described above, the current deliverable also describes the 1st version of the integrated Tec4MaaSEs platform as conceptualized to provide the primary user interface for managing Manufacturing as a Service (MaaS) operation. The content presented in this deliverable is subject to refinement based on further technical advancements and feedback from all partners and tools during the validation phase of the project.

The demonstrator described in the context of this deliverable, aiming to provide the closed alpha release (1st prototype) of the integrated Tec4MaaSEs platform, can be accessed via the following URL for accessing the platform: https://platform.t4m.atc.gr/dashboard

The Organization Registration Page is accessible here: https://platform.t4m.atc.gr/organization-management/register

Credentials for demo purposes are provided upon request.

1.2 Relation with other deliverables

The current deliverable is the 1st version of the Tec4MaaSEs Integrated platform aiming to provide the initial version of the integrated platform; thus it is the direct result of Task 4.4 and is incorporating the results of all technical WPs involved in the implementation phase, i.e., WP2, WP3 and WP4.

As a 1st version it encapsules results from individual tool development results and lays the ground for the upcoming Deliverable “D4.2 – Tec4MaaSEs Integrated platform v2”, which will be released in M34 and will include the final version of the platform.

The current document is also closely related to:

- Deliverable “*D2.6 - Tec4MaaSEs architecture blueprint and specifications v2*”, due for M24, as the next (final) version of the architecture will benefit from issues highlighted in the 1st platform version

- Deliverable "*D3.1 - Digital twins models and services for factories and value networks v1*", also due for M18, presenting the DT models and services at factory level, which includes models and services created according to operation of the platform at the current moment.
- Deliverable "*D3.3 - Analytics for resource-subservice matching and service composition v1*", due also for M18, which will host the current status of analytics and the way these interoperate with the platform
- Deliverable "*D5.1 - Pilot validation plan and assessment report*", due for M18, as the validation plan and the foreseen assessment report will be based upon the current status of the system and the way tools integrated together.
- Deliverable "*D5.2 - Pilots execution and evaluation v1*", due for M22, since the 1st round of Pilots execution and the evaluation of Value Networks operation will be structured around the way the 1st version of the system operates as depicted in the current Deliverable.

1.3 Structure of the document

The document is structured as follows:

- **Section 2** introduces the Software and System Design principles and includes description of Operational environment, an overview of the architecture and the key platform functionalities. These are complemented by the components that are integrated in the platform, which are thoroughly presented in the Appendix A, within the "Components v0.5" Annex including the status of each tool developed in the context of the project.
- **Section 3** is devoted to a User Guide (in form of a Manual) which describes theTec4MaaSEs Central User Interface, for the Analytics Dashboard, the Supply Chain Monitoring and the AAS Model Generator and for the Information Model Framework.
- **Section 4** summarizes the conclusions of the current document, including the outcomes of the current phase and an overview of what is anticipated in the next project period.

2 Software and System Design principles

The section presents the latest system architecture, illustrating the relationships and interactions between its various components. Complementing this, it highlights the key functionalities of the T4M platform, accompanied by their corresponding user stories, to provide a clear understanding of how technical design translates into tangible user benefits and system capabilities. This section also contains some main principles of technical design which have been applied to the development of the T4M solution. Selecting design principles is critical for creating complex software structures and doing it properly at the initial stages of the project leads to better results in the long term in terms of scalability, availability, reliability, and reduced maintenance costs.

2.1 System Architecture Overview

Based on *"D2.5 System Architecture v1"*, the latest T4M system architecture can be depicted in detail below in Figure 1.

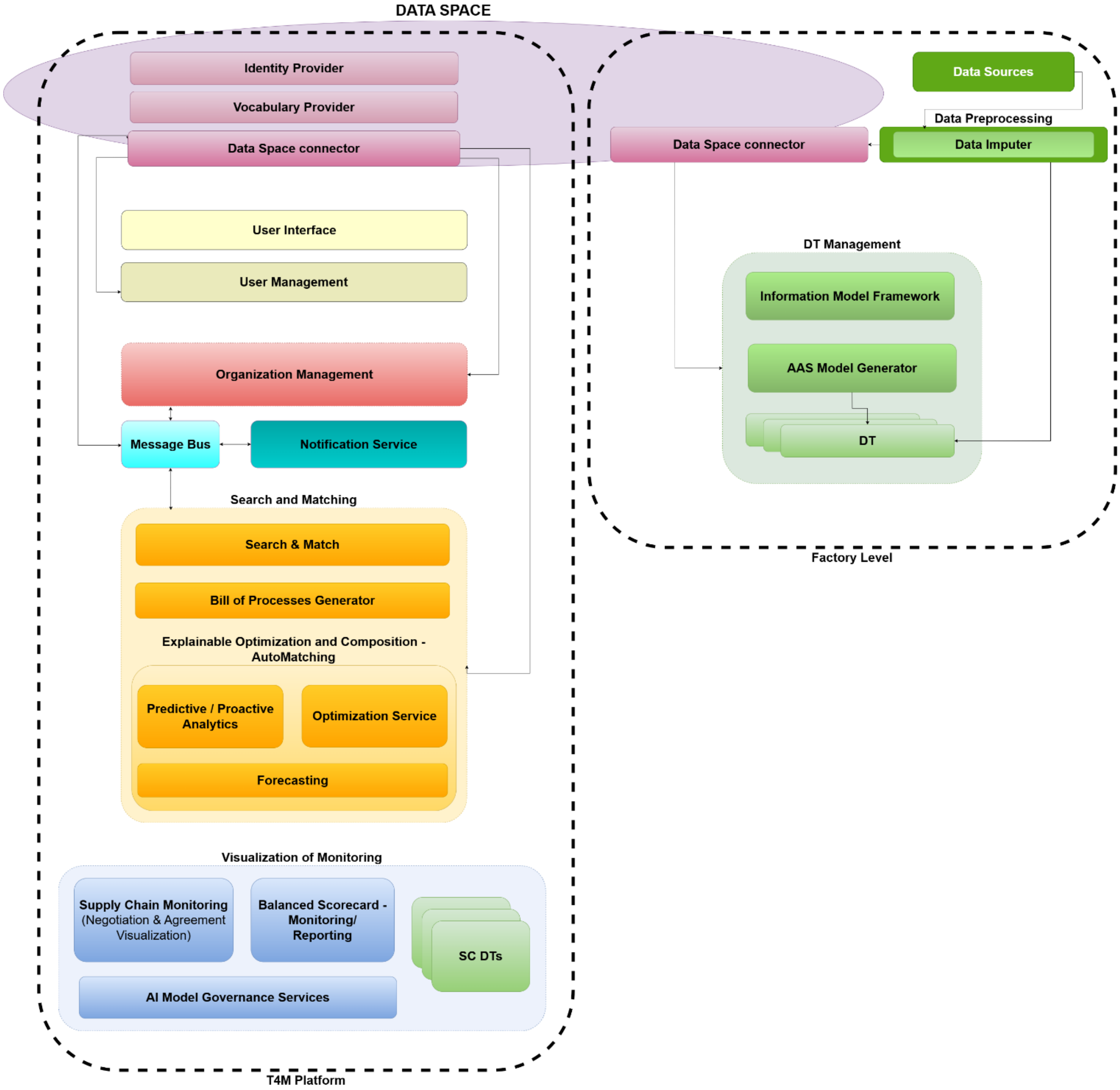

Figure 1: T4M Architecture

This latest version bares some differences from the one presented in D2.5. This version was updated to focus on the practical implementation aspects required for the current prototype. Deliverable 2.6 will provide a more comprehensive and finalized system architecture, including design decisions and a detailed comparison of the two versions.

2.2 Key platform functionalities

This deliverable presents version 0.5 (v0.5) of the T4M platform, a functioning intermediate prototype developed to demonstrate the technical progress achieved to date, validate key functionalities, and gather early input from end-users.

This version integrates the core components necessary to showcase the platform's architecture and integration strategy, serving as a foundation for confirming essential concepts. The final release, version 1.0, will be delivered by the end of the project, featuring the complete functionality set, enhanced performance, improved interoperability, and refinements based on testing and stakeholder feedback.

The table below, Table 1, outlines the key functionalities included in version 0.5 of the T4M platform. These represent initial implementations of core features that address specific user stories, as defined in *"D2.1 Reference cases and actionable models for reconfigurable value networks and service decomposition v1"*, and workflows across various value networks (VNs). While these functionalities establish the foundation for end-to-end operations, they are not yet fully finalized or integrated across all components. Several features will be updated, extended, or refined in upcoming releases, and some components may currently rely on placeholder or standalone implementations pending full integration.

Table 1: Key platform functionalities v0.5

#	User Stories	Title	VNs	Main Components
1	US0.1 & US0.4	User & Role Management Operations, Account Management and Access Operations	All	User Management, User Interface
2	US0.2	Supply Chain Configuration and Overview	All	Supply Chain Monitoring
3	US2.O2	Manage Data Access	VN2	AAS Model Generator, Data Space Connector
4	--	Preparatory Activities => Capabilities DTs	VN2	AAS Model Generator, Digital Twin, Data Space Connector
5	US3.3	Information Model Creation and Update	VN3	Information Model Framework
6	US1.1, US2.O1, US3.1	Provider/Consumer Organization Registration	All	Organization Management, User Interface
7	US1.3, US2.O3, US3.1	Validate Onboarding Process	All	Organization Management, Data Space Connector
8	US1.4, US1.5, US2.P1, US2.P2 (UC2.P2.1)	Request and Extract Manufacturing Service - initial version	VN1, VN2	Search&Match (& Bill of Processes Generator for VN2)
9	US2.P2 (UC2.P2.2), US2.P3	Trigger Optimization, Match and Display Provider Configurations & Request Service Quotation	VN2	Optimization Service, Predictive/Proactive Analytics
10	US1.7 (UC1.8)	Upload Production Schedule, Inventory and forecasted ETAs	VN1	Forecasting
11	--	Manual Search – find providers based on location and Manufacturing Service	All	Organization Management, User Interface
12	US1.8, US3.4, US3.5, US3.6, US3.7	Review, and Negotiate Manufacturing Service Request - one negotiation loop	VN1, VN3	Supply Chain Monitoring (& Information Model Framework for VN3)

2.3 Operational Environment

The operational environment for hosting and facilitating the components and layers of the T4M ecosystem was set upon a dedicated Kubernetes cluster (Version: v1.32.4), named "T4M cloud [1]." The choice of Kubernetes itself indicates a strong commitment to modern, cloud-native infrastructure, a decision that inherently requires specialized knowledge and significant setup.

This "T4M cloud" is a carefully configured cluster hosted on Azure, comprising two Linux server worker nodes. Each cluster node consists of 8 vCPUs and 32 GB of RAM and further complemented by 500 GB of shared disk storage for the cluster (see Figure 2). This selection of resources reflects an in-depth analysis of the T4M components' resource requirements, meticulously ensuring ample capacity for both current and future demands. As shown by the well-below-capacity CPU and memory utilization in Figure 3, which provides a real-time overview of CPU, memory, and pod utilization across the cluster, the resource allocation is optimized for efficiency and scalability.

Name	CPU	Memory	Disk	Taints	Roles	Version	Age	Conditions
aks-agentpool-32358320-vmss000009				0		v1.32.4	5h45m	Ready
aks-agentpool-32358320-vmss00000a				0		v1.32.4	5h45m	Ready

Figure 2: Kubernetes Node Overview

Every component is deployed, maintained, and scaled on these 2 nodes. Additionally, the Lens IDE was used to provide a graphical user interface for administering the Kubernetes cluster. This means each distinct component within the T4M ecosystem has been containerized and orchestrated. This involves writing Dockerfiles, defining Kubernetes Deployment, Service, and potentially Ingress resources for each component [2]. The use of Lens IDE further highlights the active administration and fine-tuning required to manage such a complex environment, providing a graphical interface to navigate the intricacies of pods, services, and namespaces [3].

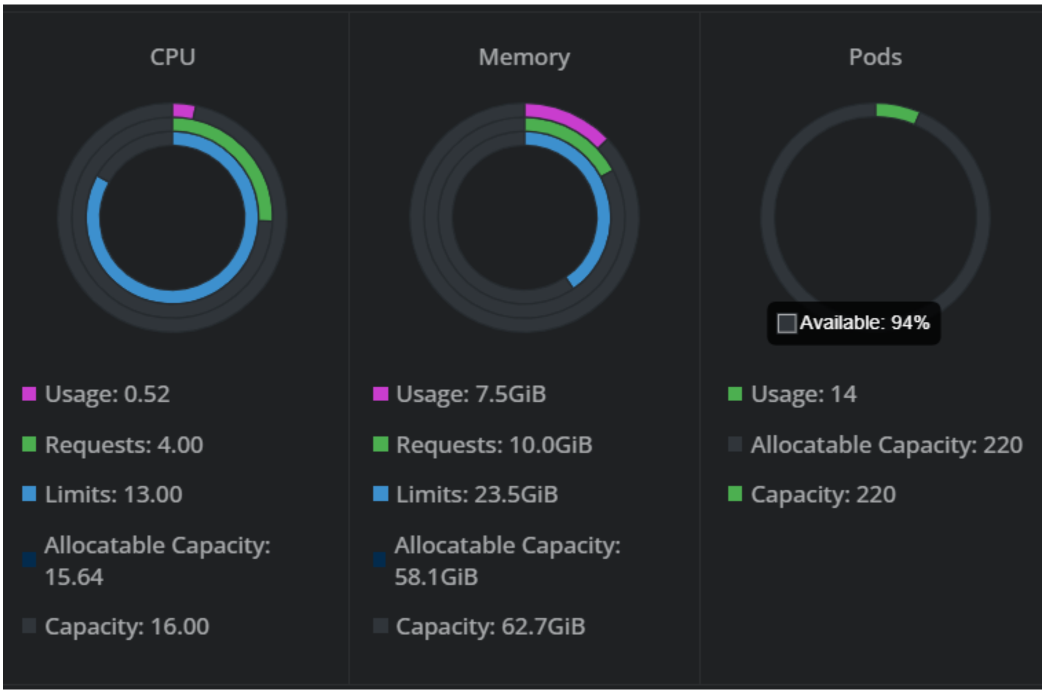

Figure 3: Cluster Resource Usage

Figure 4 lists the ingress controllers deployed across multiple namespaces such as tec4maases and monitoring. Each ingress exposes services like Keycloak, MinIO, and Grafana via LoadBalancer IPs, along with their mapped rules and uptime information. Each of these exposures requires specific Kubernetes Ingress rules, certificate management, and LoadBalancer provisioning, representing a significant layer of deployment complexity.

Name	Namespace	LoadBalancers	Rules	Age
conduktor-ingress	tec4maases	4.225.34.0	https://conduktor.t4m.atc.gr/ → conduktor-console...	52m
grafana-ingress	monitoring	4.225.34.0	https://monitoring.t4m.atc.gr/ → loki-stack-grafana:...	133m
keycloak-ingress	tec4maases	4.225.34.0	https://kc.t4m.atc.gr/ → keycloak-service:8080	3d1h
minio-api-ingress	tec4maases	4.225.34.0	https://minio.api.t4m.atc.gr/ → minio-api-service:90...	3d1h
minio-console-ingress	tec4maases	4.225.34.0	https://minio.t4m.atc.gr/ → minio-console-service:9...	3d1h
notification-serv-ingress	tec4maases	4.225.34.0	https://services.t4m.atc.gr/api/notifications → notifi... https://services.t4m.atc.gr/api/event-mappings → n... https://services.t4m.atc.gr/api/notification-service ...	2d22h
organization-management-ingress	tec4maases	4.225.34.0	https://services.t4m.atc.gr/api/organization → orga... https://services.t4m.atc.gr/organization → organiza...	3h22m
user-manager-ingress	tec4maases	4.225.34.0	https://services.t4m.atc.gr/api/users → user-manag... https://services.t4m.atc.gr/api/admin → user-mana... https://services.t4m.atc.gr/api/user-manager → use...	3d3h

Figure 4: Ingress Endpoints and Services

Each component is mapped to a dedicated deployment configuration that handles the continuous integration and deployment, scale up and monitoring of the component (Figure 5).

tec4maases | Search Deployments... | 13 items

Name	Namespace	Pods	Replicas	Age	Status
bill-of-processes-deployment	tec4maases	1/1	1	3h10m	Running
conduktor-postgres-deployment	tec4maases	1/1	1	66m	Running
forecasting-deployment	tec4maases	1/1	1	173m	Running
keycloak-deployment	tec4maases	1/1	1	4h40m	Running
keycloak-postgres-deployment	tec4maases	1/1	1	5h1m	Running
minio-deployment	tec4maases	1/1	1	3d1h	Running
notification-serv-deployment	tec4maases	1/1	1	5h7m	Running
org-mongo-deployment	tec4maases	1/1	1	4h33m	Running
organization-management-deployment	tec4maases	1/1	1	4h14m	Running
postgres-notif-deployment	tec4maases	1/1	1	5h7m	Running
search-and-match-deployment	tec4maases	1/1	1	95m	Running
sm-mongo-deployment	tec4maases	1/1	1	3h54m	Running
user-manager-deployment	tec4maases	1/1	1	3d22h	Running

Figure 5: Deployed components

Additionally, a Grafana and Prometheus instance is deployed to effectively monitor the performance of each component in the environment and report incidents [4], [5]. Figure 6 provides a real-time (or near real-time) view of a Spring Boot application running within the Kubernetes environment, with a focus on its startup process, performance metrics, and detailed log messages.

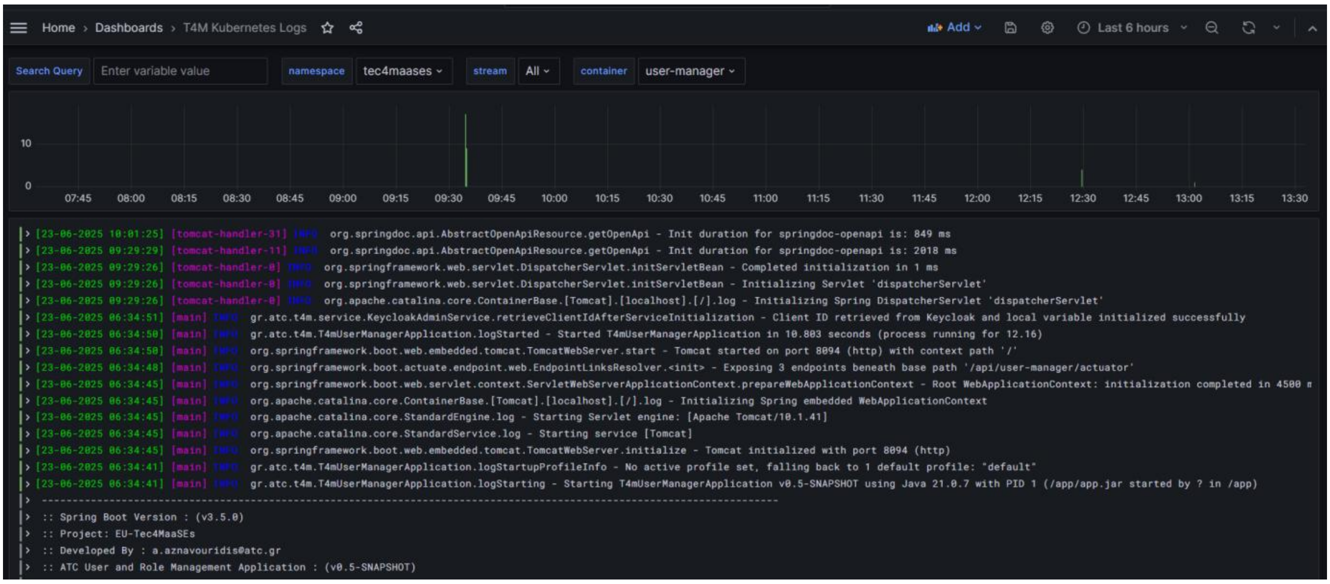

Figure 6: Monitoring

Finally, combining GitHub CI/CD features with the GitHub Container Registry, an end-to-end Continuous Integration and Continuous Deployment mechanism is established, enabling automated deployments upon code pushes for components hosted on GitHub [6]. However, not all components follow this flow—some may pull images from alternative registries or be hosted externally due to various access, integration, or infrastructure-related considerations. Future steps aim to gradually address and reduce these limitations where feasible.

3 Manual – User Guide

This section provides user-oriented guidance through a series of annotated screenshots that illustrate the key functionalities of the T4M platform from an end-user perspective. Each image highlights specific user actions and interactions across the system, covering the full range of features available in the current version, v0.5. The goal is to offer a clear, visual walkthrough of how users engage with the platform, supporting both onboarding and operational understanding. There is a correlation between this section and Table 1: Key platform functionalities v0.5, mainly for the functionalities with UI.

3.1 User Manual for T4M Central User Interface

This sub-section, which covers #1, #6, #8 and #11 from Table 1, serves as a comprehensive user manual, specifically detailing the functionalities of the T4M Central User Interface. Through a series of annotated screenshots, users will be guided step-by-step through the primary interactive elements and workflows. The essential user journeys will be covered, from initial access, registration and service requests to managing users and user roles, ensuring a thorough understanding of the v0.5 platform's capabilities. This guide is designed to facilitate seamless onboarding and provide a clear reference for everyday operational use of the T4M Central UI. Throughout the multiple forms that the user can fill in, there are "i – info icons" that the user can click or hover over to get information explaining each field.

The T4M UI is accessible via this link: https://platform.t4m.atc.gr/dashboard

The Organization Registration Page is accessible here: https://platform.t4m.atc.gr/organization-management/register

Credentials for demo purposes are provided upon request.

3.1.1 Super Admin Creation

The System Administrator creates Super Admins who will be able to access the whole system, create users and user role.

The System Administrator accesses Keycloak and the Users Dashboard, where they can select "Add user" (Figure 7).

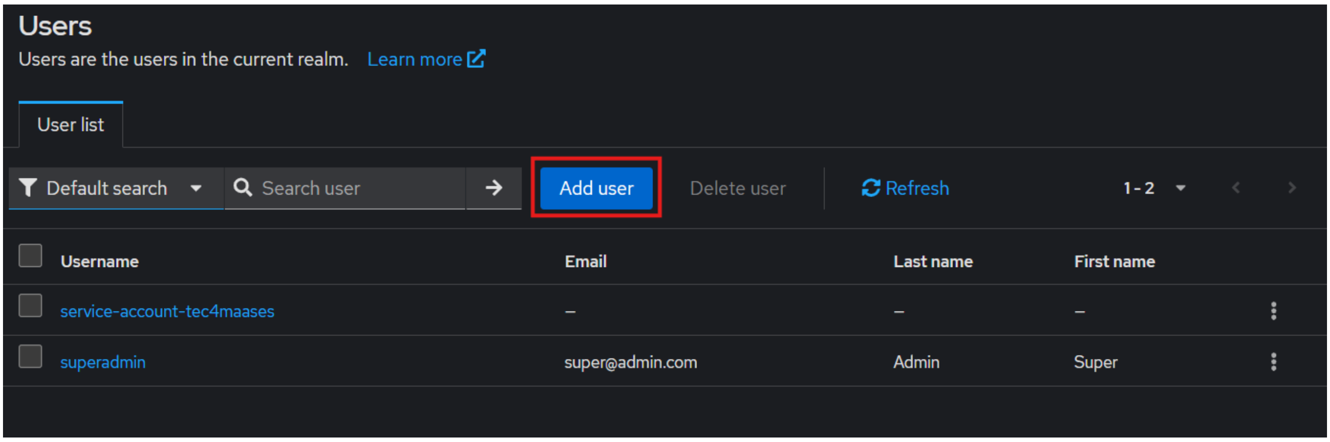

Figure 7: Users Dashboard in Keycloak

Then, they fill in the initial required information (Figure 8) and view the newly created user by selecting them for the list of users (Figure 9).

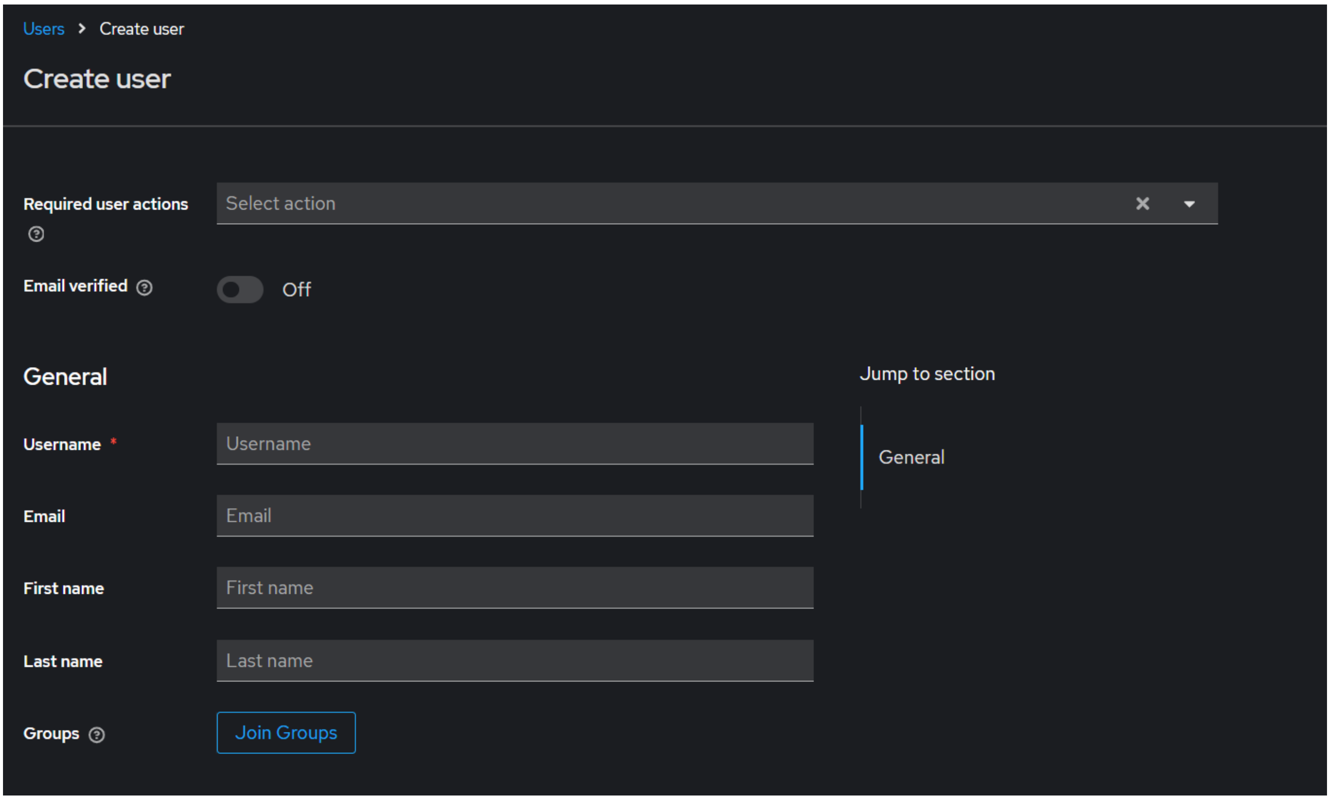

Figure 8: Create User in Keycloak

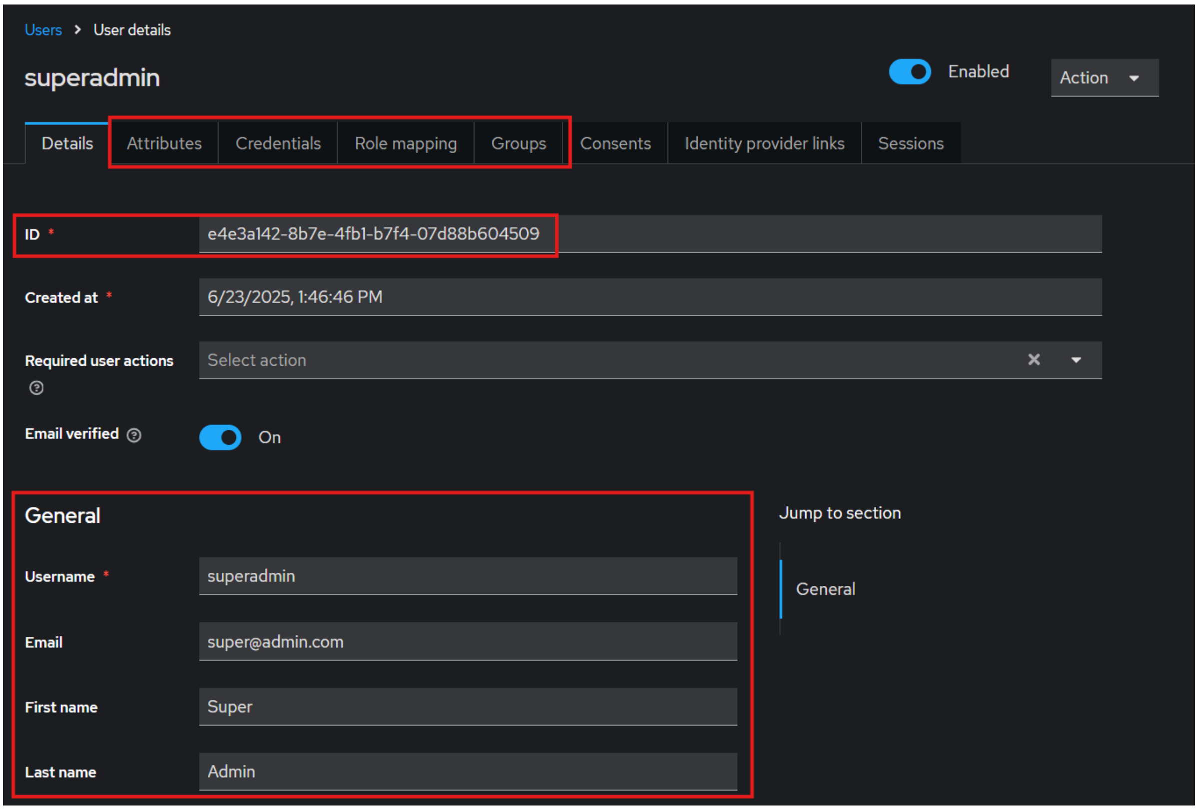

Figure 9: View User in Keycloak

From there and the tab “Credentials”, they set up the password (Figure 10).

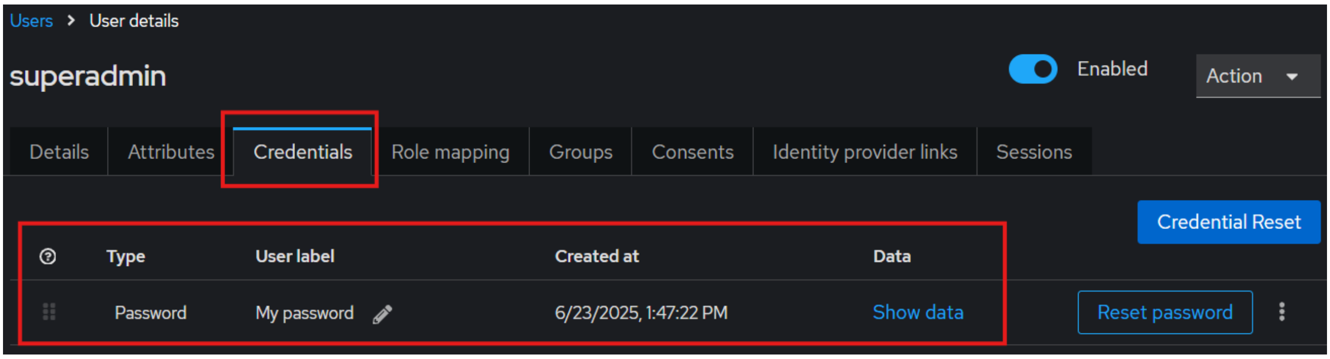

Figure 10: Set User Credentials

Finally, the System Administrator should select the appropriate role for a "Super Admin" user from the tab "Role Mapping" (Figure 11).

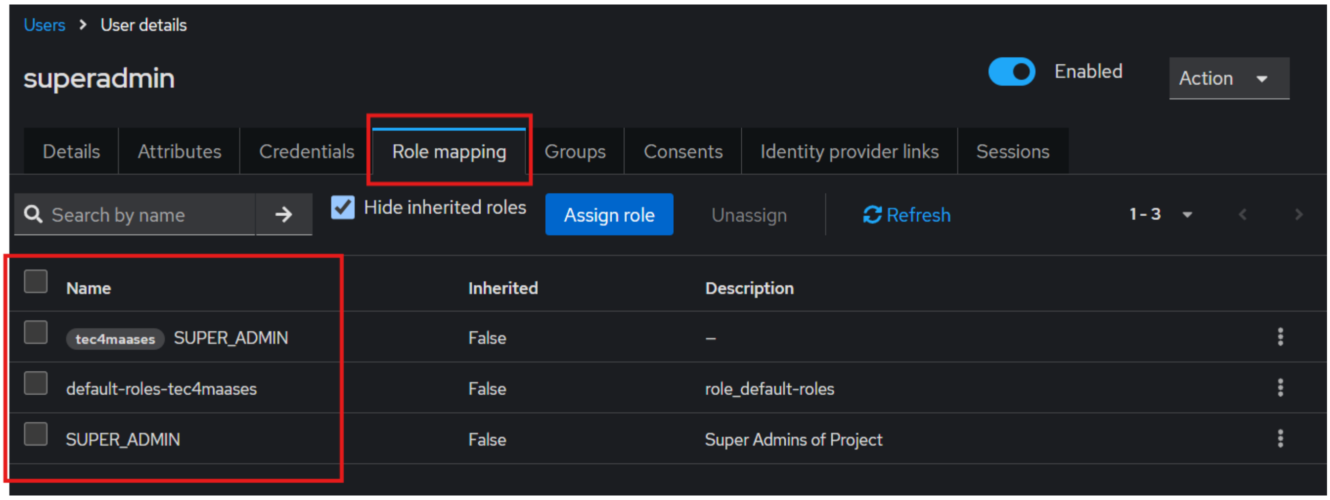

Figure 11: Set User Role

The Super Admin can now access the T4M platform and create other users and user roles.

3.1.2 Authentication Page

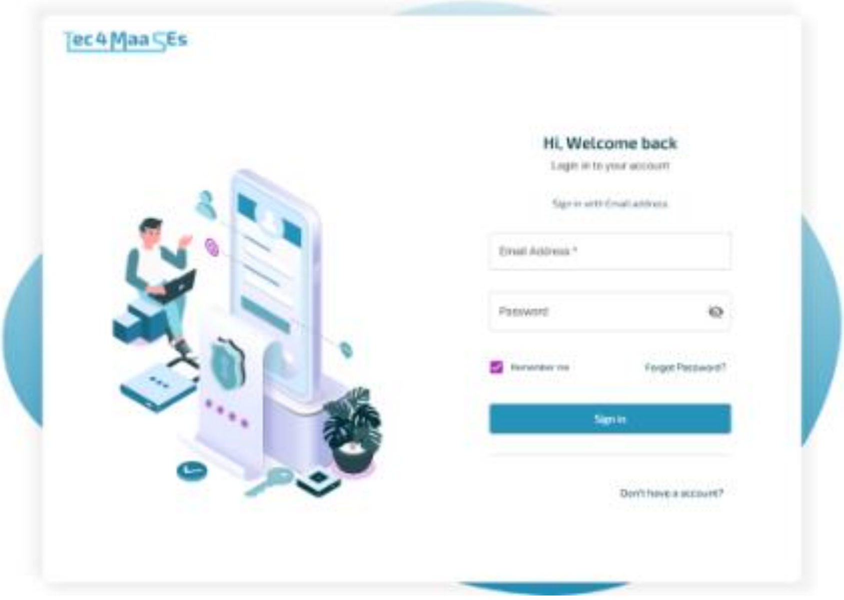

Figure 12: Authentication Page

On this page, as seen in Figure 12, the user is required to enter their credentials—specifically, their email address and password—into the designated input fields. Once the information has been provided, clicking the Sign In button will authenticate the user and grant access to the application.

3.1.3 User List Page

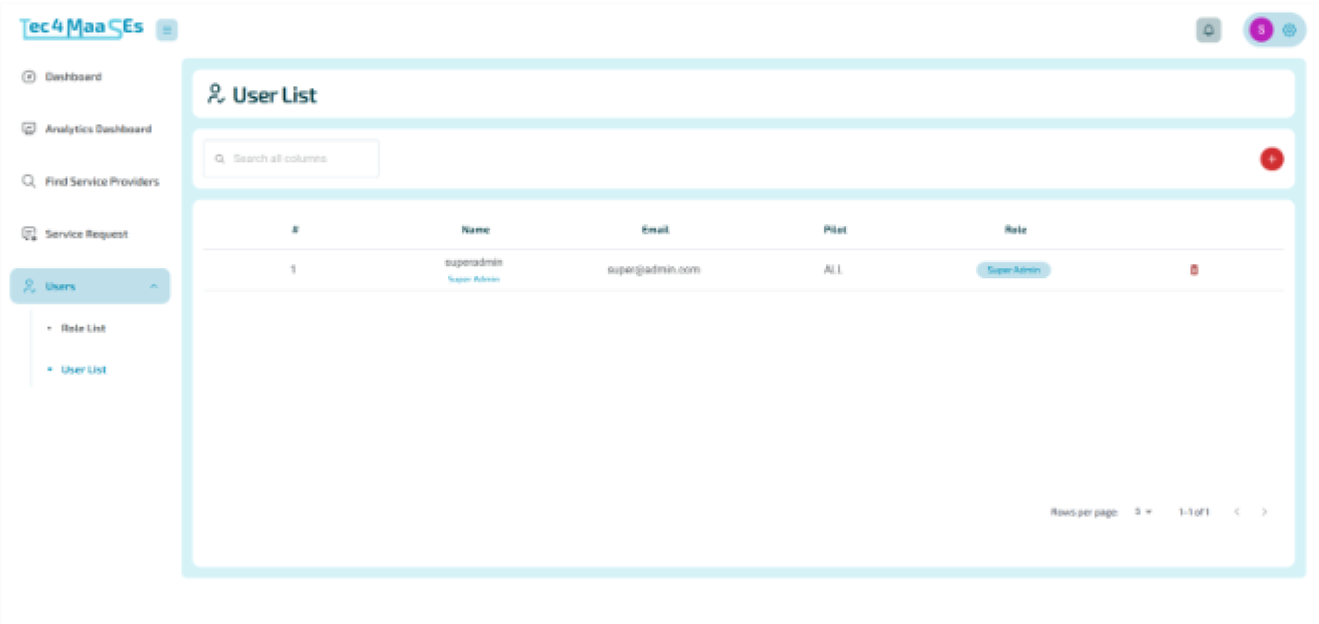

Figure 13: User List Page

The above page (Figure 13), along with the **Users** menu item, is accessible exclusively to Admins and Super Admin users.

By clicking the **"Users"** menu item in the sidebar and then selecting the **"User List"** option, the user is navigated to the **User List** page. This page displays a table containing all registered users along with their associated information. This page allows Super Admins to view and manage user data efficiently within the application.

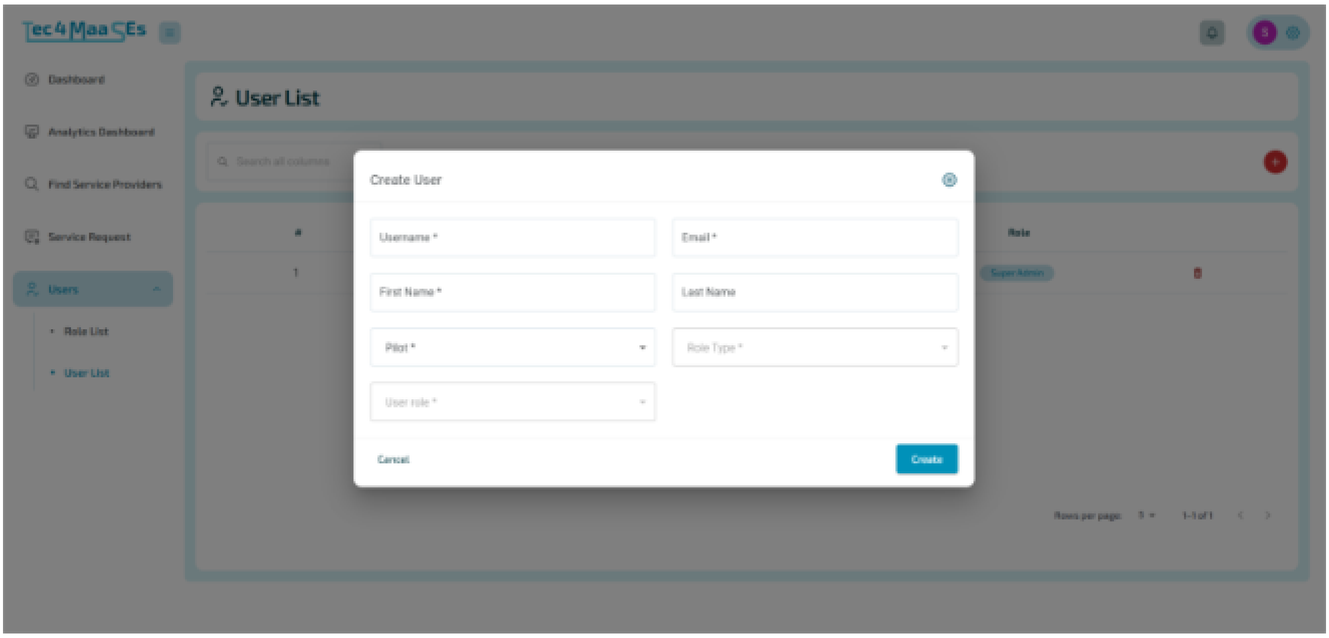

Figure 14: Create User on T4M platform

When the user clicks the red plus icon, a pop-up window will appear prompting them to complete a form to create a new user (Figure 14). After creating a new user, the user will receive an email with an activation link (Figure 15), which will redirect them to the platform to set their password. After that, the user is ready to log in.

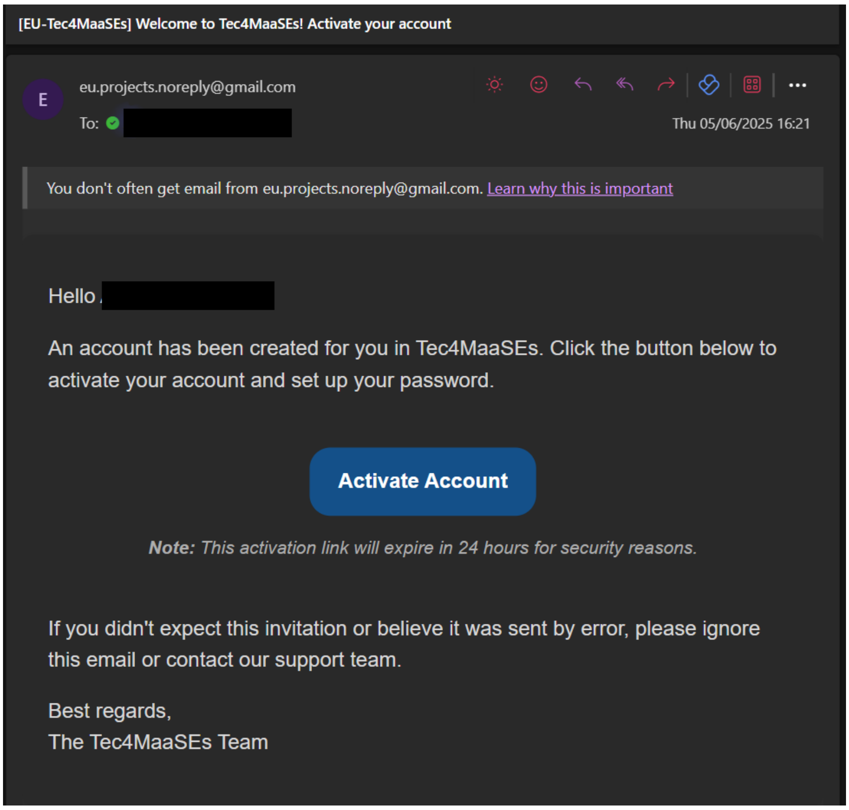

Figure 15: Email with activation link

3.1.4 Role List Page

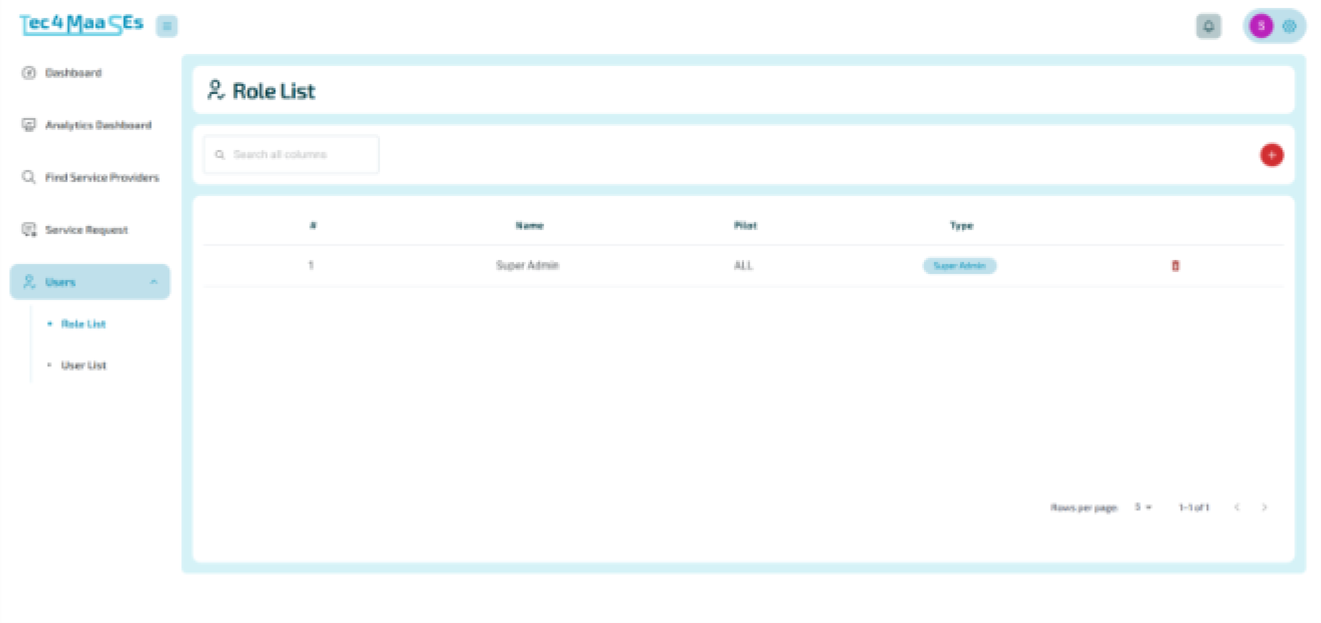

Figure 16: Role List Page

The above page (Figure 16), along with the **Users** menu item, is accessible exclusively to Admins and Super Admin users.

By clicking the **"Users"** menu item in the sidebar and then selecting the **"Role List"** option, the user is navigated to the **Role List** page. This page displays a list of all users along with their assigned roles within the application.

This feature allows Super Admins to view and manage user roles, ensuring appropriate access control across the platform.

When the user clicks the red plus icon, a pop-up window will appear prompting them to complete a form to create a new user role (Figure 17).

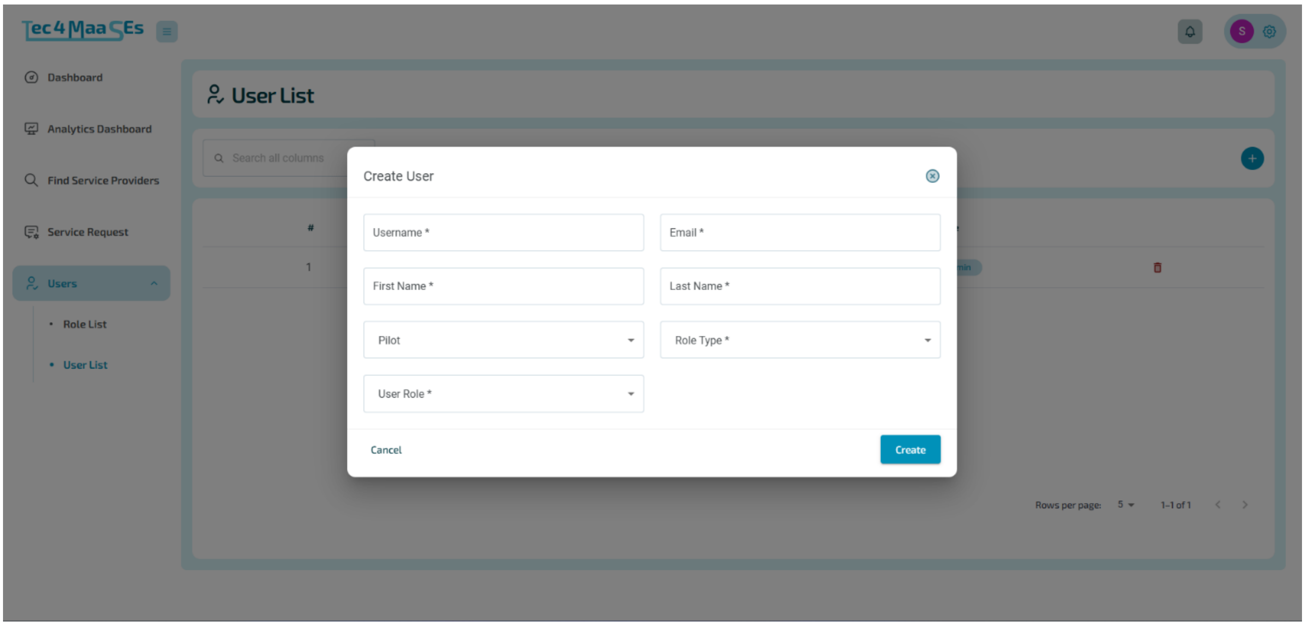

Figure 17: Create User Role on T4M platform

3.1.5 Organization Register Page

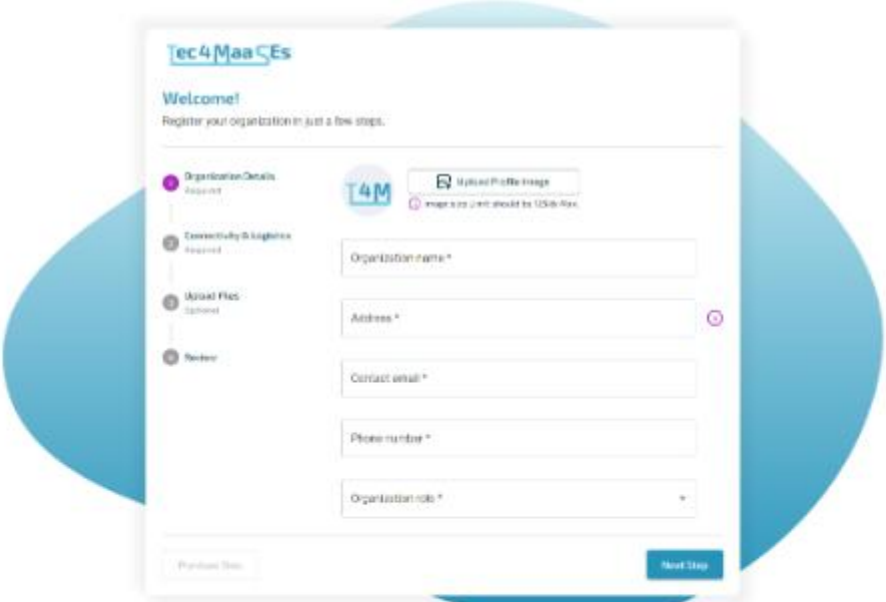

Figure 18: Organization Register Step 1

After logging in, if the user does not belong to an organization, they are requested to register their organization. As a result, they are directed to the **Organization Registration** page, which presents a multi-step wizard form for registering a new organization.

In Step 1 (Figure 18), the user is prompted to provide the following information:

- **Upload Profile Image** *(optional)* – Allows the user to upload a profile image or logo for the organization.
- **Organization Name** – The official name of the organization.
- **Address** – The physical location or mailing address of the organization.
- **Contact Email** – The primary email address for communication.
- **Phone Number** – A contact phone number for the organization.
- **Organization Role** – A multi-select input where the user can assign one or more roles that describe the organization's function within the system, Provider and/or Consumer.

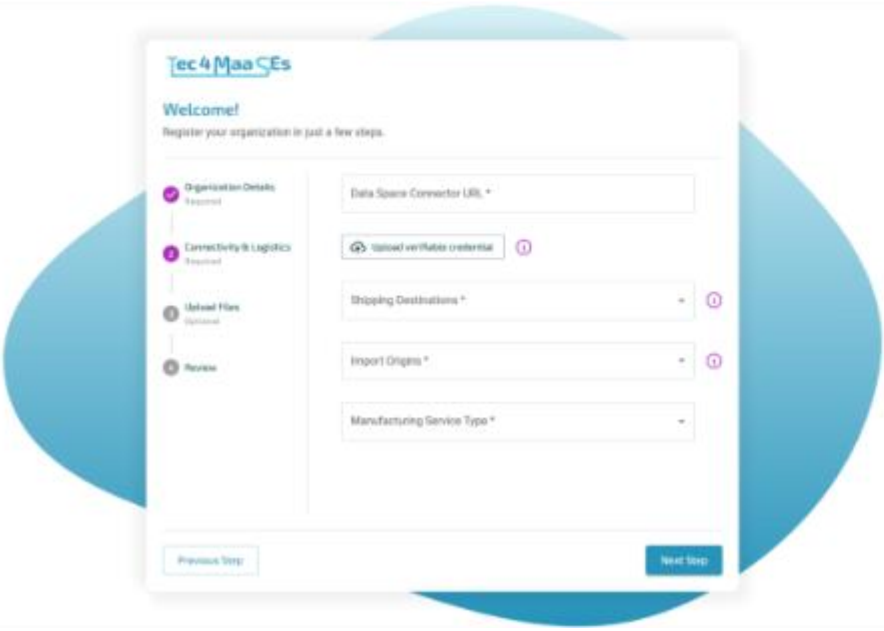

Figure 19: Organization Register Step 2.

In Step 2 (Figure 19), the user is prompted to provide the following information:

- **Data Space Connector URL** – The endpoint used to connect the organization to the data space network.
- **Upload Verifiable Credentials** (optional) – Allows the user to upload digital credentials that can verify the organization's identity or capabilities – implemented in the next iteration.
- **Shipping Destinations** – A multi-select input where the user can specify the geographic regions or countries to which the organization ships products or services, as a provider.
- **Import Origins** – A multi-select input where the user can identify the regions or countries in which the organization can accept products or services, as a consumer.
- **Manufacturing Service Type** – Specifies the type of manufacturing services the organization provides, in case of a provider organization.

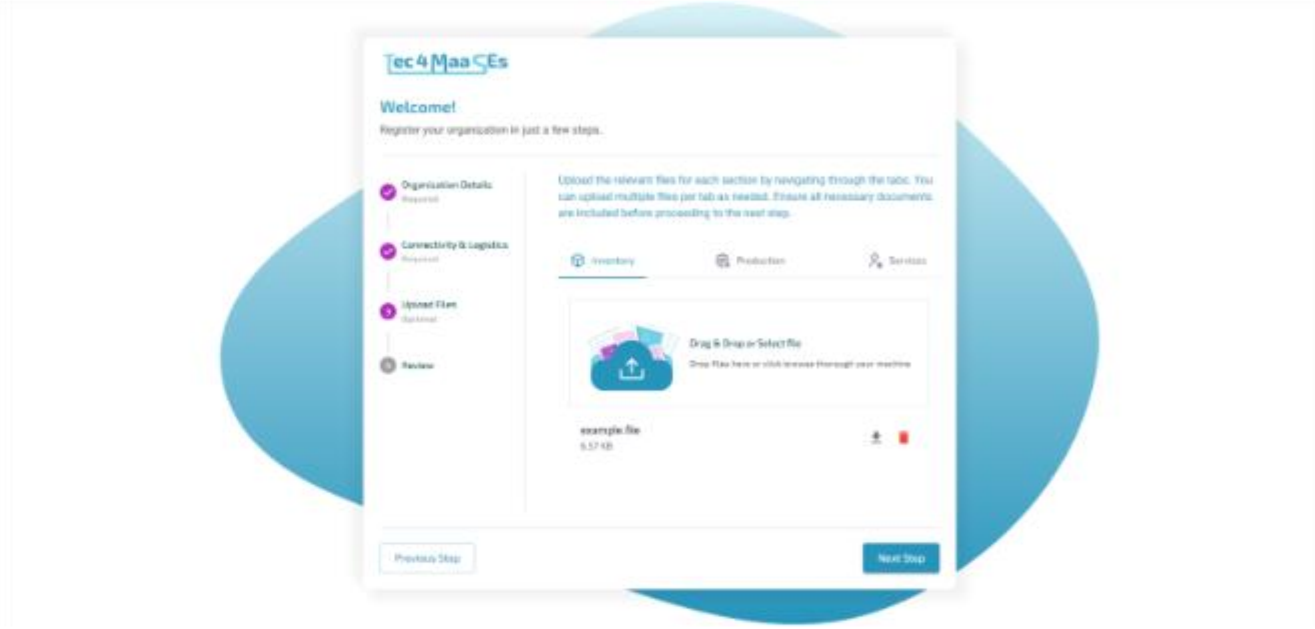

Figure 20: Organization Register Step 3.

In Step 3 (Figure 20), the user can optionally upload or drag and drop files related to **Inventory**, **Production**, or **Services**. Additionally, users have the option to:

- **Download** uploaded files for review
- **Remove** any previously uploaded files

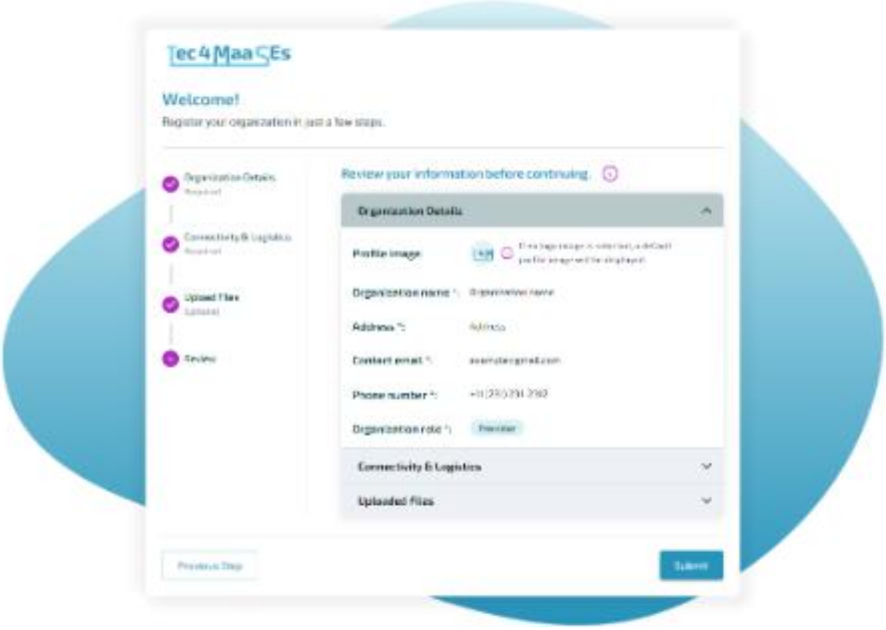

Figure 21: Organization Register Step 4.1

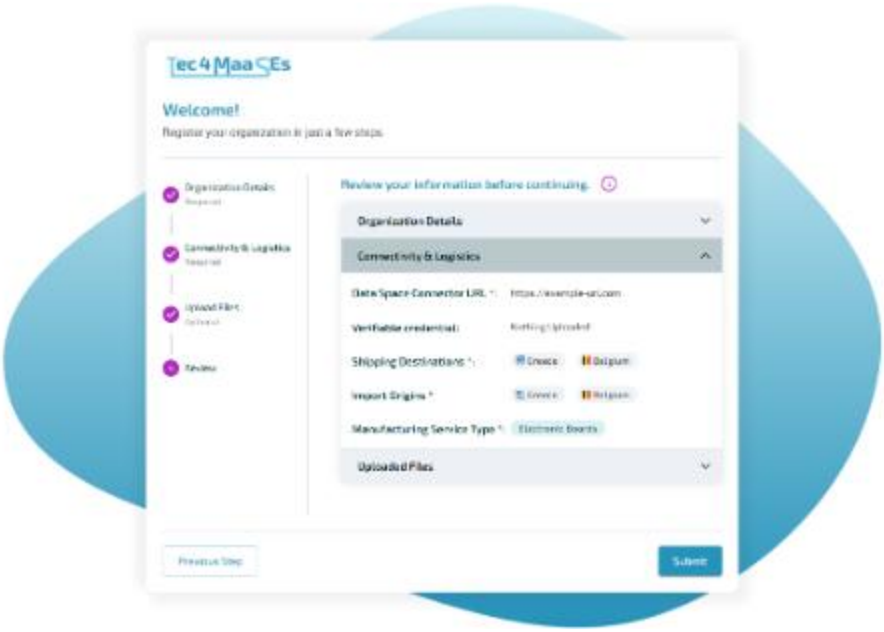

Figure 22: Organization Register Step 4.2

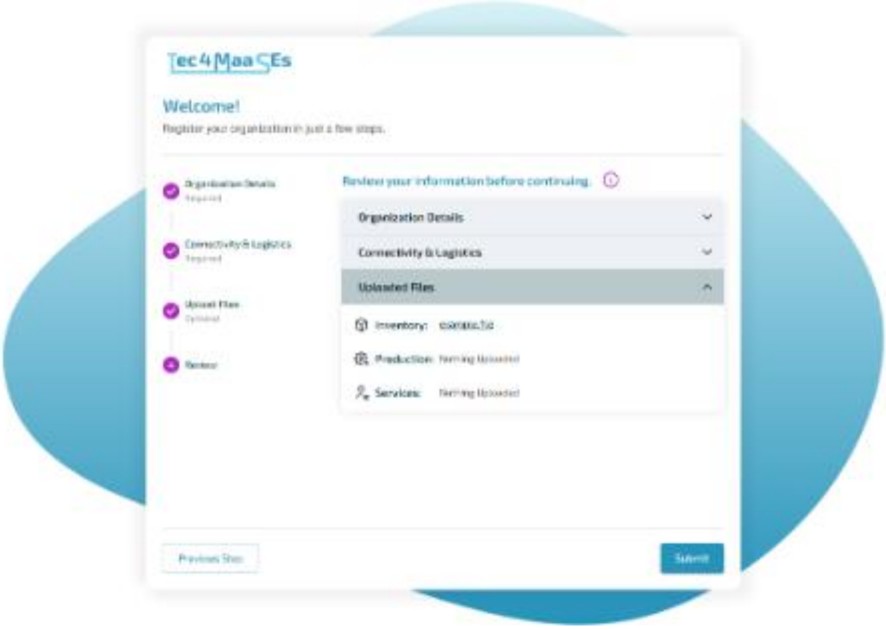

Figure 23: Organization Register Step 4.3

In Step 4 (Figure 21, Figure 22 and Figure 23), the user is presented with a summary of all the information entered in the previous steps. This review step allows the user to verify the accuracy and completeness of their request before submission.

The user can make changes by either:

- Clicking the **"Previous"** button to navigate back step-by-step, or
- Directly selecting any of the numbered steps in the wizard to quickly return to and edit specific sections.

Once all information has been reviewed and confirmed to be correct, the user can proceed by clicking the **"Submit"** button to finalize the registration.

Figure 24: Organization Register - Submission

After submission, a **confirmation message** is displayed on the screen, notifying the user that the registration was successful (Figure 24).

3.1.6 Welcome Page

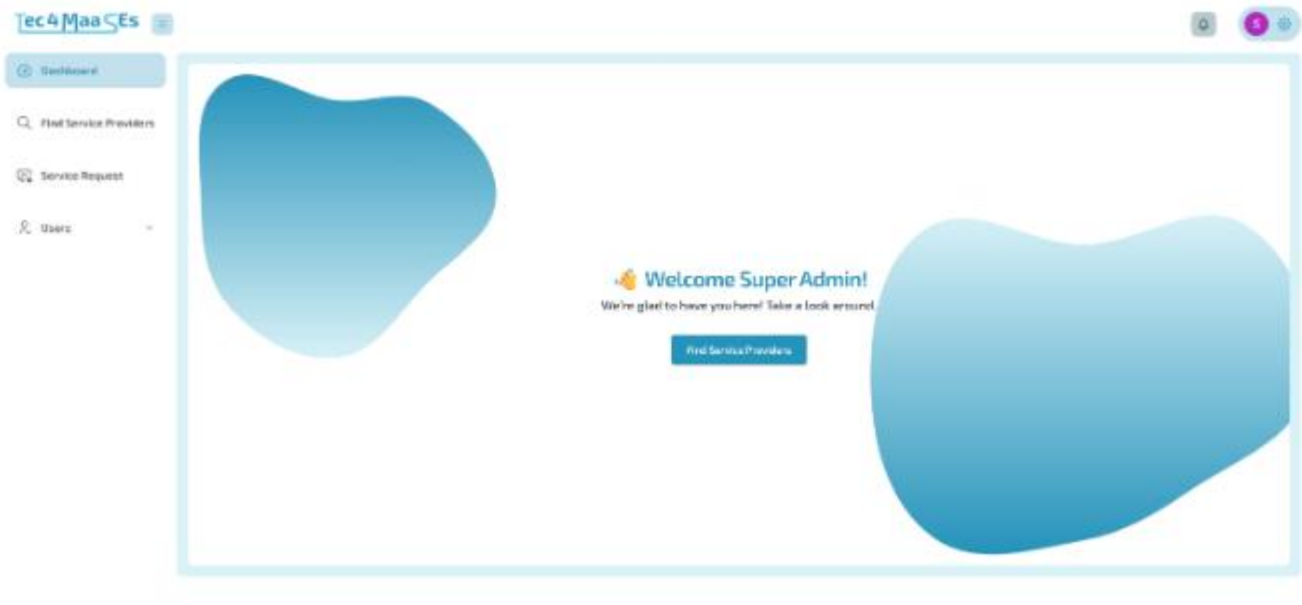

Figure 25: Welcome Page

Upon successful organization registration and sign-in, the user is directed to the welcome page of the dashboard (Figure 25). From this screen, the user can navigate through the application using the sidebar menu items, which are located on the left side of the interface. Additionally, a "**Find Service Providers**" button is displayed at the center of the welcome page, providing another way for the user to begin exploring available services within the application.

3.1.7 Find Service Providers Page

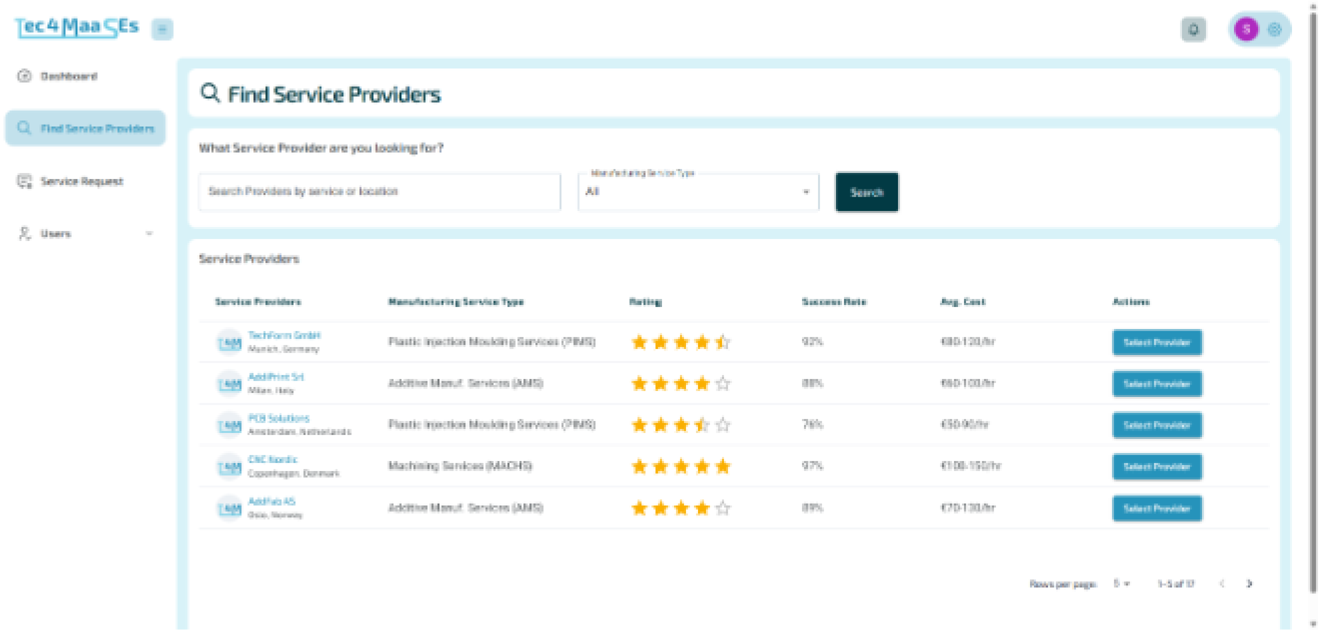

Figure 26: Find Service Providers

When the user clicks the "Find Service Providers" button located at the center of the welcome page, or selects the corresponding option from the sidebar menu, they are directed to the Find Service Providers page (Figure 26). On this page, a data table is displayed, listing all available service providers. The user can browse through this table to view and select the provider that best suits their needs.

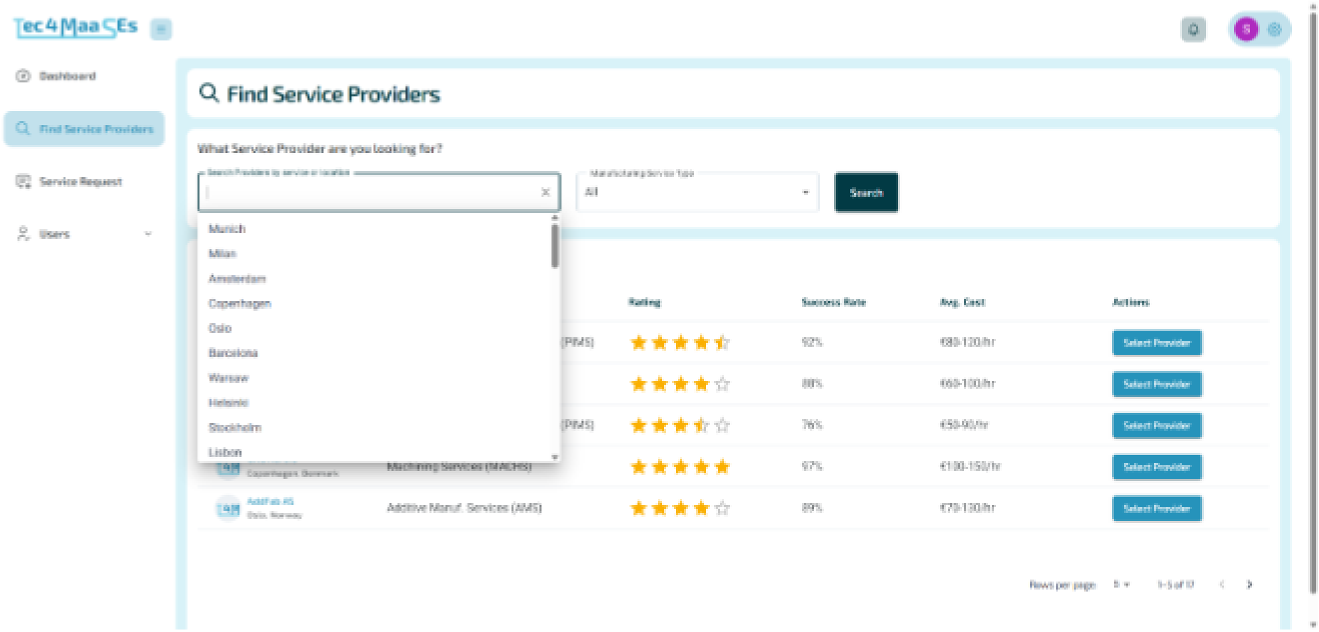

Figure 27: Find Service Providers - Select location

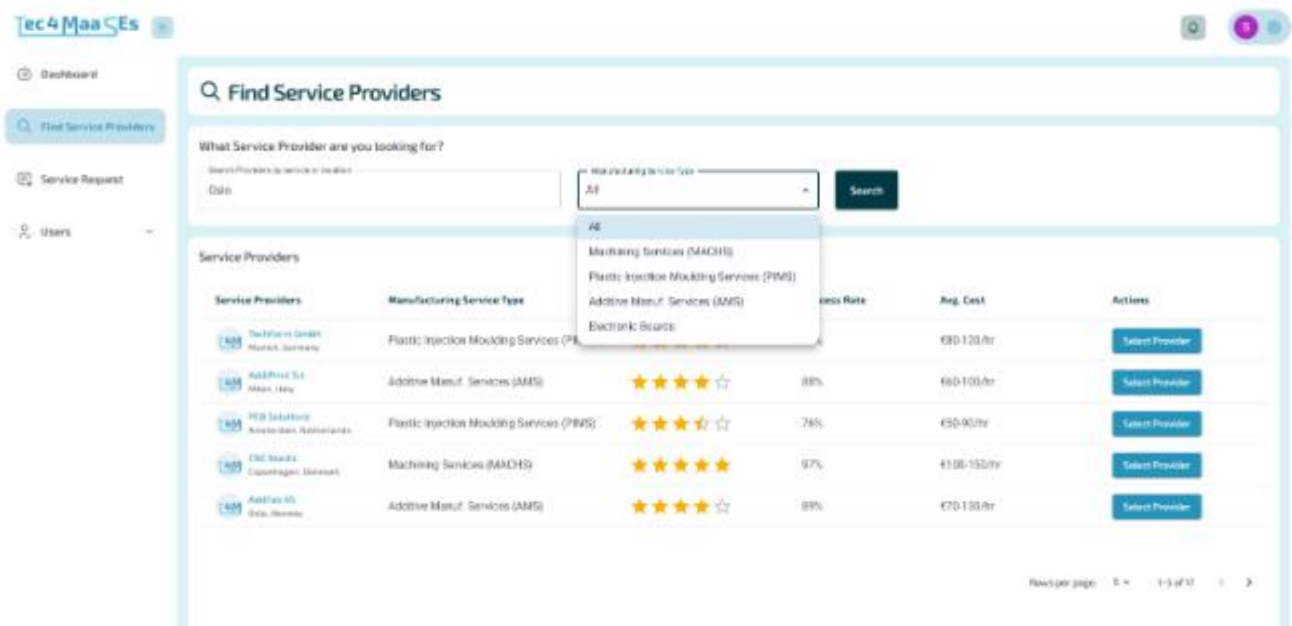

Figure 28: Find Service Providers - Select Manufacturing Service

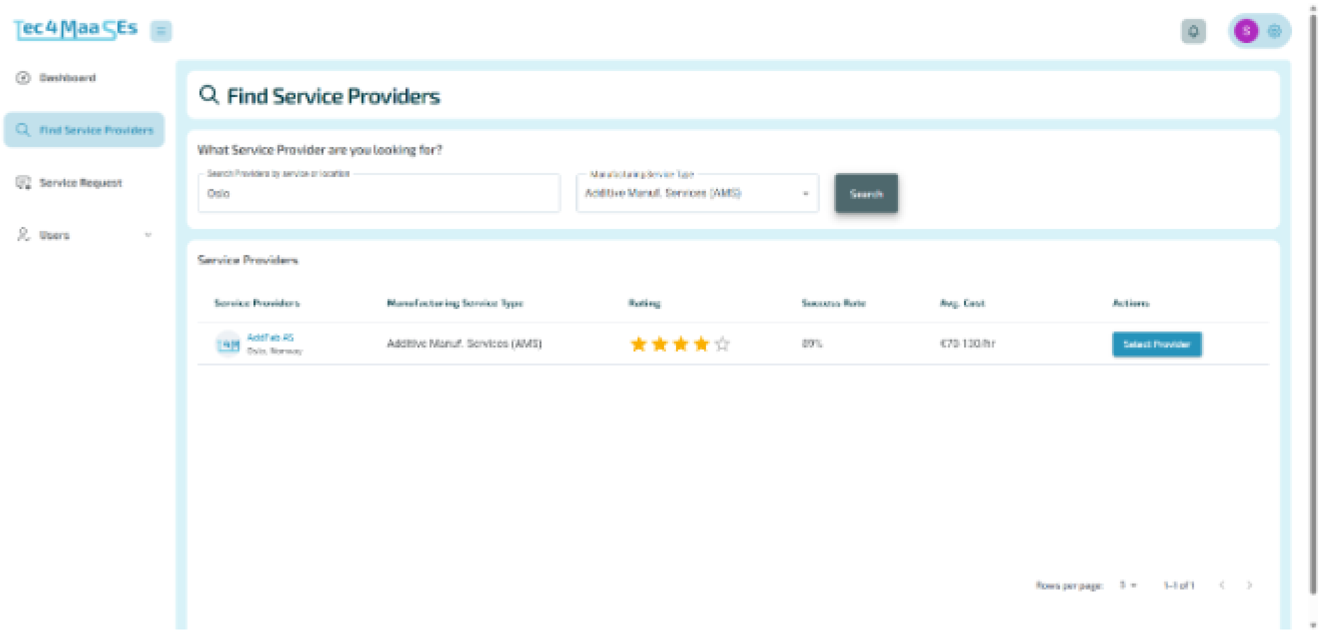

Figure 29: Find Service Providers - Results

Users can utilize the "Search Providers by Service or Location" autocomplete field, as well as the "Manufacturing Service Type" dropdown menu, as seen in Figure 27 and Figure 28. These input fields allow users to filter the results and locate specific providers based on service offerings or geographic location (Figure 29).

3.1.8 Service Request Page

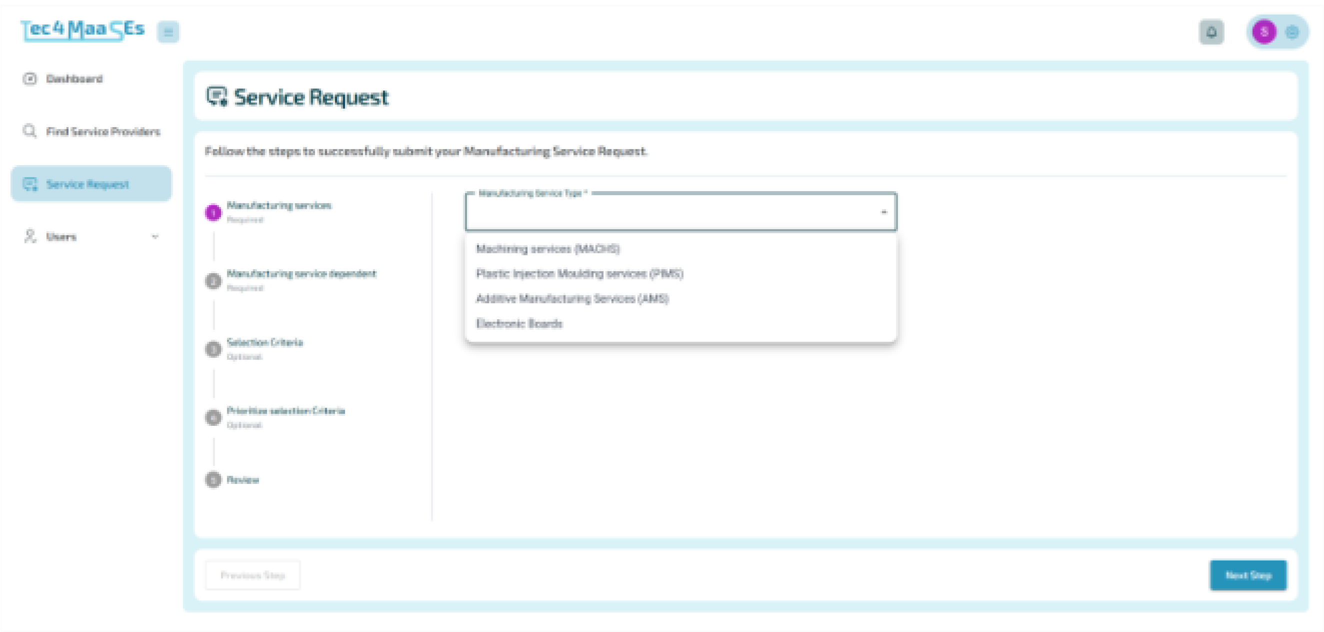

Figure 30: Service Request Step 1

When the user clicks the "Service Request" menu item from the sidebar, they are navigated to the Service Request page. Here, the user is presented with the first step of a multi-step wizard-style request form. In this step, the user is required to select one of the available options from the Manufacturing Service Type dropdown menu. The available options can be seen in Figure 30. Additionally, the user has the option to upload a file to accompany the request; however, this upload is optional and not required to proceed to the next step.

3.1.8.1 Service request for Electronic Boards

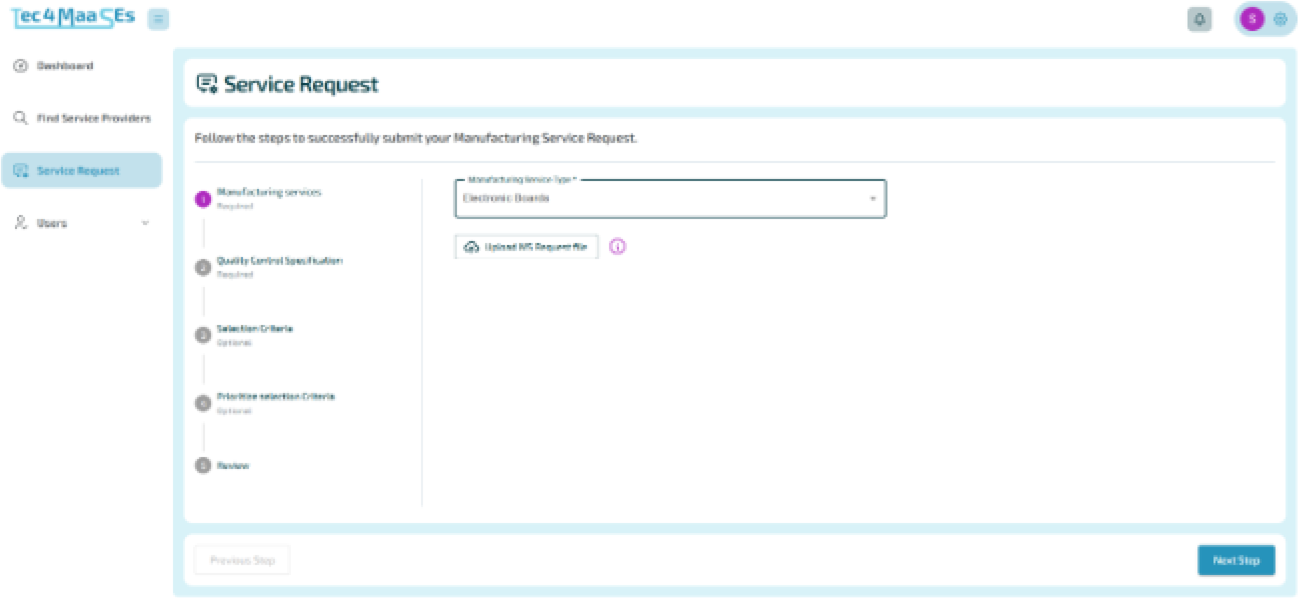

Figure 31: Service Request – Electronic Boards

If the user selects "**Electronic Boards**" from the **Manufacturing Service Type** dropdown in Step 1, the content of Step 2 will dynamically change. Instead of the standard Manufacturing Service Dependent section, the second step will display the Quality Control Specification form. Additionally, the label for the file upload field will update to "Upload MS Request File" to reflect the specific requirements for this service type (Figure 31).

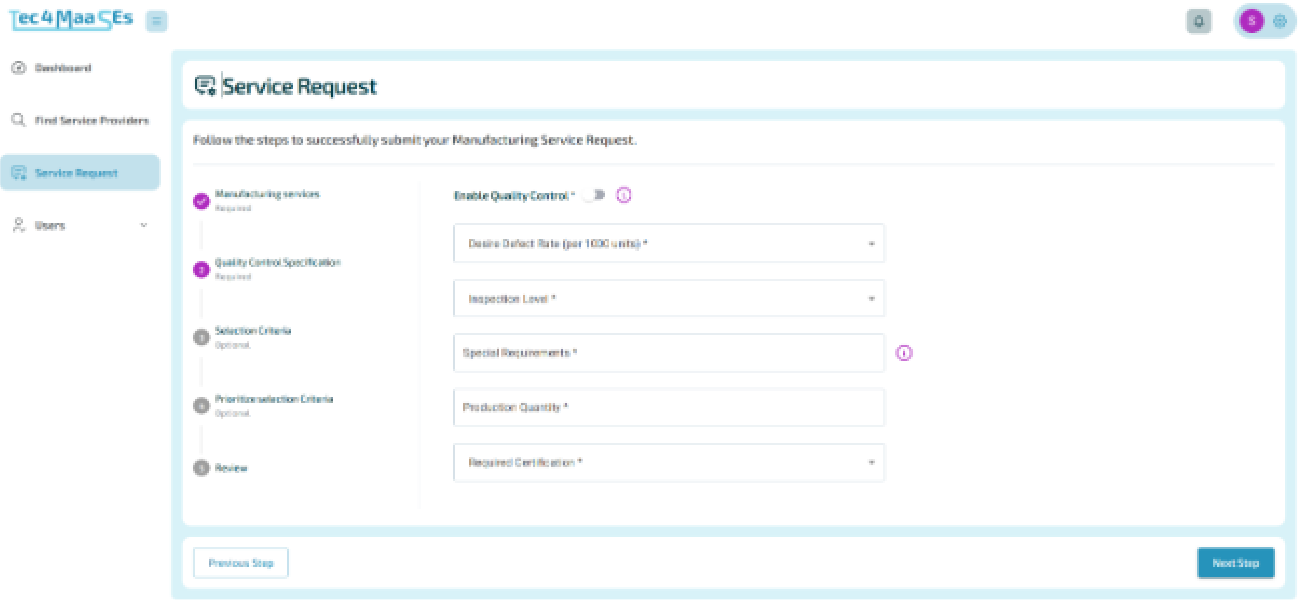

Figure 32: Service Request Step 2 – Quality Control Specification

In Step 2 (Figure 32), they will be presented with additional fields as part of the **Quality Control Specification** section in Step 2. These fields include:

- **Enable Quality Control** – ensure that the delivered electronic boards meet defined quality standards.
- **Desired Defect Rate (per 1000 units)** – Specifies the acceptable defect rate threshold.
- **Inspection Level** – Indicates the rigor or depth of quality inspection required.

- **Special Requirements** – Allows the user to define any specific conditions or constraints relevant to quality control.
- **Production Quantity** – States the total number of units to be produced.
- **Required Certification** – Identifies any necessary certifications that must be met by the service provider.

3.1.8.2 Service request for Machining Services (MACHS)

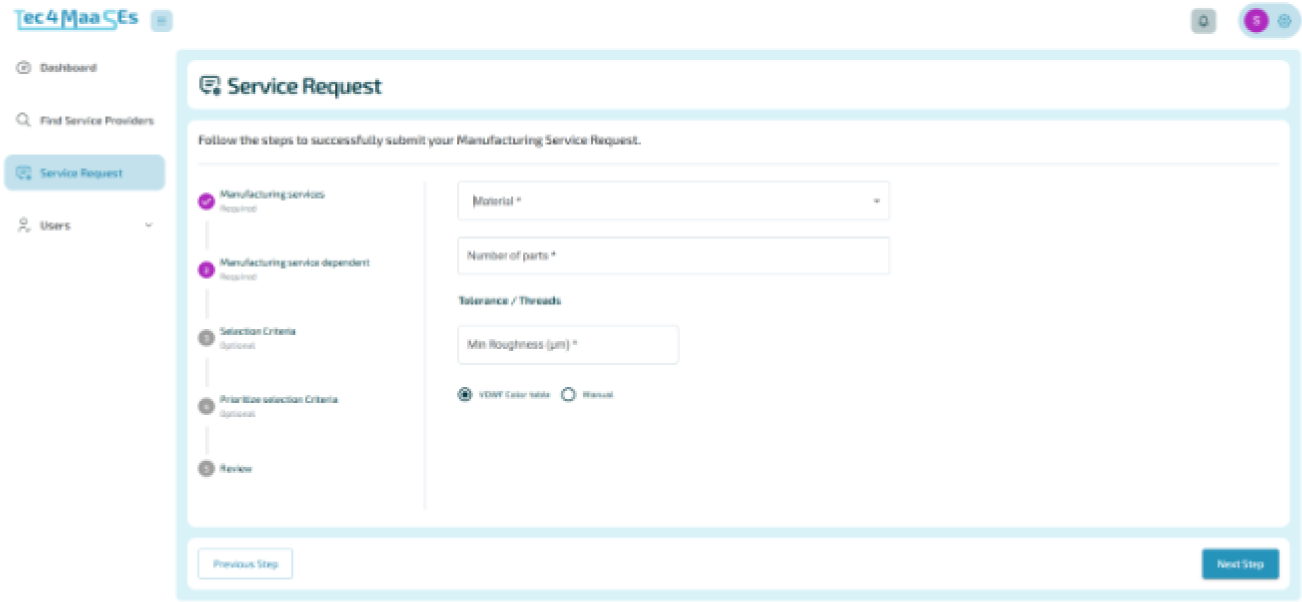

Figure 33: Service Request Step 2 – Machining Services, Manufacturing Service Dependant

If the user selects **"Machining Services (MACHS)"** from the **Manufacturing Service Type** dropdown in Step 1, after uploading a CAD file in Step 1, they will be required to complete a specific set of fields in Step 2 (Figure 33). These include:

- **Material** – A dropdown menu to select the material for the machining process.
- **Number of Parts** – A numeric input to specify the quantity of parts needed.
- **Minimum Roughness (µm)** – An input field to define the desired surface roughness.
- Under the **Tolerance / Threads** section, the user can choose between two options:
- **VDWF Color Table** – Uses predefined tolerance standards.
- **Manual** – Allows the user to provide custom specifications.

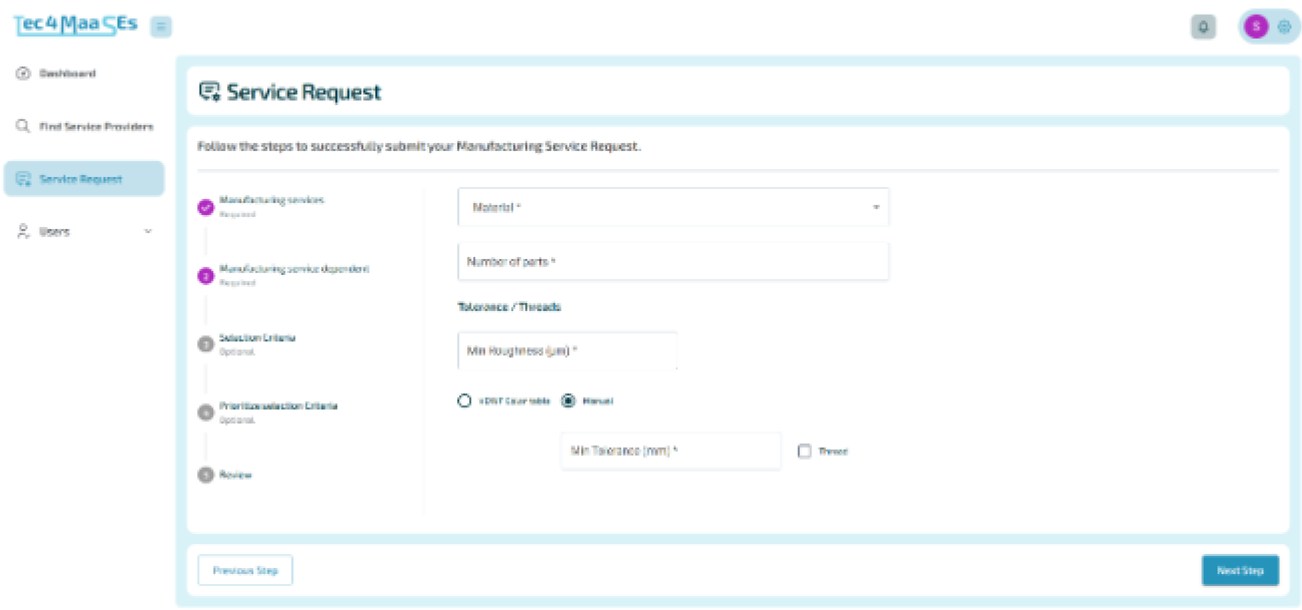

Figure 34: Service Request Step 2 – Machining Services, Manufacturing Service Dependant, Manual

If the **Manual** option is selected (Figure 34), two additional fields become visible:

- **Minimum Tolerance** – Specifies the minimum tolerance value required.
- **Thread** – A checkbox to indicate whether threading is required in the machining process.

3.1.8.3 Service request for Additive Manufacturing Services (AMS)

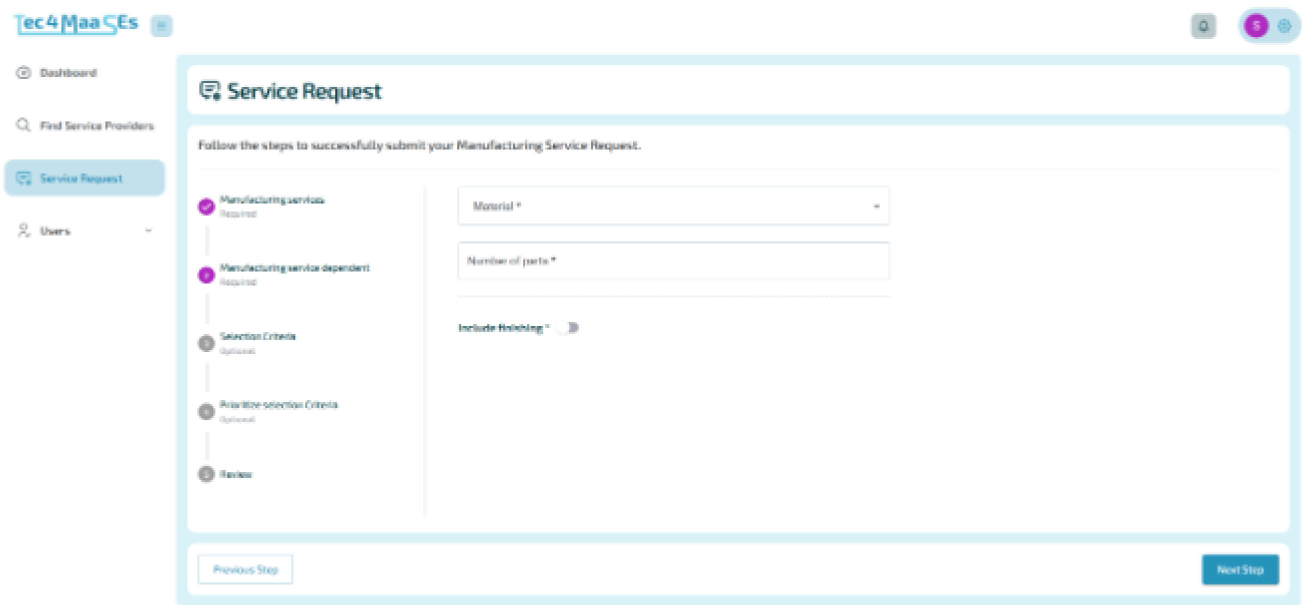

Figure 35: Service Request - Additive Manufacturing Services, Manufacturing Service Dependant

If the user selects the **"Additive Manufacturing Services (AMS)"** option from the **Manufacturing Service Type** dropdown in Step 1, after uploading a CAD file in Step 1, Step 2 will display the following fields (Figure 35):

- **Material** – A dropdown menu used to select the material for the machining process.
- **Number of Parts** – A numeric input to specify the quantity of parts required.
- **Include Finishing** – An optional toggle switch.

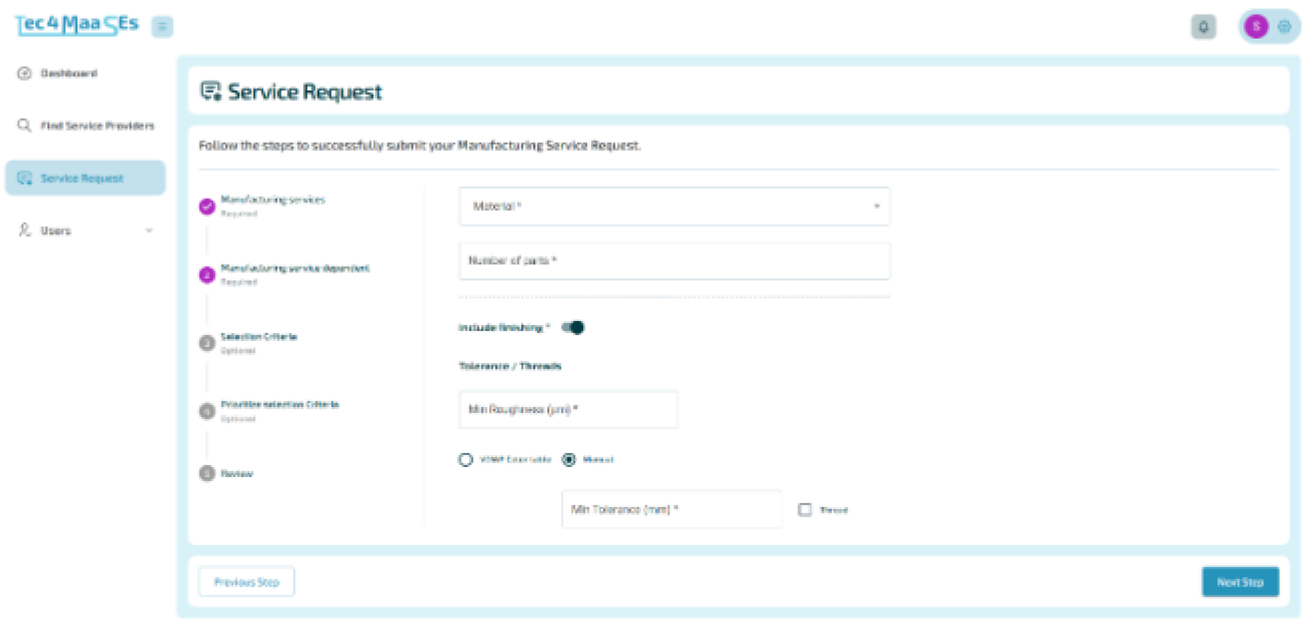

Figure 36: Service Request - Additive Manufacturing Services, Manufacturing Service Dependant, Finishing included

If **Include Finishing** switch is enabled (Figure 36), the following additional field becomes visible:

- **Minimum Roughness (µm)** – An input field to define the desired surface roughness of the finished parts.

Under the **Tolerance / Threads** section, the user must select one of the following methods to define tolerances:

- **VDWF Color Table** – Utilizes standardized tolerance classifications.
- **Manual** – Allows the user to input custom specifications.

If **Manual** is selected, two additional fields appear:

- **Minimum Tolerance** – Specifies the exact tolerance requirement for the parts.
- **Thread** – A checkbox indicating whether threading is required in the machining process.

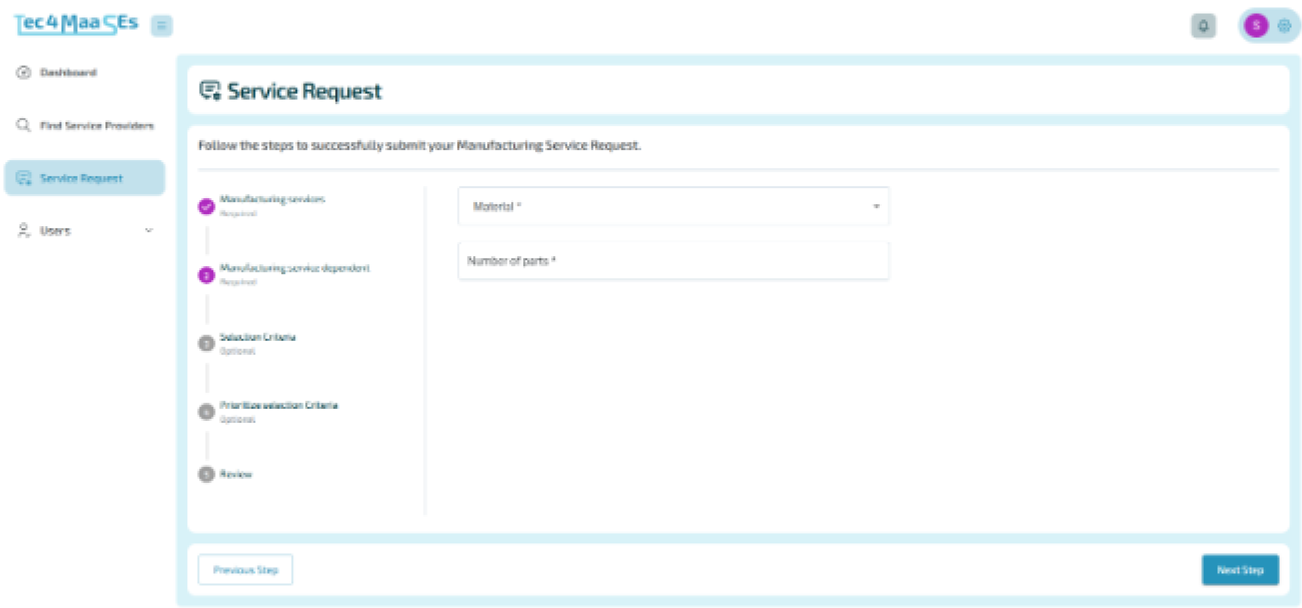

Figure 37: Service Request - General

If any other option is selected from the **Manufacturing Service Type** dropdown in Step 1, then Step 2 will display only the following fields (Figure 37):

- **Material** – A dropdown menu used to select the material for the manufacturing process.

- **Number of Parts** – A numeric input to specify the quantity of parts required.

3.1.8.4 Selection Criteria and Submission

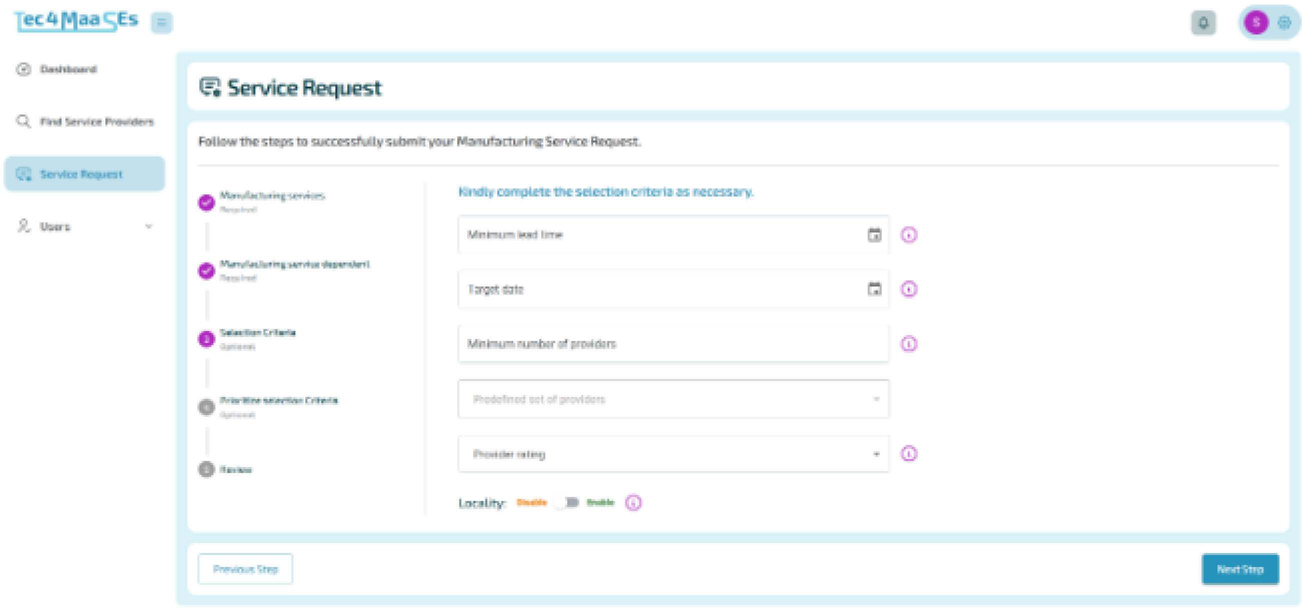

Figure 38: Service Request Step 3 - Selection Criteria

In Step 3, the user can define optional **selection criteria** to further tailor the service provider search (Figure 38). These criteria are not mandatory. At a later stage, this input of criteria can be incorporated with the analytics the user can interact with in the Analytics Dashboards. The available options include:

- **Minimum Lead Time** – Specifies the earliest acceptable delivery date.
- **Target Date** – Defines the preferred completion date for the service.
- **Minimum Number of Providers** – Sets the minimum number of providers that should fulfil the criteria.
- **Provider Rating (1 to 5 stars)** – Allows the user to set a minimum acceptable provider rating.
- **Locality (Switch)** – An optional toggle that, when enabled, prioritizes results to providers within the same geographic region as the user.

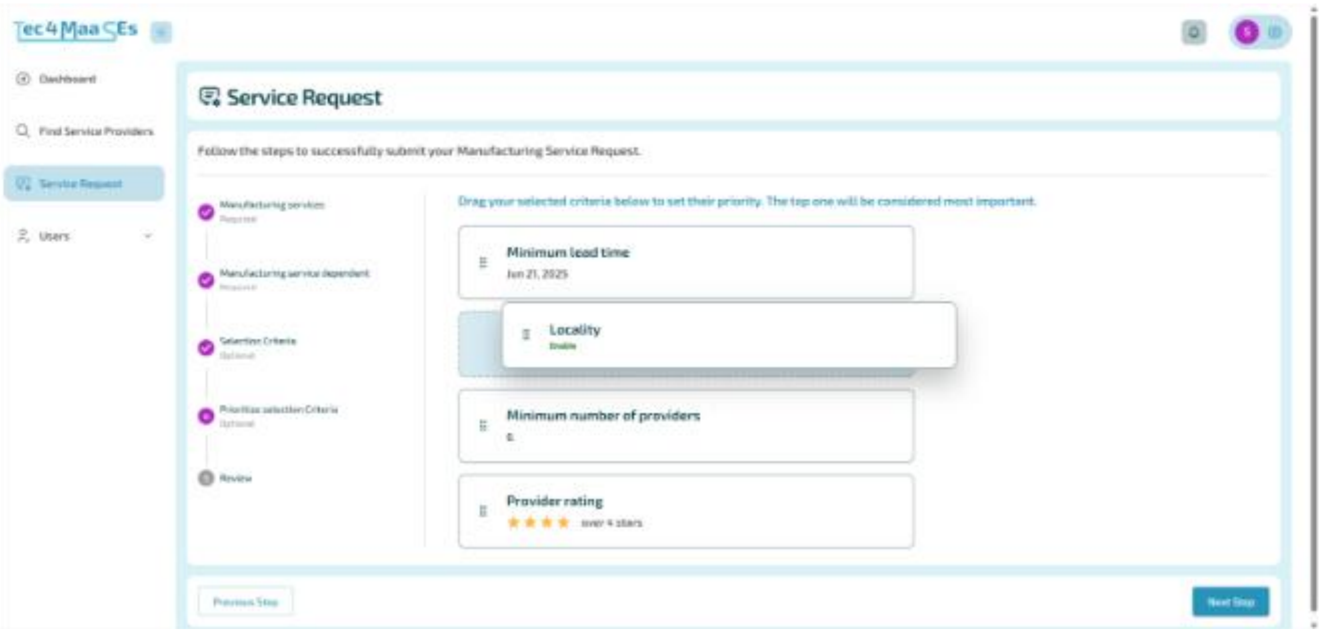

Figure 39: Service Request Step 4 - Prioritization of Selection Criteria

In Step 4, the user is asked to **prioritize** the selection criteria specified in the previous step (Figure 39). This is done by dragging and dropping the chosen criteria into a ranked list, where the **top item is considered the most important** during the provider matching process. Only the criteria selected in Step 3 will appear in the list, and their order directly influences how providers are scored and recommended.

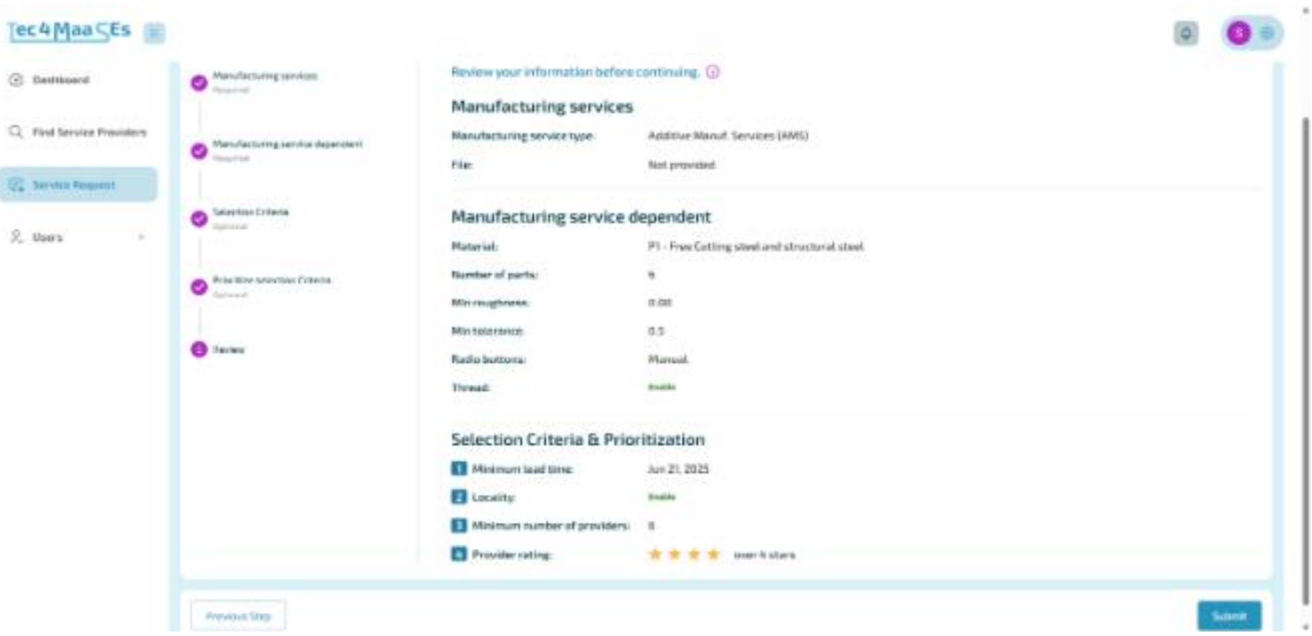

Figure 40: Service Request Step 5 - Review

In Step 5, the user is presented with a **summary of all the information** entered in the previous steps (Figure 40). This review step allows the user to verify the accuracy and completeness of their request before submission. The user can make changes by either:

- Clicking the **"Previous"** button to navigate back step-by-step, or
- Directly selecting any of the numbered steps in the wizard to quickly return to and edit specific sections.

Once all information has been reviewed and confirmed to be correct, the user can proceed by clicking the **"Submit"** button to finalize and send their service request.

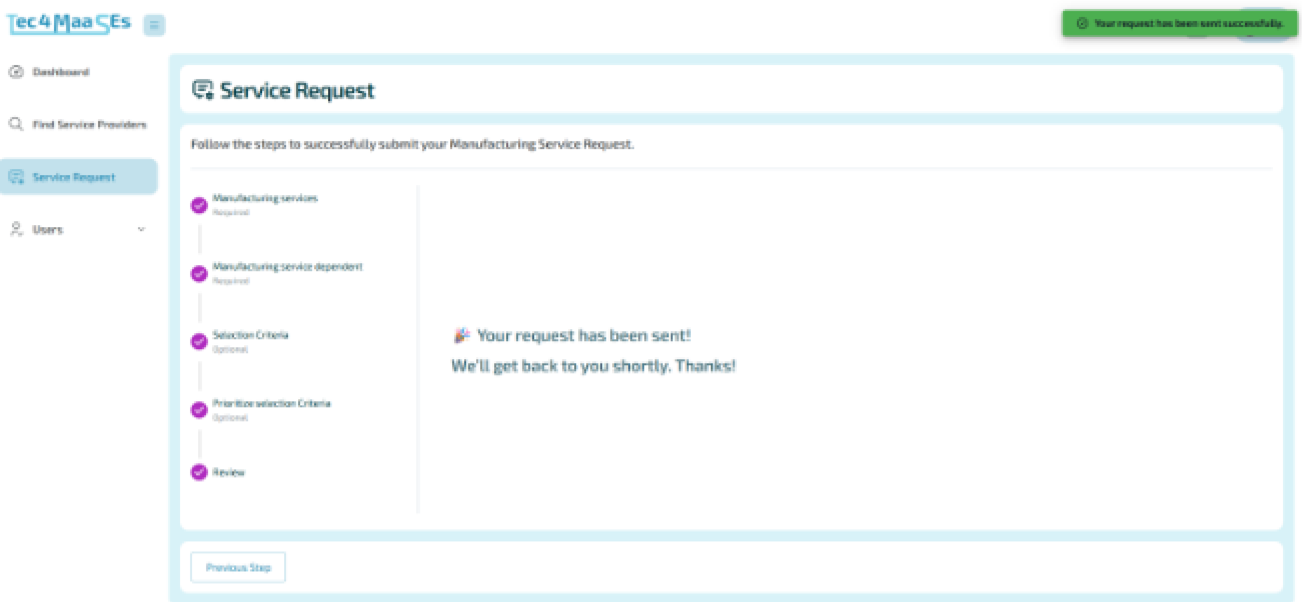

Figure 41: Service Request - Successful Submission

After submission, a **confirmation message** is displayed on the screen, notifying the user that the request has been successfully sent to the server (Figure 41).

3.1.9 Logout

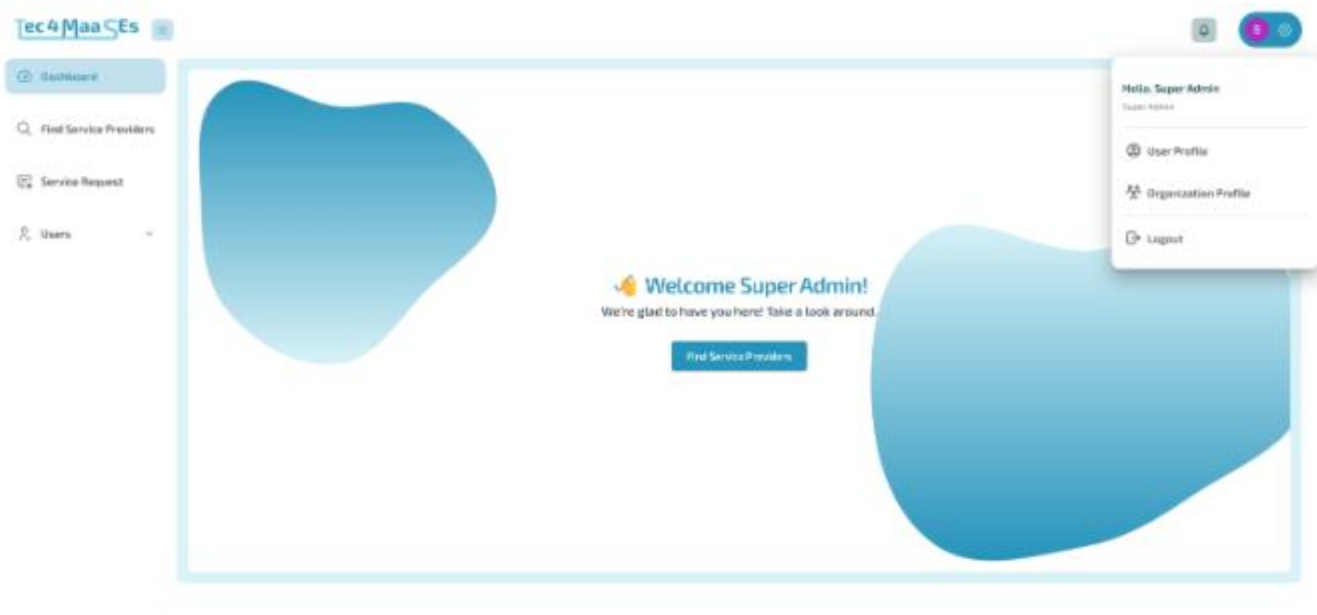

Figure 42: Logout

When the user clicks on the **Profile** section located in the top-right corner of the page, a popup menu will appear (Figure 42). From this menu, the user can click the **Logout** button to securely sign out of the application.

3.2 User Manual for Analytics Dashboards

This user manual, which covers #9 from Table 1, provides a comprehensive guide to the Analytics (Power BI) dashboards integrated into the T4M Central User Interface. These dashboards are designed to deliver real-time, interactive visual insights that enhance visibility, traceability, and strategic control over operations, provider performance, optimization decisions, and resource utilization. The dashboards are organized into distinct analytical views, each tailored to specific user needs within the platform, from general overviews and provider information to decision-support tools and detailed composition and post-optimality summary. Each section of this manual walks through the functionalities, data structure, and user interactions within these dashboards. The dashboards can be applied for all the VNs; however, the generated data for the visualizations are currently related to VN2.

3.2.1 Platform Overview Dashboard

The Platform Overview Dashboard serves as the central hub for understanding the current state of operations across the T4M platform. It provides a real-time summary of service request activity, offering both a high-level performance snapshot and granular status insights to all logged-in users of the T4M platform (Figure 43).

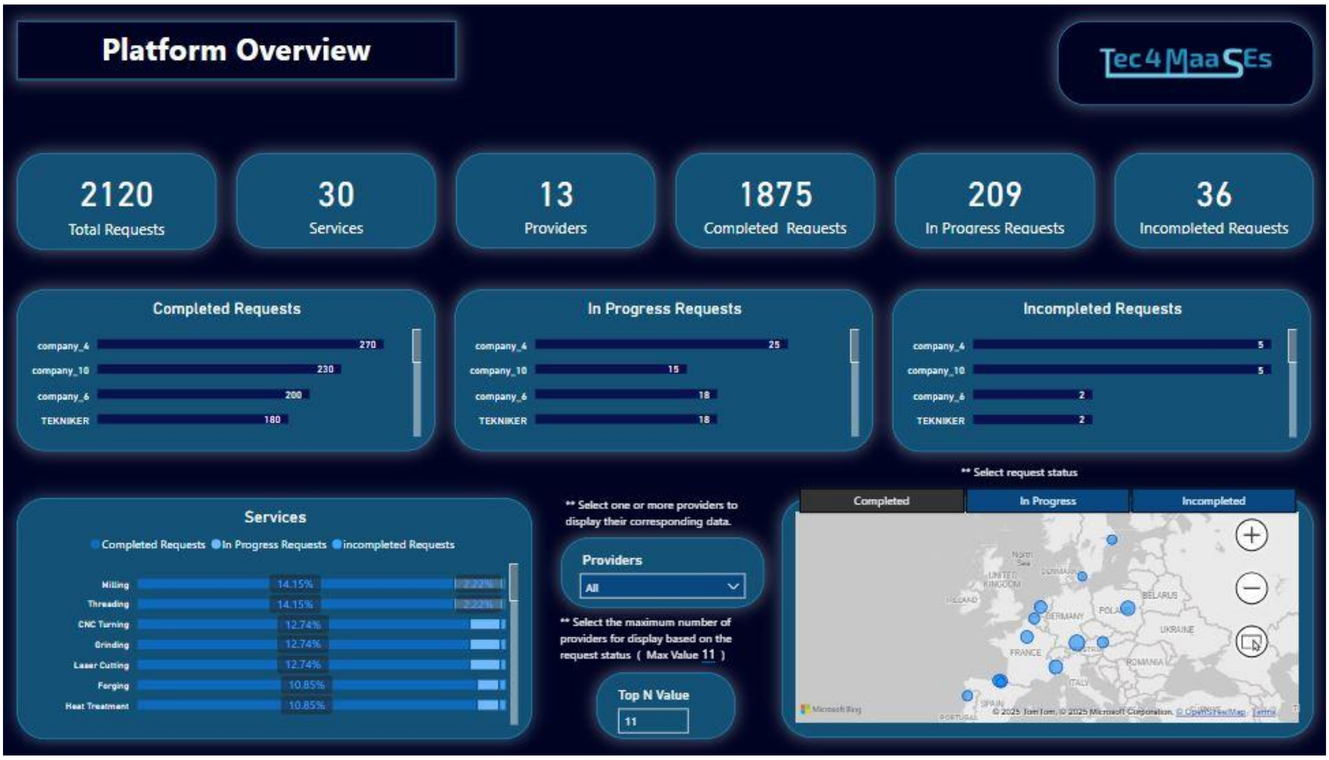

Figure 43: Platform Overview Dashboard Analytics

At the top of the dashboard (Figure 44), a set of performance indicator cards summarize key metrics: the total number of service requests submitted to the platform, the number of distinct services offered, and the count of active providers. The dashboard also distinguishes the volume of requests by their status—completed, in progress, and incomplete—giving immediate visibility into operational throughput.

Figure 44: General Statistics

Further down (Figure 45), three horizontal bar charts provide a breakdown of request statuses by the provider. These charts reveal how requests are distributed and help identify which providers are handling the highest volume of tasks across each request status category.

Figure 45: Requests Status

To the left (Figure 46), the dashboard includes a chart highlighting how services perform in terms of completion rate. This chart displays comparative request distributions across service types such as Milling, CNC Turning, and Threading (VN2 services), allowing users to assess workload balance and operational coverage by process type. If multiple services are available, a dropdown bar allows users to scroll through them.

Figure 46: Services Status

An interactive filter allows users to focus on data associated with one or more providers (Figure 47). A “Top N” input box enables refined analysis by displaying only the top-ranked providers based on a selected metric. This is especially useful for identifying high-contributing or high-performing providers quickly.

Figure 47: Providers filtering

A geographical map completes the dashboard, offering a spatial perspective on platform activity (Figure 48). Each data point corresponds to a specific country or region, and hovering over a location reveals a detailed status summary for that area. This geospatial visualization helps uncover regional trends and possible geographic performance disparities.

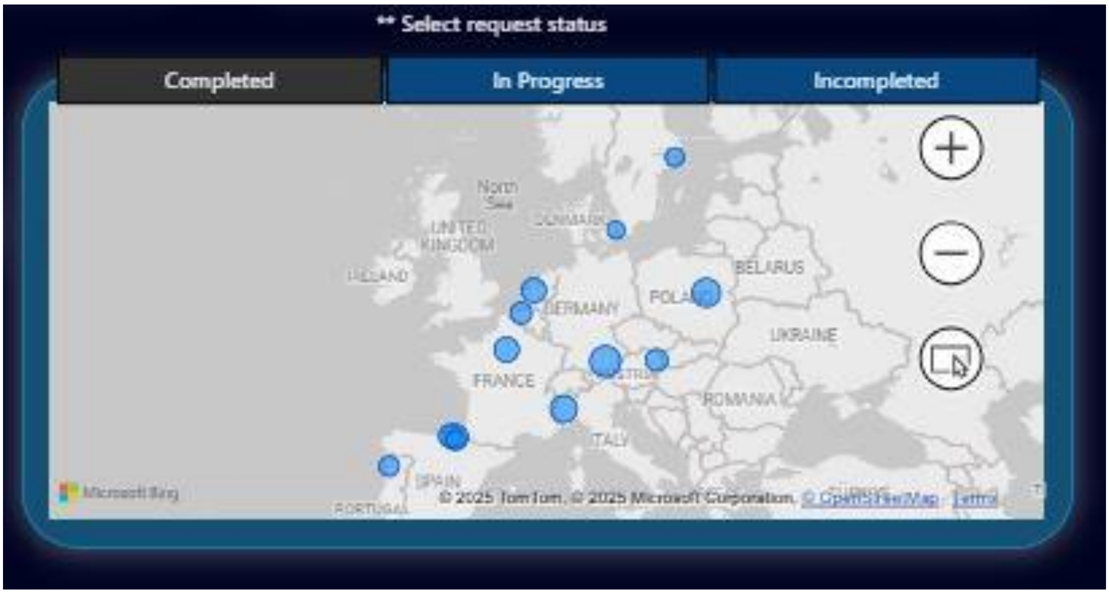

Figure 48: Geospatial visualization

3.2.2 Provider Overview Dashboard

The Provider Overview Dashboard focuses on the performance of individual service providers, helping T4M platform administrators monitor the efficiency and contribution of each participant in the T4M ecosystem (Figure 49).

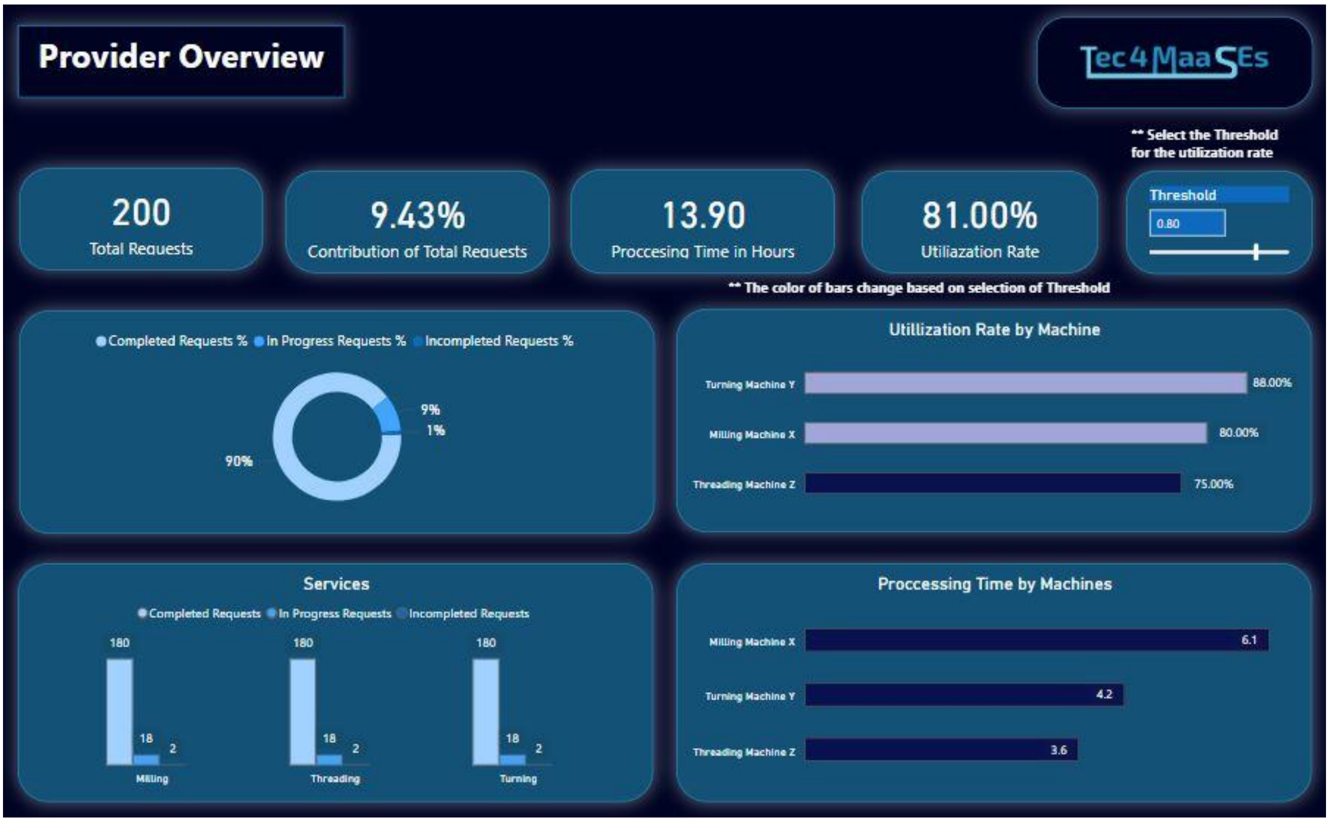

Figure 49: Provider Overview Dashboard Analytics

Initially, the dashboard presents some indicators that capture the total number of requests associated with a provider, the proportion of these requests in relation to the total platform activity, the average processing time per request in hours, and the provider's machine utilization rate (Figure 50). These figures offer a quick performance benchmark for comparing different providers.

Figure 50: General Provider Metrics

A large donut chart visually breaks down request statuses, showing the proportion of completed, in-progress, and incomplete requests (Figure 51). This makes it easy to assess a provider's reliability and responsiveness at a glance. Service-specific bar charts underneath the donut chart display request status distributions for each process type, such as Milling or Turning. This chart is helpful to understand how each provider performs in different service domains.

Figure 51: Requests and Services Overview

To the right, the dashboard highlights machine-level performance (Figure 52). A threshold allows users to define a target utilization rate. Once set, the utilization chart below visually distinguishes machines that fall above or below this threshold, with color-coded bars. This makes underutilized or overutilized equipment immediately apparent. Beneath the latter, another bar chart displays the average processing time for each machine. This helps identify the most and least efficient equipment in a provider's lineup. The dashboard's interactive nature allows users to click on any chart element, such as a machine or service, to filter the remaining dashboard components accordingly, making it easy to isolate and explore specific performance factors.

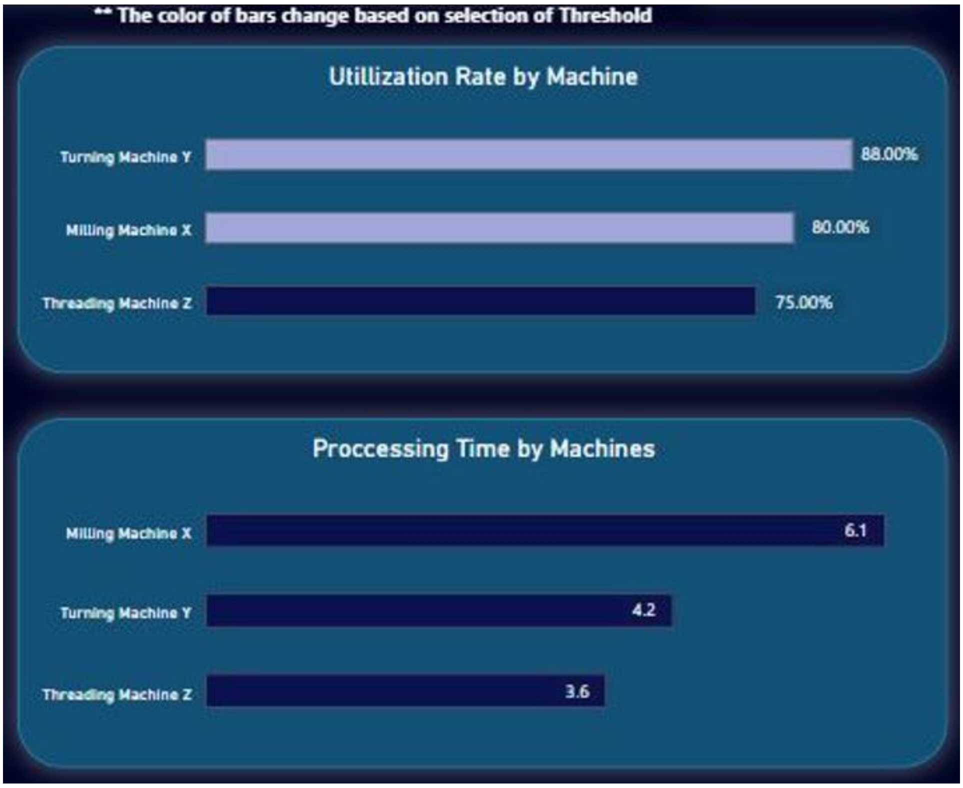

Figure 52: Provider Machines Overview

3.2.3 Post-Optimization Dashboard

This dashboard is designed as a decision-support interface, offering consumers, who have submitted a service request and received an initial composition result, the ability to interactively explore and refine optimization results based on personal preferences, quality requirements, and provider preferences (Figure 53).

Figure 53: Post-Optimality Analysis Dashboard

Initially, consumers observe a summary table displaying the compositions returned by the optimization service (Figure 54).

*** Total Supply Chains

Composition	Number of Providers	Providers	Quality Score	Completion Date
composition_1	1	Techniker	1.15	2/12/2024
composition_2	2	Techniker, URA	2.10	1/12/2024
composition_3	1	URA	3.65	1/12/2024
composition_4	2	Erreka, URA	4.45	2/12/2024
composition_5	1	Erreka	4.65	2/12/2024

Figure 54: Optimization Summary

They can either select a composition directly or proceed to a filtering and/or ranking phase (Figure 55). In this phase consumers can categorize providers into three preference groups: disliked, neutral, or preferred. Each group is selectable through a search box or predefined options. This classification directly influences the outcomes of the analysis and allows the system to evaluate trade-offs accordingly.

Figure 55: Provider Preference for the Post-Optimality Analysis

A set of parameters in the central panel allow users to define the maximum number of providers considered per composition and the minimum acceptable quality score (Figure 56). These filters help narrow down the solution space and ensure the results reflect strategic priorities. Two tables beneath this panel present filtered and ranked compositions based on the selected inputs. Each row represents an alternative composition, including details such as the involved providers, quality scores, and completion dates.

Figure 56: Filtering Criteria and the Filtered Summary

On the right side of the dashboard, users can assign custom weights to the evaluation criteria: number of providers, quality score, and preference alignment (Figure 57). These weights must sum to one, and a real-time indicator confirms when a valid configuration is achieved. The weights allow users to test different prioritization scenarios and observe their effects on solution rankings. In the next iteration, selection criteria already prioritized by the user during the service request can be incorporated here. A secondary table includes a TOPSIS score, which quantifies the relative performance of each solution based on a multi-criteria ranking approach (Figure 57).

Figure 57: Ranking Weights and its Ranked Table

All dashboards within the T4M platform are developed using Microsoft Power BI and are seamlessly integrated to provide real-time, data-driven insights. Once ready for external visibility, dashboards are published via secure or public URLs, with access strictly governed by user roles and associated privileges. Access control is enforced using organizational accounts and embedded credentials. This ensures that users only see data relevant to their permissions, protecting sensitive information while maintaining usability across diverse stakeholder groups. Dashboards are shared as read-only views via web browsers, enabling intuitive, cross-team collaboration without requiring technical expertise. For extended analysis or reporting, data can be exported in JSON format to support third-party tools or internal evaluation processes.

3.3 User Manual for Supply Chain Monitoring – Balanced Scorecard – AI Model Governance Service

This user manual provides a guide to the Supply Chain Monitoring, Balanced Scorecard, and AI Model Governance services offered through the MIRA platform. These services are designed to function as an external extension to the T4M platform, allowing users to model their resources, processes, and supply chains as digital twins and monitor their performance in real time. They also support the visualization of negotiations and agreements within the value network. The Balanced Scorecard tool supports the monitoring of key performance indicators across key strategic areas, while the AI Governance service focuses on explainability and responsible management of AI-driven decisions. This manual provides an overview of the main features, data structures, and user interactions for each service. Although applicable to all value networks (VNs), the current examples focus primarily on VN3.

3.3.1 Supply Chain Monitoring

To support the Supply Chain Monitoring module, particularly the negotiation loops between MaaS providers and consumers, which covers #12 from Table 1, a dedicated Messaging system has been implemented within the MIRA platform. Once a collaboration has been established, users can access the messaging functionality through the collaborator's profile page, under the Communications tab (see Figure 58 and Figure 59).

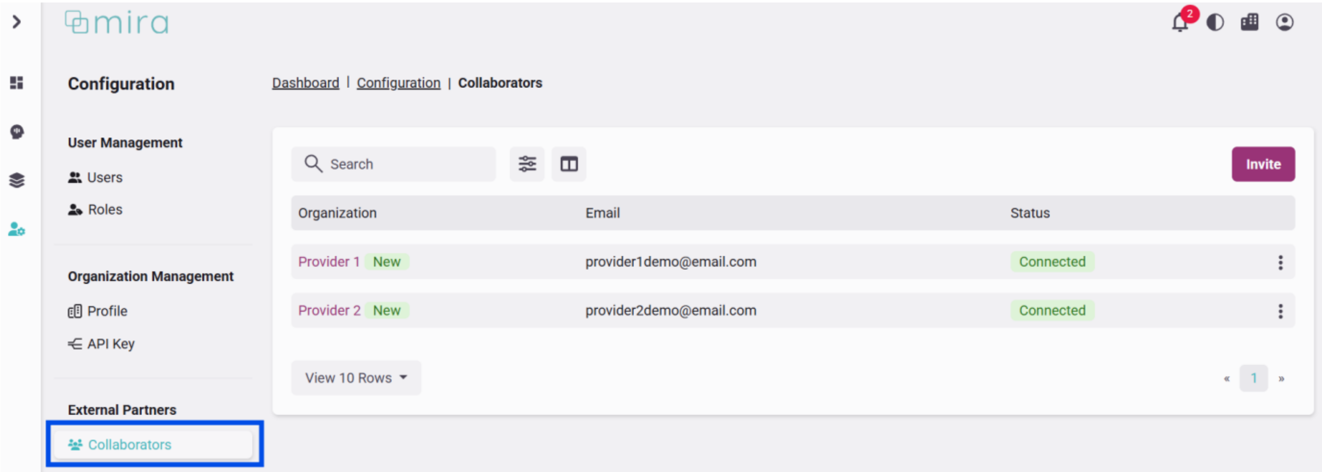

Figure 58: Collaborators List View

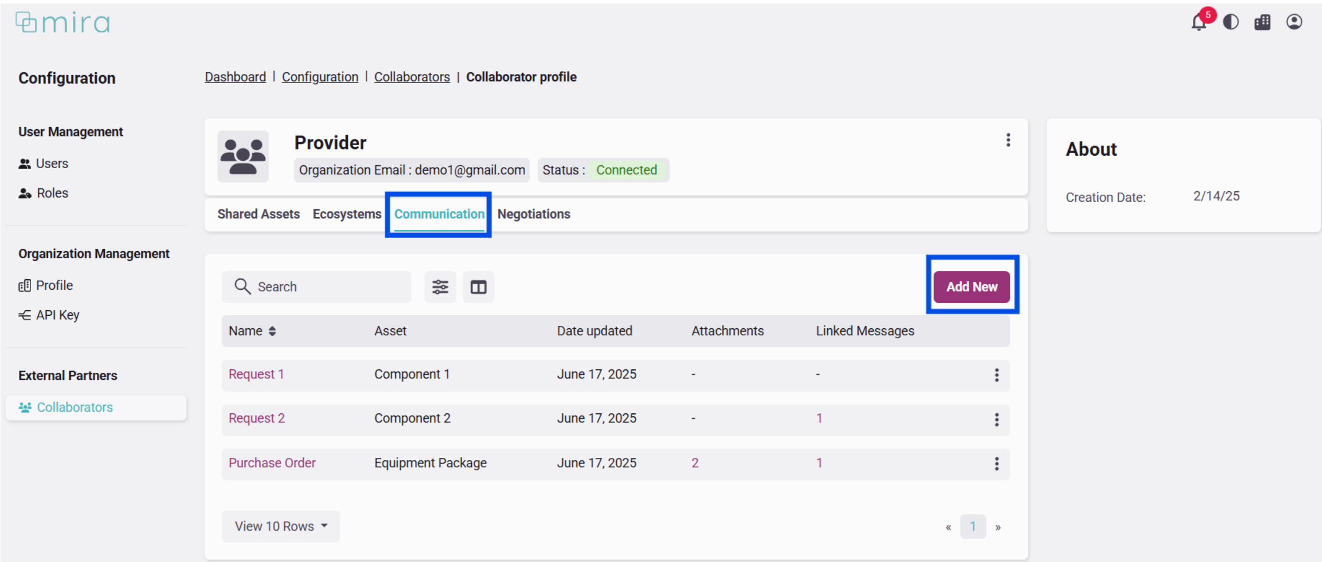

Figure 59: Accessing Messages in Collaborator Interface

From this view, a new message can be initiated by clicking the **Add New** button. The user needs to define a title for the message and choose whether to link it to a specific asset (resource), as presented in Figure 60. After this step, a form appears allowing the user to compose and send a blank message to the collaborator (see Figure 61).

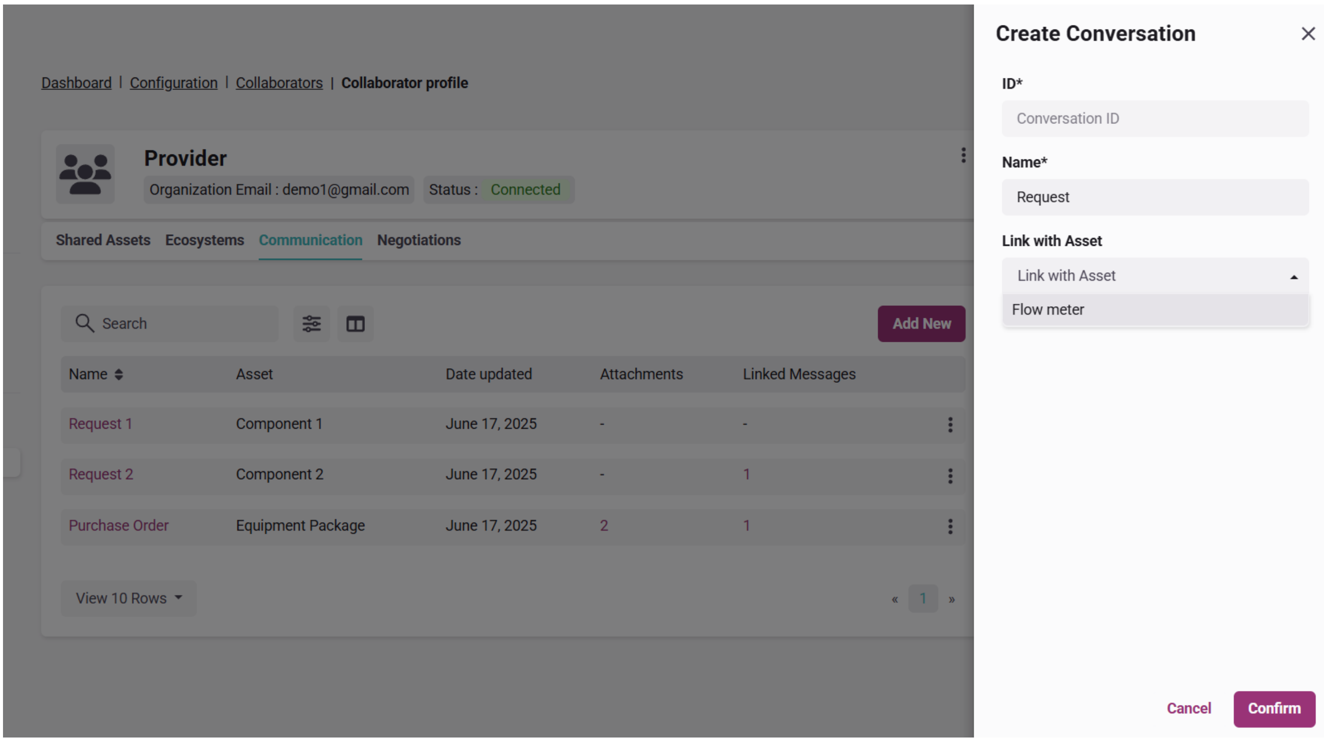

Figure 60: Creating a New Message

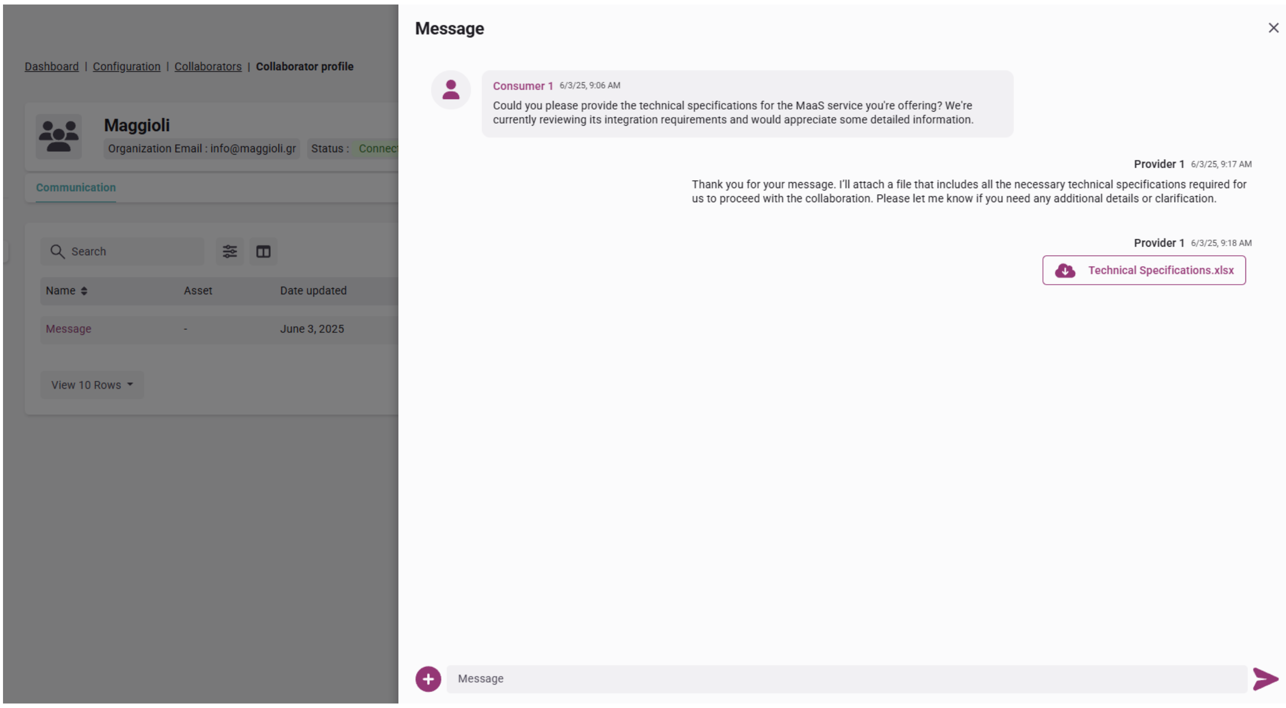

Figure 61: User Interface for New Message Composition

In addition to free-text messages, users can also send a predefined template developed based on the Purchase Request Notification Submodel. This structured template includes fields such as order quantity, delivery date, and currency. Users can complete the fields and send the Request Order Template directly to the collaborator, as depicted in Figure 62 and Figure 63.

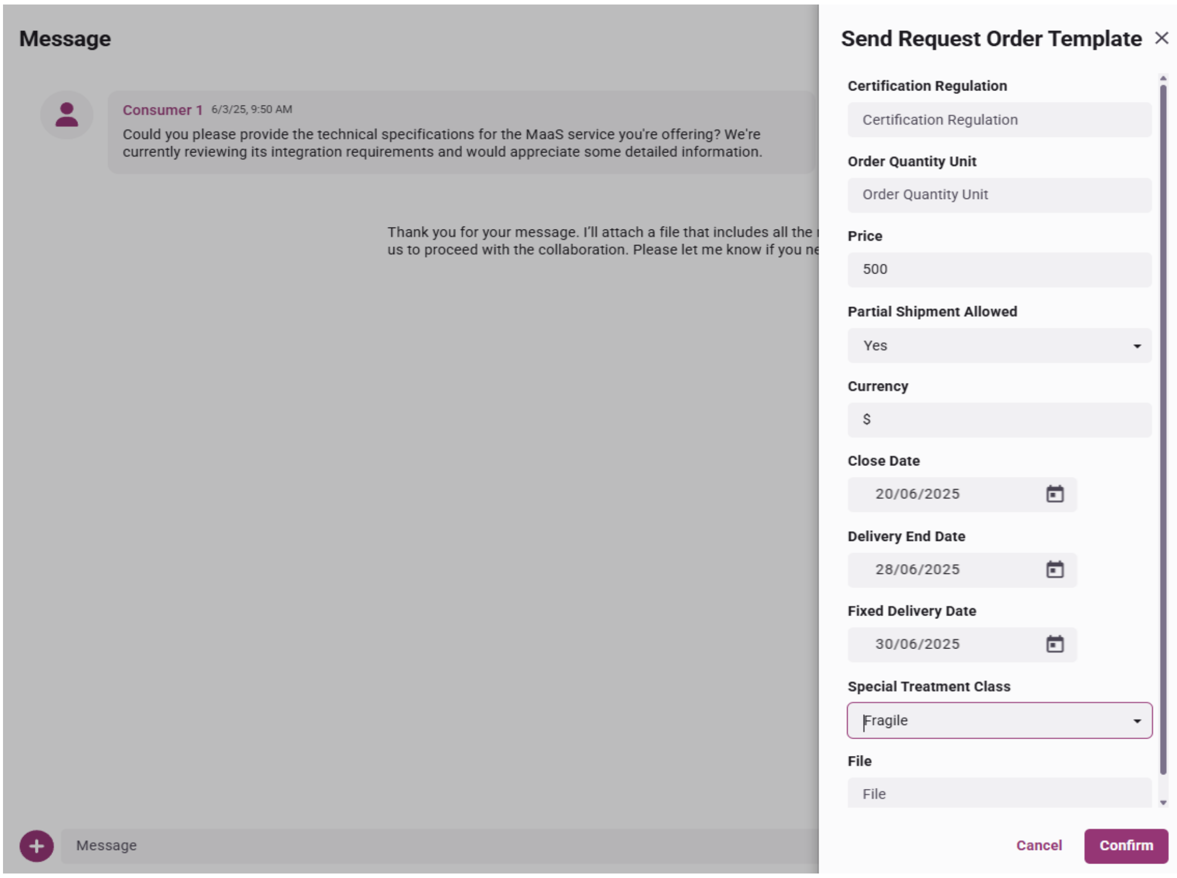

Figure 62: Creating the Request Order Template

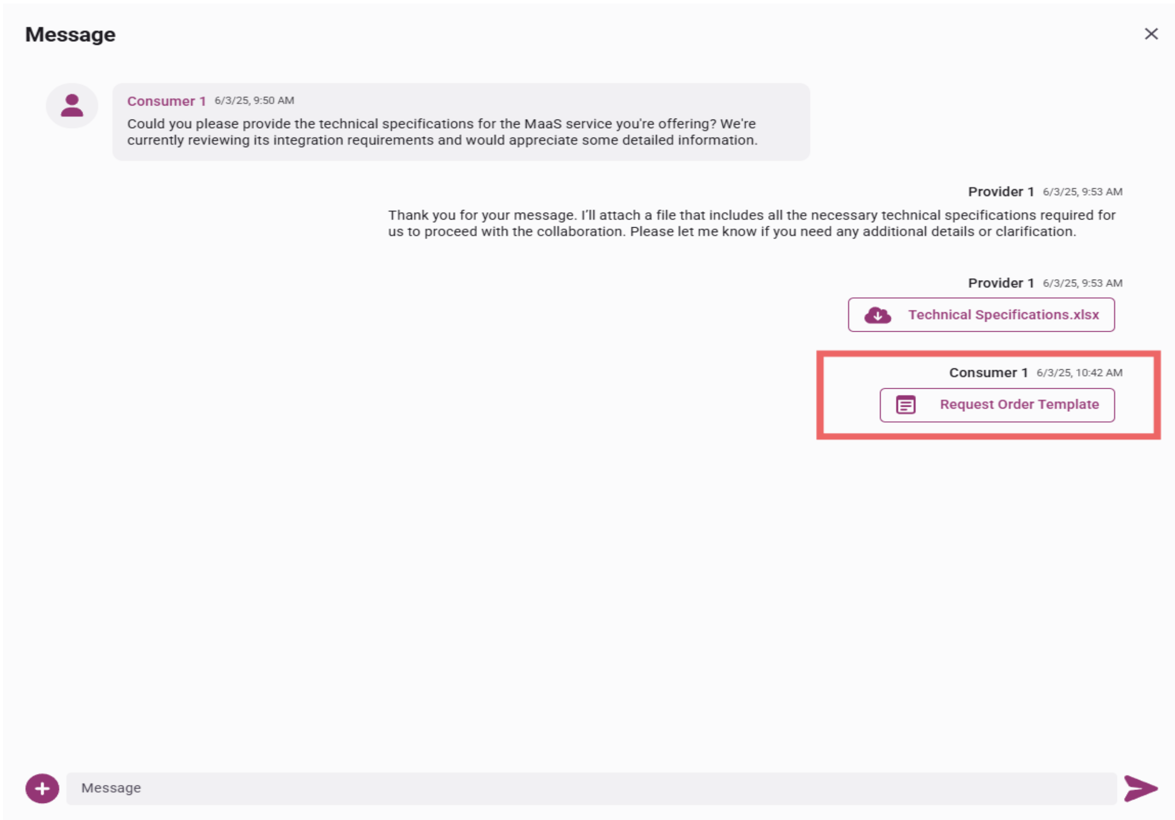

Figure 63: Exchanging Template Message

Furthermore, within the same view, users can attach files or documents to messages and link related messages within the same conversation thread. As Figure 64 shows, all linked messages and attached documents are displayed alongside each message entry in the message list, providing clearer context and helping users maintain an organized view of the negotiation history between organizations.

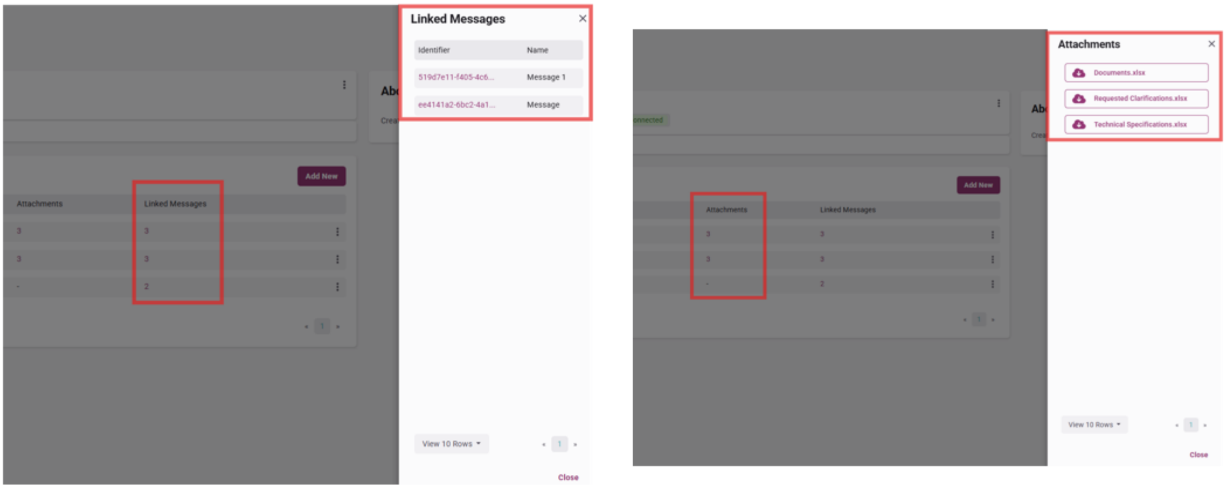

Figure 64: Display of Attachments and Linked Messages

3.3.2 Balanced Scorecard

The Balanced Scorecard tool in the MIRA platform helps users evaluate the performance of shared resources by linking actions to measurable results. KPIs are represented as telemetries, that are tied to specific entities, allowing users to explore correlations, uncover impactful relationships, and make informed, data-driven decisions.

To initiate this process, the user navigates to a telemetry's profile page within the MIRA interface. In the Linked Telemetries tab, the user can select the **Add** button to begin linking that telemetry with others available within the organization (see Figure 65). MIRA displays all existing telemetries in the system and highlights those that show high correlation with the selected telemetry. This helps users focus on the most impactful connections suggested by the system while also allowing them to manually add any additional telemetries they wish to explore for correlation (see Figure 66).

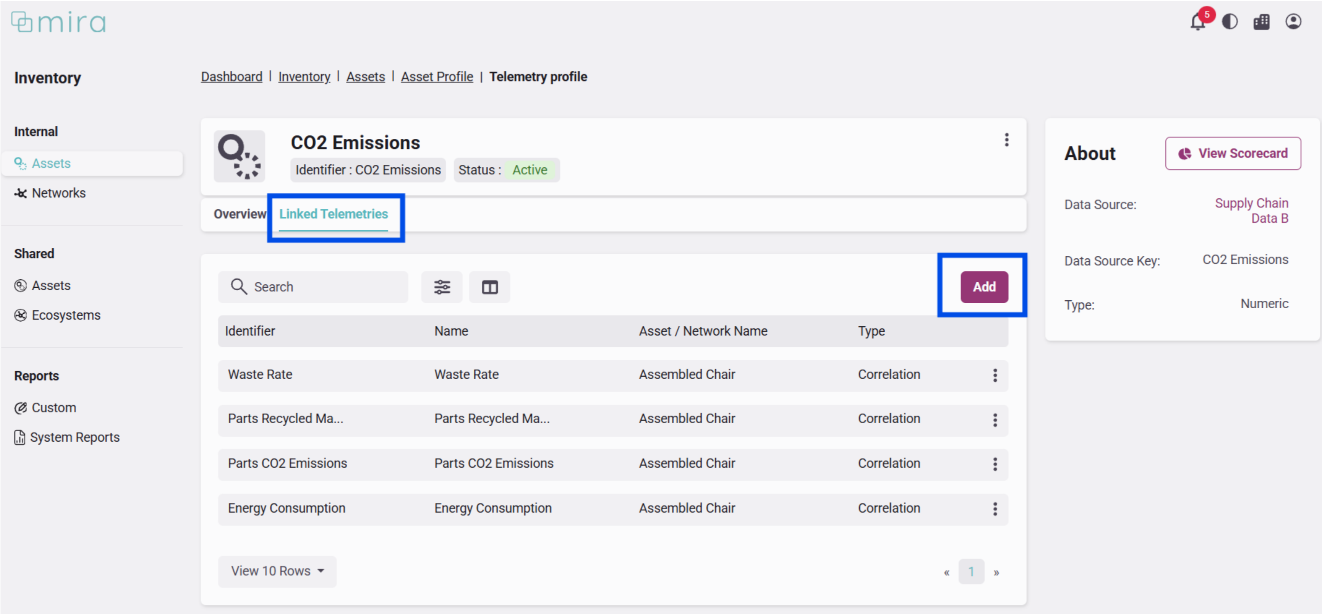

Figure 65: Telemetry Table View for Initiating Linking Process

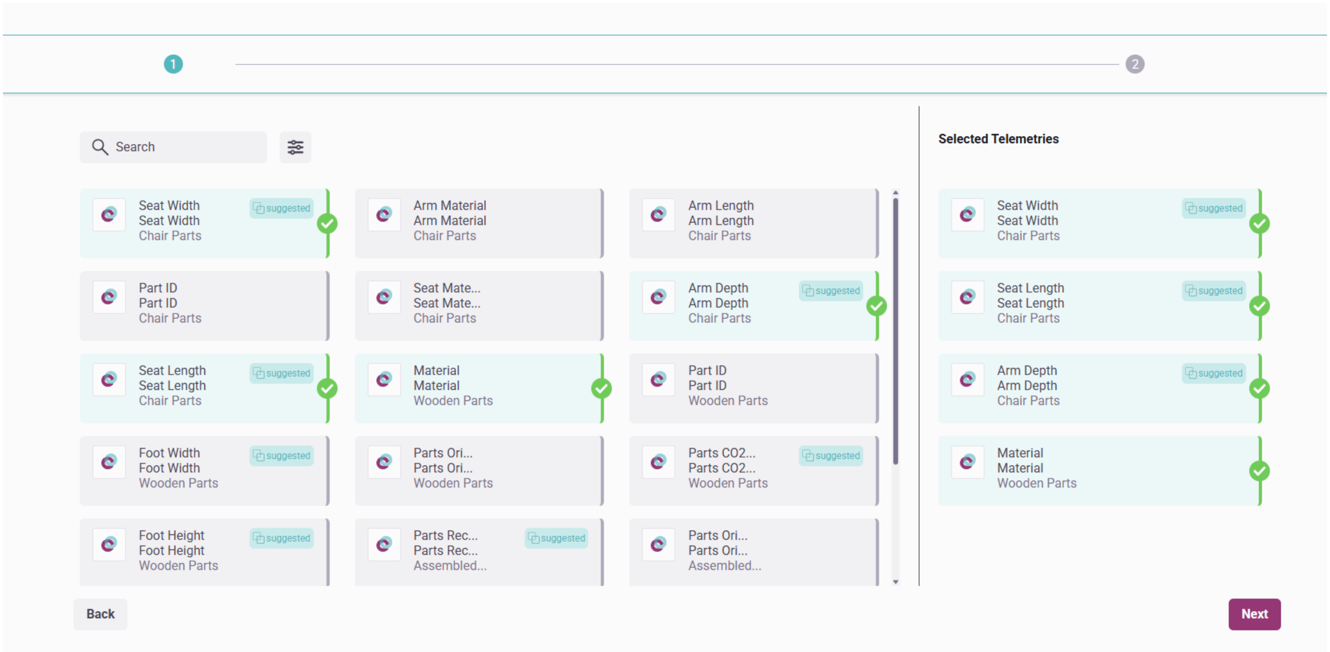

Figure 66: User Interface for Creating Links Between Telemetries

Once the relevant telemetries are selected, the user proceeds by clicking the View Scorecard button. This action redirects them to Superset; a visualization environment integrated with the platform. In Superset, the user can explore the correlation data in detail. The interface presents correlation percentages between the selected telemetry and the others, indicating whether the correlation is positive or negative, as presented in Figure 67. Additionally, if the user selects the **False** option in the **Dependencies** field, Superset will display an extended view showing correlations between the selected telemetry and all KPIs embedded in their organisation, enabling broader insight across the full set of available indicators (see Figure 68).

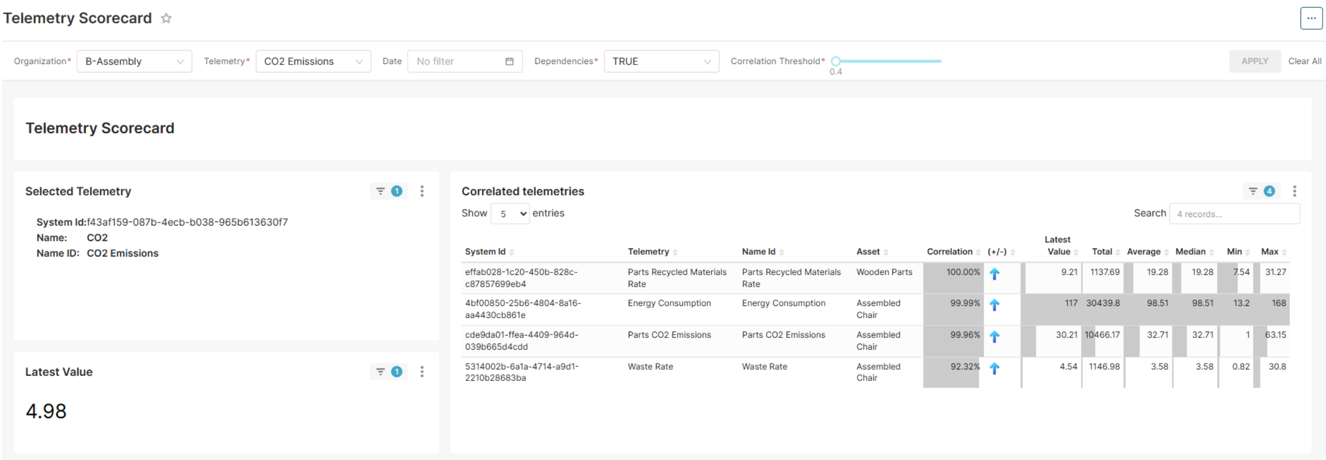

Figure 67: Correlation View for Linked Telemetries in Superset

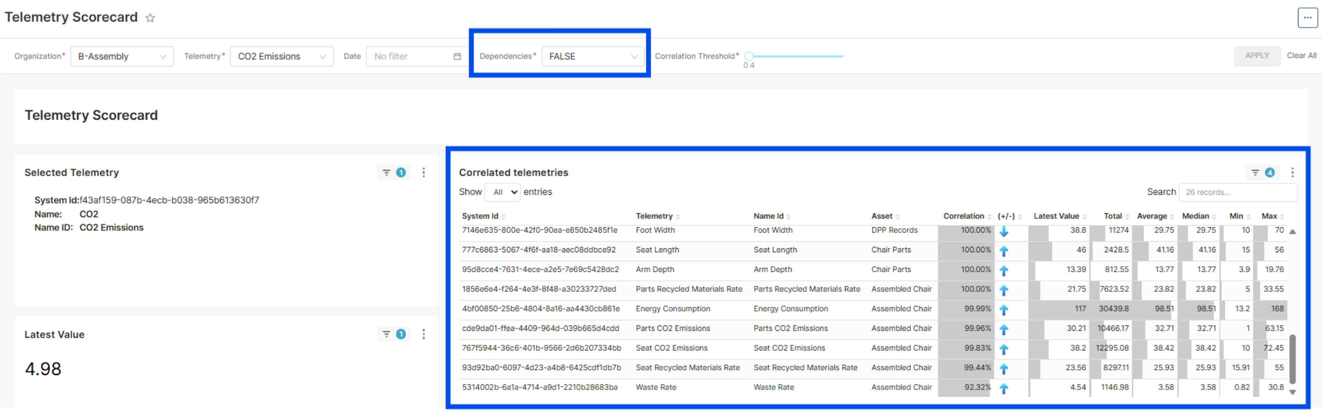

Figure 68: Correlation Between Selected Telemetry and All Organisational Telemetries

Beyond the telemetry profile view, users can also access an alternative visualization of KPI correlations by navigating to Inventory/ System Reports and selecting the **Correlations Heatmap** report (see Figure 69). This redirects them again to Superset, where they can manually select different KPIs (telemetries) available in their organisation. As illustrated in Figure 70, the resulting heatmap displays correlation percentages across the selected telemetries, offering an overview of how various KPIs relate to one another.

Figure 69: Accessing the Correlations Heatmap Report via MIRA System Reports

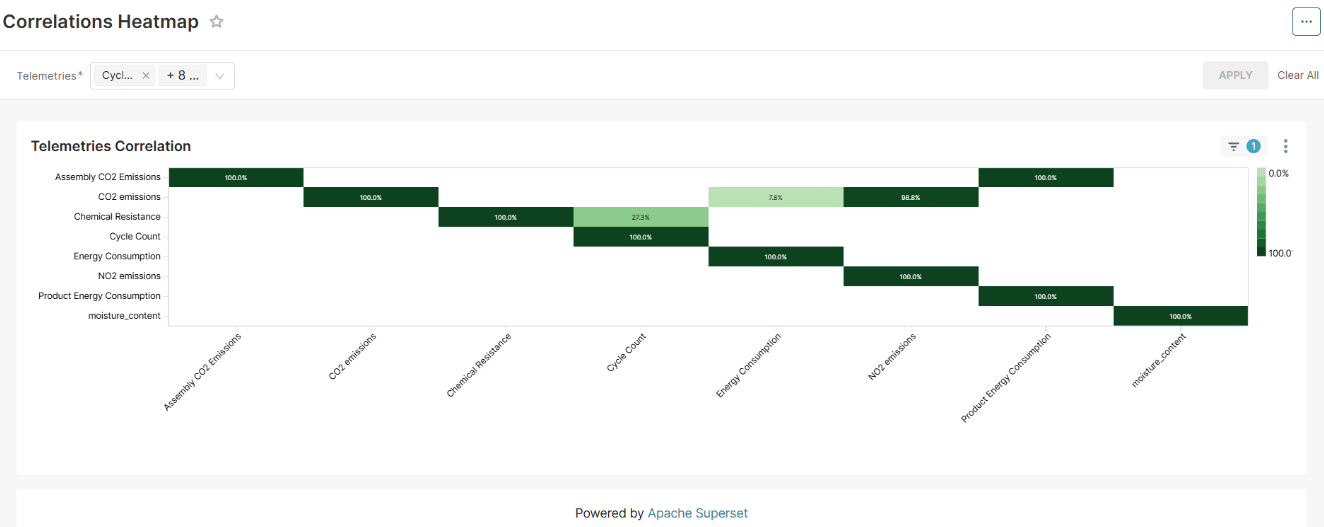

Figure 70: Correlations Heatmap Overview

3.3.3 AI Model Governance Service

To support the documentation of AI Models used during Service execution, a dedicated AI Model feature has been implemented within the MIRA platform. Users can access the AI Model feature, after selecting the desired Service, navigating to the Service's profile page and selecting the tab AI Model (see Figure 71, Figure 72 and Figure 73).

As shown in Figure 73, the Service Profile is consisted of two more tabs, the Basic Info tab where basic information about the Service is displayed, and the under-development Documentation tab, where documentation about the AI Model such as technical and performance requirements and custom fields that the Organization Administrator will be able to add, will be displayed.

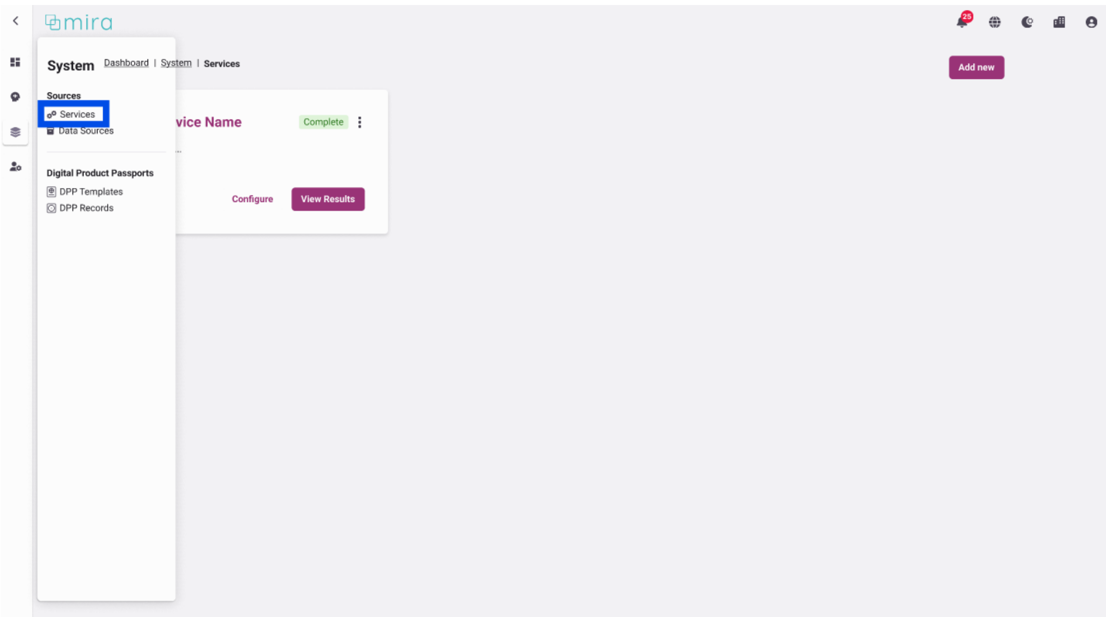

Figure 71: Accessing Services User Interface

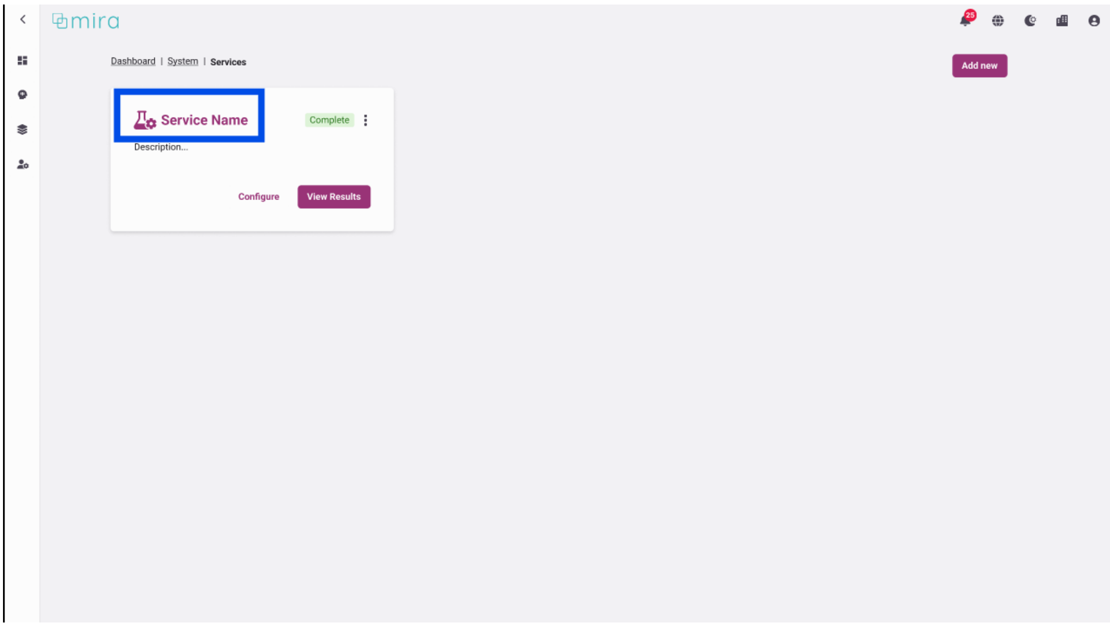

Figure 72: Available Services View

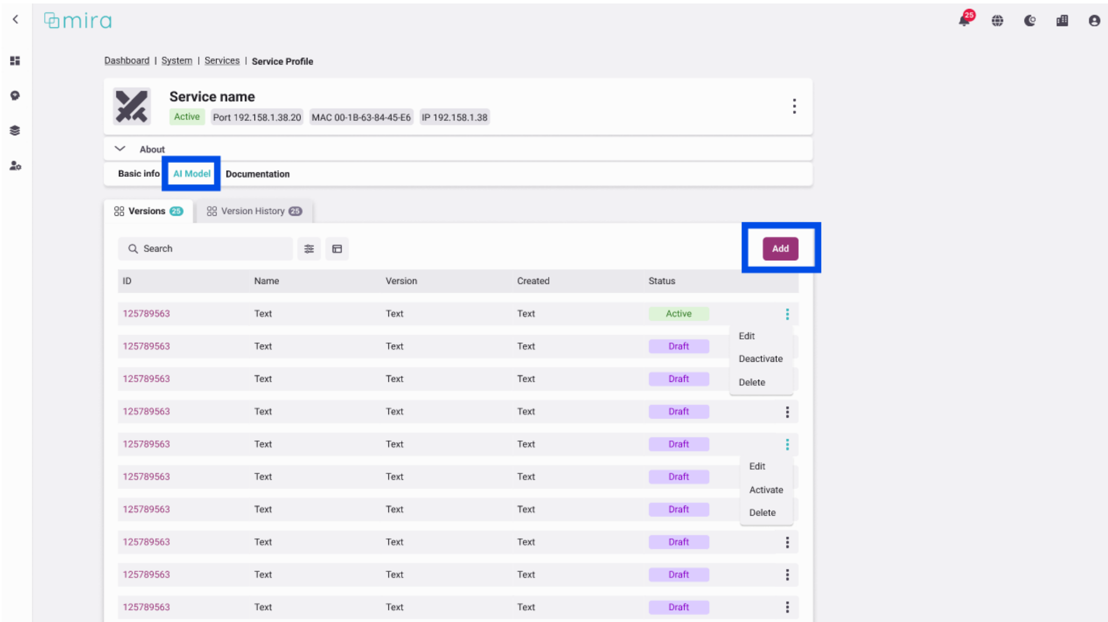

Figure 73: Table with the Available AI Model Versions with Status Active of Draft

As shown in Figure 73, all the AI Model Versions with Status Active or Draft are displayed on the table. Only one Version can be Active at a time, but multiple Versions with Status Draft can coexist. The actions available for the Active Version are Edit, Deactivate and Delete whereas for the Draft Versions Edit, Activate and Delete.

A new AI Model Version can be instantiated by the Organization Administrator by clicking the Add button and then defining the AI Model Version's Basic Information, its Name, Version, Type and Description (see Figure 74). After clicking on the Next button, custom fields can be added to document the Version's specific characteristics. In this step, the custom field's Name, Type (String, Numeric) and Value (see Figure 75) are defined. After clicking the Confirm button, the AI Model Version is added with Status Draft, and the Organization Administrator is directed to the AI Model Version Profile's Overview tab. The Version's Status can be changed into Active by selecting Activate from the Version's actions. When a Version is activated, the previously activated one is automatically deactivated and moved to the Version History subtab of the AI Model tab (see Figure 76). Deactivated Versions are read-only and can only be deleted.

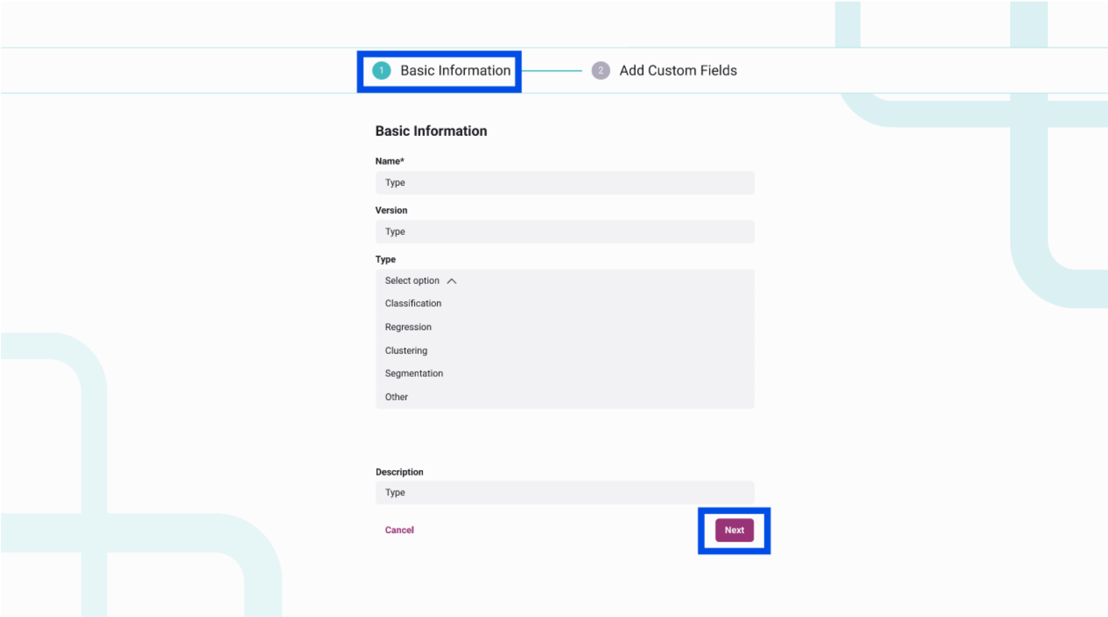

Figure 74: Filling in the Version's Basic Information

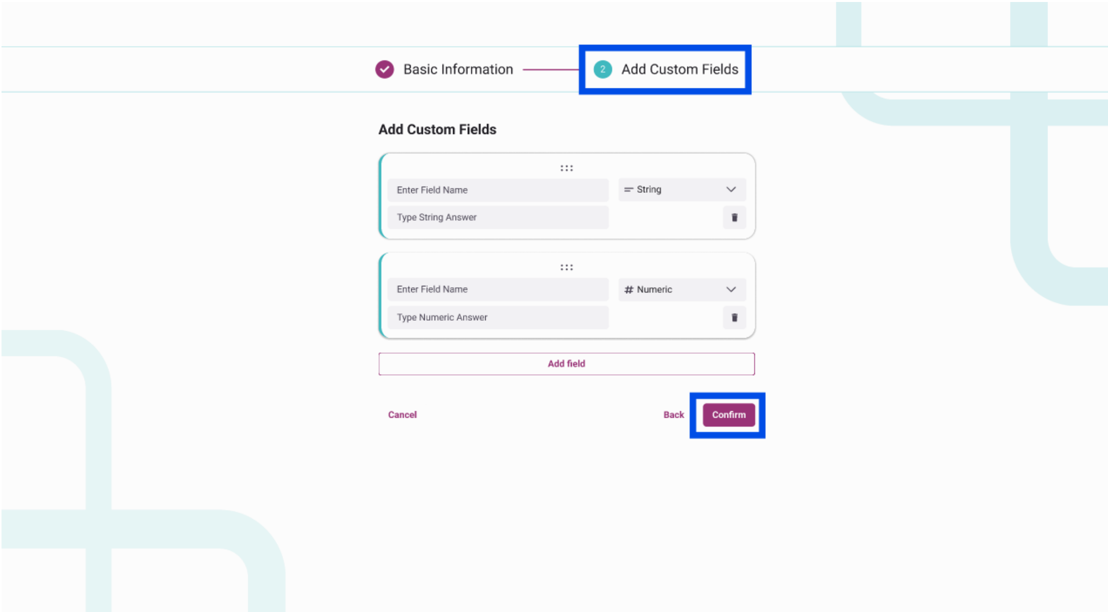

Figure 75: Filling in the AI Model Version's Custom Fields

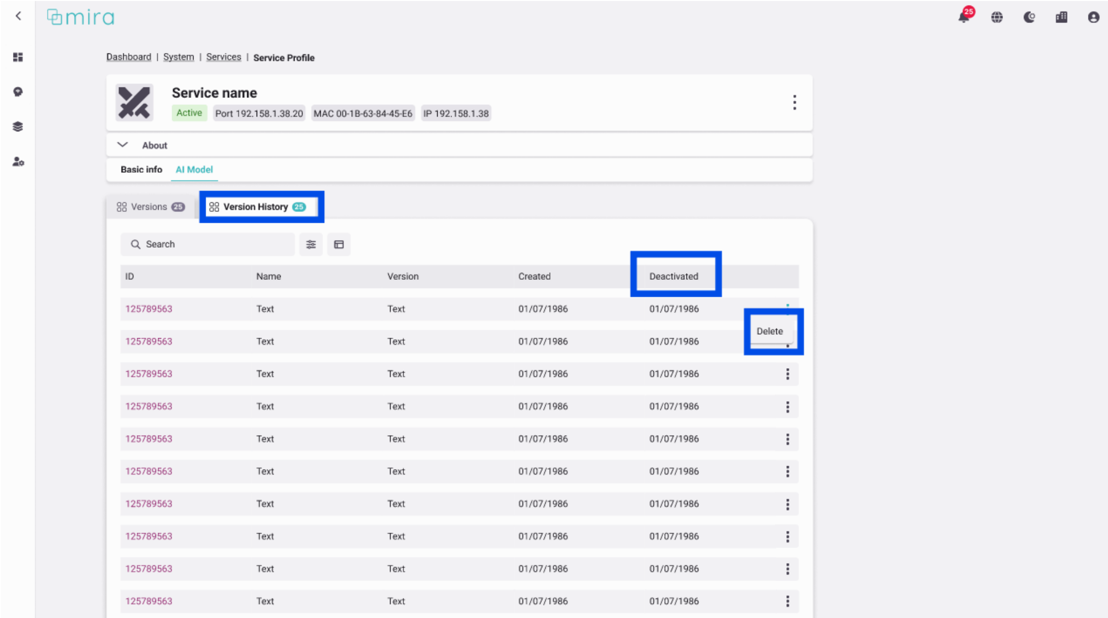

Figure 76: Table with the Deactivated AI Model Versions

On the Overview tab of the AI Model Version Profile, all the fields that were filled in when the Version was added are displayed (see Figure 77).

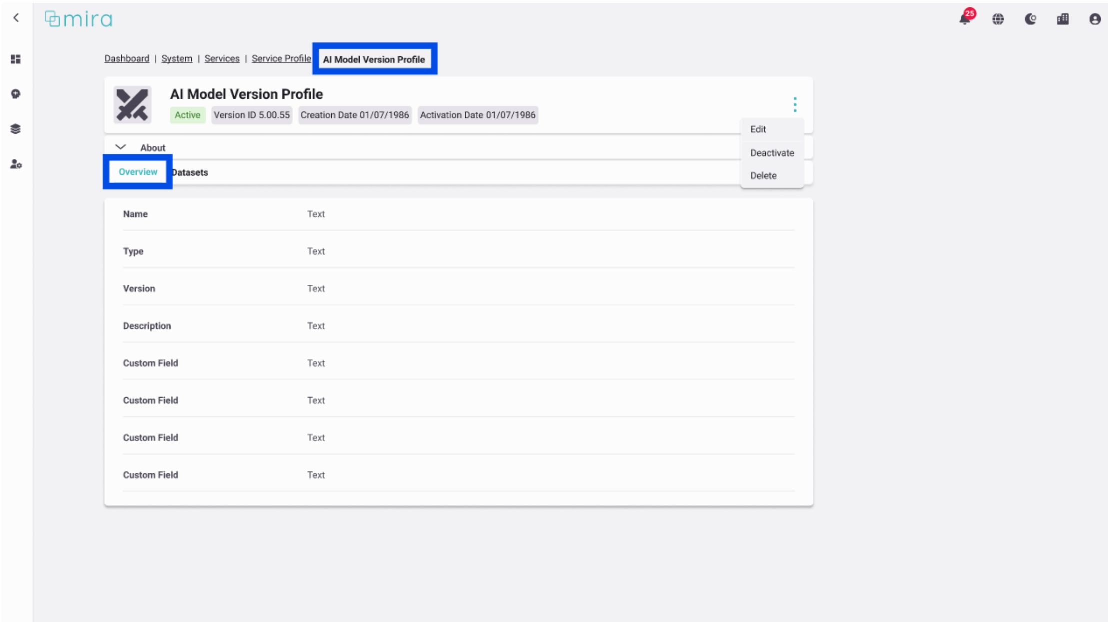

Figure 77: View of the AI Model Version's Information

On the Datasets tab of the AI Model Version Profile, a table with descriptive metadata on the datasets that were used for training or validating or some other purpose for the AI Model Version are depicted (see Figure 78). Each row contains information about the datasets that were used for training, validation or some other purpose for the AI Model Version.

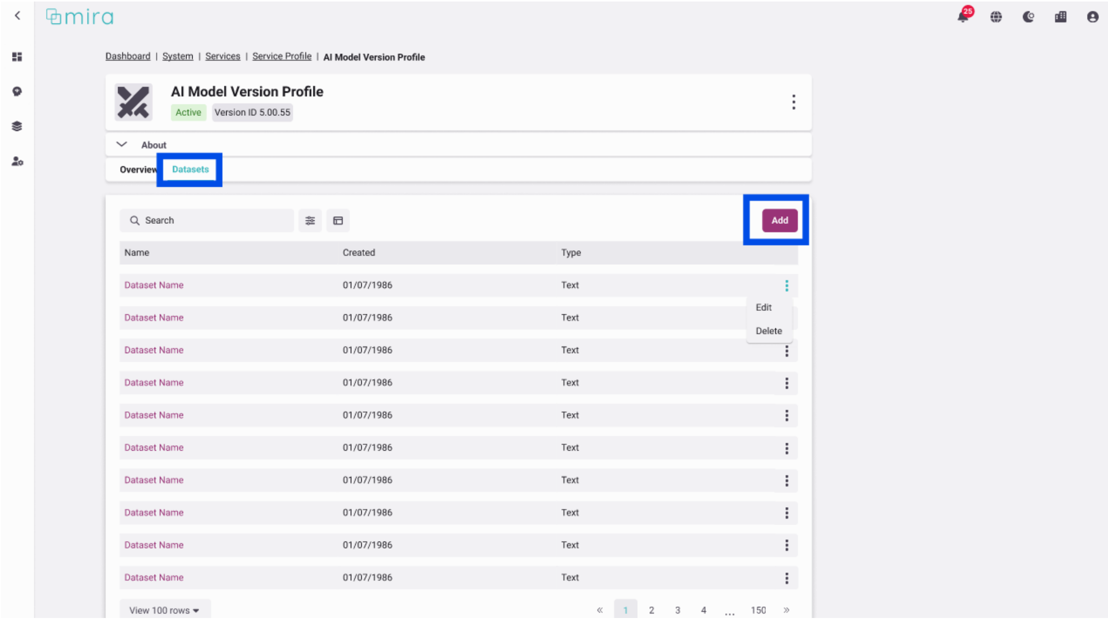

Figure 78: Table with Descriptive Metadata Used for the AI Model Version

Information about the Datasets used for the AI Model Version can be added by the Organization Administrator by clicking on the Add button, filling in the fields and clicking on the confirm button (see Figure 79).

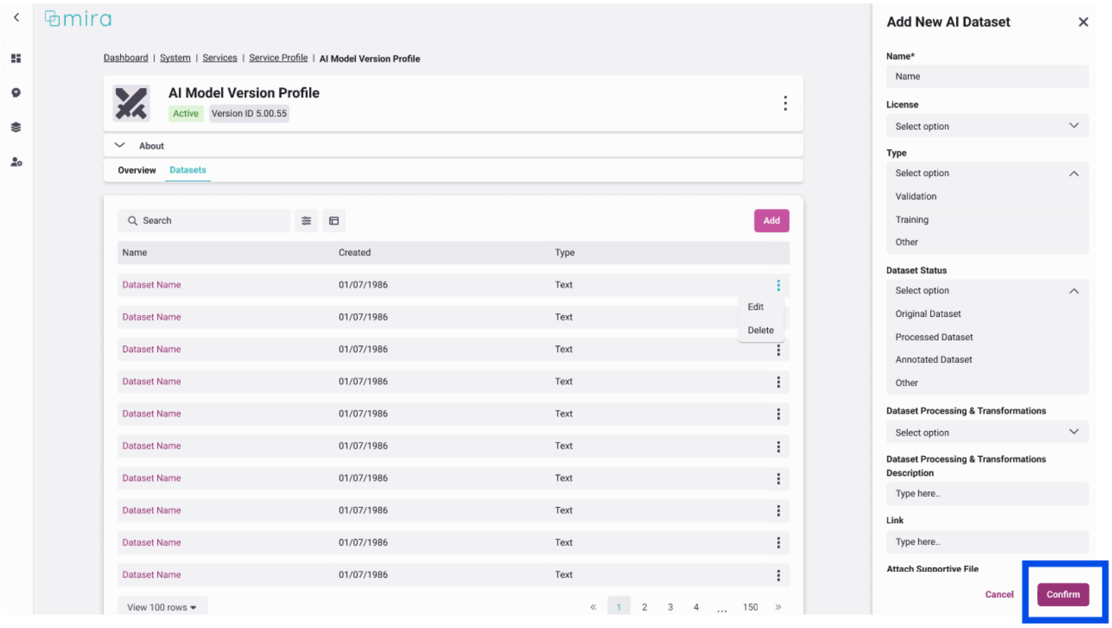

Figure 79: Adding new Dataset Information on the AI Model Version Profile

The information about the Dataset that were added, are displayed on the Dataset Profile as depicted on Figure 80.

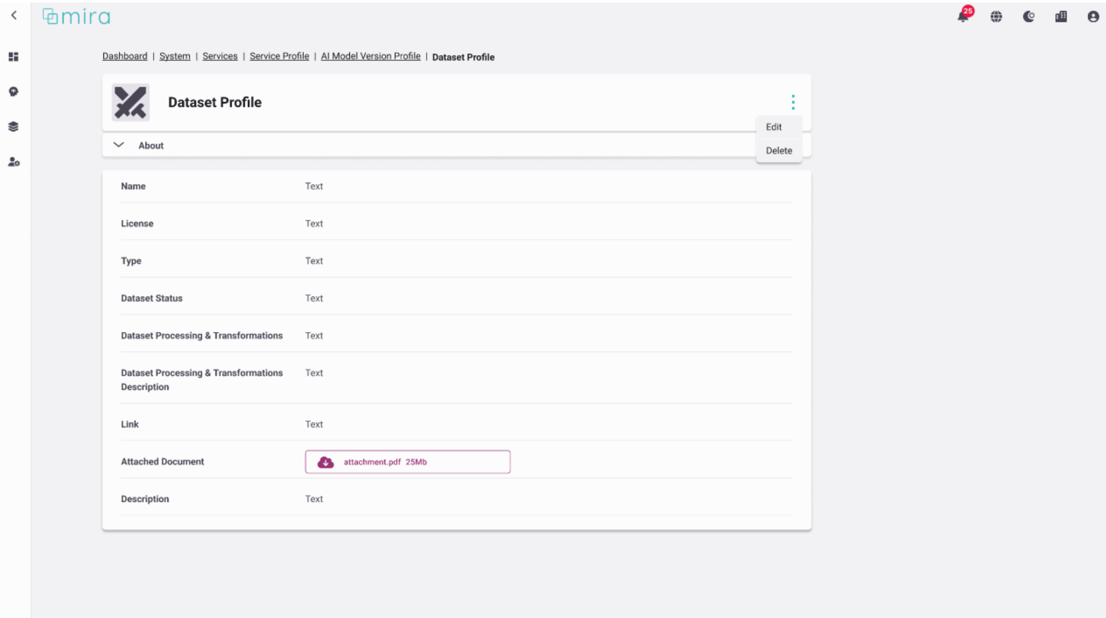

Figure 80: Dataset Profile User Interface

3.4 User Manual for AAS Model Generator – FA³ST CreAItor

The AAS Model Generator component, called FA³ST CreAItor, is described in detail in Deliverable 3.1 Digital twin models and services for factories and value networks v1. In this deliverable the focus is on the user's perspective while using the User Interface of this component to cover #4 from Table 1.

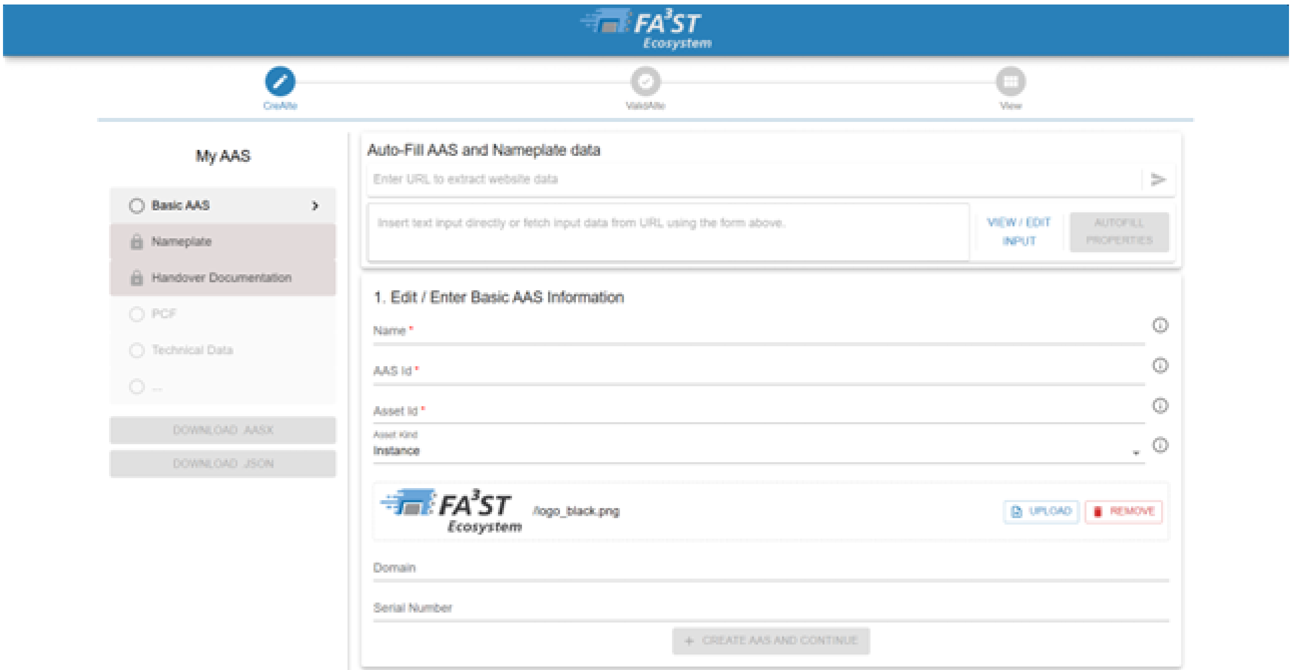

Figure 81: Start Screen of the FA³ST CreAItor

The start screen of the FA³ST CreAItor is organized into three main sections, as seen in Figure 81. On the left, users select the Submodel Templates (SMTs) they wish to instantiate, though these remain locked until basic AAS information—such as model name and asset ID—is provided. On the right, users can upload input data via URL, text, or files (CSV, PDF, TXT). URLs are processed through a web scraper, while other inputs go through an LLM-based extraction pipeline. If successful, extracted values are auto-filled into SMTs like Nameplate and Handover Documentation, but only become visible after the AAS information is entered. The third section includes a form for inputting this required AAS data, enabling the SMT instantiation process.

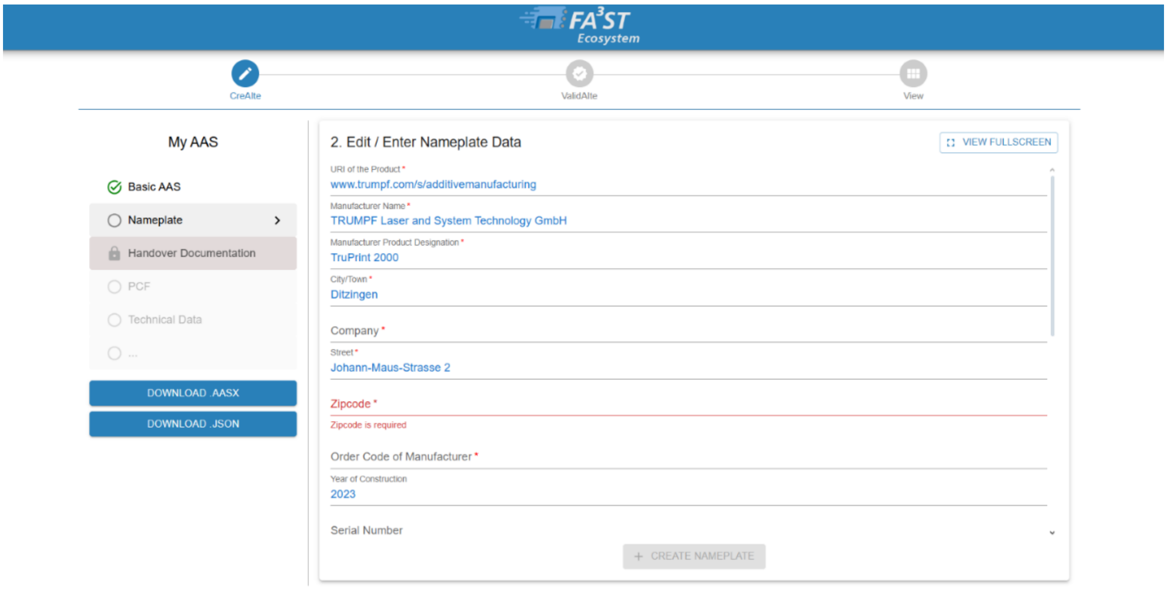

Figure 82: Digital Nameplate SMT Screen of the FA³ST CreAItor

The Digital Nameplate SMT instantiation screen, as seen in Figure 82, features a structured form on the right, pre-filled with values extracted from the user's input data. Users can review, edit, or remove these values, with key fields marked by red asterisks to indicate recommended elements per the SMT specification. Once finalized, clicking "Create Nameplate" adds the instantiated Submodel to the current AAS. On the left, users see an overview of all already-added Submodels. After instantiation, the full AAS model can be exported in AASX or JSON format, ensuring compatibility with tools like the FA³ST Service or AASX Package Explorer.

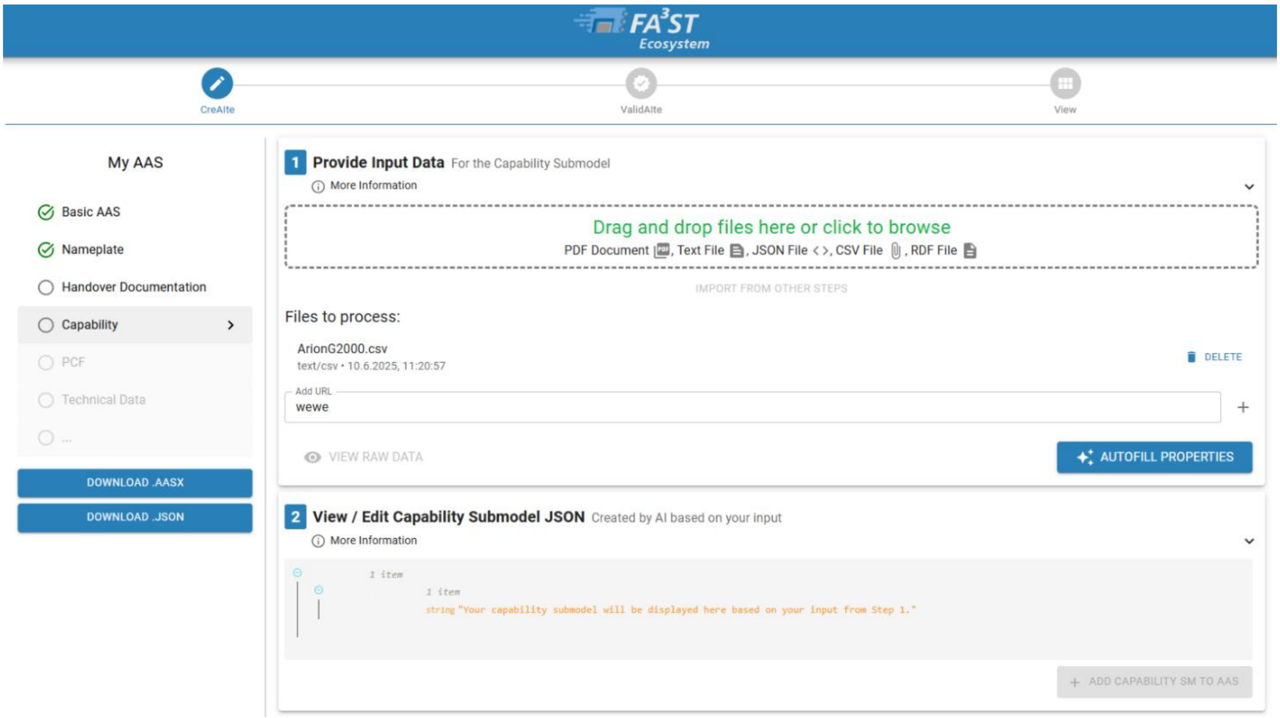

Figure 83: Capability Description SMT Screen of the FA³ST CreAItor

The Capability Description SMT interface is divided into three main areas, as seen in Figure 83. On the right, users can upload input files (e.g., PDFs, CSVs) and optionally domain-specific ontologies in RDF format to guide the SMT instantiation. If no ontology is provided, a default one is used. Clicking "Create Capability SM" triggers automated instantiation, and the resulting AAS JSON is shown above for user review and annotation. Once confirmed, users can add it to their AAS. On the left, users can export the completed AAS in either AASX or JSON format using buttons at the bottom of the sidebar.

3.5 User Manual for Information Model Framework (IMF) Editor

IMF Editor is based on yEd, a graph modelling tool with freeware license that supports GraphML, an open language for representing graphs, equipped with custom modelling palettes for IMF. The IMF palettes are open source.

This manual covers #5 from Table 1.

Getting started:

1. Download yEd: https://www.yworks.com/products/yed.
2. Download IMF Editor palettes: https://gitlab.com/imf-lab/tool/imf-graphml
3. Open yEd and load the two IMF palettes IMF-Elements.graphml and IMF-Relations.graphml (Go to Edit -> Manage palette (Figure 84) -> Import section (Figure 85) -> [Move imports to "Displayed Palette Sections"]) (Figure 86).

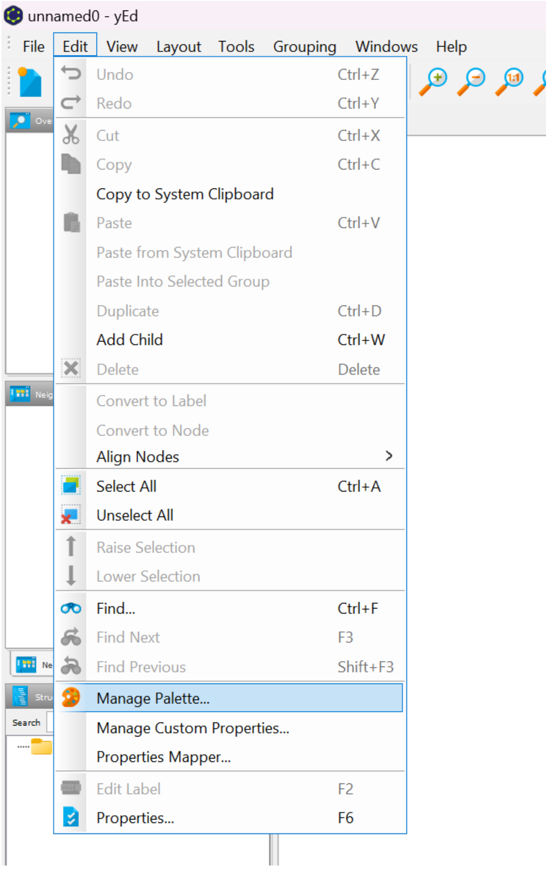

Figure 84: Step 3 Edit > Manage Palette...

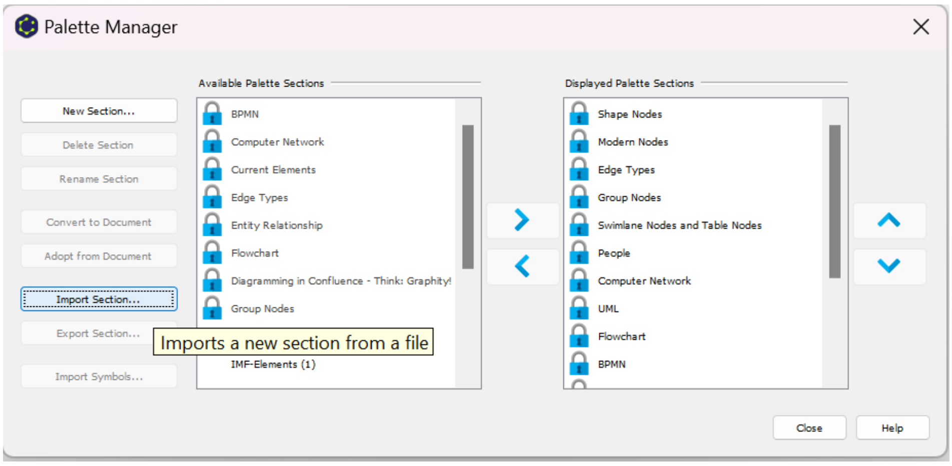

Figure 85: Step 3 Import section

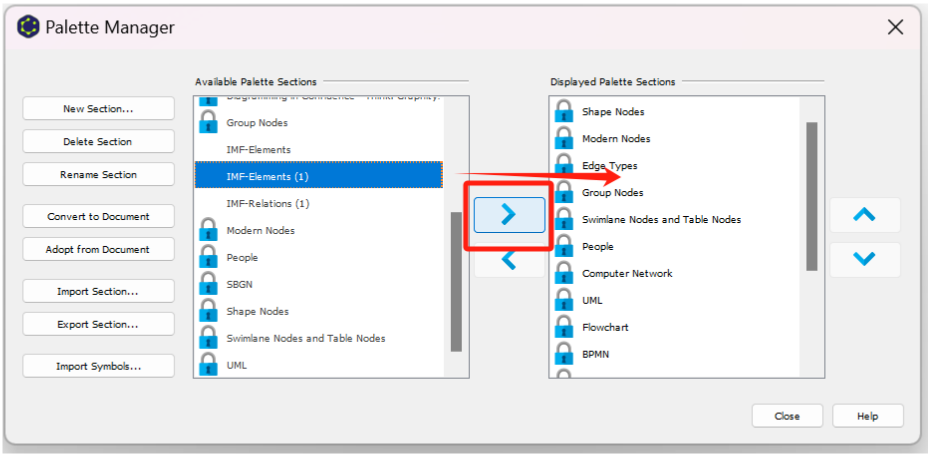

Figure 86: Move imports to "Displayed Palette Sections"

4. Create your IMF model by constructing a graph in yEd using the two palettes only. Save your work as a GraphML file (Figure 87).
5. The GraphML file is converted to RDF/XML format using the XSL Transformation script found in the Git repo: graphml2rdf.xsl using an XSLT service that supports the XSLT 2.0 standard.

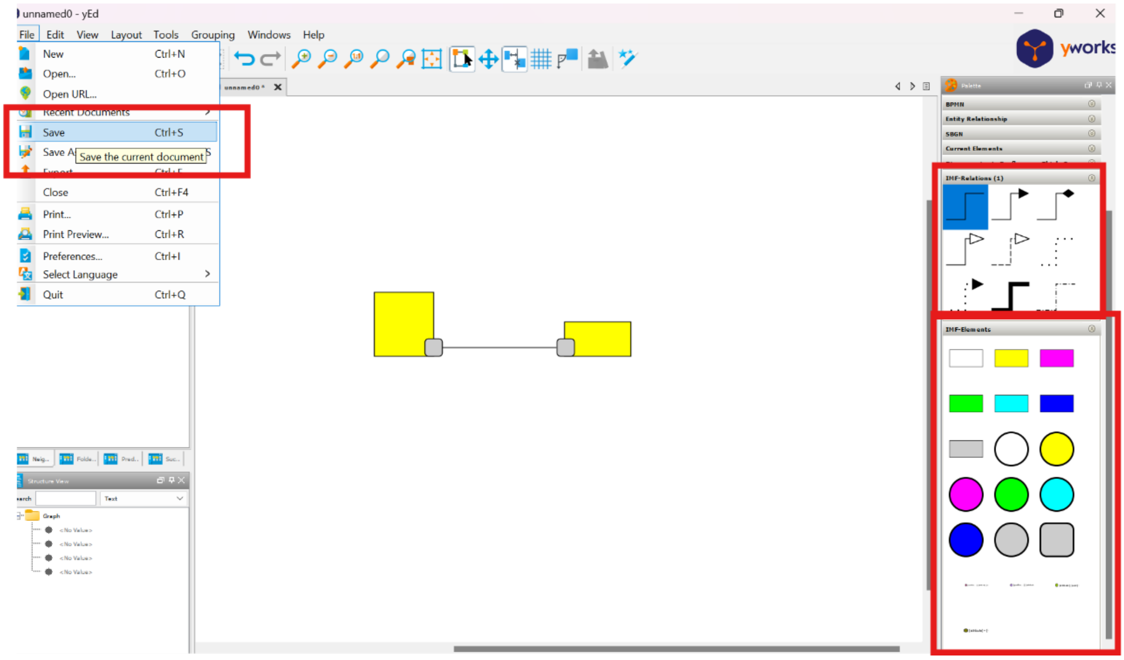

Figure 87: Step 4 Create and Save your IMF model

Conclusions

This document serves as the report following the current version of the 1st integrated prototype of the Tec4MaaSEs platform available for demonstration. It provides an overview of the available functionalities, showcasing the initial capabilities of the platform and it is aimed to integrate all tools of Tec4MaaSEs System. While this version offers a first look at the core features, it is intended as a starting point for further validation and development. The Tec4MaaSEs platform will act as a testbed for stakeholders to explore its capabilities, evaluate its approach, and provide feedback for future refinements.

The current implementation represents only a partial realization of the platform’s full potential. Tec4MaaSEs is designed with flexible and modular architecture, allowing for continuous improvements, feature expansions, and integrations with additional tools per Value Network. The System remains adaptable to evolving MaaS needs, ensuring scalability and efficiency for its users.

Future iterations will integrate the platform with Supply Chain Monitoring to keep the platform updated on the statuses of negotiation and orders as they take place in the Supply Chain Monitoring Tool. Next versions will also carefully integrate the triggering of the Optimization Service and the visualization of its results to the Analytics Dashboard, seamlessly for each Use Case. An important future functionality is the selection by the end user of a service composition, as provided by the results of the Optimization Service, which will trigger the Negotiation process in the Supply Chain Monitoring. These foreseen steps are driven by further development of project tools and focus on enhancing stability, usability, and automation, further refining the platform’s capabilities to meet industry standards and Value Network needs.

Upcoming versions of the Tec4MaaSEs platform will introduce new functionalities, including the rating of collaborators based on specific criteria, the history of collaborations and their current statuses, history of service requests and a further integration with the Dat Space Connector to receive real time data that have to do with the capacities of the provider organizations. These updates will support broader testing and validation activities, ensuring that the platform evolves to meet all functionalities foreseen in the Grant Agreement and the needs of Industry professionals. The ongoing development cycle will incorporate stakeholder feedback and scheduled work across technical work packages, leading to an enhanced release in future iterations of the platform.

References

[1] “Kubernetes Documentation | Kubernetes.” [Online]. Available: https://kubernetes.io/docs/home/

[2] “Ingress | Kubernetes.” [Online]. Available: https://kubernetes.io/docs/concepts/services-networking/ingress/

[3] “Lens | The Kubernetes IDE.” [Online]. Available: https://k8slens.dev/

[4] “Technical documentation | Grafana Labs.” [Online]. Available: https://grafana.com/docs/

[5] “Overview | Prometheus.” [Online]. Available: https://prometheus.io/docs/introduction/overview/

[6] “GitHub Actions documentation - GitHub Docs.” [Online]. Available: https://docs.github.com/en/actions

Appendix A: Components v0.5

Table 2: Component links

Component	Git Registry	Access / Documentation Link(s)
Digital Twins: FA³ST Service	FA³ST GitHub	SwaggerHub API Docs *(Only partial APIs implemented)*
AAS Model Generator (FA³ST CreAItor)	Private	N/A
Information Model Framework Editor (IMF Editor, yEd custom)	IMF Editor GitLab	Desktop application
User Interface	N/A	T4M Frontend
User Management	N/A	Users API, Admin API, User Manager API, Keycloak
Organization Management	N/A	Organization API, Org Web Interface
Message Bus	N/A	kafka.t4m.atc.gr:9092, Conduktor UI
Notification Service	N/A	Notifications API, Event Mappings, Notification Service API
Search & Match	N/A	BoPG API, EB-MaaS API, MACH-MaaS API, Materials API, Search Match API
Dataspace Connector (Tekniker)	Docker Package *(EU-Tec4MaaSEs access required)*	Not public; no Swagger; available upon deployment
Bill of Processes Generator	Docker Package *(EU-Tec4MaaSEs access required)*	Private
Optimization Service	GitHub Repository	Swagger UI
Forecasting	Private	Private
Predictive / Proactive Analytics	N/A	N/A
Supply Chain Monitoring / Balance Scorecard / AI Model Governance	Resides on MIRA (not public)	MIRA Swagger
Grafana Monitoring	N/A	Grafana Dashboard

1. Digital Twins: FA³ST Service

FA³ST Service[1] enables the execution of Digital Twins (DTs) based on the Asset Administration Shell (AAS) specification, allowing users to interact with DTs as a service. It supports hosting multiple AASs, submodels, and concept descriptions, and facilitates synchronization between digital twins and physical assets through value/property synchronization and operation delegation over protocols like OPC UA, MQTT, and HTTP. Its flexible and open architecture allows different implementations for key components such as persistence (in-memory, file-based, MongoDB), file storage, message bus (internal or MQTT), endpoints (HTTP, OPC UA), and asset connections. Configuration is simplified through JSON-based config files and environment variables, and the service can be deployed via a command-line interface (JAR), Docker container, or embedded as a Java library.

Hardware requirements for FA³ST Service depend on the complexity and scale of the deployment. A minimal setup typically consumes around 500MB of RAM and has low CPU usage unless numerous user requests or asset subscriptions are active. Factors like the size of the AAS model, number of asset connections, and the choice of interface implementations influence resource demands. The API is determined by the enabled endpoints (HTTP and/or OPC UA), with the HTTP API offering full functionality and two proprietary calls—/reset and /import—for managing the server state and importing models. The OPC UA API, however, supports only basic operations like reading and updating properties, without support for structural modifications.

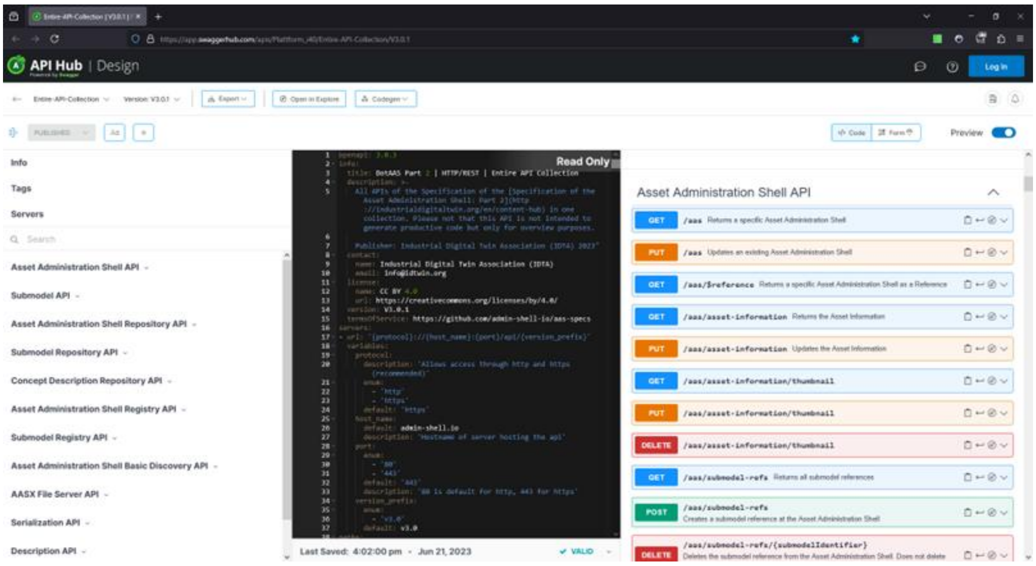

Figure 88: Official AAS Swagger API

2. AAS Model Generator – FA³ST CreAItor

The current version of the AAS Model Generator, called FA³ST CreAItor, enables the semi-automated instantiation of Submodel Templates provided by the IDTA, allowing non-expert users to build their own standard compliant Asset Administration Shell (AAS) models. It currently supports Instantiation of the following Submodel Templates: Digital Nameplate for Industrial Equipment, Handover

[1] https://faaast-service.readthedocs.io/en/latest

Documentation, and Capability Description. The component leverages large language models (LLMs) to process unstructured textual input and extract relevant attribute values for the instantiation process. It supports both OpenAI models and locally hosted LLMs with Ollama. The component offers a user-friendly interface for annotation and completion of extracted values, streamlining the AAS modeling process. It is designed as a stand-alone service and does not expose an external API, aligning with its role in the T4M project context. It is intentionally designed as a stand-alone service to support participants of the T4M platform in generating the AAS models relevant to their specific use cases.

FA³ST CreAItor is containerized and deployable via Docker, allowing local execution with configurable options to specify which LLMs should be used. It can also be accessed as a hosted web service provided by FhG IOSB. The component's hardware requirements are defined by how the LLMs it uses are hosted. Two options are available: OpenAI API – LLMs are hosted externally, no additional local hardware is required, or locally (via Ollama) – requirements depend on the model size. For example: Gemma3 4B requires approximately 2.3 GB VRAM, LLaMA 3.2 3B approximately 6.9 GB, LLaMA 4 Scout approximately 76.2 GB, and LLaMA 3.3 70B approximately 161 GB. For most use cases, it is recommended to use GPT-4o-mini (2024-07-18) via the OpenAI API as a performant and cost-efficient option.

3. Information Model Framework Editor

The IMF Editor is a tool for modelling engineering systems within the Information Modelling Framework (IMF). Its core functionality centers on placing IMF elements (block, terminal, and connector) and defining relations between them. Users can customize elements by modifying their names, aspect types, and sizes, and can extend them with custom attributes that include fields for name, value, unit, and attribute qualifier. All elements and relations are visually represented as shapes and lines on a canvas. The modelling experience is designed to be intuitive and user-friendly. Users can export the graphical model into RDF format for further usage.

The current IMF Editor is based completely on yEd, a graph modelling tool that supports the GraphML, an open language for representing graphs. The IMF Editor is equipped with two sets of predefined palettes that users must use to build IMF models. IMF models in GraphML format may be transformed to RDF format following the IMF OWL ontology using an XSL transformation.

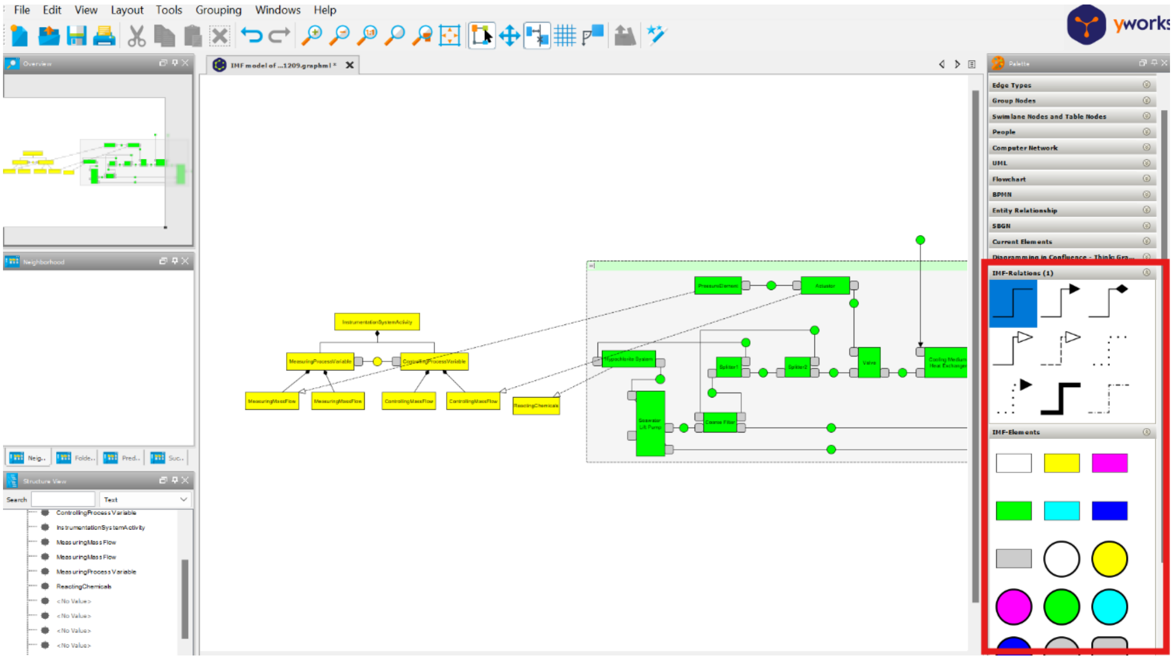

Figure 89: IMF Editor: yEd equipped with two palettes (highlighted by red box).

4. User Interface

The User Interface is a React.js-based front-end that provides access to key functionalities of the platform, including User Management, Organization Management, and Search & Match. In version 0.5, it allows users to create and manage accounts, register organizations, submit service requests through interactive forms, upload supporting files, browse and filter providers, and access role-based views. It connects with back-end services via REST APIs and integrates with Keycloak for secure authentication and authorization.

The UI is packaged in a Docker container for consistent deployment. It requires minimal hardware (e.g., 1 vCPU, 2GB RAM) and can be deployed independently or as part of a larger docker-compose setup. Environment-specific configurations (e.g., API endpoints, Keycloak settings) are provided through runtime environment variables or configuration files.

5. User Management

The User Management component is a Java Spring Boot service that handles user and role management for the T4M platform. It integrates with Keycloak for identity and access management, supporting authentication, authorization, and user provisioning. Version 0.5 includes the full set of role definitions and core user management functionalities such as user creation and activation, role assignment, and access control enforcement. This ensures consistent and secure handling of user permissions across the platform.

The component is deployed as a Docker container and operates alongside a Keycloak instance. It requires standard resources (e.g., 2 vCPUs, 2GB RAM, 10GB disk storage) and exposes RESTful APIs for integration with other services. Configuration is managed via Spring profiles and Keycloak realms.

6. Organization Management

The component is a Java Spring Boot service backed by a MongoDB database, developed to support basic organizational onboarding and discovery functionalities. It enables the registration of organizations, viewing a list of available providers, and filtering providers based on defined criteria. In this version (v0.5), it also integrates with a Dataspace Connector instance to validate provided URLs and retrieve datasets, specifically focusing on the "capabilities" aspect of the offering metadata. This allows for early validation of cross-component integration and discovery flows.

The component is fully containerized using Docker, enabling easy deployment across environments. It requires moderate system resources (e.g., 4 vCPUs, 2GB RAM, 50GB disk storage) and depends on a running MongoDB instance. It communicates externally with a Dataspace Connector via HTTP endpoints.

7. Message Bus

The Kafka Message Bus serves as the backbone for asynchronous communication between components within the T4M platform. It enables decoupled data exchange through a publish-subscribe model, facilitating scalability and modularity. In version 0.5, it supports key messaging topics for system-level events, service interactions, and integration with components such as the Bill of Processes Generator and Search & Match.

Kafka Cluster is deployed in a Dockerized setup, using the KRaft algorithm for controller election. Kafka Schema Registry is also deployed alongside with Kafka Cluster to validate only specific schemas of events. Finally, to manage the Kafka Cluster with its topics and messages, Conduktor UI is utilized to provide an overview of the cluster's brokers. Kafka cluster requires moderate resources (e.g., 4 vCPUs, 6GB RAM, 50GB disk storage) depending on message volume. Topics and consumer groups are pre-configured for the prototype stage. The setup supports secure communication and can be extended with schema registry and monitoring tools in later versions.

The following are tables describing the data schema of the events (Table 3) and the defined topics for version 0.5 (Table 4). An additional variable, named "organization" will be added in the next iteration to distinguish the relevant organization for each event.

Table 3: Event Data Schema

Variable Name	Type	Values	Comments
ID	String	String	Generated automatically
type	String	String	e.g. Decomposition of a Service
description	String	String	Optional - Can provide more

			information regarding the specific request of an event type.
sourceComponent	String	String	Component generated the Event
timestamp	ZonedDateTime	Datetime in UTC	Pattern: "yyyy-MM-dd'T'HH:mm:ss'Z'"
priority	Enumeration	High / Mid / Low	-
data	Object	Object	Generic Object. Depends on the topic. Can be null if event does not generate any related data

Table 4: T4M Platform topics

Topics
service-decomposition-finished
supply-chain-negotiation
supply-chain-order-status
stakeholders-matched
dataspace-organization-onboarding-completed
service-composition-finished
post-optimization-finished

8. Notification Service

The Notification Service is a Java Spring Boot application designed to deliver real-time updates to users based on platform events. It consumes messages from the Kafka Message Bus, processes relevant events, and pushes notifications to the User Interface via WebSocket connections. In version 0.5, it supports basic user notification scenarios such as service request updates and (de)composition results, enhancing the responsiveness of the user interface.

The service is deployed as a Docker container and requires connectivity to the Kafka Message Bus. It uses WebSocket channels to communicate with the User Interface and is designed to handle multiple concurrent connections. Recommended runtime resources are 2 vCPUs, 4GB RAM and 50GB disk storage. It integrates with the platform's authentication layer to ensure secure notification delivery. Configuration is environment-driven using Spring profiles.

9. Search & Match

The Search & Match component is a Java Spring Boot service that enables users to create service requests by submitting structured forms and supporting documents. It stores request data in MongoDB and initiates downstream processing by triggering the Bill of Processes Generator for service decomposition. Results are received asynchronously via the Kafka Message Bus and stored for further use. Additionally, it can initiate optimization and analytics workflows based on the submitted request data.

The component is containerized using Docker for ease of deployment and integration. It requires a running MongoDB instance and access to the Kafka Message Bus for inter-service communication. Suggested runtime resources are 2 vCPUs, 2GB RAM and 50GB disk storage. REST APIs are exposed for user interaction and downstream service coordination. Configuration is managed via Spring profiles and environment variables.

10. Dataspace Connector

The Tekniker Dataspace Connector is a Java service that enables companies to manage the datasets they offer or request through a Data Space in an interoperable and sovereign manner. It provides an internal REST API to manage datasets, an external REST API to publish datasets, negotiate usage policies and transfer data following the Dataspace Protocol and it handles metadata related to datasets in a PostgreSQL Database.

This service itself is composed of a Nginx, the TDC backend, the TDC UI, the TDC Swagger UI, a PostgreSQL Database, a PgAdmin and a Dozzle that can be easily deployed using Docker containers. All configurations are managed using environmental variables. Suggested runtime resources are 4 vCPUs, 8GB RAM and 50GB disk storage.

11. Bill of Processes Generator

The Bill of Processes Generator component uses a Python REST API that provides two main functionalities: service decomposition and process time estimation. It uses 3D CAD files along with information of the material as inputs. Additional 2D sketches can be provided to provide additional information, although those inputs can also be included in the JSON payload used as input.

The component is containerized using Docker for ease of deployment and integration. It requires a link to a MinIO storage where the files will be stored, as well as a Kafka broker to publish the response for time-consuming processes. All configurations are managed using environment variables. Suggested runtime resources are 4 vCPUs, 8GB RAM and 50GB disk storage.

12. Optimization Service

The optimization service written in Python serves as a powerful backend component within a larger digital decision-making ecosystem. Leveraging Python's rich ecosystem of scientific libraries such as NumPy, SciPy, and Pyomo, the service can efficiently solve complex combinatorial optimization problems tailored to the project-specific use cases. Its modular structure allows for flexible integration with external platforms, making it well-suited to operate as a microservice that receives input data and returns results through a standardized interface.

The integration with optEngine's API enables this Python-based service to be triggered automatically by incoming requests. When optEngine initiates a call to the service, it passes input data in a structured JSON format, which typically includes decision variables, constraints, objective functions, and any relevant contextual parameters. The Python service parses this input, formulates the optimization problem accordingly, and solves it. This interaction ensures that the optimization logic remains decoupled from the Digital Twin or frontend system, promoting better maintainability and scalability.

After solving the optimization problem, the Python service packages the results - such as optimal values, objective outcomes, and diagnostic metadata - into a JSON output format that conforms to optEngine's API specification. This output is then returned via the API, allowing optEngine to relay the results back into the Digital Twin environment or downstream services. This architecture not only streamlines the optimization workflow but also supports real-time responsiveness and transparency, critical for applications in dynamic and data-rich environments.

All required parameters for the optimization service are appended in a dictionary named "data" of the following format:

```
{
    "date": "2024-12-01 18:00",                          #String of format "YYYY-MM-DD hh:mm"
    "objectives": [],                                    #A list of objectives
    "machines":     {                                    #The available machines
        "machine_id": {                                  #ID of the machine
            "company": "company_name",                   #Name of the industrial provider
            "services": [],                              #List of eligible services
            "processingTimes": [],                       #List of nominal processing times
            "timeWindows": [["2024-12-02 09:00, Infinity]],  #List of time intervals of availability
            "transportationTimes": {}                    #Transportation times from "machine_id" to
                                                         all other machines
        }
    },
    "requests":{                                         #Production demand
        "request_id":                                    #ID of the request
        "date": "2024-12-03",                            #String of format "YYYY-MM-DD hh:mm"
        "dueDate": "2024-12-07",                         #String of format "YYYY-MM-DD hh:mm"
        "services": [],                                  #List of services in order of processing
        "parts": 10,                                     #Copies of the same product
        "transportationTimes": {}                        #Transportation times from customer to all
                                                         machines.
    }
}
```

Subsequently, the optimization service generates alternative compositions – schedules for the same demand. The JSON output provides a dictionary named "data", including the alternative compositions in the following format:

```
{
    "composition_1": {                     #ID of composition – generated by optEngine
        "objectives":{                     #Values of the objectives
            "objective_1": 1628
```

```
                }
                "schedule":{                                          #Schedule of composition
                        "request_ID":{                                #ID of request
                                "part_1":{                                    #Number of copy
                                        "Service_1":{                         #Name of service
                                                "machine": "machine_id",              #Assigned machine
                                                "start": "2024-12-02 09:00",          #String of format "YYYY-MM-DD hh:mm"
                                                "end": "2024-12-02 09:07",            #String of format "YYYY-MM-DD hh:mm"
                                                "provider": "provider_name"           #Provider of machine
                                        }
                                }
                        }
                }
        }
}
```


13. Forecasting

The Forecasting component is a Python-based REST API service that provides advanced time series forecasting capabilities for predicting demand patterns within the T4M ecosystem. While demonstrated in VN1 for Electronic Board (EB) demand prediction, the service is designed to be domain-agnostic and can forecast any type of time series data provided in the appropriate format. Built using the Django REST framework, it implements multiple autoregressive models that leverage previous time steps (lags) of historical data to generate accurate predictions.

These include ARIMA, Simple Exponential Smoothing (SES), Holt's linear trend method, and Exponential Smoothing with trend and seasonality, allowing the system to capture complex temporal dependencies and cyclic behaviours in the data. This enables manufacturers and service providers to anticipate future requirements and optimize planning processes across different time horizons. The service accepts historical time series data in a standardized JSON format and returns forecasts for user-specified horizons (1-12 months ahead), supporting both immediate operational scheduling and medium-term capacity planning decisions across various manufacturing contexts.

The Forecasting service is fully containerized using Docker and designed for cloud deployment within the T4M platform. It utilizes a PostgreSQL database for storing historical data, autoregressive model parameters, and forecast results. The service exposes RESTful endpoints for data ingestion, model training with lag selection, forecast generation, and result retrieval. Hardware requirements are moderate, with recommended specifications of 4 vCPUs, 4GB RAM, and 20GB disk storage to handle the computational demands of autoregressive model fitting and concurrent forecasting requests. All configurations, including model hyperparameters and lag orders, are managed through environment variables, ensuring flexibility across different deployment scenarios.

The API provides endpoints for:

- `/forecast/train` - Train autoregressive models on historical data

- `/forecast/predict` - Generate forecasts for specified horizons
- `/forecast/models` - List available autoregressive models and their performance metrics
- `/forecast/results` - Retrieve stored forecast results

Example Input/Output:

The forecasting service accepts JSON input containing historical time series data and returns predicted values for future periods. Below is an example of the data format using Electronic Board (EB) demand data:

Input Format:


```
{
        "forecasting_horizon": 12,
        "data": [{
                "year": 2023,
                "month": 1,
                "historical_value": 116.73},
        {
                "year": 2023,
                "month": 2,
                 "historical_value": 117.003},
         {
                "year": 2023,
                "month": 3,
                "historical_value": 148.771 },
        {
                "year": 2023,
                "month": 4,
                "historical_value": 117.423}]
}
```


Output Format:


```
{
        "forecast_year": 2025,
        "forecast_horizon": 12,
        "forecasts": [{
                "year": 2025,
                "month": 1,
                "forecast": 116.73 },
        {
                "year": 2025,
                "month": 2,
                "forecast": 117.003},
        {
                "year": 2025,
                "month": 3,
                "forecast": 148.771},
        {
                "year": 2025,
```

"month": 4,
"forecast": 117.423}]
}

14. Predictive/Proactive Analytics

The analytics service operates as part of a cloud-hosted decision-making ecosystem, with multiple Python scripts coordinated through Power BI, which acts as the central orchestrator. These scripts carry out specialized data processing tasks, enabling the transformation of raw data into actionable insights tailored to specific use cases. The service is designed for flexibility and scalability, supporting various analytical workflows tailored to dynamic, data-driven environments. During the filtering and ranking phases, users can apply weights to different decision criteria, such as maximum number of providers in a value chain, perceived quality of service, and other custom preferences (e.g. provider preferences), to prioritize what's most important for their use case. The Python scripts compute relevance scores and generate ranked lists based on these inputs, ensuring that the results align with specific goals or constraints. In geospatial analysis, the service normalizes address components (e.g., street, postal code, municipality, country) into a standardized format and converts them to accurate latitude and longitude coordinates. Geolocation data is stored in JSON format, enriching the dataset for further mapping and refinement. All analytical outputs are stored in a MongoDB database hosted within the T4M environment. Power BI dashboards are configured to retrieve the data directly from MongoDB, enabling seamless, up-to-date visualization of key analytics, such as provider distributions, platform metrics, and consumer overviews.

Deployed within the cloud infrastructure, the service is structured around multiple specialized scripts, each dedicated to a specific analytical phase such as filtering, ranking, geospatial processing, and visualizations. These phases enable the system to transform raw input data into valuable insights based on configurable criteria and user-defined preferences.

The interactive dashboards are shared through public web links, giving stakeholders direct access to explore the results in real time, and there is no need for backend credentials. With simple navigation and clear visualizations, they help users quickly grasp key insights to support well-informed decisions. Built on a cloud-native architecture, the system ensures responsiveness, transparency, and adaptability, crucial qualities for data-driven tools in fast-paced, information-heavy settings.

15. Supply Chain Monitoring – Balance Scorecard – AI Model Governance Service

These three components are services provided by MIRA. Supply Chain Monitoring allows users to manage and monitor their collaborations, orders and shared data among the value network they have built. It is the interface where negotiations take place with the support of a messaging system, which allows the standardization of specific structured messages as templates. Balanced Scorecard is an evaluation tool to assess performance of the processes within an organization and correlate actions and their outcomes by setting up KPIs. The AI Model Governance Service is a tool to document, analyze and explain the algorithms, processes, training and usage of services that utilize artificial intelligence models. The interface of this tool is ideal to support documentation and user manuals for services, even if they have no AI involved.

All MIRA components are hosted independently so there is no particular set of system requirements for an organization to access its services. The setup of an organization account is the only prerequisite. The involved technologies include a Neo4j database for the support of business relationships among systems entities, a set of PostgreSQL instances for user management, system logs, and data warehousing, Angular as the main framework for the user interface, and Apache components like HOP and Superset for ETL processes and systemic data analytics.

In order to use MIRA or interact with its API, authentication is required. The swagger of MIRA API can be found in https://mira-solution.com/mira-service/swagger-ui/index.html. It includes numerous endpoints that handle I/O with all interactive services and system components.

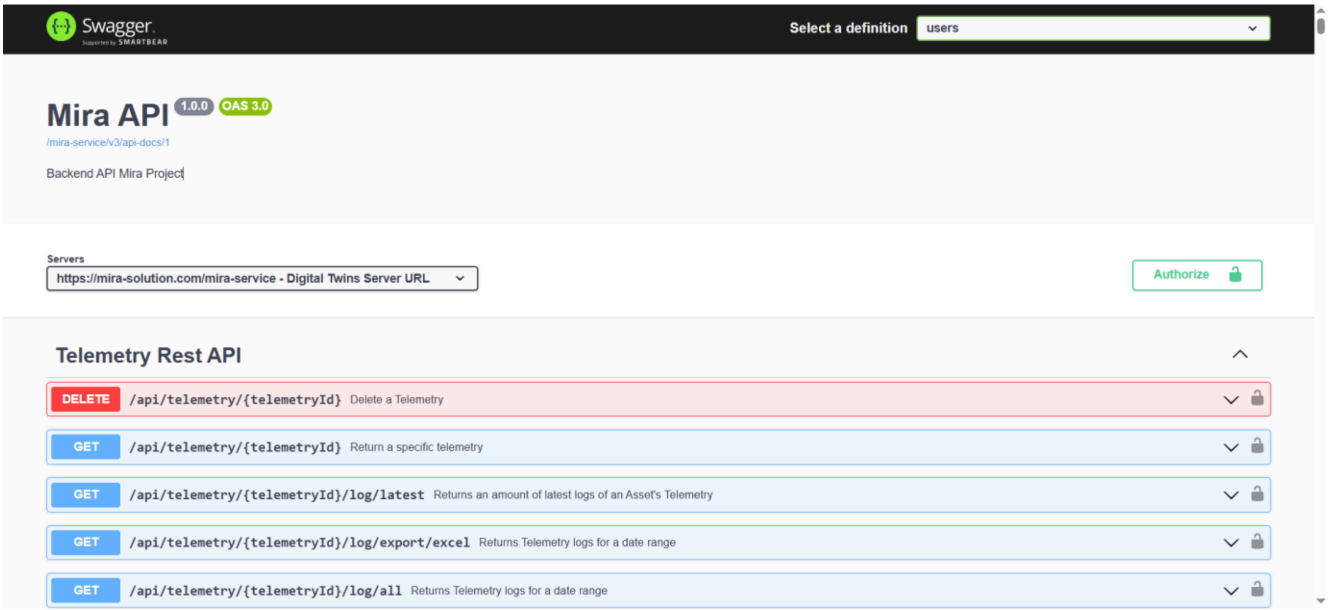

Figure 90: Swagger on MIRA API