Ref. Ares(2025)8258031 - 30/09/2025

ACCURATE

D2.2 - Matchmaking model and DT registry service

Actual Submission Date:	30/09/2025
Produced by:	IAO: Michael Hertwig, Joachim Lentes; ES: Giovanni Borzi, Simone Gregori

Accurate

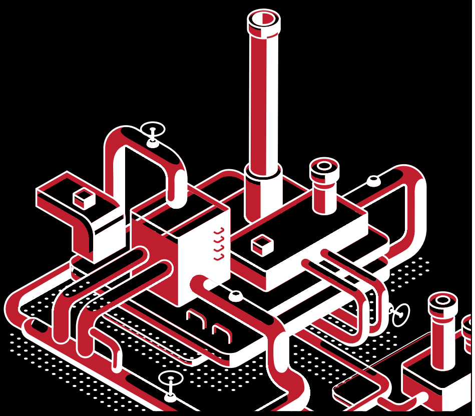
https://accurateproject.eu/

HORIZON-CL4-2023-TWINTRANSITION-01-07

Grant Agreement no.: 101138269
Start date of project: 01 November 2023 - Duration: 36 months

Funded by the European Union

The ACCURATE project is funded by the European Union, under Grant Agreement number 101138269. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



DELIVERABLE FACTSHEET

Deliverable D2.2	
Nature of the Deliverable:	Report
Due date of the Deliverable:	M22 – 30/09/2025
Actual Submission Date:	M22 – 30/09/2025
Produced by:	Fraunhofer IAO
Contributors:	Joachim Lentes, Michael Hertwig, Giovanni Borzi, Simone Gregori
Work Package Leader Responsible:	Fraunhofer IAO
Reviewed by:	Simona Bica, Mircea Vasile (SIMAVI), Valeria Borodin (IMT)

Dissemination level	
X	PU = Public
	PP = Restricted to other programme participants (including the EC)
	RE = Restricted to a group of the consortium (including the EC)
	CO = Confidential, only members of the consortium (including the EC)

Contents

Terms and abbreviations 7
Public summary 8
0 Introduction 9
  0.1 About this deliverable 9
  0.2 Document structure 9
  0.3 Relation with other tasks and deliverables 9
1 Matchmaking Service 11
  1.1 State of the Art and technical foundation 11
    1.1.1 Service-oriented manufacturing concepts towards a working definition of MaaS 11
    1.1.2 Ontology-based knowledge representation of the manufacturing domain 14
    1.1.3 Matchmaking of manufacturing services 14
  1.2 Concept of the Matchmaking 16
    1.2.1 Manufacturing Service Provider 17
    1.2.2 MaaS customer 20
    1.2.3 Matchmaking approach supporting MaaS 21
  1.3 Technical realization of the Matchmaking 25
    1.3.1 Matchmaking procedure 25
    1.3.2 Implementation as Software System 27
    1.3.3 Marketplace-oriented realisation 29
  1.4 Hybridization of the approach by adding data-driven functionality 37
    1.4.1 Conventional Approach with ChatGPT 37
    1.4.2 Open-Source Approach 40
  1.5 Scientific contributions from WP2 about the Matchmaking 43
2 Digital Twin Registry 45
  2.1 Introduction 45
  2.2 System architecture 47
  2.3 DTs integrations 49
    2.3.1 Rosenbrock example 51
    2.3.2 Available endpoints 51
    2.3.3 API Workflow 53
    2.3.4 Example API workflow 54
  2.4 Key differentiators 56
3 Use Case oriented Testing and validation 58
  3.1 Validation approach for Matchmaking functionality 59
  3.2 Validation approach for the Digital Twin Registry (DT registry) 61

4 Conclusion and further steps 62
5 References 63

Tables

Table 1: Overview on various service-based paradigms in manufacturing 13
Table 2. Translation of capabilities to characteristics – example based on a Fused deposition modeller (3D Printer)....................................19
Table 3: Task fulfilment by the chatbots 38
Table 4: Alternatives considered for the Knowledge Graph technology stack selection 41
Table 5: Technology stack selection for the ACCURATE Matchmaking Assistant 42
Table 6. List of publications with contributions from Work Package 2 43
Table 7. Input parameters of the Rosenbrock DT model 51
Table 8. Output parameter of the Rosenbrock DT model 51
Table 9. Structured representation of core endpoints 51
Table 10. Structured representation of session control endpoints 52
Table 11. Structured representation of logs and metrics endpoints 52
Table 12. Structured representation of ontology and keywords 53
Table 13. Structured representation of test endpoints 53
Table 14. Mitigation strategies to address different challenges on the way to 58

Figures

Figure 1. Document Structure 9
Figure 2. Motivation for a Matchmaking function in the Maas Ecosystem 17
Figure 3. Schematic perspective – Translating a physical asset to a MfgS and enabling Manufacturing on demand 18
Figure 4. Relationship between characteristics (descriptive elements of MfgS) and capabilities involved in the MfgS 19
Figure 5. Simplified translation of requirements (from output data) and conditions (from input data) to characteristics 21
Figure 6: Clustering of requirements raised by MaaS customers 21
Figure 7: Capability Matchmaking by Järvenpää et al. (Jarvenpaa et al. 2017) 22
Figure 8: Product-manufacturing service-matchmaking as utilized in ACCURATE 22
Figure 9. Three steps matchmaking with a human-centred final decision stage 23
Figure 10:Simplified excerpt of the ontology for matchmaking 24
Figure 11: Essential concepts for the matchmaking of products and services 25
Figure 12: UML activity diagram of the matchmaking procedure 26
Figure 13: Log in-page of the first version of the matchmaking system 28
Figure 14: Editor for service offering management 28
Figure 15: Editor for requirements 29
Figure 16: Overview about the calculated product-service-matches 29
Figure 17: Exemplary screenshot of Swagger with web service documentation of the marketplace-oriented solution. 31
Figure 18. sorting of scientific publication to the application fields 37
Figure 19: Conversation to list the manufacturing services contained in the data model (ontology) 39
Figure 20: Conversation to get an overview about the requirements’ fulfilments by the services in the model 39
Figure 21: Result of a conversation to determine the most sustainable service 40
Figure 22: Conceptual architecture of the ACCURATE Matchmaking Assistant via Open Source LLM 40
Figure 23: Screenshot of a prototype of the Open-Source-based ACCURATE-Chatbot-Assistant . 43
Figure 24: Digital Twin section on Register, showing details of Two-Machine model 45
Figure 25: Orchestrators section on Register app 46
Figure 26: Results section on Register app 46
Figure 27: System architecture of the Register application 48
Figure 28: Overview of the Register APIs presented in the Swagger UI 49
Figure 29: Section of the Swagger interface showing the endpoints generated for the Rosenbrock test case 50
Figure 30. Use cases as described in Deliverable D7.1 – base for validation with industrial data (a - project-internal use cases, b - project-external use cases) 58
Figure 31. Insight on requirement categories that are needed to text the functionality 60
Figure 32. Simplified IDEF0 diagram with specific technical capabilities of involved technical assets 60
Figure 33. Representation of requirements and characteristics to perform matchmaking (matching request & offer) 60

Terms and abbreviations

API	Application Programming Interface
CMfg	Cloud manufacturing
CRUD	Create, Read, Update, Delete (operation or procedure)
cURL	Client for URLs / ***Curl*** URL Request Library (program library)
CSV	Comma Separated Value
D	Deliverable
DSS	Decision Support System
DT	Digital Twin
GPT	Generative Pre-Trained Transformer
GraphRAG	knowledge Graph Retrieval-Augmented Generation
Icam	Integrated Computer Aided Manufacturing
IDEF0	Integration DEFinition for Function ModellingDefinition for Process Modeling
JSON	JavaScript Object Notation
LLM	Large Language Model
MaaS	Manufacturing as a Service
MAS	Multi-Agent System
MASON	Manufacturing’s Semantics Ontology
MfgS	Manufacturing Service (physical manufacturing performed service-orientated)
MGrid	Manufacturing Grid
MSDL	Manufacturing Service Description Language
NASA	National Aeronautics and Space Administration
OBMm	Ontology-Based Matchmaking
OWA	Open World Assumption
OWL	Ontology Web Language
OWL-DL	Ontology Web Language - Description Logic
PaaS	Production as a Service
QoS	Quality of Service
QUDT	Quantity, Unit, Dimension and Type
RAG	Retrieval Augmented Generation
RDF	Resource Description Framework
SOA	State of the Art
SPARQL	SPARQL Protocol And RDF Query Language.
SQWRL	Semantic (Query) Web Rule Language
SWRL	Semantic Web Rule Language
T	Task
UDDI	Universal Description, Discovery, and Integration ()
WP	Work package
WSDL	Web Service Description Languages

Public summary

This deliverable, **D2.2- Matchmaking model and DT registry**, presents two important results of the development work performed within ACCURATE. On one hand, it depicts the *semantic matchmaking service* and additional functions that represent the cornerstone enabling the match of needed Manufacturing Service and demands of the customer in the Manufacturing-as-a-Service (MaaS) paradigm. On the other hand, the digital twin registry represents the backbone framework managing and orchestrating different models and digital twins that ensure finding the optimal Manufacturing Service or even optimizing the interaction along the value chain.

Significant part of the ACCURATE project, the report outlines the methodology, requirements, and ontology modelling approaches that enable seamless integration of manufacturing processes and services.

The deliverable highlights the following key achievements:

1. **Requirements Collection and Specification**: Using structured methodologies such as Icam (Integrated Computer Aided Manufacturing) DEFinition for Function Modelling (IDEF0) diagrams and the Linked Open Terms (LOT) approach, the report consolidates functional and non-functional requirements derived from real-world use cases.
2. **Ontology Design and Development**: A modular ontology structure is created to represent concepts like production processes, supply chains, and key performance indicators. This includes alignment with existing standards and integration with external ontologies for enhanced interoperability.
3. **Semantic Matchmaking**: The ontology facilitates semantic matchmaking by aligning service providers with client requirements, ensuring flexibility, resilience, and sustainability in manufacturing operations.
4. **Alignment with Project Goals**: The deliverable establishes the foundation for future development phases, including ontology implementation and integration into the ACCURATE ecosystem.

The ontology and accompanying tools are designed to support a range of stakeholders, from manufacturing companies to service providers, ensuring adaptability across diverse industrial sectors. This work contributes to advancing MaaS by leveraging semantic technologies for supporting interoperability and enabling the integration and orchestration of manufacturing operations vertically and horizontally across supply chains.

0 Introduction

0.1 About this deliverable

D2.2 - Matchmaking model and DT registry service is a report focused on the specification of the ontology concept and the ontology modelling work. In addition, the interfaces to exchange data with other data sources and functions that are going to be developed in other WPs are discussed and shaped. This may reduce efforts in integration of the matchmaking function to a MaaS-supportive platform with multiple supportive services to evaluate 'Manufacturing as a Service'-elements.

The deliverable summarizes all work of WP2 that has been performed from M14 until the date of submission the report. The report depicts the methodology, the development work and the results generated, as presented in the following.

0.2 Document structure

The document is structured like a scientific report as illustrated in Figure 1. As the focus of the deliverable is on the outcome of the development work, the description of the developed functionality is in the centre of the document. The work package was sub-divided in five individual tasks corresponding to the description of WP2. The development work of Matchmaking service is described in Chapter 1. It mainly considers the works for the tasks T2.3 and T2.4. The work of tasks T2.1 and T2.2 was documented in Deliverable D2.1. The other outcome is to depict the development of Digital Twin registry (See Chapter 2), including an overview of the existing models and functions. The validation of the implemented functionality is documented in Chapter 3. Based on the development stage of both components, Chapter 4 provides a summary and an outlook on further developments. Chapter 5 documents all the sources used in drafting the report. The appendix (Chapter 6) contains additional content.

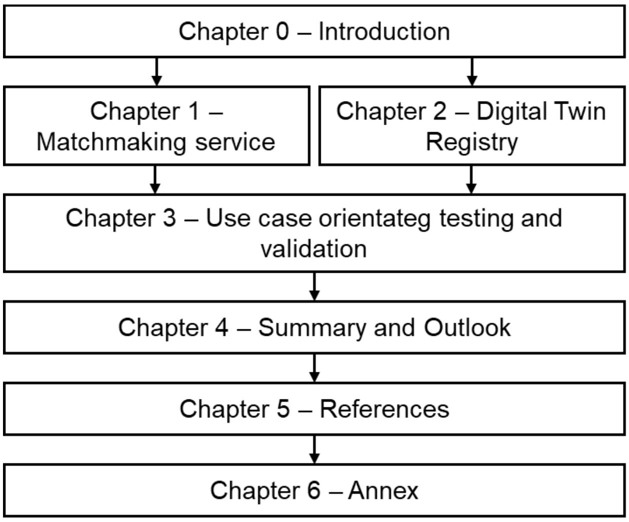

Chapter 0 – Introduction
Chapter 1 – Matchmaking service
Chapter 2 – Digital Twin Registry
Chapter 3 – Use case orientateg testing and validation
Chapter 4 – Summary and Outlook
Chapter 5 – References
Chapter 6 – Annex

Figure 1. Document Structure

0.3 Relation with other tasks and deliverables

Task T2.1 provided the basis for all work in the entire WP2, as the requirements are defined in T2.1. Task 2.2 was focused on the conceptual work, where the requirements have been translated into a technical concept. It laid an important foundation for the development of the matchmaking functions, as documented in deliverable D2.1. The answer to requirements from tasks T2.3 to T2.5 is the core of the deliverable D2.2. Its outcome – the developed functions – will be used in WP5 and WP6, where the integration and usage part will be addressed. The aim is making the function interoperable with the functions developed in WP3 and WP4 and for further developments within the ACCURATE MaaS Ecosystem.

As WP7 is focusing on the demonstration and use cases, it is important to correlate the abilities with the requirements of WP7 for operational purposes. The use cases defined in D7.1 are referred to for validation and testing. To ensure that the approaches developed in WP2 are supportive for the use cases as defined in WP7, the validation need to be founded on these settings. As the requirements as collected in Task T2.1 are always challenged against the use cases a deviation may be small. As a mitigation, during integration work in WP6 adaptions can be addressed and the function advanced. Furthermore, the semantic data model can be extended, if elements are missing.

1 Matchmaking Service

The matchmaking service is an important part of the ACCURATE Marketplace. This functionality allows users to find manufacturing services (MfgS) that match their requests and corresponding requirements. Only when the needs of the manufacturing industry can be precisely fulfilled by the available manufacturing services, the concept of Manufacturing as a Service (MaaS) will become economically viable. The acceptance of MaaS is a basic requirement to establish servitized manufacturing with relevant market share within the manufacturing landscape.

A matching process cannot take place without considering the general conditions and existing restrictions. For instance, the provider or requester may not have all the necessary data and information with sufficient quality and level of detail, which must then be generated using models and calculation functions. The ACCURATE platform offers additional functions for this purpose, which are developed in WP3 and WP4 or provided by project external parties.

The transformation of manufacturing towards a servitized offering of manufacturing capabilities is a result of the Industry 5.0 paradigm. It combines the idea of automated and connected industrial entities, as formulated in Industry 4.0 with an integration of a human-centred approach. The human becomes a valuable element within the value creation process. Considering this, the decision maker will get a stronger role which results in remaining the final authority on selection. Supported by a technological solution the human worker can actively contribute based on their expertise and experience but supported by tools considering large amount of data. Nevertheless, the approach offers support by narrowing down the selection range based on the available requirements through appropriate filtering of the offer.

The following subchapters provide detailed information based on the previously content from deliverable **D2.1**, with a focus on technical implementation and functionality.

1.1 State of the Art and technical foundation

In recent years, the field of matchmaking has evolved significantly, driven by advancements in technology and methodology. The following sub-chapters present the results derived from the current state of the art and what technical foundation might be used to develop a suitable function. To perform this analysis, a structured literature review was employed to systematically collect and analyse a diverse range of sources, ensuring a comprehensive understanding of the various techniques and frameworks utilized in the field and described previously in D2.1.

1.1.1 Service-oriented manufacturing concepts towards a working definition of MaaS

The term "Outsourcing" is used to describe the practice of industrial and services companies contracting out certain products or services to external companies that can perform these products or services in a superior or more efficient manner (Klein-Schneider und Beutler 2013). In the so-called Manufacturing Grid (MGrid), companies engage in cooperative activities through the coordinated (but not centralised) utilization, integration, and interoperability of a system of spatially distributed and heterogeneous manufacturing resources (including design, manufacturing, human, and application system resources utilizing grid, information, computer and advanced management, and advanced manufacturing technologies (Tedaldi und Miragliotta 2021; Tao und Qi 2019; Tao et al. 2011).

Cloud manufacturing (CMfg) enables the transformation of the manufacturing industry from production-oriented manufacturing to service-oriented manufacturing (Ren et al. 2017).

Henzel and Herzwurm (Henzel and Herzwurm 2018) conducted an extensive literature review on CMfg and identified the following seven characteristics:

I. Networked environment and collaboration among users;
II. Service and requirement orientation;
III. Interoperability among systems;
IV. Effective realization of intelligence by Knowledge and Data;
V. Virtualization principle;
VI. Scalability/Pay-as-you-go and
VII. Highly reliant on Trust and Security.

The concept is frequently erroneously regarded as a synonym for Manufacturing as a Service (MaaS), rather MaaS constitutes an integral part of CMfg (Bulut et al. 2021). The term 'Production as a Service' (PaaS) is also frequently employed as a synonym for MaaS, for instance, in Balta or Hermann (Balta et al.; Hermann et al. 2020). The emphasis is predominantly on small-batch production, which, in our opinion, is not incompatible with the MaaS approach. From a terminological perspective, MaaS can be considered a subset of PaaS, given that manufacturing constitutes a sub-area of production (Heizer et al. 2017; Groover 2020).

In the context of MaaS, CMfg was described by Liu et al. (Liu et al. 2019) as follows: “A model for enabling aggregation of distributed manufacturing resources (e.g., manufacturing software tools, manufacturing equipment, and manufacturing capabilities) and ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing services that can be rapidly provisioned and released with minimal management effort or service operator and provider interaction”. Tebaldi and Miragliotta characterized Cloud Manufacturing (Tedaldi und Miragliotta 2021). Considering this, some elements of this characterization will be important for realizing MaaS in industrial practice. The relevant characteristics enabling MaaS can be summarised as:

1. Centralised management of resources by the cloud operator (i.e., conversion of user requests into tasks, allocation and scheduling);
2. Intensive exchange of information between resource provider, user, and cloud operator;
3. On-demand availability of resources;
4. Service orientation and flexibility (high adaptability for the user in terms of product, delivery date, volume, mix, fast response time, flexible contractual relationships);
5. Resources are pooled and the user generally has no control or knowledge of the exact location of the resources provided;
6. Services are available everywhere and accessible via standard devices (e.g., smartphone, laptop);
7. Dynamic with uncertainty, rapid elasticity, and scalability.

Table 1 provides an overview of the various service-based paradigms with a manufacturing focus that have been derived from the concepts in the literature. As distinctive aspects, four main categories can be derived:

- Geographic demarcation.
  Geographic demarcation describes how much the location of demand and supply differ. This distinction can be logical, corresponding to an alternative production line or machine within the company or factory. It can refer to different locations within the same organization, or different locations and organizations are possible.
- Temporal demarcation.
  Temporal demarcation focuses on the time dimension of the collaboration. Is the collaboration demand-driven and short-term, or are the links established for the long term?

- Value chain coverage.
  Value chain coverage distinguishes between the level of detail. The smallest unit of a manufacturing service is the execution of a manufacturing process (e.g., producing geometry by milling). Bundles can be created by combining several different manufacturing services. These can in turn be combined with manufacturing parts and components. A further combination enables the supply chain to be mapped with different processes.
- Industrial perspective.
  Currently, companies are typically assigned to an industry or an industrial sector. The category of industrial perspectives describes whether the collaboration takes place within industry or whether services are offered to other industrial sectors.

To differentiate between the concepts, the fulfilment of certain characteristics has been analysed.

Table 1: Overview on various service-based paradigms in manufacturing

Service-based production paradigm		**MaaS**	**(PaaS)**	**CMfg**	**MGrid**	**Outsourcing**
Demarcation characteristics	**Values**	**Degree of fulfilment: fulfilled (ü), under certain conditions (w), and not fulfilled(û)**				
Geographical demarcation	Physical distribution (different locations)	ü	ü	ü	ü	ü
	Logical distribution and physical distribution	ü	ü	ü	ü	û
	Logical distribution (within the same facility)	ü	ü	ü	û	û
Temporal demarcation	Short-term (demand-oriented collaboration)	ü	ü	ü	û	û
	Long-term (contracting)	û	w	ü	ü	ü
Value chain coverage	Different value creation steps	û	û	û	ü	ü
	Components / systems	û	ü	ü	ü	ü
	Mfg service bundle	ü	ü	ü	ü	û
	Mfg service	ü	ü	û	û	û
Industrial Perspective	Cross industrial collaboration	ü	w	ü	û	û
	Industry internal collaboration	ü	ü	ü	ü	ü

Based on all the literature review, it was important to derive a working definition of Manufacturing as a Service. A working definition describes the current understanding of a term that is used as fundamental understanding. The definition is derived from the current review of literature, because the work of different researchers resulted multiple understandings which allows multiple interpretations. A joint understanding based on converged perspectives was required to result in a reliable baseline, aligning all development work, but especially for WP2. Therefore, the resulting definition is as follows (Schuseil et al. 2024):

> “MaaS represents a service-based manufacturing concept enabled by CMfg and thus managed via a centralized CMfg platform. A distinguishing characteristic of this concept is the capacity to disperse manufacturing services across both geographical and logical boundaries. Primarily demand-oriented, MaaS is characterized by short-term collaboration. MaaS provides both individual manufacturing services and combinations of such services (service bundles).”

1.1.2 Ontology-based knowledge representation of the manufacturing domain

Zhang et al. (Zhang et al. 2015) observe that widely used SOA standards such as Web Service Description Languages (WSDL) and Universal Description, Discovery, and Integration (UDDI) employ syntactic, text-based descriptions that are inadequate in terms of their semantic expressiveness, failing to adequately represent and describe web service concepts such as the capabilities of manufacturing services. The use of semantic web technology enables the resolution of semantic deficiencies and the facilitation of matchmaking through the annotation of web services with generalised ontologies.

Landolfi et al. describe the advantages of using ontologies to represent knowledge in the so-called Semantic Web (Landolfi et al. 2018). In contrast to conventional data models, an ontology is founded upon the Open World Assumption (OWA). An ontology allows the input of both known and unknown information, as well as incomplete knowledge. It enables the user and ontology applications to comprehend the meaning of the data, with applications able to infer additional information based on the ontology's disclosure and reasoning/inference rules. Furthermore, an ontology facilitates the sharing, integration and inference of knowledge and data. Most identified initiatives utilise OWL (particularly OWL-DL) as the formalisation language. Other initiatives are directed towards the formalisation of a core/upper ontology, which could provide the fundamental building blocks necessary for the description of a diverse range of manufacturing services, including MSDL (Manufacturing Service Description Language) or MASON (Manufacturing’s Semantics Ontology) (Landolfi et al. 2018).

1.1.3 Matchmaking of manufacturing services

In their work, Jang et al. define manufacturing services as any activity in which physical equipment is employed for the processing of raw or in-process materials (Jang et al. 2008). The discovery of services is formalised as a process of identifying matches between an order and the services advertised (Zhang et al. 2007).

Service discovery and matching in MGrids based on keywords as described by Yu’an et al. (Yu'an et al. 2009) are not ideal, which is why the idea of developing a semantics-based matching approach emerged in the mid-2000s. Ameri and Dutta propose a matchmaking algorithm designed to connect buyers and sellers of manufacturing services based on their semantic similarities with regard to manufacturing capabilities. They describe a manufacturing service as a manufacturing capability provided by a manufacturing process and by one or more manufacturing resources. They use Manufacturing Service Description Language (MSDL) to formally represent manufacturing services in an ontology, as MSDL compensates for the lack of a complete vocabulary for describing manufacturing services in other semantic description languages. A tree matching approach is used to represent MSDL descriptions as directed, labelled trees. In a realistic scenario, an enquiry may comprise several services that must be matched with a corresponding number of advertisements available in the supply pool. In this scenario, there may be several potential configurations of the supply chain for a given enquiry. The optimal configuration is the one that exhibits the highest similarity value. Accordingly, an optimisation model has been developed which assigns the requested services to the offered services in such a way that the overall similarity of the supply chain is maximised and the size of the supply chain, that is to say the total number of suppliers involved in the supply chain, is minimised (Ameri und Dutta 2008).

Yu'an et al introduce a matching method based on OWL-S, including a three-layer model consisting of requirements for service functions or methods (Method), constraints of object properties of the task (Objects), and requirements of Quality of Service (QoS) (Yu'an et al. 2009).

ManuHub, described by Cai et al. (Cai et al. 2011) is a tool that facilitates the modelling of distributed manufacturing services based on a formal ontological specification. This specification formalises the structural knowledge of manufacturing services using Ontology Web Language (OWL), while the constraint knowledge between structural knowledge is formalised using Semantic Web Rule Language (SWRL). A novel approach to semantic matchmaking for the retrieval of required manufacturing services is proposed, based on the formal specification of service capabilities. The work presented in their paper builds upon the feature-based semantic matchmaking approach originally proposed by Ameri and Dutta. It introduces a simplified and efficient algorithm that assumes that either the total number of super classes subsuming the compared concepts or the total number of subclasses subsumed by the compared concepts in a shared ontological taxonomy is known. Thus, it overcomes the drawback of the assumption of full and complete information. Three distinct matchmaking strategies are employed to categorise the varying meanings of concepts, namely: an exact match, a similar match and the best first match. Other authors use fuzzy approaches to describe semantic similarity matching (Sun et al. 2008; Yu'an et al. 2009; Liu et al. 2012; Ameri und Patil 2012). In Ameri and Patil (Ameri und Patil 2012) the semantic similarity is measured by taking into account Service, Supplier and Part. They also build their work on MSDL.

In their subsequent work, Zhang et al. employ an OWL-based manufacturing service ontology and propose a personalised manufacturing service recommendation method. Building upon the preliminary work (Cai et al. 2011), they introduce a collaborative filtering method to automate the semantic annotation of manufacturing service capabilities and implement a personalised recommendation (Zhang et al. 2015). Lartigau et al. (Lartigau et al. 2015) present a CMfg approach based on QoS evaluation (considering price, manufacturing time, reliability, maintainability, availability, and eco-impact) and extending it with geo-perspective correlation from one cloud service to another for transportation impact analysis. Järvenpää et al. (Jarvenpaa et al. 2017, 2018; Jarvenpaa et al. 2023) elaborate on an OWL-based capability matchmaking, which supports rapid configuration and reconfiguration of production systems. As the work of Järvenpää is a fundamental prerequisite for a matchmaking to find suitable service providers for potential service consumers and vice versa at a meta level, our work will take this preliminary work (OWL-based capability matchmaking) as a basis.

A multi-level matchmaking to facilitate the semantic matchmaking of manufacturing services is put forth by Zhao et al. (Zhao et al. 2017), comprising feature matching based on semantic similarity, numerical matching based on rules, and feature matching based on task decomposition. Siltala et al. (2019) introduce a method to check the connectability of manufacturing resources in addition to the capability-based matchmaking by Järvenpää et al. (2017, 2018, 2023). For this they focus on the interface connectivity of the resources and develop an interface ontology, which is queried with SPARQL for the interface matchmaking. Vennesland et al. (Vennesland et al. 2019) propose a semantic matching algorithm for capacity exchange, which, based on the formally described supply chain resources, returns a ranked list of relevant suppliers given a customer query. Interestingly, they identify the following relevant parameters to find suppliers whose offered resources match a consumer query: process, material, machine, certifications, capacity, and calendar availability.

Delaram et al. (Delaram et al. 2021) model the resource allocation problem as a matching game within a public cloud manufacturing environment, thereby considering the preferences of those searching, whereas Sparr et al. (Sparr et al. 2021) leverage a Multi-Agent System (MAS) for an auction resulting in MaaS service provider selection for CMfg. Matchmaking is done via SPARQL, which also serves to create the needed operation templates.

Summarizing the state of the art, it must be said, that various developers did already work on different aspects. However, we did identify multiple gaps. Within ACCURATE project, not all gaps will be closed holistically. To give a better understanding about the advancements, the gaps and the contribution by the project will be highlighted as follows:

- The matchmaking approaches are contributing on theoretical frameworks and standards – The standards are incorporated but only elements of relevance for the matchmaking function will be detailed. The standards itself are correct but miss some practical conditions required, like units and dimensions. In the case that all information has the same dimension and units are synchronized, this may obsolete. But in practical situation, different dimension and units are in use that may result in errors.
- Semantic systems in industry are typically expert systems and difficult to use by non-experts with relevant domain knowledge – The approach of the development is to cover the expert system to make interaction for domain experts possible. Therefore, a web interface with web technology will provide an interchange between GUI and expert system, reducing the entry level for system use.
- Existing models are too detailed or limited in speciality of application field – Enabling Manufacturing as a Service is currently limited by the machine-readable information. As most enterprise has its manufacturing processes not described in machine-readable formats the matching is difficult as either the degree of detail is too high or too low to evaluate the offering against the requirements. Therefore, the project follows a modular approach. Models of higher degree of detail can be integrated but the focus is laid on finding a suitable offeror. The functions focus on capability level, which is an intermediate level above a detailed process description with semantics.
- Limited human-centric approach – By using a hybrid approach, it is possible to perform the matchmaking based on a semantic-data structuring. The results generated by the matchmaking function could be manipulated using a chatbot like function to re-structure the data based on perspectives of the user.

1.2 Concept of the Matchmaking

To enable MaaS, two parties are required. It is necessary to have a service provider and a service customer that represents at least two contract parties. By the time passing, the number of providers and customers will increase (Figure 2). The offerings will vary based on technical, timely and other terms and conditions, that will challenge the choosing process as complexity increases. Based on the increasing complexity and the large number of offerings, it will be challenging to find suitable partners to establish an on-demand collaboration in manufacturing. If the finding of suitable partners is challenging, it may result in a low acceptance rate of MaaS. Therefore, the matchmaking is needed. It shall enable to identify suitable offers, narrow down the possibilities and support the human choosing the right partner for his task.

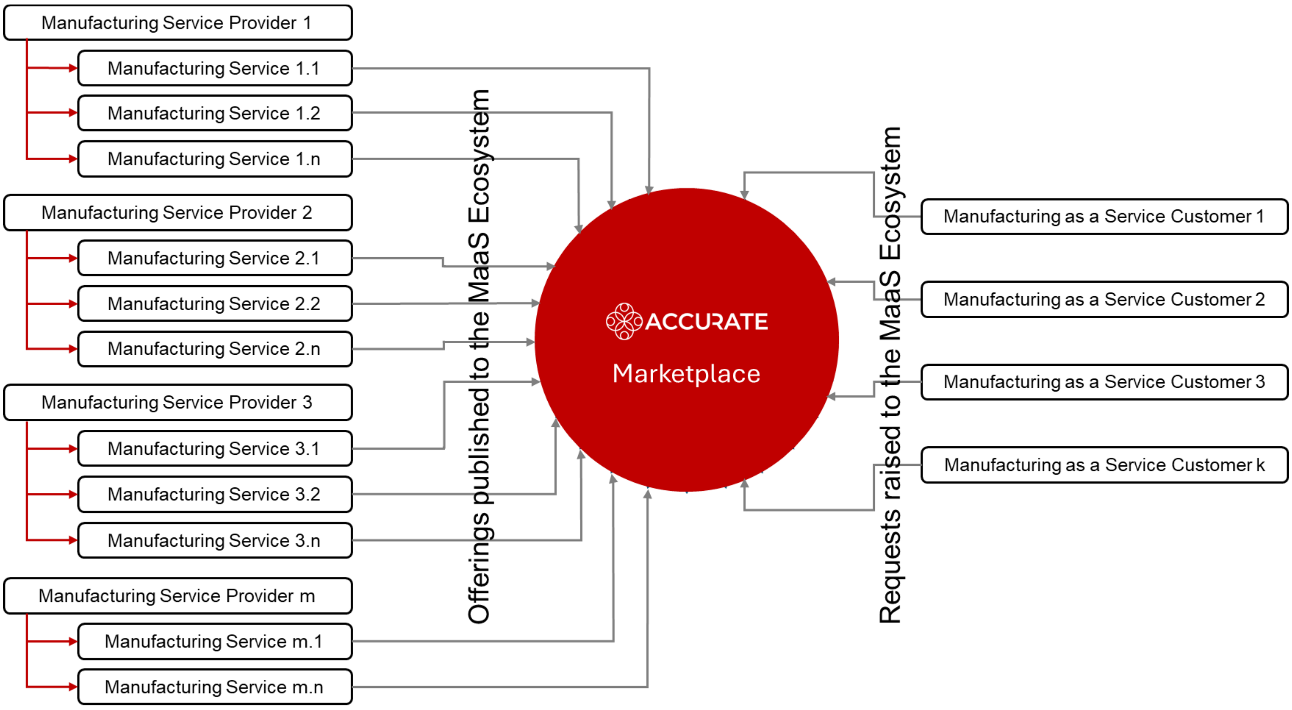

Manufacturing Service Provider 1
Manufacturing Service 1.1
Manufacturing Service 1.2
Manufacturing Service 1.n
Manufacturing Service Provider 2
Manufacturing Service 2.1
Manufacturing Service 2.2
Manufacturing Service 2.n
Manufacturing Service Provider 3
Manufacturing Service 3.1
Manufacturing Service 3.2
Manufacturing Service 3.n
Manufacturing Service Provider m
Manufacturing Service m.1
Manufacturing Service m.n
Offerings published to the MaaS Ecosystem
ACCURATE
Marketplace
Requests raised to the MaaS Ecosystem
Manufacturing as a Service Customer 1
Manufacturing as a Service Customer 2
Manufacturing as a Service Customer 3
Manufacturing as a Service Customer k

Figure 2. Motivation for a Matchmaking function in the Maas Ecosystem

1.2.1 Manufacturing Service Provider

A Manufacturing Service Provider (MfgS provider) is an entity that offers manufacturing services on demand. The service that is offered covers a single manufacturing step, a sum of manufacturing steps that can be embedded in a production or can even replace the supplier function. Becoming a MfgS provider may has different motivations. It is possible that an overcapacity is reducing the profit on turnover, a machinery cannot be utilised for manufacturing anymore or the service offering become an additional business operation. Independently, of the dedicated business operational concept, MaaS aims for increasing the flexibility, optimize resource utilization, reduce costs, or even extend the customer base.

However, becoming a MfgS provider seems to be interesting for all stakeholders in the value creation chain, multiple framework conditions may limit the applicability of the concept of MaaS to all manufacturing companies (Hertwig et al. 2025). To leverage the full potential, the companies need to know their processes and capabilities very well and need to be able to facilitate them quickly on demand. Based on the wide range of degrees in digitalization among the industries, the horizontal integration is varying. Furthermore, the way of describing processes and procedures is not standardized across industries which may allow an interchange manufacturing offering in broader range. The differences in supply chain management structures even make this more complicated because rules, regulations and standards are different or at least not harmonized. All these aspects must be considered accordingly when building a MaaS Ecosystem.

Focusing on the MfgS itself, the offering can be provided because relevant capabilities are existing. These capabilities are a result of established infrastructure, machines and software but also the availability of employees and their competencies. The physical assets and resources are the foundation as they allow to access technical, organisational and knowledge-based capabilities. By a suitable combination of these capabilities a specific manufacturing capability can be derived (Figure 3). Depending on the concrete MfgS offering, different manufacturing capabilities are combined to orchestrate the manufacturing offering. A reason for this comes with the structure of the MfgS which consists of the core service (value added to the customer) and supplemental services. These are typically needed to meet the intended quality of the service output and cover measurements, intralogistics or similar.

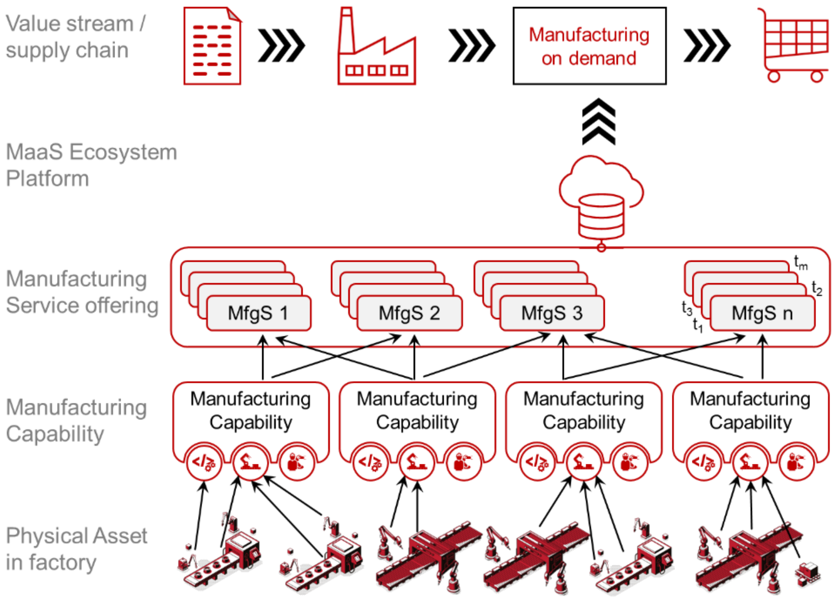


Figure 3. Schematic perspective – Translating a physical asset to a MfgS and enabling Manufacturing on demand

To enable the matchmaking of offer and request in the MaaS Ecosystem, the description of the MfgS need to be structured that it can be understood by the software-based function. As of today, manufacturing industry is highly specialized. Companies are focusing on specific target to address and fulfil required conditions with high performance parameters. Based on this, the explanation and description of the service-oriented manufacturing offering will be industry-specific. However, the description must be as widely understandable to be applicable across the industries. Only then the potentials could be leverages by starting to exchange manufacturing capabilities in a wider range. To describe the MfgS, characteristics are use that are derived from the different capabilities involved in the MfgS (Figure 4). The characteristics are used to check the compliance to the customers' needs. The customer demands have different dimensions. Following the logic of technology management dimensions, they could be sub-divided in three categories – Technology-related, Organization-related or Human-related.

Technology

Important characteristics are definitely technical. They have as strong influence on the outcome and customer satisfaction in the first place. They are targeting aspects like quality, performance, and process conditions. Typically, this need to be fulfilled or sufficiently addressed by the service.

Organization

In a complex world, besides the technical perspective the customer raises additional aspects. Depending of the reason of searching a MfgS, the time to deliver is important, including also the processing time, scheduling of transport, etc. Overarching these aspects could covered by terms like operations, compliance and process. Based on the industry and framework conditions, organisational characteristics could be next to timely aspects certifications, fulfilling of certain regulations or providing relevant documentation.

Human

Besides the technical and organizational aspects in a manufacturing company, it is required to have suitable knowledge. The experience and expertise in the staff are essential to perform high-quality MfgS. These specifics are covered in knowledge based and human-centred capabilities represented by characteristics accordingly.

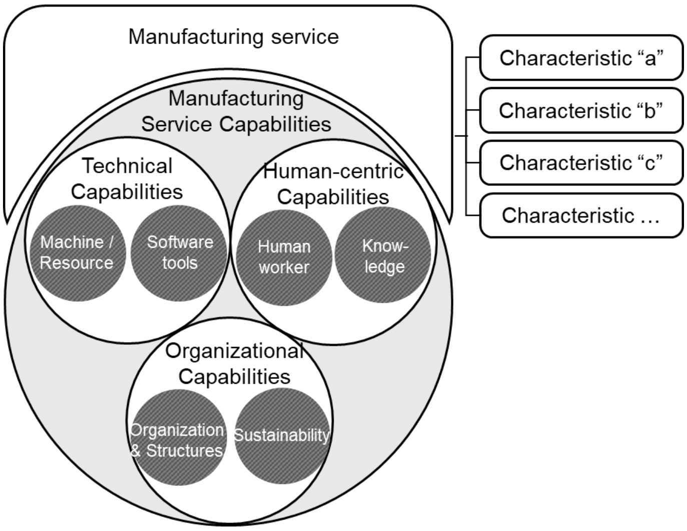


Figure 4. Relationship between characteristics (descriptive elements of MfgS) and capabilities involved in the MfgS

To describe the Manufacturing Service the technical, organizational and knowledge-based capabilities are translated to characteristics. Characteristics could be presented like limitation values [L] (e.g., maximal or minimal), aspect of choice [C] (e.g., material A, material B or material C) or binary conditions [B] (e.g., have / fulfil or no have / not fulfil). Based on an additive fused-deposition modelling machine located at Fraunhofer IAO, an example set of characteristics has been generated (Table 2).

Table 2. Translation of capabilities to characteristics – example based on a Fused deposition modeller (3D Printer)

Type of	Capability		Characteristics	Type of
Technical	Production volume - length	→	Part volume (x-axis)	L
	Production volume – width	→	Part volume (y-axis)	L
	Production volume - hight	→	Part volume (z-axis)	L
	Noise	→	Emission level	B
	Accuracy	→	Precision in x- and y-axis	L
	Layer thickness accuracy	→	Precision in z-axis	L
	Material bays	→	Number of materials	L
	Size of material bays	→	Volume of material feed	L
	Operation temperature	→	Operation condition	L
	Power requirements	→	Sustainability-relevant information	C
	Certified materials	→	Defined output quality	C
	Layer thickness	→	Precision in z-axis	L
	Type of solution washer	→	Emissions, output quality	L
Knowledge-based	GrabCad software	→	Limitation in format processing	B
	Mechanical engineers	→	Experience and knowledge	B
	Material choices	→	Strength, durability	C
Organizational	Time between confirmation and production start	→	At least 16 h	C
	Certification	→	Greengard for some materials, TISAX-2	B
	Quality management system	→	No quality management procedures	B
	Cyber security procedures	→	Only transfer via ownCloud	B

1.2.2 MaaS customer

The MaaS customer is the counterpart of the supply perspective. Due to an incident, a MfgS is needed to fulfil the contract with the end customer (providing the product in requested quality, compliant to the committed price and in time). In this case, a MfgS will be part of the value creation process, because the reason of customers searching for MfgS are various.

- Disruption of internal production processes
  A disruption of the internal production processes can be caused by an unforeseen breakdown of a machine or production line. The manufacturing capacity is significantly reduced or even eliminated. A quick replacement of the manufacturing capacity is needed to fulfil the delivery agreements (Katsaliaki et al. 2022). The missing capacity could be filled by a matching MfgS offering.
- Disruption in the supply chain
  The supply chain can be quite fragile, as shown by the interruptions during the COVID-19 pandemic (Shishodia et al. 2023). An analysis of potential risks in the supply chain allows developing specific strategies to mitigate these risks. However, some of the interruptions cannot be addressed appropriately. Reasons for that are various. So, the establishment of a second source could be too costly or too difficult to be economic viable. Besides that, the reasons for disruptions could be divided in two main clusters: (1) a breakdown of the supplier (the supplier will not deliver) or (2) disturbances of transportation from supplier to the factory (blockage of route, large disruptions of transportation chain). By falling back on a MfgS, a disturbance could be addressed shortly by replacing the supplier and/or the disturbed transportation route until normal conditions are restored again (Herold and Marzantowicz 2023).
- Short-term increases in demand of capacity
  Typically, an enterprise measures its production capacity to serve the prognosed demand. The demand is volatile so it might be possible to changing fit of the capacity to fulfil the customers' demands. Not fulfilling customer demands leads to losses as there is an overcapacity or too small capacity. In case of larger customer demands compared to the established capacity, the manufacturer can increase its capacity by booking suitable MfgS.

The MaaS customer has demands regarding the service offering. The demands are represented by a complex set of conditions and requirements. Talking about a MfgS service, it can be described as a transformative action which may comprise a single or multiple steps. A simplified example could be the transformation of a rectangular block into a tubular component by means of machining (Figure 5).

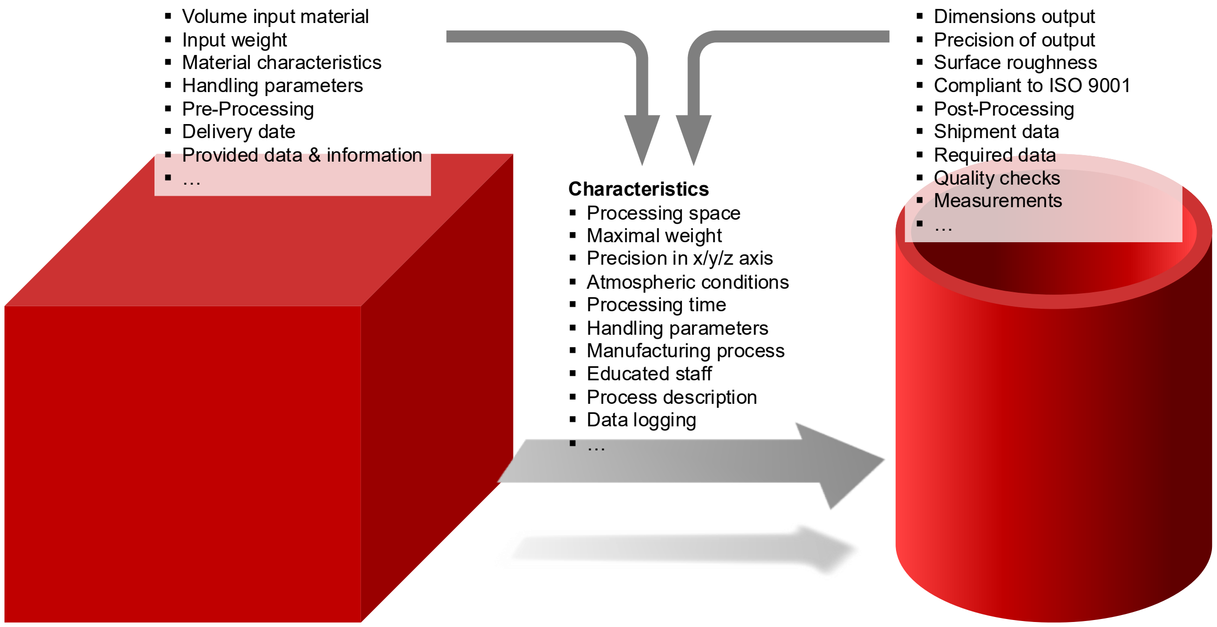


Figure 5. Simplified translation of requirements (from output data) and conditions (from input data) to characteristics

The customer requirements can be clustered in six dimensions. Whereas three each could be assigned to two different perspectives that are interconnected (Figure 6). The first perspective can be described by the magic triangle of business finds an optimum between quality, time, and cost. *Quality* covers aspects about the product like form, volume, material, and tolerances. *Cost* is considering financial efforts needed. As a replacement by a manufacturing service has time-based constraints, this addressed in the aspect *time*. Secondly, a successful offering requires the aspects technology, people and structure which represents the human-factors dimensions. The product-specific requirements may influence the required *technology* utilized. *People* are driving the processes and influence sustainability aspects of the organization. Industries must follow certain regulations, certification and fulfil external framework conditions. The requirements derived from this are combined in the aspect *structure*.

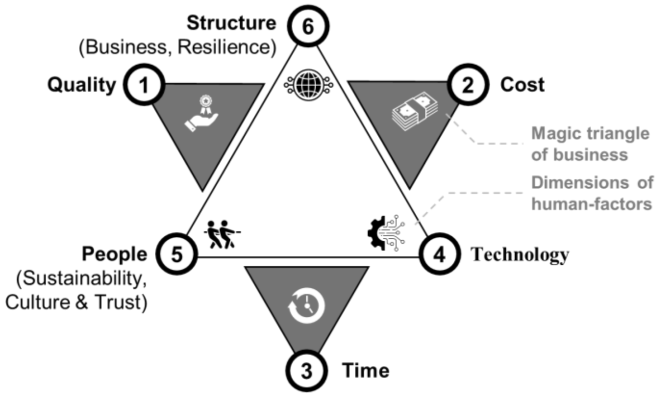

Structure
(Business, Resilience)
6
Quality
1
2
Cost
Magic triangle of business
Dimensions of human-factors
People
5
(Sustainability, Culture & Trust)
4
Technology
3
Time

Figure 6: Clustering of requirements raised by MaaS customers

1.2.3 Matchmaking approach supporting MaaS

Matchmaking could be performed automatically or as support for the human decision maker. In this case, the human decision maker shall be enabled to find a suitable fitting manufacturing service quick by a systematic support. The matchmaking will only support the person in their tasks. The beneficial aspect of this approach is the optimal use of the advanced of both parties – machine and human (Wilson and Daugherty

2018). A machine can process large data sets and provide an understandable representation. The human decision maker can evaluate different constraints according to the current situation and strategic perspective (Ozkizitan and Hassel 2020).

The decisions must be compliant with the strategic goals of the organization. To be transparent and comprehensible in the way of shaping the choice, the matching criteria must be adapted accordingly. Following the matching approach as presented previously, the decision maker has different interaction possibilities. In a first step, the requirements formulation has a strong influence on the identified relevant manufacturing services. Furthermore, the decision maker can adapt the weighting of different influence aspects by the secondary ontological matching which results in coverage probabilities. Based on this, a rating is done about the quality of fit. The weighting allows to set a higher priority on certain aspects. Finally, the decision maker can analyse the manufacturing service details and evaluate whether it matches the compliance requirements of the organization and product.

As the matching is currently focused on optimal fit of the service offering, an advancement could be to provide a profile of the service offeror, this would offer an additional lever for controllability of the matching results and decision criteria by the human decision makers.

Semantic product requirements and process attributes are crucial for the accurate and efficient matchmaking of manufacturing capabilities. The matchmaking process involves comparing the product requirements with the capabilities of available manufacturing services, ensuring compatibility and optimal resource utilization.

Figure 7 illustrates the capability matchmaking of Järvenpää et al. (Jarvenpaa et al. 2017). This approach was used as a base to build on. Considering the discussions and previous development work the approach was advanced to the matchmaking “chain” that is used in ACCURATE to match to-be-manufactured products by means of their related requirements and manufacturing services (Figure 8). To realize a logical and-condition, the requirements of a product are grouped in requirement sets. The capability matchmaking leverages a capability model and catalogue for the matchmaking, ensuring that the right manufacturing processes are selected for the given product requirements.

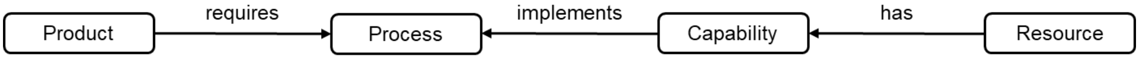


Figure 7: Capability Matchmaking by Järvenpää et al. (Jarvenpaa et al. 2017)

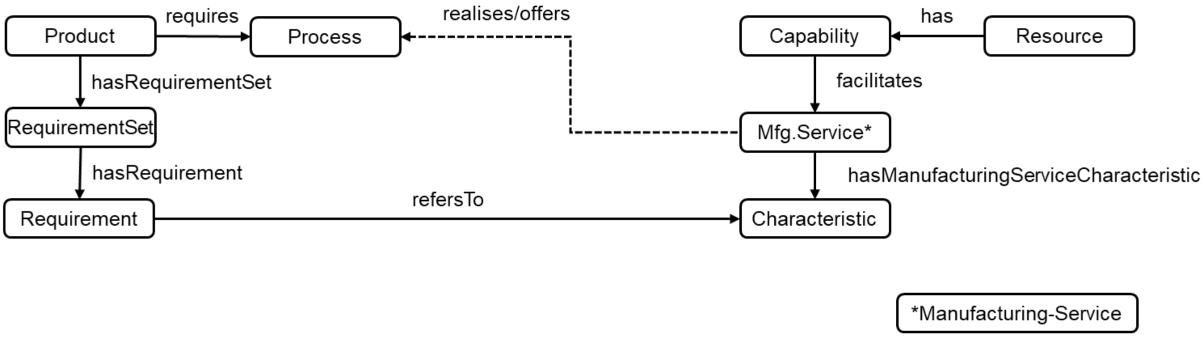


Figure 8: Product-manufacturing service-matchmaking as utilized in ACCURATE

Essential requirements on the Ontology-Based Matchmaking (OBMm) are to have the possibility to define must-requirements on the selected service and to weight these requirements at least in clusters, therefore enabling the users to put emphasis on specific requirements or requirement clusters as sustainability or resilience based on the specific situation. Consequently, the ontology concept to represent requirements has attributes to describe its must-characteristic and its importance for the decision situation by a weighting, which is a crisp double-value. In future work, these crisp values for the weighting may be advanced towards a linguistic variable in the sense of fuzzy-set theory or related. In addition, requirements have a Boolean attribute isNegated to realise logical not-operations.

The OBMm is divided into two steps as illustrated in Figure 9. At first the available services are filtered by the must requirements based on their characteristics, and then in a second step, the remaining services are ranked in accordance with their fulfilment of the requirements to support the decision maker. Humans as decision makers are therefore not patronized by the approach for OBMm but supported with a ranking of applicable available manufacturing services of which they can choose based on their preferences or the specific decision-situation, respectively.

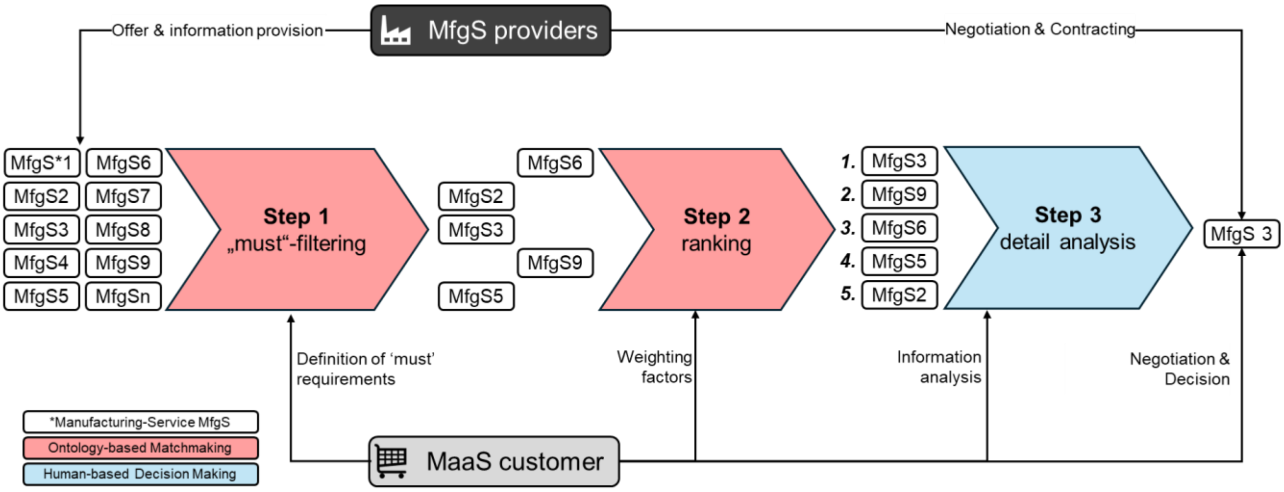


Figure 9. Three steps matchmaking with a human-centred final decision stage

For the OBMm, requirements refer to characteristics of manufacturing services. Both, requirements and characteristics may be quantifiable or not, but have reference values for the comparison. If they are quantifiable, they refer to a quantity, which at least has a value and a measurement unit and may be derived from ontology for Quantities, Units, Dimension and Types (QUDT) of the National Aeronautics and Space Administration (NASA) (QUDT 2019). If they are not quantifiable, they refer to an arbitrary instance of a class of the ontology. For quantified comparisons, requirements can request for equality but may be based on a minimum or maximum condition with included or excluded boundary values (so they can be closed or open as intervals). Thus, to define quantified requirements which are based on an interval, two requirements have to be defined by the users, one for the minimum and one for the maximum condition. Non-quantifiable requirements are fulfilled, if the characteristic "has" an arbitrary instance of the ontology class which is defined by the reference value of the requirement. A simplified excerpt of the ontology for matchmaking is shown in Figure 10.

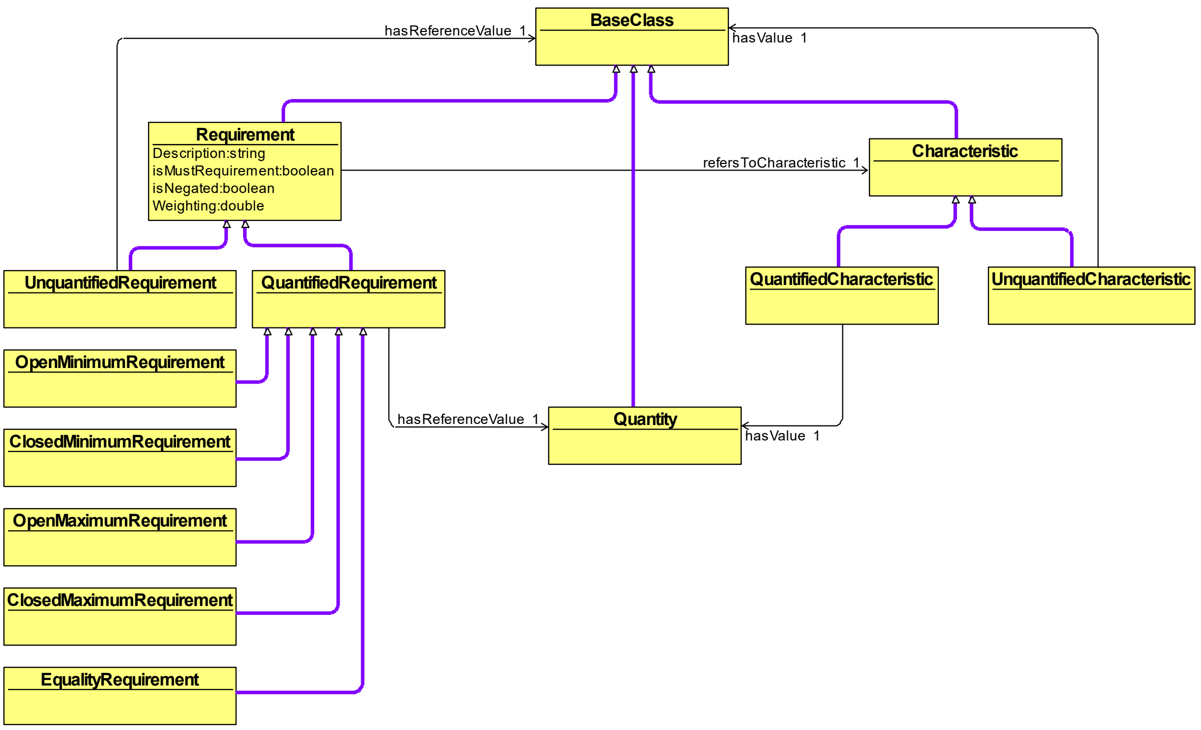


Figure 10:Simplified excerpt of the ontology for matchmaking

To support the matchmaking between products and manufacturing services, a related concept ProductServiceMatching with two specific attributes is declared. The attributes correspond to the results of the two steps of the matchmaking procedure.

In the first step of the matchmaking procedure, each service is analysed if its characteristics fulfil the set of requirements. If all related must-requirements are fulfilled, the attribute isPossible is set to true, if not all requirements are fulfilled, it's set to false.

For the second step of the matchmaking, each ProductServiceMatching which is possible for the defined requirements set is analysed for the fulfilment of the requirements of the set. For this, for requirements, fulfilment degrees are calculated with respect to the characteristics of the service. For this, for each requirement fulfilment, its fulfilment proximity is multiplied with the weighting of the respective requirement. For unquantified requirements and equality-related quantified requirements, the fulfilment proximity is set to 1 if the requirement is fulfilled, and to 0 if it is not fulfilled. For quantified requirements, which are not equality-related, the fulfilment proximity corresponds to the absolute value of the difference between the values of the quantities of the requirement and of the characteristic under consideration of the unit of measurement. To take account of deviating orders of magnitude, two measures are taken. Firstly, the weightings of the requirements are defined on identical scales and normalized subsequently. Secondly, the normalized weightings are scaled to compensate the order of magnitude of the quantity of the characteristic. Possible product-service-matchings are then provided in a ranking based on their matching degree as the sum of the fulfilment degrees of the requirements fulfilments of the product-service-matching to the user, who may then select a service based on his preferences.

The third step, as presented in Figure 9, is the human-centred decision-making process. The human may have a deep dive into information about the MfgS. This could be done by checking the documents provided by the offeror. Alternatively, the decision maker may utilize functions developed in the WP3 and WP4 to gather a better understanding about the performance and fit his requirements.

Essential concepts for the matching of product and services are summarized in Figure 11 together with their essential relations.

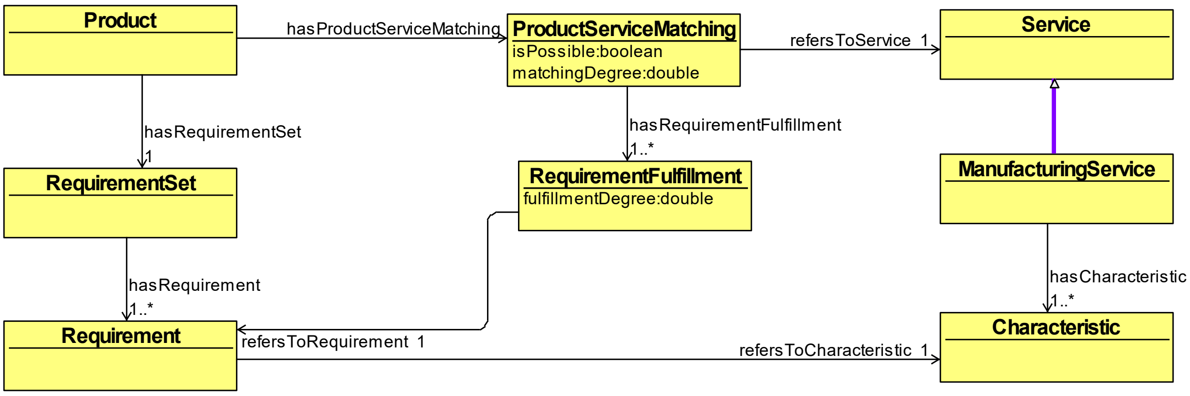


Figure 11: Essential concepts for the matchmaking of products and services

1.3 Technical realization of the Matchmaking

As introduced in the report D2.1 of the project ACCURATE, the technical realization of the matchmaking is done in steps, starting with functional services with a local ontology.

For the first realization step, the following basic decisions were made:

- To store and process the ontology locally, the well-known programming library Apache Jena is used, which besides handling and processing ontologies e.g., in OWL supports the work with SPARQL Protocol And RDF Query Language (SPARQL) on the ontology directly.
- To simplify the implementation of the semantic software services, the programming library OntTools by Fraunhofer IAO is used. This library was developed over years in projects like the Fraunhofer Innovation Cluster “Digital Production”, and the EU-funded projects amePLM (FP7 285171) and QU4LITY (H2020 825030).
- As the usage of Apache Jena implies the usage of the programming language Java for the services, the framework Vaadin is used to realise the graphical user interfaces (GUI) of this step. This GUI enables the prototyping, showcasing and discussion of the services to and with the intended users within the consortium and beyond.

The OntTools library provides support classes like an OntoAdapter, which decouples the ontology processing library (Apache Jena) and tools for work with quantities in the Quantities, Units, Dimensions, and Types (QUDT)-ontology from the system-to-be-developed, thereby supporting a technology-agnostic implementation, as well as helper classes like OntoHelper, which provide comfort methods to simplify and accelerate implementation.

1.3.1 Matchmaking procedure

As shown in the activity diagram in Figure 12, in the matchmaking process for Manufacturing-as-a-Service systems, each available manufacturing service is iteratively evaluated against the requirements of a specific product as defined by means of a requirement set. This involves iterating over all product requirements.

For each requirement, the related manufacturing service characteristic quantity is retrieved. If this characteristic quantity is not directly defined, an attempt is made to calculate it by calling a web service. If this is not possible—for example, due to the absence of a suitable web service or insufficient data—the requirement is considered not fulfilled, and the process continues with the next requirement.

Next, it is checked whether the requirement is fulfilled based on the characteristic quantity. If the requirement is fulfilled, the matching degree for this requirement-characteristic combination is calculated. If not fulfilled, the matching degree is set to zero.

If the unmet requirement is a must-requirement, the product-service combination is marked as not possible, and the iteration proceeds to the next manufacturing service.

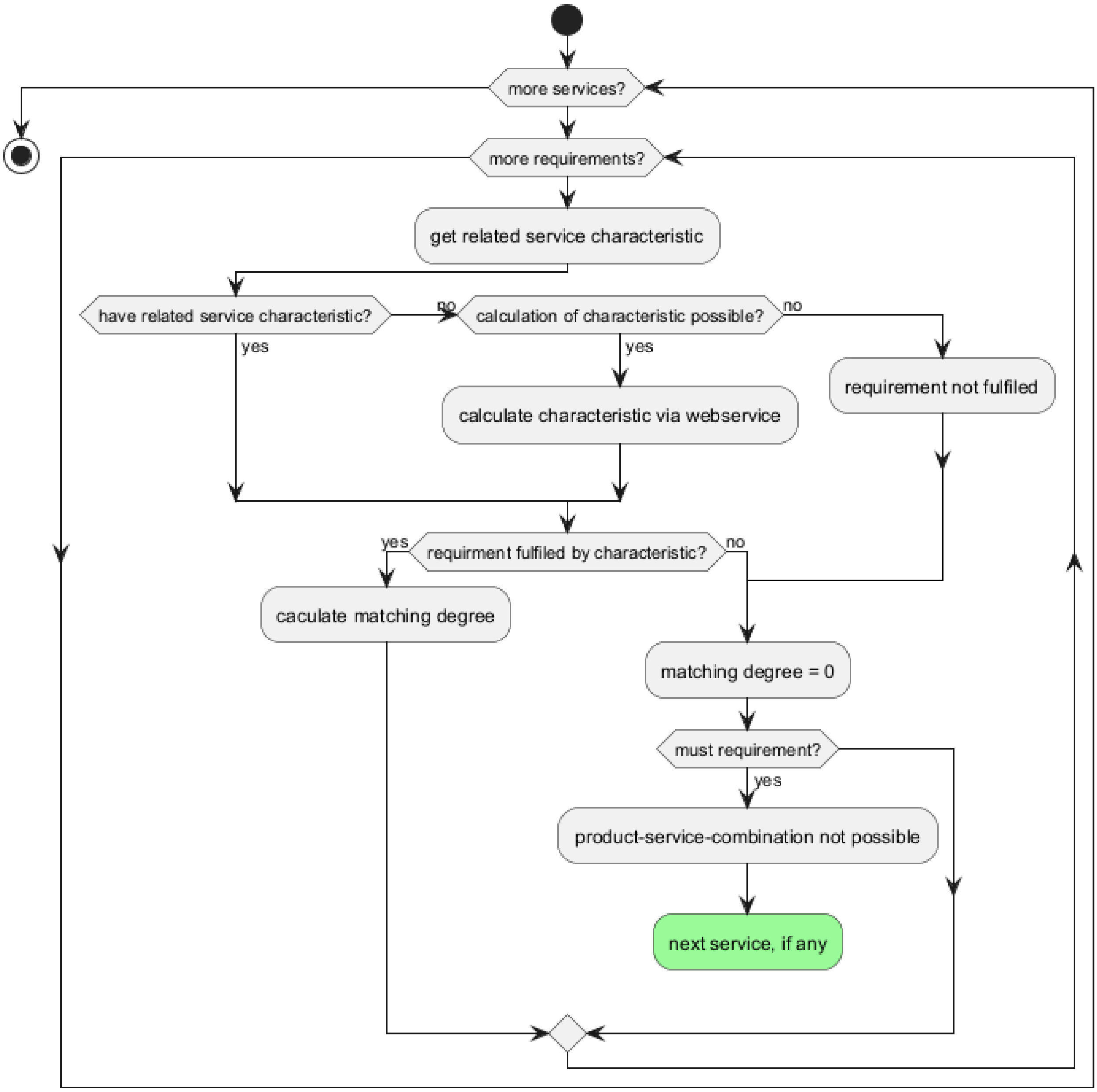

more services?
more requirements?
get related service characteristic
have related service characteristic?
no
calculation of characteristic possible?
no
yes
yes
calculate characteristic via webservice
requirement not fulfiled
yes
requirment fulfiled by characteristic?
no
caculate matching degree
matching degree = 0
must requirement?
yes
product-service-combination not possible
next service, if any

Figure 12: UML activity diagram of the matchmaking procedure

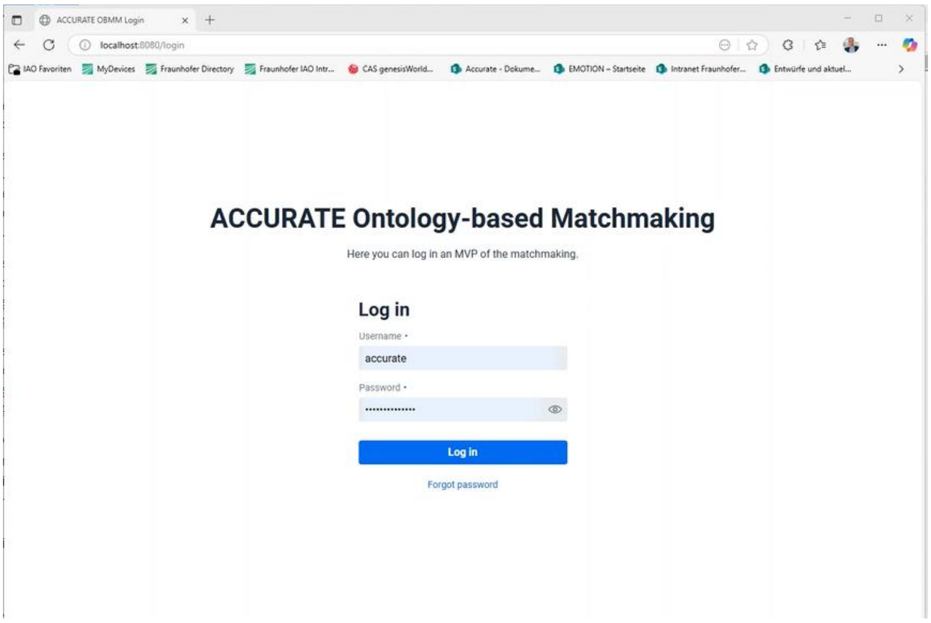
1.3.2 Implementation as Software System

The implementation of the overall matchmaking was done by classes in the programming language Java, as "the" programming language" for semantic (web) solutions. As introduced in chapter 1.3, as programming libraries especially Apache Jena and Vaadin were used.
For the matchmaking procedure itself, a class OntoMatchmaker was developed. Some of its method signatures are shown in the following:

```
// returns ProductServiceMatches where requirements are fulfilled
public List<String> makeACCURATEMatch(String product)

// returns ProductServiceMatches where requirements are fulfilled
and besides calculates matching degrees
public List<String> makeACCURATEMatchRanking(String product)

// returns a double value with the matching degree of a service
related to a list of requirements
private double calculateMatchingDegree(List<String> requirements,
String service)

// returns true if a service fulfils a given requirement. If a
requirement cannot be proven as there is no related service
characteristic it is seen as not fulfilled -> answer is false
private boolean checkRequirementFulfillmentByService(String
requirement, String service)

// returns true if a characteristic of a service fits to a
quantified requirement and fulfils it.
boolean checkQuantifiedRequirementFulfillmentByCharacteristic(
String requirement, String characteristic)

// returns true if a characteristic of a service fits to an
unquantified requirement and fulfils it.
boolean checkUnquantifiedRequirementFulfillmentByCharacteristic(
String requirement, String characteristic)
```

By leveraging Vaadin, the functionalities of the matchmaking are provided via an internet browser access, so in principle remotely as a webserver and user operation system agnostic. The realised system provides a graphical user interface which enables the editing of requirements, forming a requirement set, which is related to the manufacturing of a product by means of the (manufacturing) services offered in a Manufacturing-as-a-Service-system, of manufacturing services with their characteristics, the matchmaking, and the analysis of the product-service-matches as determined by the matchmaking procedure.

The developed system provides a user management with a role-based access management, as shown in Figure 13.



ACCURATE Ontology-based Matchmaking

Here you can log in an MVP of the matchmaking.

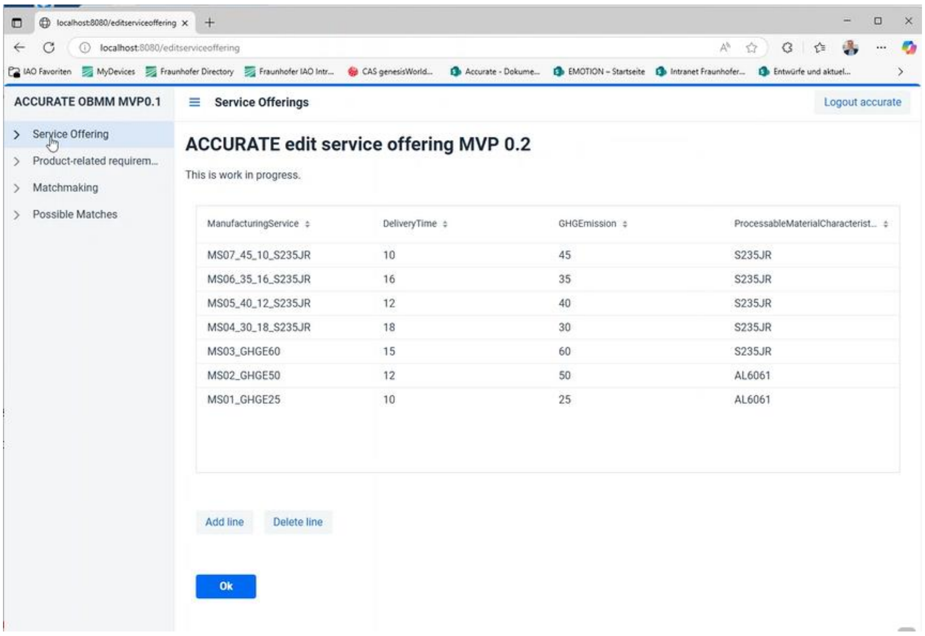
Log in

Username * accurate

Password * ••••••••••••••

Figure 13: Log in-page of the first version of the matchmaking system

The following figures provide exemplary screenshots of the graphical user interface of the first version of the matchmaking system via an internet browser, so Figure 14 for the editing of service offerings which are available in the MaaS-system.



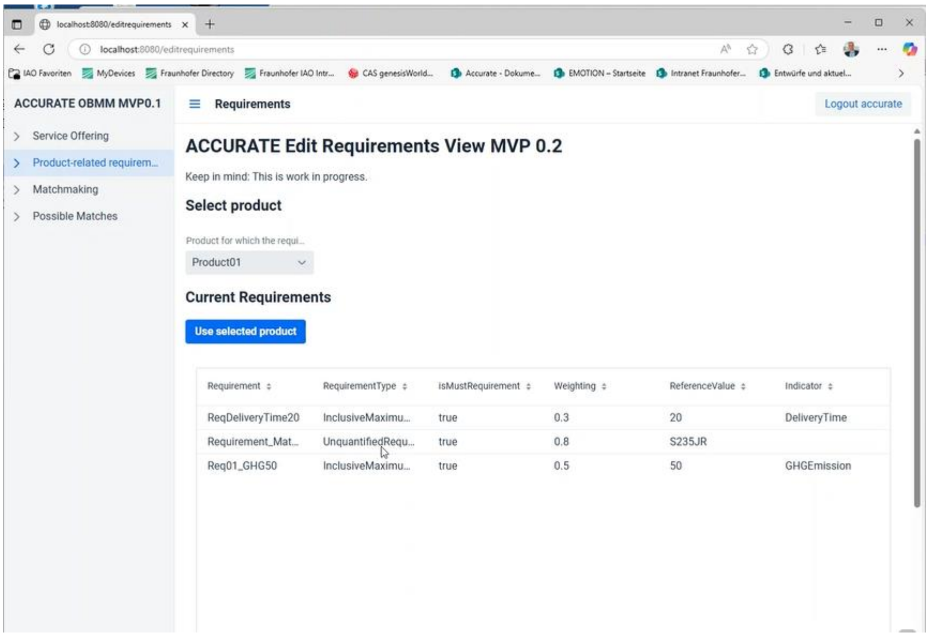
ACCURATE edit service offering MVP 0.2

This is work in progress.

ManufacturingService	DeliveryTime	GHGEmission	ProcessableMaterialCharacterist...
MS07_45_10_S235JR	10	45	S235JR
MS06_35_16_S235JR	16	35	S235JR
MS05_40_12_S235JR	12	40	S235JR
MS04_30_18_S235JR	18	30	S235JR
MS03_GHGE60	15	60	S235JR
MS02_GHGE50	12	50	AL6061
MS01_GHGE25	10	25	AL6061

Figure 14: Editor for service offering management

The editor for requirements to manufacture products, shown in Figure 15, also enables the selection of the product that has to be manufactured and based on this the addition, edition, and removal of related requirements on the manufacturing of the product. For the requirements, also weightings can be defined and the requirements marked as must criteria (requirements).



ACCURATE Edit Requirements View MVP 0.2

Keep in mind: This is work in progress.

Select product

Product for which the requi...: Product01

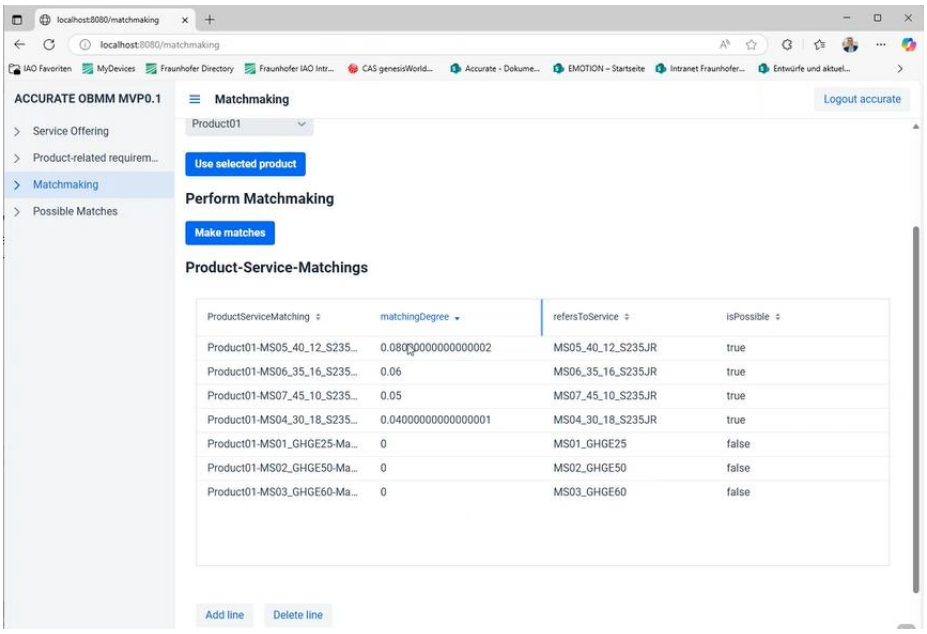
Current Requirements

Use selected product

Requirement	RequirementType	isMustRequirement	Weighting	ReferenceValue	Indicator
ReqDeliveryTime20	InclusiveMaximu...	true	0.3	20	DeliveryTime
Requirement_Mat...	UnquantifiedRequ...	true	0.8	S235JR	
Req01_GHG50	InclusiveMaximu...	true	0.5	50	GHGEmission

Figure 15: Editor for requirements

An overview about calculated product-service-matches, including their principal feasibility and matching degree as provided by the matchmaking system is shown by Figure 16.



Product01

Use selected product

Perform Matchmaking

Make matches

Product-Service-Matchings

ProductServiceMatching	matchingDegree	refersToService	isPossible
Product01-MS05_40_12_S235...	0.08000000000000002	MS05_40_12_S235JR	true
Product01-MS06_35_16_S235...	0.06	MS06_35_16_S235JR	true
Product01-MS07_45_10_S235...	0.05	MS07_45_10_S235JR	true
Product01-MS04_30_18_S235...	0.04000000000000001	MS04_30_18_S235JR	true
Product01-MS01_GHGE25-Ma...	0	MS01_GHGE25	false
Product01-MS02_GHGE50-Ma...	0	MS02_GHGE50	false
Product01-MS03_GHGE60-Ma...	0	MS03_GHGE60	false

Figure 16: Overview about the calculated product-service-matches

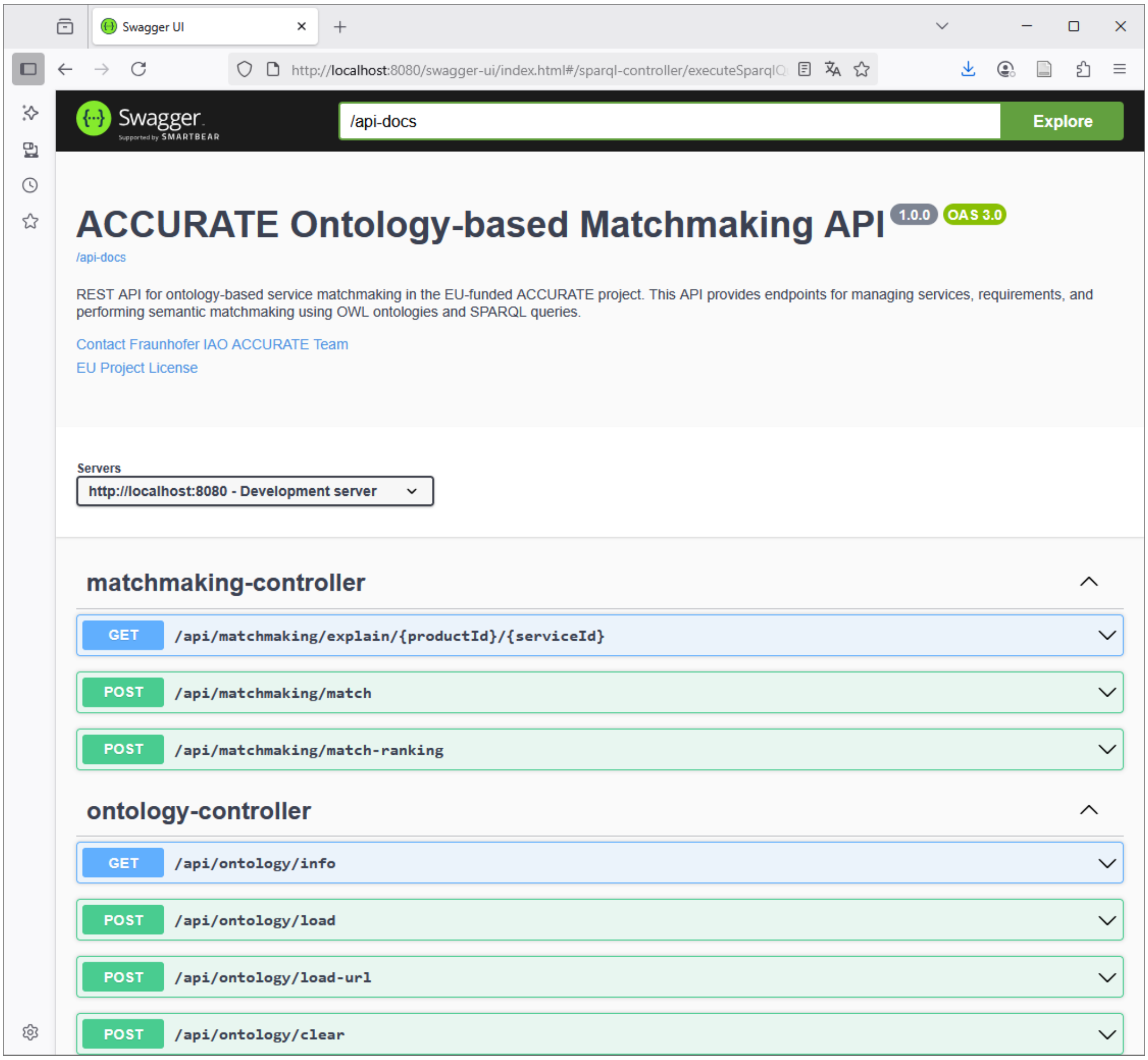
1.3.3 Marketplace-oriented realisation

For the marketplace-oriented realisation, the matchmaking solution as introduced in chapter 1.3.2 has to be movable as a software container as well as to be accessible remotely. To realise this, in a first step, the Vaadin application, which is already based on Spring Boot as well as a graphical user interface for internet browsers was transferred into a Docker container. For this, not only the respective fat JAR has to be contained in the Docker image, but also the ontology definitions by means of the related OWL-files as well as the needed ontpolicy.rdf file, which is needed by Apache Jena to load local ontologies, which are not publicly available on the internet.

Furthermore, the matchmaking application was extended by web services to allow the remote access with other tools than the provided Vaadin-GUI. Therefore, web services covering the matchmaking, requirements, and products, services as well as ontology management and SPARQL-requests were developed. Related endpoints are realized accordingly. For this, the following controllers were implemented:

- matchmaking-controller,
- ontology-controller,
- requirement-management-controller,
- service-management-controller, and
- sparql-controller.

The current web services are documented with Swagger, as shown in Figure 17 and summarized in the following with exemplary values in related request bodies. The implementation may be advanced in the frame of WP6 and WP7, concerning the ACCURATE-marketplace and the industrial pilot cases.


Swagger UI
http://localhost:8080/swagger-ui/index.html#/sparql-controller/executeSparqlQ
Swagger
Supported by SMARTBEAR
/api-docs
Explore
1.0.0
OAS 3.0

ACCURATE Ontology-based Matchmaking API

/api-docs

REST API for ontology-based service matchmaking in the EU-funded ACCURATE project. This API provides endpoints for managing services, requirements, and performing semantic matchmaking using OWL ontologies and SPARQL queries.

Contact Fraunhofer IAO ACCURATE Team

EU Project License

Servers

http://localhost:8080 - Development server

matchmaking-controller

GET /api/matchmaking/explain/{productId}/{serviceId}

POST /api/matchmaking/match

POST /api/matchmaking/match-ranking

ontology-controller

GET /api/ontology/info

POST /api/ontology/load

POST /api/ontology/load-url

POST /api/ontology/clear

Figure 17: Exemplary screenshot of Swagger with web service documentation of the marketplace-oriented solution.

matchmaking-controller

```
GET/api/matchmaking/explain/{productId}/{serviceId}
{
  "isPossible": true,
  "matchingDegree": 0
}

POST/api/matchmaking/match
{
  "productUri": "string"
}

POST/api/matchmaking/match-ranking
{
  "productUri": "string"
}
```

ontology-controller

```
GET/api/ontology/info

POST/api/ontology/load
{
  "file": "string"
}

POST/api/ontology/load-url
{
  "url": "string"
}

POST/api/ontology/clear
```

requirement-management-controller

```
DELETE/api/requirements/requirements/{requirementId}

GET/api/requirements/products
[
  {
    "uri": "string",
    "name": "string",
    "description": "string",
    "requirements": [
      {
        "uri": "string",
        "name": "string",
        "description": "string",
        "requirementType": "string",
        "referenceValue": "string",
        "unit": "string",
        "numericValue": 0,
        "isMustRequirement": true,
        "weighting": 0
      }
    ]
  }
]
```

GET/api/requirements/products/{productId}/requirements

```
[
  {
    "uri": "string",
    "name": "string",
    "description": "string",
    "requirementType": "string",
    "referenceValue": "string",
    "unit": "string",
    "numericValue": 0,
    "isMustRequirement": true,
    "weighting": 0
  }
]
```

GET/api/requirements/products/{productId}

```
{
  "uri": "string",
  "name": "string",
  "description": "string",
  "requirements": [
    {
      "uri": "string",
      "name": "string",
      "description": "string",
      "requirementType": "string",
      "referenceValue": "string",
      "unit": "string",
      "numericValue": 0,
      "isMustRequirement": true,
      "weighting": 0
    }
  ]
}
```

POST/api/requirements/products

```
{
  "uri": "string",
  "name": "string",
  "description": "string",
  "requirements": [
    {
      "uri": "string",
      "name": "string",
      "description": "string",
      "requirementType": "string",
      "referenceValue": "string",
      "unit": "string",
      "numericValue": 0,
      "isMustRequirement": true,
      "weighting": 0
    }
  ]
}
```

POST/api/requirements/products/{productId}/requirements

```
{
  "uri": "string",
  "name": "string",
  "description": "string",
  "requirementType": "string",
  "referenceValue": "string",
  "unit": "string",
  "numericValue": 0,
  "isMustRequirement": true,
  "weighting": 0
}
```

service-management-controller

DELETE/api/services/{serviceId}
GET/api/services/{serviceId}

```
{
  "uri":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Service_1
23",
  "name": "3D Printing Service",
  "description": "High-precision 3D printing service for
prototypes",
  "serviceType":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Manufactu
ringService",
  "characteristics": [
    {
      "uri": "string",
      "name": "Precision",
      "characteristicType":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Precision
Characteristic",
      "value": "string",
      "unit": "mm",
      "numericValue": 0.1
    }
  ]
}
```

GET/api/services

```
[
  {
    "uri":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Service_1
23",
    "name": "3D Printing Service",
    "description": "High-precision 3D printing service for
prototypes",
    "serviceType":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Manufactu
ringService",
    "characteristics": [
      {
        "uri": "string",
        "name": "Precision",
        "characteristicType":
"http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Precision
Characteristic",
        "value": "string",
        "unit": "mm",
        "numericValue": 0.1
      }
    ]
  }
]
```

POST/api/services

```
{
  "uri": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Service_123",
  "name": "3D Printing Service",
  "description": "High-precision 3D printing service for prototypes",
  "serviceType": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#ManufacturingService",
  "characteristics": [
    {
      "uri": "string",
      "name": "Precision",
      "characteristicType": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#PrecisionCharacteristic",
      "value": "string",
      "unit": "mm",
      "numericValue": 0.1
    }
  ]
}
```

PUT/api/services/{serviceId}

```
{
  "uri": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#Service_123",
  "name": "3D Printing Service",
  "description": "High-precision 3D printing service for prototypes",
  "serviceType": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#ManufacturingService",
  "characteristics": [
    {
      "uri": "string",
      "name": "Precision",
      "characteristicType": "http://accurate.de.iao.fraunhofer.de/ontologies/ACCURATE#PrecisionCharacteristic",
      "value": "string",
      "unit": "mm",
      "numericValue": 0.1
    }
  ]
}
```

sparql-controller

POST/api/sparql/update

```
{
  "query": "string",
  "queryType": "UPDATE"
}
```

POST/api/sparql/query

```
  {
    "query": "string",
    "queryType": "string"
  }
```

In the course of WP6 and WP7, the current system solution may be transferred into a three-container solution, if needed, to satisfy requirements and conditions by industrial users. As the main approach of WP2 is to do research and development (RTD) for a Research and Innovation Action (RIA) this is not done in advance, as WP2 already shows the principal feasibility of the approaches by means of technology validations and demonstrations, respectively. When needed, the three-container solution in WP6/WP7 could consist of containers for SPARQL server, matchmaking tool, and frontend, e.g., by

- **Container 1: Fuseki SPARQL Endpoint**
  - **Basis**: Official Fuseki Docker Images
  - **Purpose**: Ontology storage, SPARQL queries, data persistence
  - **Ports**: 3030 (standard Fuseki)
- **Container 2: Matchmaking Service**
  - **Basis**: OpenJDK + Spring Boot
  - **Purpose**: REST APIs for matchmaking, ontology management
  - **Ports**: 8080
  - **APIs**:
    - POST /api/matchmaking/match - Basic matching
    - POST /api/matchmaking/match-ranking - Ranking-based matching
    - GET/POST/PUT/DELETE /api/services - Service management
    - GET/POST/PUT/DELETE /api/requirements - Requirements management
    - POST /api/sparql - Direct SPARQL queries
- **Container 3: Web UI (optional)**
  - **Base**: OpenJDK + Spring Boot + Thymeleaf
  - **Purpose**: Demo/test interface
  - **Ports**: 8081

Based on the complexity of the matchmaking algorithm and proceeding, it is not intended to realize the matchmaking of products with their requirements solely with SPARQL but to provide a related service written in Java, Python or any other programming language with reliable programming libraries for ontology processing, like Apache Jena as used in the current Java implementation of the matchmaking service. The matchmaking approach can be characterized by three phases that involve several kinds of comparison and related calculations, by the indirect relation from requirements as well as characteristics to actual quantitative values via QUDT-quantities with the related measurement units, and by the possibility to conduct the matchmaking via a graphical usage interface. The application of QUDT results in multipliers and offsets concerning SI-units of measurement, which enables to automatically compare not only different measurement units like the ones of the metric (e.g. Meters) with the imperial system of measurements units (e.g. Inches), but also to automatically compare scaled units like Millimetres) with base units like Meters.

Furthermore, during the development of WP6 and WP7 the APIs as described above may be advanced to appropriately support the realization on the ACCURATE-marketplace and the industrial pilots, respectively.

1.4 Hybridization of the approach by adding data-driven functionality

In the following, a conventional approach using ChatGPT is introduced, then an Open Source-based approach leveraging Llama, and then information about the transfer to the marketplace is given.

1.4.1 Conventional Approach with ChatGPT

For the discussion of the hybridization, the starting point was a structured literature review, too. In contrast to the detailed analysis of state of the art, presented in chapter 1.1, all relevant scientific articles and papers have been identified by a defined search term in Web of Science and Scopus. But the following steps with check of title and abstract have been performed differently. To derive the results the following search term was applied to literature databases:

```
( TITLE-ABS-KEY ( hybrid AND matchmaking ) OR TITLE-ABS-KEY ( semantic
AND matchmaking ) OR TITLE-ABS-KEY ( logic-based AND matchmaking )
OR TITLE-ABS-KEY ( non-logic-based AND matchmaking ) OR TITLE-ABS-
KEY ( syntactic AND matchmaking ) )
```

Based on the search term, it was possible to identify 1050 scientific publications (Figure 18). All papers have been sub-divided into their main relevance. Based on this, most of the publications were dedicated to web services (695), whereas the minority (59) was addressing aspects of relevance for the project. The clusters "production" and "manufacturing" have been analysed again, which left in total only 15 sources.

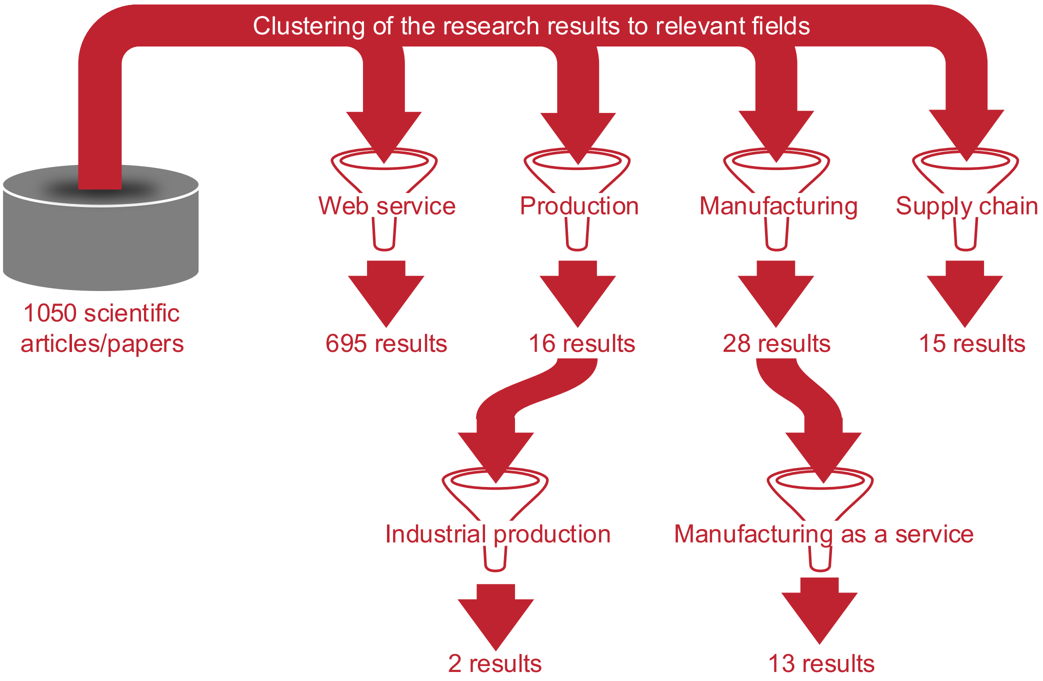


Figure 18. sorting of scientific publication to the application fields

The publications do not contain any combinations of data-driven and semantic approaches. All publications identify optimization issues that are to be addressed using various methods. Ye and Ming-Hua (Ye and Ming-Hua 2010) propose a three-stage optimization process, pursuing specific approaches for matching in each stage. They combine semantic similarity with boundary condition checks. Similarly, Lui et al. (Lui et al 2012) focus on evaluating the similarity of service attributes. They use a fuzzy approach for the membership function, whereby the scope is very narrowly defined with material selection for casting processes. Ameri and Dutta (Ameri and Dutta 2008) also use similarity evaluation. They use MSDL and a directly labelled tree for matching. Since the use case is not comparable, only inspiration can be derived. Based on MSDL, the approach of Ameri and Patil is extended by agents, which represents a hybridization. The combination of semantic modelling and agent-based approaches is also proposed by Nizamis et al. (Nizamis et al 2018), but they even implement a rule-based matchmaking engine. It shall evaluate the suppliers based on different requesters' criteria. In contrast, multi-level ontologies connected with a service-oriented architecture is the approach proposed by Delamer and Lastra (Delamer and Lastra 2006) Jirkovský et al. (Jirkovský et al. 2019) also attempt to establish matchmaking using three elements – product, process and resource – with the help of SQWRL. The approach appears innovative, but since there is no discussion of tried-and-tested approaches, its practical applicability is questionable. In addition, the SQWRL query must be created manually each time.

This source analysis shows that no concrete implementation of hybrid approaches has been successfully used and further developed to date. However, it can be stated that the use of a semantic model in combination with various other methods can offer positive added value in terms of the desired goal. This is because additional functions can be linked or developed that are not possible with ontologies alone as semantic representations.

Based on the analysis of the state of the art, a chatbot approach using Large Language Models (LLMs) was used to hybridize the ontology-based approach. As described below, for this, trials were first carried out before implementing an approach which can be operated on the ACCURATE-marketplace.

Objective of the trials was to investigate the analysability of the operational semantic data model, i.e., the ontology of ACCURATE, in the sense of a knowledge graph, by Large Language Models. For this, static context injection of the knowledge graph by means of the system prompt was done as a kind of GraphRAG (knowledge Graph Retrieval-Augmented Generation). FhGenie as Fraunhofer-internal instance of a chatbot was used for the trials as it ensures data safety and security combined with a high reliability and low costs for the project. As LLMs GPT 4o mini and GPT 4.1 were used. The context “You are an assistant in a company with outstanding expertise in Knowledge-Based Supply Chain Management, who supports supply chain managers in planning and operating a Manufacturing-as-a-Service-System. Use the following data in OWL for an ontology-based evaluation of a manufacturing-as-a-service system:” followed by the content of the OWL-file as listed in the appendix of this document was injected in the system prompt.

The chatbots, set-up as described above, had to perform three tasks and answer related questions, respectively:

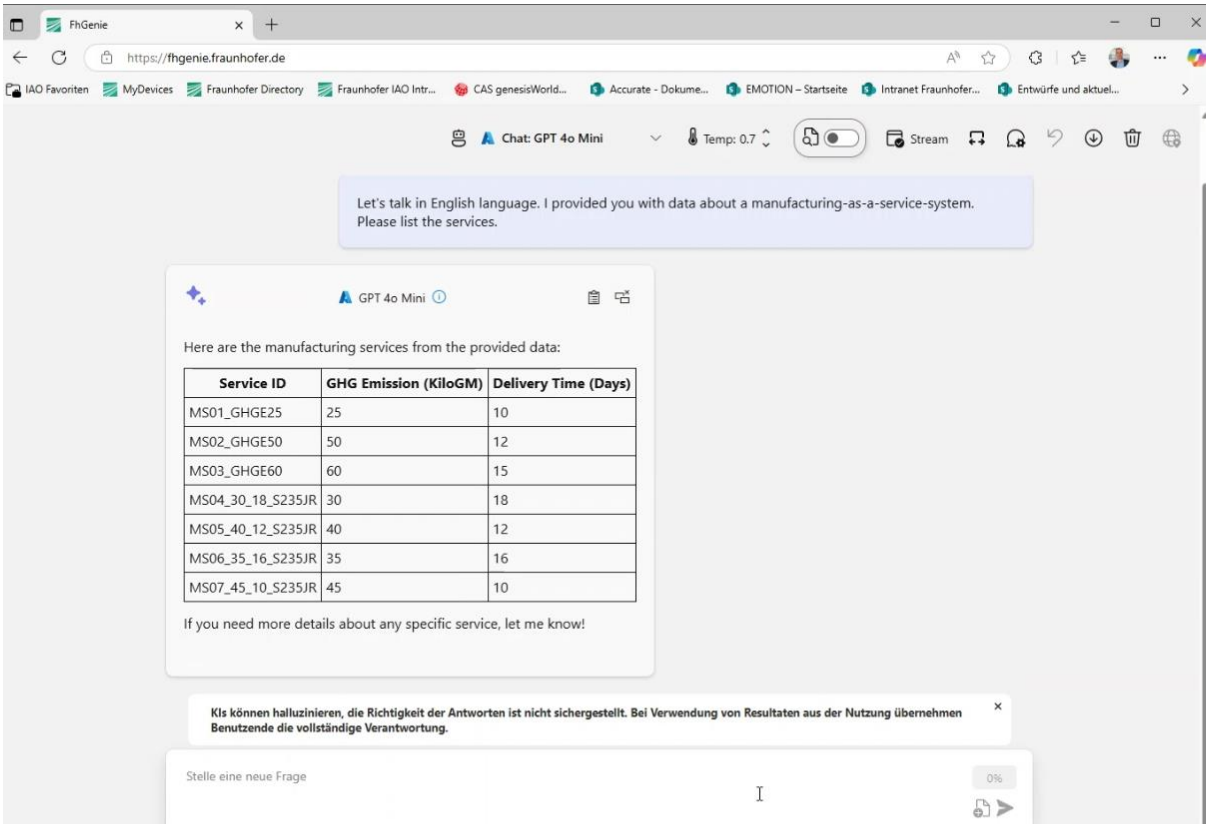
- List the services in the data model.
- Summarize the fulfilment of the requirements contained in the model by the services and their characteristics.
- Suggest the most sustainable service and explain the decision.

As summarized in the following table, both chatbot realisations were able to perform the three tasks.

Table 3: Task fulfilment by the chatbots

Task	GPT 4o mini	GPT 4.1
List services	✓	✓
Check requirements fulfilment	✓	✓
Suggest most sustainable service	✓	✓

Figure 19 to Figure 21 show screenshots from conversations done to perform the tasks by the chatbots.



Chat: GPT 4o Mini

Temp: 0.7

Let's talk in English language. I provided you with data about a manufacturing-as-a-service-system. Please list the services.

GPT 4o Mini

Here are the manufacturing services from the provided data:

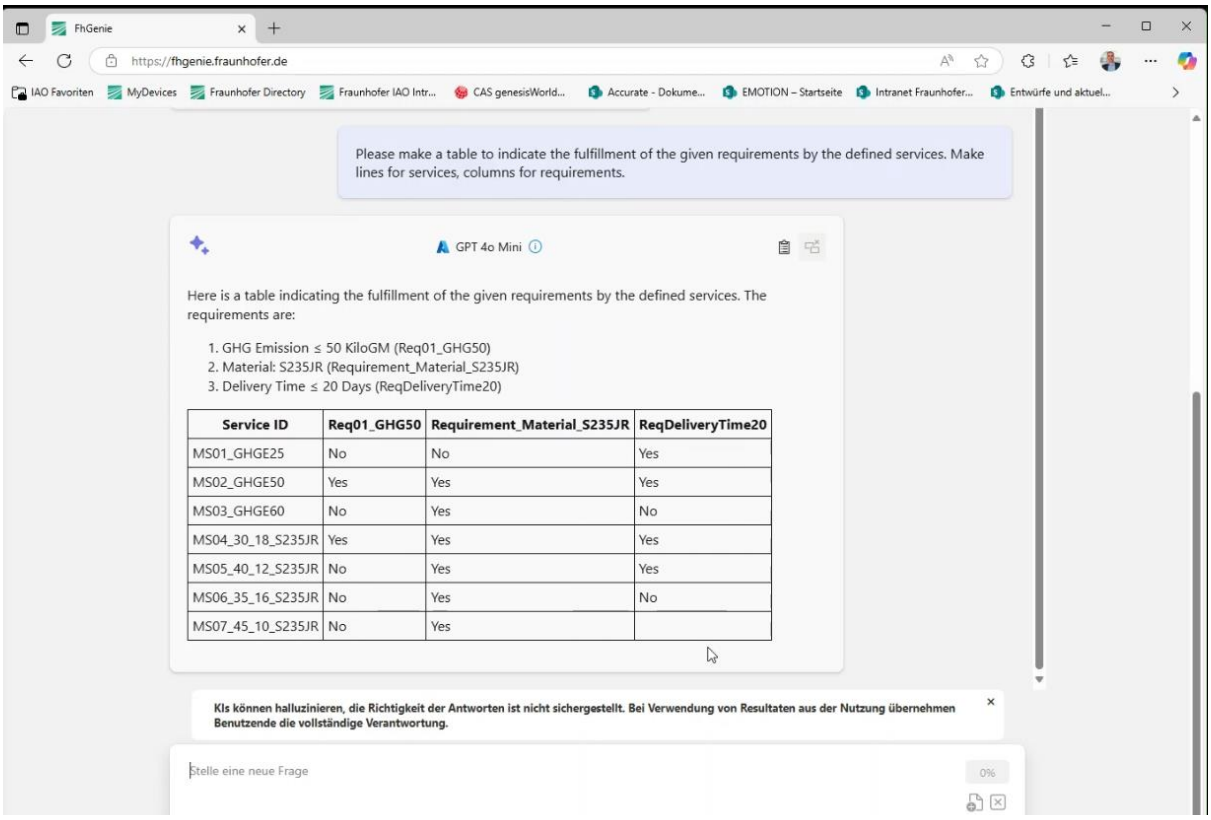
Service ID	GHG Emission (KiloGM)	Delivery Time (Days)
MS01_GHGE25	25	10
MS02_GHGE50	50	12
MS03_GHGE60	60	15
MS04_30_18_S235JR	30	18
MS05_40_12_S235JR	40	12
MS06_35_16_S235JR	35	16
MS07_45_10_S235JR	45	10

If you need more details about any specific service, let me know!

KIs können halluzinieren, die Richtigkeit der Antworten ist nicht sichergestellt. Bei Verwendung von Resultaten aus der Nutzung übernehmen Benutzende die vollständige Verantwortung.

Stelle eine neue Frage

Figure 19: Conversation to list the manufacturing services contained in the data model (ontology)



Please make a table to indicate the fulfillment of the given requirements by the defined services. Make lines for services, columns for requirements.

GPT 4o Mini

Here is a table indicating the fulfillment of the given requirements by the defined services. The requirements are:

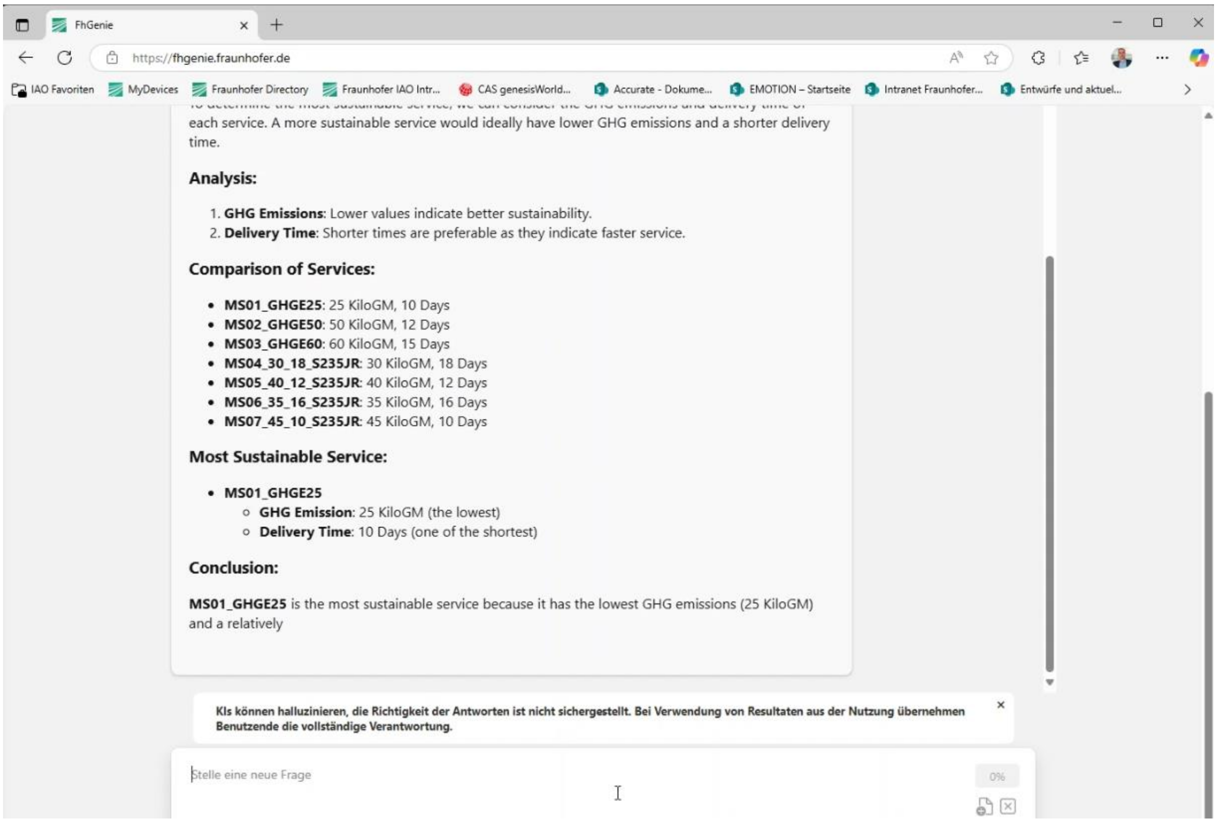
1. GHG Emission ≤ 50 KiloGM (Req01_GHG50)
2. Material: S235JR (Requirement_Material_S235JR)
3. Delivery Time ≤ 20 Days (ReqDeliveryTime20)

Service ID	Req01_GHG50	Requirement_Material_S235JR	ReqDeliveryTime20
MS01_GHGE25	No	No	Yes
MS02_GHGE50	Yes	Yes	Yes
MS03_GHGE60	No	Yes	No
MS04_30_18_S235JR	Yes	Yes	Yes
MS05_40_12_S235JR	No	Yes	Yes
MS06_35_16_S235JR	No	Yes	No
MS07_45_10_S235JR	No	Yes	

KIs können halluzinieren, die Richtigkeit der Antworten ist nicht sichergestellt. Bei Verwendung von Resultaten aus der Nutzung übernehmen Benutzende die vollständige Verantwortung.

Stelle eine neue Frage

Figure 20: Conversation to get an overview about the requirements' fulfilments by the services in the model



each service. A more sustainable service would ideally have lower GHG emissions and a shorter delivery time.

Analysis:

1. **GHG Emissions**: Lower values indicate better sustainability.
2. **Delivery Time**: Shorter times are preferable as they indicate faster service.

Comparison of Services:

- **MS01_GHGE25**: 25 KiloGM, 10 Days
- **MS02_GHGE50**: 50 KiloGM, 12 Days
- **MS03_GHGE60**: 60 KiloGM, 15 Days
- **MS04_30_18_S235JR**: 30 KiloGM, 18 Days
- **MS05_40_12_S235JR**: 40 KiloGM, 12 Days
- **MS06_35_16_S235JR**: 35 KiloGM, 16 Days
- **MS07_45_10_S235JR**: 45 KiloGM, 10 Days

Most Sustainable Service:

- **MS01_GHGE25**
  - **GHG Emission**: 25 KiloGM (the lowest)
  - **Delivery Time**: 10 Days (one of the shortest)

Conclusion:

MS01_GHGE25 is the most sustainable service because it has the lowest GHG emissions (25 KiloGM) and a relatively

KIs können halluzinieren, die Richtigkeit der Antworten ist nicht sichergestellt. Bei Verwendung von Resultaten aus der Nutzung übernehmen Benutzende die vollständige Verantwortung.

Stelle eine neue Frage

Figure 21: Result of a conversation to determine the most sustainable service

1.4.2 Open-Source Approach

To transfer the system approach to open source and preparing the shift towards the marketplace of ACCURATE by means of docker containers, the architecture shown in Figure 22 was developed.

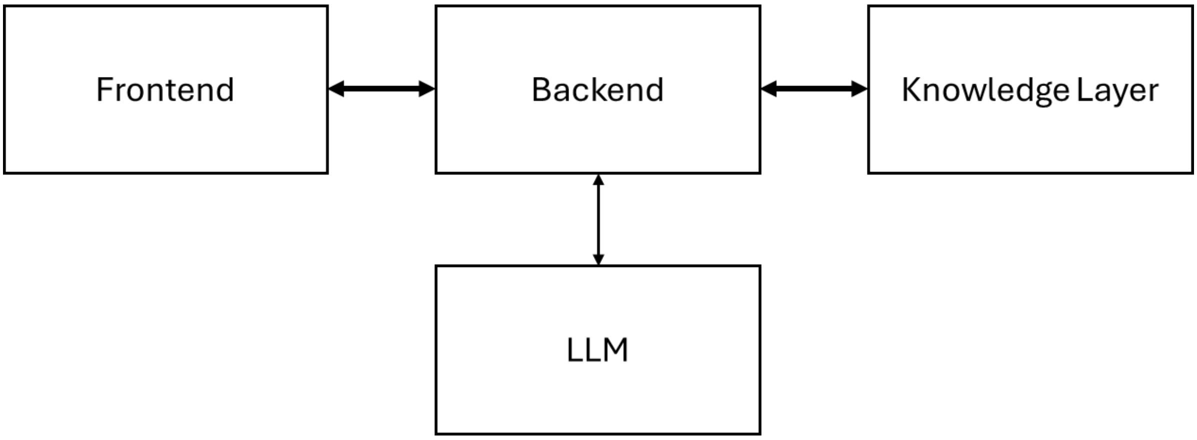


Figure 22: Conceptual architecture of the ACCURATE Matchmaking Assistant via Open Source LLM

In the following, the technology selection for the ACCURATE Matchmaking Assistant is introduced.

1. **Large Language Model Selection**

The Large Language Model (LLM) **Llama 3.1 8B** was selected based on performance and open source considerations.

Llama 3.1 8B provides a good balance between computational efficiency and reasoning capability for manufacturing domain queries. Thereby, 8B parameters are typically sufficient for domain-specific tasks while maintaining sub-second response times on consumer hardware. Furthermore, it offers multilingual capability, which is an asset concerning European projects, as it supports German, English, French, Italian, and other EU languages

Concerning open source compliance, the LLMs custom license allows commercial use and modification. Besides, it supports reproducibility as full model weights and training methodology are publicly available. The local deployment ensures data never leaves EU infrastructure.

As an alternative, Mistral 7B was considered. Whereby this solution offers advantages as faster inference and its European origin by a French company, it also has disadvantages as a typically lower performance on complex reasoning tasks. Therefore, it is selected as fallback option for resource-constrained deployments.

2. Deployment Framework Selection

As deployment framework Ollama was selected based on its ease of deployment and Application Programming Interface (API) compatibility. The one-command installation of Ollama supports research reproducibility. Furthermore, its automatic quantization reduces memory footprint substantially without significant quality loss. Additionally, its docker availability ensures consistent deployment across research institutions. Besides, its OpenAI-compatible REST API enables easy integration with existing research tools, whereby the standardized interface facilitates comparison studies with proprietary models.

3. Knowledge Graph Technology Stack Selection: Apache Jena Fuseki

For the Knowledge Graph Technology Apache Jena Fuseki was selected based on its Semantic Web Standards Compliance and research community adoption. Apache Jena is compliant with RDF, RDFS, OWL 2, and SPARQL 1.1. Furthermore, it offers built-in support for OWL reasoning, which is essential for the manufacturing and supply chain ontology inference in ACCURATE. The interoperability offered by Apache Jena ensures compatibility with other European research projects using semantic technologies. An additional benefit of selecting Apache Jena is its adoption in the research community, which can be seen by the vast number of citations for Apache Jena and the high number of European-funded projects, which use Jena. Besides, there are applications of Apache Jena by major global companies.

Alternatives that were considered during the selection of the Knowledge Graph technology stack are summarized in Table 4.

Table 4: Alternatives considered for the Knowledge Graph technology stack selection

Technology	Pros	Cons	Decision Factor
GraphDB Free	Better performance, visualization	Proprietary core, licensing complexity	Rejected: IP concerns

Technology	Pros	Cons	Decision Factor
Stardog	Enterprise features	Commercial license required	Rejected: Cost
Virtuoso Open	High performance	Complex configuration	Rejected: Maintenance overhead

4. Retrieval Augmented Generation approach

Based on the structured knowledge advantage, a SPARQL-based Retrieval Augmented Generation (RAG) approach is selected in comparison with vector embeddings. Manufacturing ontologies contain precise numerical relationships like tolerances, emissions, and lead times, that are better preserved in symbolic representation than vector embeddings.

5. Backend Framework Selection

FastAPI was chosen as Backend framework because of several research integration benefits: The automatic OpenAPI Schema facilitates integration with research tools and other EU projects. The type validation reduces integration errors substantially compared to untyped frameworks, and the async Support is essential for handling concurrent SPARQL queries and LLM calls.

6. Frontend Framework Selection

For the Frontend framework, StreamLit was selected because of rapid prototyping advantages and research validation features. Rapid prototyping advantages result from typically faster development compared to React/Vue.js for research prototypes. Furthermore, StreamLit supports scientific computing integration by its native support for the python programming libraries Pandas, Matplotlib, and Plotly. Additionally, StreamLit offers deployment simplicity by its single-command deployment, which is an advantage for research demonstrations and reproductions. Moreover, StreamLit facilitates research validation by query logging, simple realization of A/B testing and by its export capabilities in multiple formats like CSV, JSON and PDF.

7. Summary

Table 5 summarizes the technology stack selection.

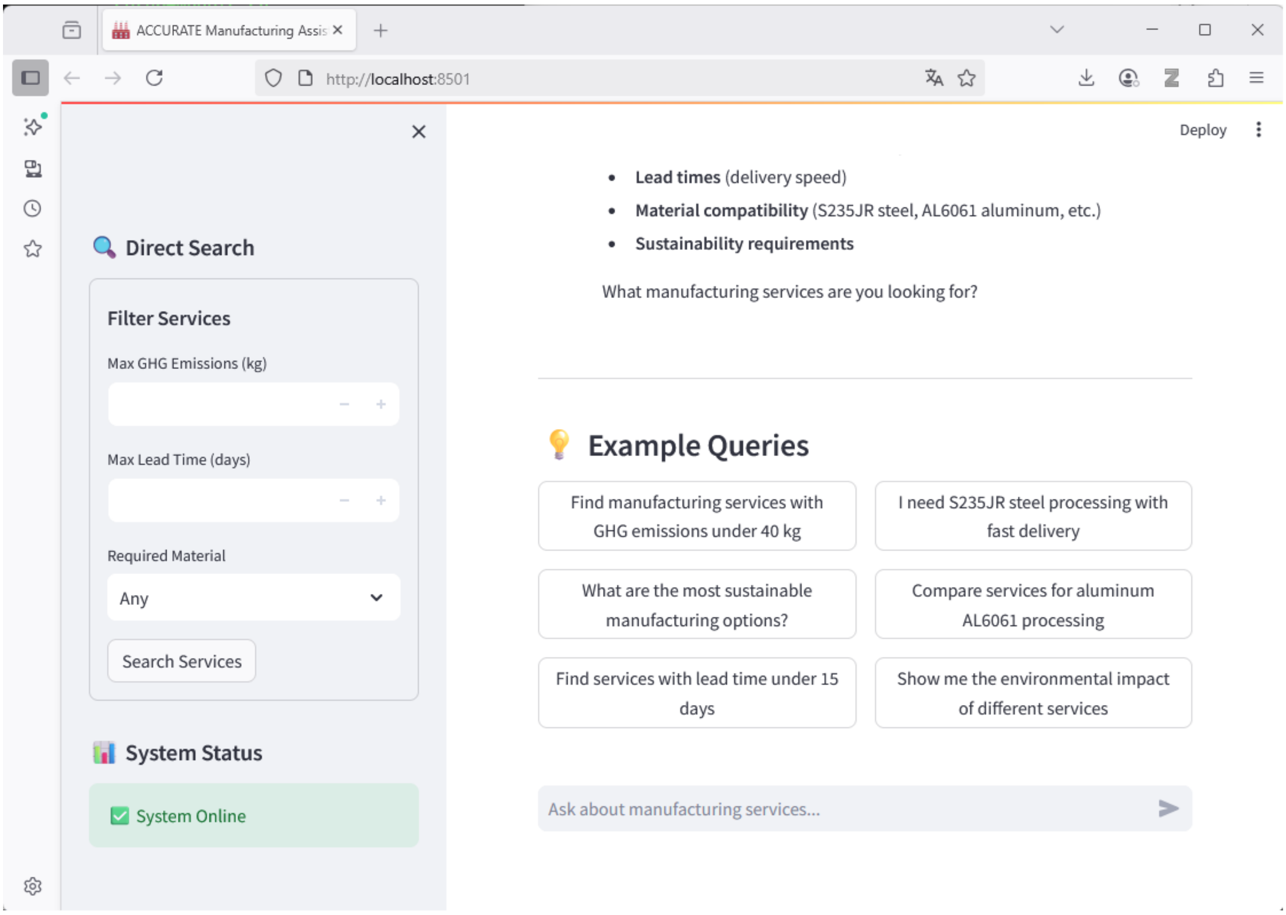
Table 5: Technology stack selection for the ACCURATE Matchmaking Assistant

Component	Selection	Alternative
LLM	Ollama + Llama 3.1	vLLM + Mistral
Ontology Server	Apache Jena Fuseki	GraphDB Free
RAG Framework	LangChain	Haystack
Backend	FastAPI	Flask
Frontend	StreamLit	Gradio
Vector DB	Chroma	Weaviate

The selected technology stack aims for a reasonable balance of:

1. Scientific Rigor: Open-source components with established research validation
2. European Compliance: GDPR-compliant, locally deployable architecture
3. Scalability: Microservices-ready architecture for multi-institutional deployment
4. Cost Effectiveness: substantial cost reduction compared to proprietary alternatives

Figure 23 shows a screenshot of a functional prototype of the Open-Source-based chatbot “ACCURATE-Assistant” for matchmaking in MaaS-systems. The prototype will be advanced in the course of WP6 and WP7, which are concerned with the ACCURATE-marketplace/framework and the industrial pilots.


ACCURATE Manufacturing Assis
http://localhost:8501
Deploy
Direct Search
Filter Services
Max GHG Emissions (kg)
Max Lead Time (days)
Required Material
Any
Search Services
System Status
System Online
Example Queries
Find manufacturing services with GHG emissions under 40 kg
I need S235JR steel processing with fast delivery
What are the most sustainable manufacturing options?
Compare services for aluminum AL6061 processing
Find services with lead time under 15 days
Show me the environmental impact of different services
Ask about manufacturing services...

- **Lead times** (delivery speed)
- **Material compatibility** (S235JR steel, AL6061 aluminum, etc.)
- **Sustainability requirements**

What manufacturing services are you looking for?

Figure 23: Screenshot of a prototype of the Open-Source-based ACCURATE-Chatbot-Assistant

1.5 Scientific contributions from WP2 about the Matchmaking

To discuss the approach with the dedicated scientific community, it was a strong aim to present the concept as well as intermediary steps of development at conferences. Therefore, different contributions have been made (Table 6), sometimes only based on the work of WP2 or in collaboration with other partners.

Table 6. List of publications with contributions from Work Package 2

Proposed Citation	Status
Schuseil, F., Hertwig, M., Lentes, J., Zimmermann, N., Hölzle, K. (2024). A semantic matchmaking approach to empower human decision-making in Manufacturing-as-a-Service scenarios. In: Tareq Ahram and Waldemar Karwowski (eds) Human Factors in Design, Engineering, and Computing. AHFE	Published (Open Access)

(2024) International Conference. AHFE Open Access, vol 159. AHFE International, USA. http://doi.org/10.54941/ahfe1005751	
Hertwig, M.; Schuseil, F.; Lentes, J.; Borodin, V.; Duran-Mateluna, C.; Dolgui, A.; Thevenin, S. (2025) Ontology-based matchmaking and scheduling for Manufacturing as a Service, Procedia CIRP, Volume 134, 2025, Pages 372-377. https://doi.org/10.1016/j.procir.2025.02.144.	Published (Open Access)
Lentes, J.; Hertwig, M.; Schuseil, F. (2025) Ontology-Based Matchmaking for Manufacturing-as-a-Service, 11th IFAC Conference for Manufacturing Modelling, Management and Control 2025, Trondheim, 30 June to 4 July 2025	Presented at conference, publication still open (Open Access)
Hertwig, M.; Lentes, J.; Komenda, T. (2025) Flexible Wertschöpfung mit MaaS / Flexible Value Creation by use of MaaS. Werkstatttechnik Online (German), 115 (6) 2025. DOI: doi.org/10.37544/1436-4980-2025-06-107	Published
Hertwig, M.; Lentes, J.; Riedel, O. (2025) Manufacturing as a Service – Current Research Work and Future Directions, 28th International Conference on Production Research, Chia, 14 to 17 July 2025	Presented at conference, publication still open (Open Access)
Vathoopan, M.; Boudjadar, J.; Hertwig, M.; Lentes, J. (2025) Ontology-driven Simulations for Quantified Service Discovery in Manufacturing Ecosystems, 2025 29th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)	Presented at conference, publication still open (Open Access)

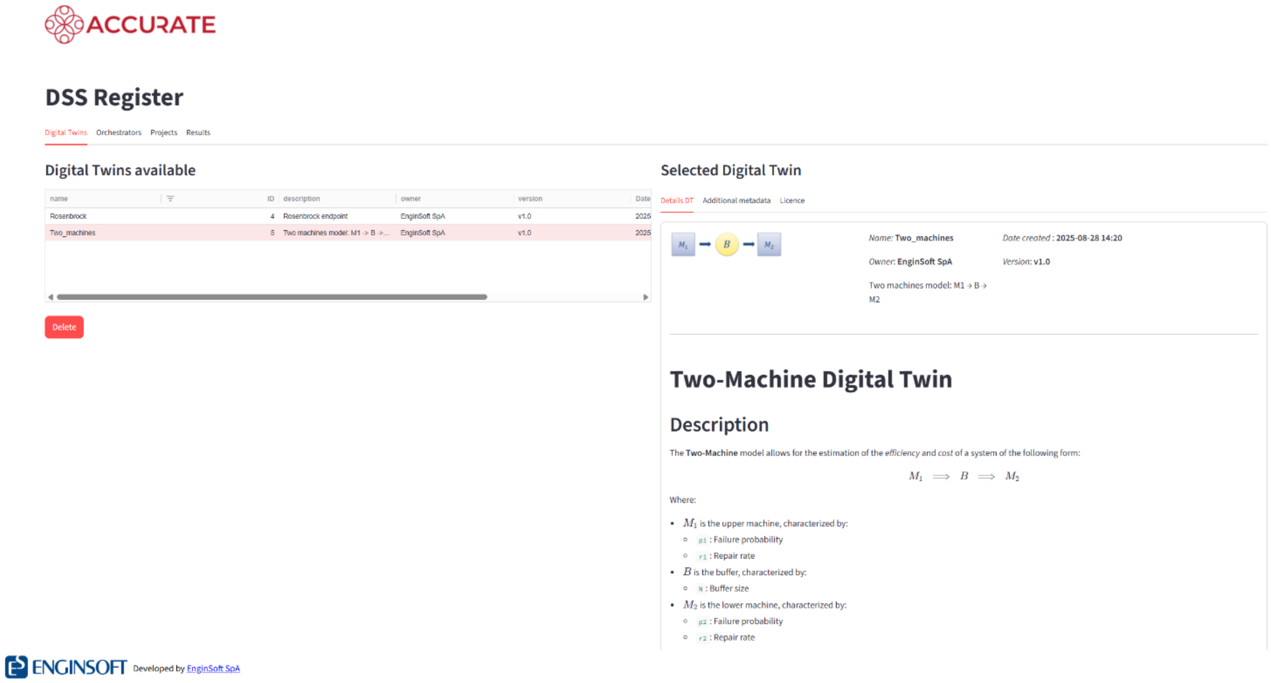
2 Digital Twin Registry

2.1 Introduction

The Register application provides a unified environment for browsing, managing, and documenting Digital Twins (DTs), orchestrators, and their related projects and results. In this context, a Digital Twin is an executable model that mirrors the behaviour of a real-world entity, such as a machine, an entire production line, or a logistics network. To combine and coordinate these models, the system relies on orchestrators. An orchestrator can be seen as a workflow made up of interconnected nodes, where each node may represent a Digital Twin or, more broadly, any computational component exposing its outputs through APIs. By exchanging data through defined input and output ports, orchestrators link multiple DTs into coherent, integrated workflows. From this, projects are built by pairing an orchestrator with a decision support task, such as single or multi-objective optimization or a full factorial design of experiments.

Its main features include:

- **Browsing DTs and nodes** (**Figure 24**)**:** users can navigate through the list of registered DTs and computational nodes, exploring their metadata such as name, owner, and creation date. A tabbed interface allows for a deeper dive into the details, where one can review a full description of the DT or node, inspect all custom metadata defined at creation time, and access licensing details.



DSS Register

Digital Twins available

name	ID	description	owner	version	Date
Rosenbrock	4	Rosenbrock endpoint	EnginSoft SpA	v1.0	2025
Two_machines	5	Two machines model: M1 -> B ->...	EnginSoft SpA	v1.0	2025

Selected Digital Twin

Name: **Two_machines**
Date created : **2025-08-28 14:20**
Owner: **EnginSoft SpA**
Version: **v1.0**
Two machines model: M1 → B → M2

Two-Machine Digital Twin

Description

The **Two-Machine** model allows for the estimation of the *efficiency* and *cost* of a system of the following form:

$$M_1 \implies B \implies M_2$$

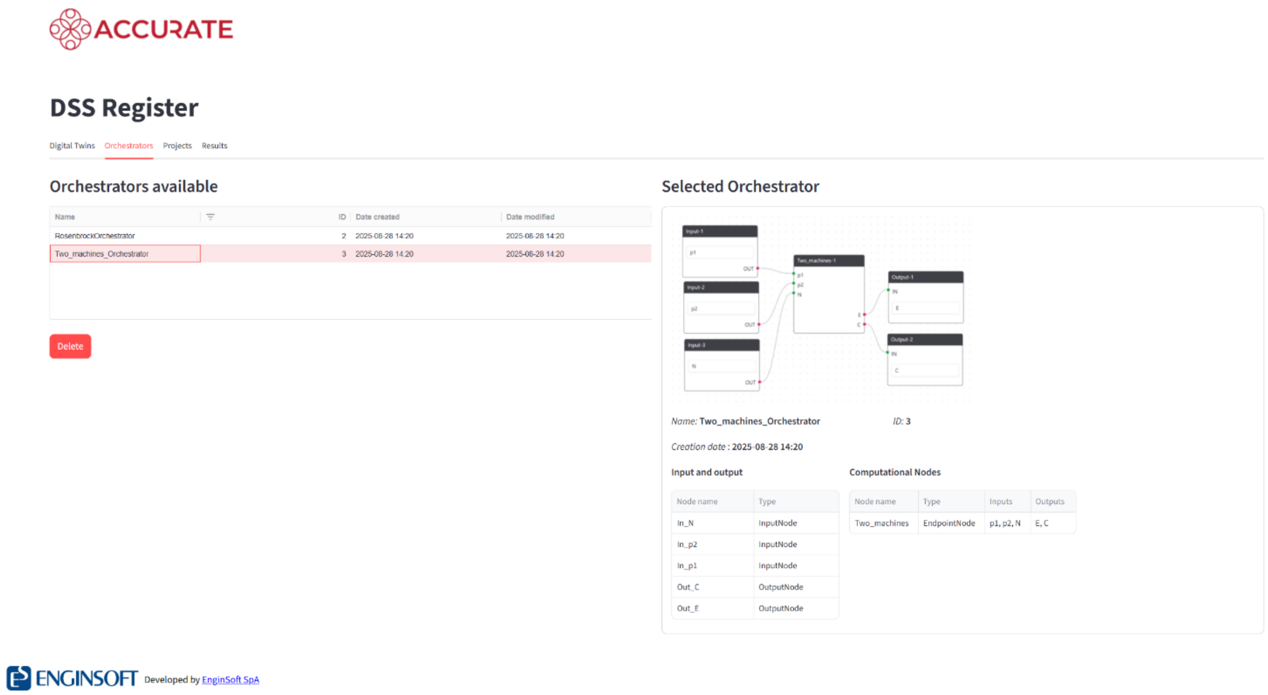
Where:

- M_1 is the upper machine, characterized by:
  - `p1` : Failure probability
  - `r1` : Repair rate
- B is the buffer, characterized by:
  - `N` : Buffer size
- M_2 is the lower machine, characterized by:
  - `p2` : Failure probability
  - `r2` : Repair rate

Developed by EnginSoft SpA

Figure 24: Digital Twin section on Register, showing details of Two-Machine model

- **Exploring orchestrators and projects** (**Figure 25**)**:** users are able to see defined orchestrators (i.e., workflows) and the projects that are built upon them, obtaining a high-level overview of how DTs and nodes are orchestrated.
- **Viewing project results** (**Figure 26**): the Register enables users to inspect the results produced by each project. When a project is selected, its associated output file is displayed as a tabular preview of the corresponding CSV in the right-hand panel, providing a quick and interactive way to review outcomes directly within the interface.



DSS Register

Orchestrators available

Name	ID	Date created	Date modified
RosenbrockOrchestrator	2	2025-08-28 14:20	2025-08-28 14:20
Two_machines_Orchestrator	3	2025-08-28 14:20	2025-08-28 14:20

Selected Orchestrator

Name: Two_machines_Orchestrator *ID:* 3

Creation date : 2025-08-28 14:20

Input and output

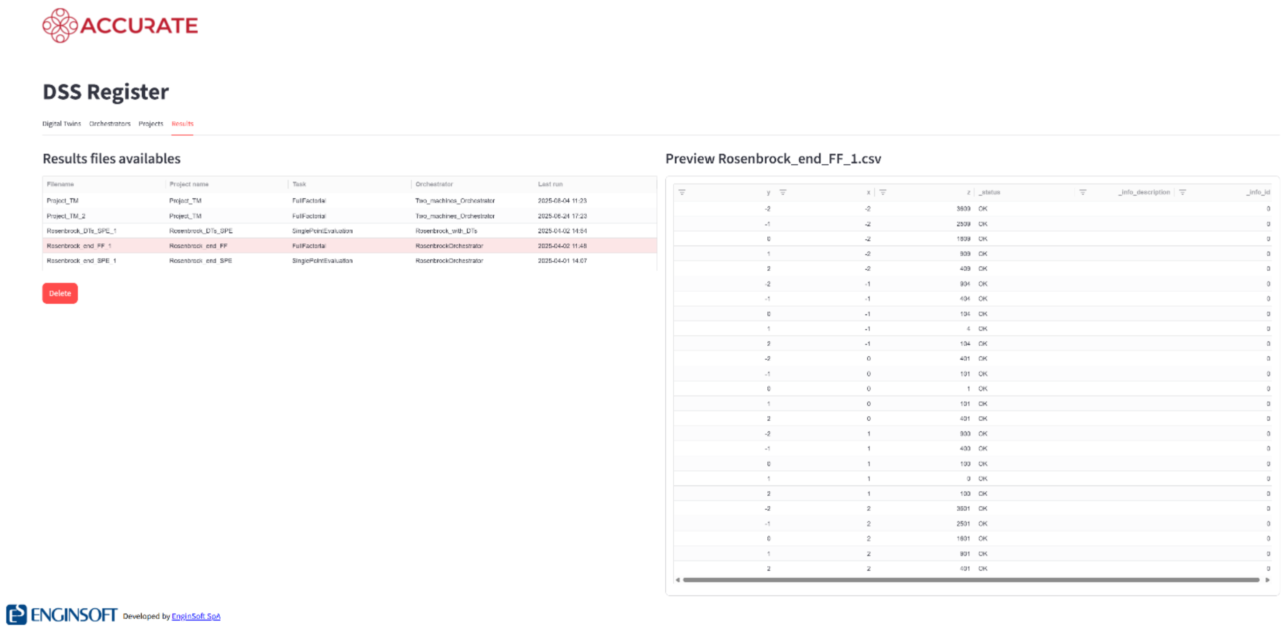
Node name	Type
In_N	InputNode
In_p2	InputNode
In_p1	InputNode
Out_C	OutputNode
Out_E	OutputNode

Computational Nodes

Node name	Type	Inputs	Outputs
Two_machines	EndpointNode	p1, p2, N	E, C

Developed by EnginSoft SpA

Figure 25: Orchestrators section on Register app



DSS Register

Results files availables

Filename	Project name	Task	Orchestrator	Last run
Project_TM	Project_TM	FullFactorial	Two_machines_Orchestrator	2025-08-04 11:23
Project_TM_2	Project_TM	FullFactorial	Two_machines_Orchestrator	2025-08-24 17:23
Rosenbrock_DTs_SPE_1	Rosenbrock_DTs_SPE	SinglePointEvaluation	Rosenbrock_with_DTs	2025-04-02 14:54
Rosenbrock_end_FF_1	Rosenbrock_end_FF	FullFactorial	RosenbrockOrchestrator	2025-04-02 11:48
Rosenbrock_end_SPE_1	Rosenbrock_end_SPE	SinglePointEvaluation	RosenbrockOrchestrator	2025-04-01 14:07

Preview Rosenbrock_end_FF_1.csv

y	x	z	_status	_info_description	_info_id
-2	-2	3609	OK		0
-1	-2	2509	OK		0
0	-2	1609	OK		0
1	-2	909	OK		0
2	-2	409	OK		0
-2	-1	904	OK		0
-1	-1	404	OK		0
0	-1	104	OK		0
1	-1	4	OK		0
2	-1	104	OK		0
-2	0	401	OK		0
-1	0	101	OK		0
0	0	1	OK		0
1	0	101	OK		0
2	0	401	OK		0
-2	1	900	OK		0
-1	1	400	OK		0
0	1	100	OK		0
1	1	0	OK		0
2	1	100	OK		0
-2	2	3601	OK		0
-1	2	2501	OK		0
0	2	1601	OK		0
1	2	901	OK		0
2	2	401	OK		0

Developed by EnginSoft SpA

Figure 26: Results section on Register app

Each model is characterized by inputs, outputs, and facultative expressions (the latter only for computational nodes). Inputs and outputs are enriched with attributes such as default values and admissible ranges. For computational nodes, output variables can be linked to mathematical expressions that reference input variables, expressed through dictionaries mapping each output variable to its defining formula (Figure 5).

Additional information can also be attached to each model. This includes a structured description, which may contain mathematical formulations as well as detailed explanations of input and output variables, and a license statement. Both are stored in markdown format and displayed in dedicated tabs.

Finally, users can define or update custom metadata for a model at any time. These are managed as key–value pairs and extend the descriptive capabilities of the system.

In the current implementation, the Register app stores all information related to Digital Twins, computational nodes, orchestrators, and projects in MongoDB. This design replaces the previous file-based approach and brings several key advantages:

- **Unified storage**: all metadata, model definitions, and auxiliary information are maintained consistently in one database.
- **Flexibility**: MongoDB's document-oriented model naturally supports both structured data (parameters, ranges, configurations) and unstructured content (descriptions, licenses in markdown).
- **Scalability**: the system can seamlessly handle growth in the number of models, orchestrators, and projects without performance degradation.
- **Reliability**: centralized storage ensures data consistency and reduces the risks of redundancy or version mismatches.

This transition makes the backend more robust, extensible, and better suited for future evolution of the Register app.

In future developments, the Register app could support viewing the taxonomy associated with each Digital Twin. Taxonomies, created and managed in a separate DSS module (Digit Twin Builder), could be used in the Register to display model classifications according to predefined categories, domains, or functional characteristics. For example, DTs might be tagged under application domains such as circularity or sustainability, helping users quickly identify models relevant to these areas. Leveraging these taxonomies for filtering would further enable more efficient discovery and selection of relevant models within the system.

2.2 System architecture

The Register app is a standalone module that interacts with a backend server for 'read' and 'write' operations on the database. Figure 27 shows the overall architecture and components that interact with the Register app. On the server side, a dedicated component manages processing and CRUD operations. As already mentioned, these operations are performed on a MongoDB, which acts as the structured, scalable repository for Digital Twin information. The server exposes this functionality through an API, ensuring that both the Register and other DSS modules can access, retrieve, or update DT-related data consistently.

On the client side, the Register app provides a panel-based graphical user interface described above. The Register communicates with the server exclusively through the API, making the interaction modular and extensible. In addition, other DSS modules can connect to the same backend, enabling them to benefit from the shared repository without interfering with the Register's operation.

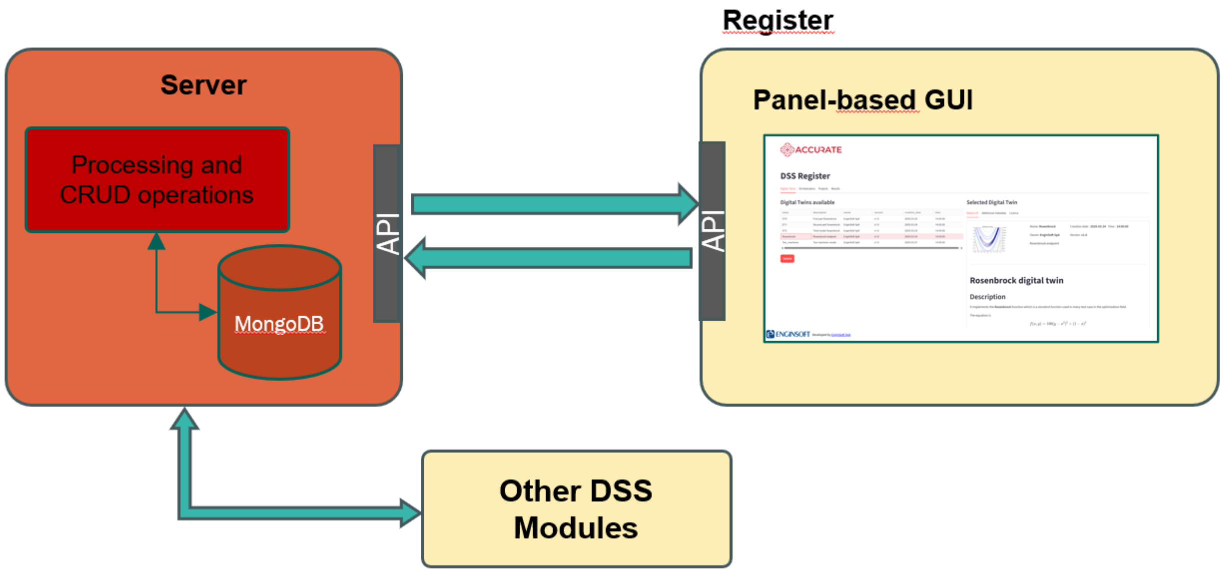
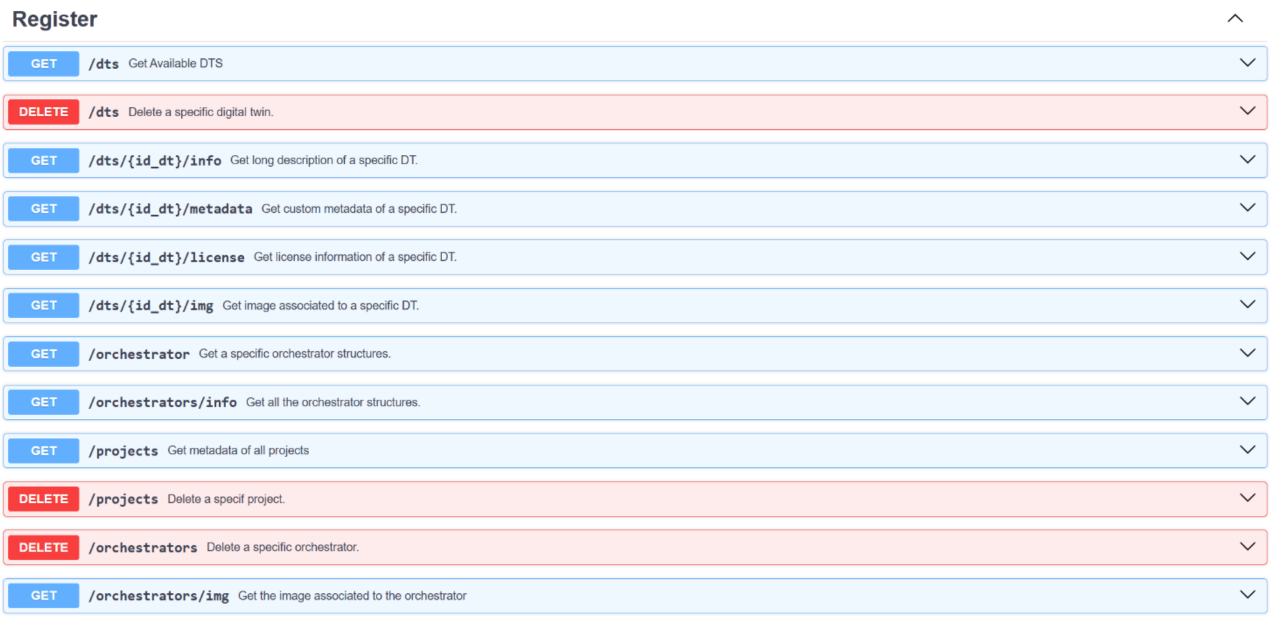


Figure 27: System architecture of the Register application

The Register is deliberately kept separate from the other DSS modules to ensure clear separation of concerns. It allows different ACCURATE Ecosystem actors to use the Decision Support System in ways that match their expertise and business goals. Engineers can focus on producing DTs, while decision makers, operators, or service providers can leverage those DTs to create value and potentially implement innovative business models.

This standalone design brings several advantages. By isolating the Register from the Builder and other DSS modules, each tool can concentrate on its own responsibility—creation on one side and exploration on the other—making development and maintenance simpler and more efficient. Because it communicates with the server exclusively through APIs, the Register can be easily integrated with other DSS modules, ensuring interoperability without tight coupling. Its modularity also makes it reusable across projects and scalable as the number of DTs grows, keeping the browsing functionality reliable even as the ecosystem expands. Finally, treating the Register as a distinct service enhances business opportunities: it enables DTs to be catalogued and shared across different actors, turning the Register into a platform that supports collaboration, value creation, and innovation.

The Register application interacts with a backend server built with the FastAPI Python framework. This server handles all operations required to retrieve information describing Digital Twins, workflows, and projects by querying the underlying MongoDB database. In addition, it exposes APIs that allow users to delete selected elements, including DTs or computational nodes, orchestrators, and projects. All available endpoints are documented and can be directly explored through the Swagger UI, accessible at the endpoint /docs (see **Figure 28**).



Register

GET /dts Get Available DTS

DELETE /dts Delete a specific digital twin.

GET /dts/{id_dt}/info Get long description of a specific DT.

GET /dts/{id_dt}/metadata Get custom metadata of a specific DT.

GET /dts/{id_dt}/license Get license information of a specific DT.

GET /dts/{id_dt}/img Get image associated to a specific DT.

GET /orchestrator Get a specific orchestrator structures.

GET /orchestrators/info Get all the orchestrator structures.

GET /projects Get metadata of all projects

DELETE /projects Delete a specif project.

DELETE /orchestrators Delete a specific orchestrator.

GET /orchestrators/img Get the image associated to the orchestrator

Figure 28: Overview of the Register APIs presented in the Swagger UI

2.3 DTs integrations

Within the ACCURATE ecosystem, Digital Twins are designed to be general-purpose. To ensure their seamless integration with the Register application and all other DSS modules, a predefined set of APIs must be implemented. This chapter provides an overview of these APIs and highlights their key functionalities.

When a user creates a Digital Twin through the template engine, the system automatically generates the set of APIs described below, derived directly from the DT's inputs and outputs. This automation greatly simplifies the development process, ensuring that every new DT is immediately equipped with standardized interfaces for communication and integration. As a result, users can focus on defining DT's logic and behaviour, while the template engine guarantees seamless interoperability with the Register application and the other DSS modules.

The APIs are developed with FastAPI and related Python libraries. They are exposed through an asynchronous REST interface, allowing users to run multiple Digital Twin instances in parallel while keeping client sessions isolated and independent. In addition, the system leverages FastAPI's modern lifespan mechanism to manage server startup and shutdown events asynchronously. This makes it possible to:

- Launch and supervise background tasks, such as periodic cleanup of idle sessions, maintenance routines, and metric tracking loops.
- Ensure a graceful shutdown of these tasks when the server terminates, preventing memory or resource leaks, incomplete operations, and inconsistent session states.

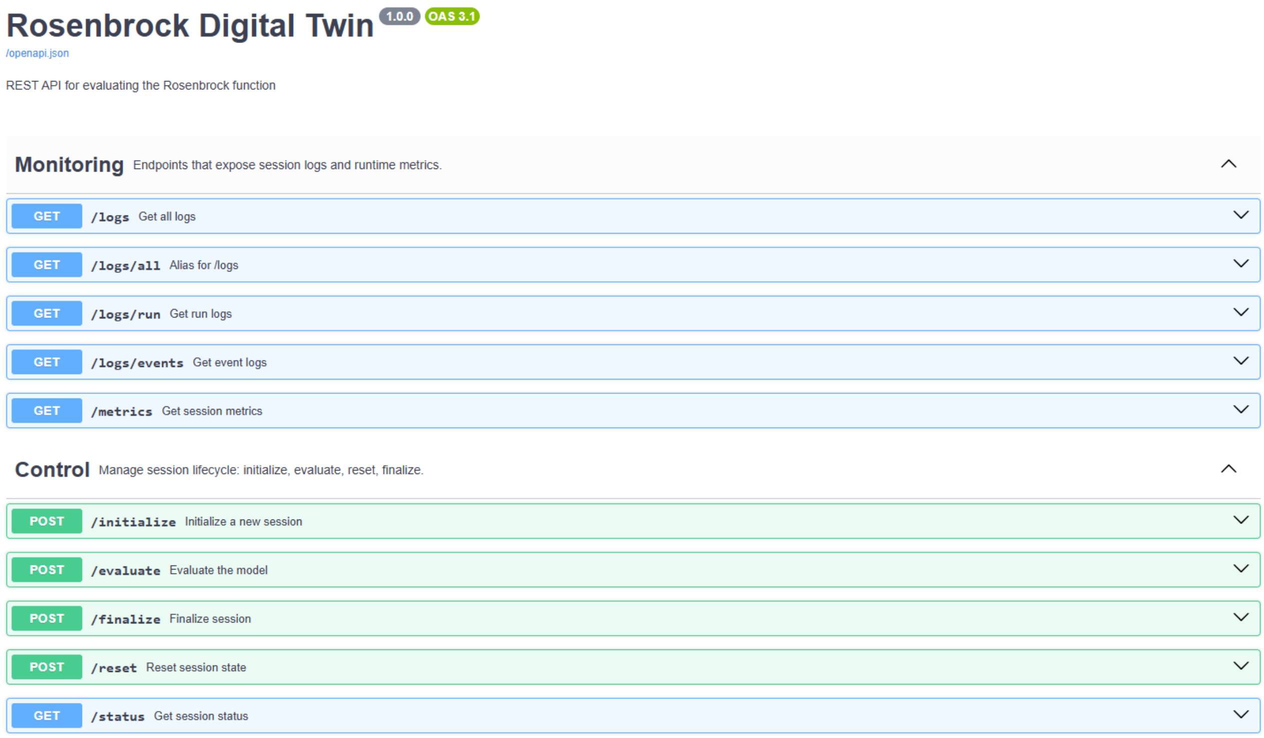
A session can be initialized by using the */initialize* endpoint, this creates and returns to the client a unique id, called session_id, that must be included in all subsequent requests to evaluate the model, retrieve logs, or finalize the session. Sessions are fully isolated, thread-safe, and managed independently to ensure concurrent client interactions do not interfere. Moreover, inactive sessions are automatically cleaned up after a configurable timeout, freeing resources and preventing stale data accumulation.

The system organizes its API into dedicated routes, each grouping related endpoints under a common scope:

- Core routes provide metadata services such as system information, health checks, and schema retrieval.
- Control routes manage the lifecycle of a session, including initialization, reset, finalization, status monitoring, and model evaluation.
- Log routes handle access to logs and runtime metrics.
- Ontology routes expose ontology data and keyword metadata associated with the Digital Twin model.
- Test routes offer static endpoints for validating model implementation and supporting debugging activities.

FastAPI automatically generates OpenAPI-compliant documentation, accessible through the */docs* endpoint with Swagger UI (**Figure** 29). This interactive interface allows developers, integrators, and testers to explore all available endpoints in a clear, human-readable format, review request and response models with their parameters and data types and even execute test calls directly from the browser using the *Try-it-out* functionality.

In the end, this set of APIs not only provides endpoints to manage multiple instances in isolation and run model evaluations but also exposes useful functionalities to the Register application. For example, the Register can verify the availability of the server handling the requests, and access session-level metrics about Digital Twin usage, such as evaluation count, execution duration, and current status.



Rosenbrock Digital Twin 1.0.0 OAS 3.1
/openapi.json

REST API for evaluating the Rosenbrock function

Monitoring Endpoints that expose session logs and runtime metrics.

GET `/logs` Get all logs
GET `/logs/all` Alias for /logs
GET `/logs/run` Get run logs
GET `/logs/events` Get event logs
GET `/metrics` Get session metrics

Control Manage session lifecycle: initialize, evaluate, reset, finalize.

POST `/initialize` Initialize a new session
POST `/evaluate` Evaluate the model
POST `/finalize` Finalize session
POST `/reset` Reset session state
GET `/status` Get session status

Figure 29: Section of the Swagger interface showing the endpoints generated for the Rosenbrock test case

2.3.1 Rosenbrock example

In this section, a more complete overview of the endpoints is presented through a demo Digital Twin model based on the Rosenbrock function, a classic non-convex benchmark widely used to evaluate optimization algorithms. Its mathematical formulation (1) is reported below, where C is a positive coefficient that influences the function's curvature and steepness.

$$f(x, y) = C \cdot (y - x^2)^2 + (1 - x)^2 \tag{1}$$

In this case, the Rosenbrock function is used as a simulated model evaluation within each client session. Clients provide x, y, and an initialization parameter C, and the API computes the corresponding value of f(x,y) in an isolated, state-managed environment.

The input and output schemas—named TwinInput and TwinOutput as defined by the template engine—are shown in Table 7 and Table 8.

Table 7. Input parameters of the Rosenbrock DT model

Field	**Type**	**Description**	**Range**	**Default**
x	float	Variable x ∈ [-10, 10]	[-10.0, 10.0]	1.0
y	float	Variable y ∈ [-10, 10]	[-10.0, 10.0]	2.0

Table 8. Output parameter of the Rosenbrock DT model

Field	**Type**	**Description**	**Range**	**Default**
z	float	Output variable $z = f(x, y)$	-	-

2.3.2 Available endpoints

Core endpoints

These endpoints (Table 9) provide general metadata, health status, and model schema information.

Table 9. Structured representation of core endpoints

Endpoint	**Method**	**Description**
/	GET	Returns a welcome message with a link to API docs.
/info	GET	Returns basic metadata about the API and model.
/health	GET	Lightweight health check to verify server status.
/schema	GET	Returns the full input/output schema definitions (Pydantic models)

/ready?session_id=	GET	Checks if a given session is initialized and ready to evaluate

Session control endpoints

These endpoints (Table 10) manage the full lifecycle of a Digital Twin session—from creation and evaluation to reset and finalization. All operations require a valid session_id to ensure isolation and state consistency.

Table 10. Structured representation of session control endpoints

Endpoint	**Method**	**Description**
/initialize	POST	Creates a new session and sets the model coefficient C. Returns a session_id.
/evaluate	POST	Evaluates the Rosenbrock function using provided x and y values within a session.
/status	GET	Returns the current status and metadata of a given session.
/reset	POST	Resets the session state, clearing all logs and previous evaluations.
/finalize	POST	Finalizes and terminates the session, freeing associated resources.

Logs and metrics endpoints

These endpoints (Table 11) provide access to detailed logs and runtime statistics for each session. Logs are automatically recorded during key lifecycle events and model evaluations. All endpoints require a valid session_id as a query parameter.

Table 11. Structured representation of logs and metrics endpoints

Endpoint	**Method**	**Description**
/logs	GET	Returns the complete session log, including evaluations and lifecycle events.
/logs/all	GET	Alias for /logs; provides full session logs.
/logs/run	GET	Returns only the run logs—i.e.,, model evaluations with inputs and outputs
/logs/events	GET	Returns session event logs (e.g.,, initialization, reset, finalize actions).
/metrics	GET	Returns session-level metrics such as run count, duration, and current status.

Endpoints for ontology and keywords

This module provides descriptive metadata about the Digital Twin model, exposing both a semantic ontology in JSON-LD format and a curated list of domain-specific keywords (Table 12).

Table 12. Structured representation of ontology and keywords

Endpoint	Method	Description
/keywords	GET	Returns a list of human-readable keywords that describe the model's domain and intent.
/ontology	GET	Returns a JSON-LD document describing the model's inputs, outputs, and operations using semantic vocabularies.

Test endpoint

The /test endpoint (Table 13) is designed for debugging and validation purposes. It performs a predefined model evaluation using fixed input values, bypassing session management. This allows developers or external tools to quickly verify that the core model logic is functioning as expected.

Table 13. Structured representation of test endpoints

Endpoint	Method	Description
/test	GET	Executes a hardcoded test using default inputs to evaluate the model without requiring a session. Useful for debugging and health checks

2.3.3 API Workflow

As already mentioned, the Digital Twin API follows a session-based interaction model, enabling clients to simulate and manage independent instances of the model. The typical workflow is outlined below:

1. **Initialize a session**: The client starts a new session by sending a request to /initialize, specifying the
2. model coefficient C.
3. **Session enters READY state**: Once initialized, the session is stored in memory and marked as ready to receive inputs.
4. **Evaluate the model**: The client sends a POST request to /evaluate, passing the session_id along with input values x and y. The API computes the Rosenbrock function.
5. **Access logs and outputs**: All inputs and computed outputs are automatically logged and can be retrieved via the /logs endpoint.
6. **Manage session state**: The session can be explicitly reset (/reset) to clear its state or finalized (/finalize) to terminate and clean up resources.
7. **Automatic cleanup**: Sessions that remain inactive beyond a configured timeout are automatically removed by a background task.

2.3.4 Example API workflow

This section shows a typical interaction with the Digital Twin API, covering session initialization, model evaluation, log inspection, and cleanup. The example assumes that the DT server is running locally on port 8001 and that cURL is used to send requests to the server.

1. Perform a test

   Run a test without initializing a session by calling the /test endpoint. This is useful to verify server reachability and ensure that inputs are processed correctly. The test inputs are predefined within the endpoint implementation, so there is no need to provide them explicitly.

```
curl -X 'GET' \
    'http://localhost:8001/test' \
    -H 'accept: application/json'
```

   Response:

```
{
    "status": "success",
    "init": {
        "C": 100
    },
    "input": {
        "x": 1,
        "y": 2
    },
    "output": {
        "z": 100
    }
}
```

2. Initialize a new session

   Start a session using the /initialize endpoint. This call allows you to set initial variables, such as constants. In this example, the parameter ‘C’ is initialized to 100 and will remain fixed for the entire session.

```
curl -X 'POST' \
    'http://localhost:8001/initialize' \0
    -H 'accept: application/json' \
    -H 'Content-Type: application/json' \
    -d '{"C": 100}'
```

   Response:

```
{
"message": "Model initialized",
"C": 100,
"session_id": "5a10e463-09ac-405f-abb2-27596d522c02"
}
```

3. Perform evaluation

Run the model evaluation passing the inputs and the session_id.

```
curl -X 'POST' \
'http://localhost:8001/evaluate?session_id=5a10e463-09ac-405f-
abb2-27596d522c02' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{"x": 1, "y": 2}'
```

Response:

```
{
    "z": 100
}
```

4. Retrieve logs

Use the endpoint /logs to retrieve all the logs produced by server in a given session.

```
curl -X 'GET' \
    'http://localhost:8001/logs?session_id=5a10e463-09ac-405f-
    abb2-27596d522c02' \
    -H 'accept: application/json'
```

Response:

```
{
    "logs": [
        {
            "start_time": "2025-07-25T09:44:21.016052+00:00",
            "status": "READY",
            "end_time": "2025-07-25T09:44:21.016052+00:00",
            "inputs": null,
            "error": null,
            "message": "Digital twin initialized."
},
{
        "start_time": "2025-07-25T09:45:35.129343+00:00",
        "status": "COMPLETED",
        "end_time": "2025-07-25T09:45:35.129961+00:00",
        "inputs": {
            "x": 1,
            "y": 2
        },
        "error": null,
        "message": null
}
                ]
}
```

5. Finalize the session

   Close the current session by calling the /finalize endpoint with the session_id. If not explicitly finalized, the session will be automatically cleaned up after a timeout.

```
curl -X 'POST' \
     'http://localhost:8001/finalize?session_id=5a10e463-09ac-
     405f-abb2-27596d522c02' \
     -H 'accept: application/json' \
     -d ''
```

   Response:

```
{
     "message": "Digital twin finalized",
     "session_id": "5a10e463-09ac-405f-abb2-27596d522c02"
}
```

2.4 Key differentiators

While many initiatives and platforms exist for cataloguing and managing Digital Twins or computational models, the Register application distinguishes itself through a combination of flexibility, scalability, and business-oriented design. Unlike systems that blend creation, deployment, and exploration within a single environment, the Register deliberately separates these concerns. Engineers can focus entirely on producing and validating DTs, while decision makers, operators, and service providers interact with the catalogue to browse, evaluate, or integrate models without interfering with the development process. This modularity not only reduces complexity and improves maintainability but also ensures that each actor operates in a space aligned with their expertise. This modular and adaptable architecture makes the Register well-suited for extension across diverse domains. Another distinctive feature of the Register is the richness of its descriptive capabilities. Beyond basic meta data such as name or owner, each DT or computational node can be documented in detail, including mathematical formulations, variable definitions, and licensing information, all presented in markdown and accessible through dedicated tabs. The possibility of adding custom metadata as key–value pairs further enhances this flexibility, allowing users to adapt the descriptive layer to domain-specific needs and ensuring the Register remains relevant across diverse contexts.

The Register goes beyond simple cataloguing by supporting and displaying orchestration info. DTs are not only listed but also connected to orchestrators and projects, giving users a high-level view of how computational nodes interact within larger workflows. This contextual layer highlights dependencies, reusability, and the role of DTs in broader processes, making it easier to design efficient solutions and support informed decision-making. Equally important is the Register’s orientation toward business opportunities. By treating it as a standalone, shareable service, the system becomes more than a technical repository: it serves as a platform for collaboration and innovation. DTs catalogued in the Register can be exchanged across different actors in the ecosystem, enabling value creation, and encouraging the emergence of new business models. Its interoperability, supported through APIs, ensures reusability across projects and organizations, avoiding the limitations of siloed systems.

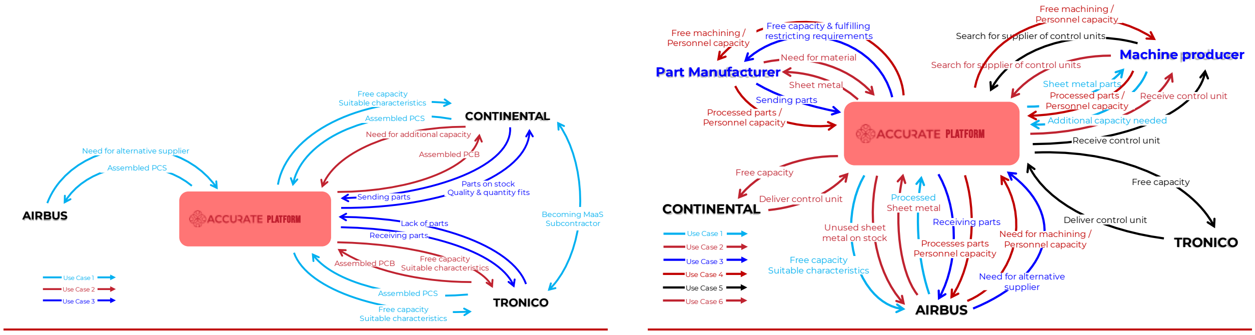
Looking ahead, the Register is also designed to be future-ready. Planned taxonomy integration will enable models to be classified under domains such as circularity or sustainability, facilitating semantic exploration and filtering. This evolution will make it easier for users to identify models relevant to their specific needs and will further strengthen the Register's role as a discovery and decision-support tool.

In the end, the Register is far more than a simple repository. Through its modular architecture, flexible data model, comprehensive descriptive features, and strong business orientation, it provides a robust and forward-looking solution. Where other initiatives often stop at maintaining a static catalogue, the Register adds context, extensibility, and interoperability, transforming it into a strategic enabler for collaboration, scalability, and innovation within the ACCURATE ecosystem.

3 Use Case oriented Testing and validation

Since the functions in chapters 1 (Matchmaking Service) and 2 (

Digital Twin Registry) were developed in accordance with the requirements collected and documented in deliverables D2.1, D3.1, and D4.1, their performance was first checked against the list of requirements. Deliverable D7.1 enables validation against practical use cases. It was possible to derive three project-internal use cases (Figure 30, a) and six project-external use cases (Figure 30, b).



a) b)

Figure 30. Use cases as described in Deliverable D7.1 – base for validation with industrial data (a - project-internal use cases, b - project-external use cases)

Each use case integrates functions developed across WP2, WP3, and WP4, which naturally increases the complexity of validation. The use case description supported a deviation from workflows to support the tool/function individual validation. Each validation requires a specific subset of data at the appropriate level of granularity, complemented by additional information depending on the scope, time horizon, and physical coverage of the functions. Achieving coherence across functions demands the alignment, orchestration, and processing of data flows. The first steps of course is the validation of the individual function. As described above, a subset of data is required. The developers need to share their demands with the pilots to gather the required data. Based on this, the validation of the matchmaking was performed with different volume of data. It was possible to show the matching of different virtual offerings and requests according to the provided data. However more challenging is the combination of multiple function. This is essential for the system/framework to operate as an integrated whole. Based on this the different developers need to synchronize to communicate with the pilots about the data needed. Therefore, each use case, as described in D7.1 will be discussed how the functions of the WPs could contribute. Based on this a synchronized description of the stages of the pilot case must be formulated, including the data handling. Utilizing the BPMN 2.0 methodology, the use cases are translated in process descriptions. In close collaboration and exchange with the pilots the use cases are checked concerning their industrial relevance.

Addressing these risks different approaches (Table 14) have been discussed to be able to offer fully functional individual results from each development work package. Some aspects of the mitigation are already incorporated in the generic validation, as set up during WP2 (See chapter 3.1).

Table 14. Mitigation strategies to address different challenges on the way to

Risk / Challenge	Mitigation strategy	Level of risk	Involved parties
No data from pilots available to validate the function	A stepwise approach has been defined that allows to validate the function with different set and volumes of data.	Low	IAO, ES (technology developers)
Limited data set to be used for demonstration	Negotiation with each pilot (within WP7) was started to define	Medium	CONTI, TRONICO, AIRBUS (Pilots)

	relevant data and potential limitation by organizational rules.		IAO, ES (technology developers)
Data from pilots are not sufficient to validate the function	Define a clear set of required information. Providing a support to the pilots in deriving and collecting this information from their systems and facilities.	Medium	IAO (technology developers)
Matching requires a two-sided perspective (offer & demand). Data from externals of the project are confidential	For the demand side, there are demonstration parts defined that give a reliable based for requirements. For the offering the focus would lay on internal project internal use cases to reduce risks	Medium	CONTI, TRONICO, AIRBUS (Pilots) IAO, ES (technology developers)
Data interchange between functions of WP2, WP3 and WP4 cannot be tested as function not available till end of the project	As different It tools need to exchange data by an API, these are defined. To be able to demonstrate connected functions, a demonstration with data input and export was defined to see different functions in a row	High	IAO, ES, IMT, AU (functions developers)
Approval of data release by pilots is not in intended schedule and functional test might be delayed	Machinery of the research organizations are used as an additional manufacturing service offering to start with an initial set of data.	Medium	IAO, AU, IMT (research organizations)

3.1 Validation approach for Matchmaking functionality

As all pilots are mainly working for customers on the make-to-order principle, the data they are processing is provided by their customers. As the data and information are confidential and intellectual property of the customers, they cannot be used to demonstrate the developed functions in public. Therefore, the validation and demonstration of the functionalities must be performed with non-customer data. By replacing the customer-specific products with synthetic products the challenge will be solved.

The products of the pilot partners Continental and Tronico are electronic components for different industries. This result in two different conditions – Continental has high volume and medium variance products and Tronico has low volume and very high variance products. Different open-source products have been identified as re-placement for the customer specific products. Open-source products have been identified, because all relevant data elements are publicly available but still technically tangible. Four open-source projects have been identified to be used as an example for a customer's part from which the request could be derived. Different requirements have been identified that are relevant to specify the manufacturing service needed. To simply the understanding of the derived requirements, they are clustered in organizational (product documentation, data sets), technical (chemicals, parts specification) knowledge-based (set up, handling) and timely aspects (timeframe) to check fit with the offered MfgS (Figure 31).

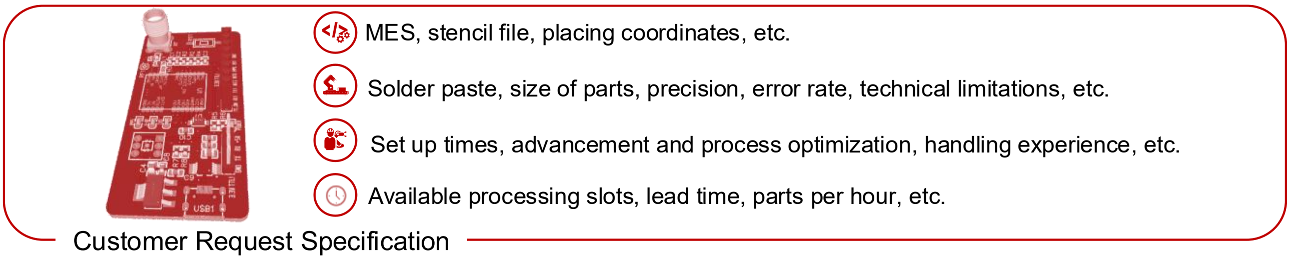

MES, stencil file, placing coordinates, etc.
Solder paste, size of parts, precision, error rate, technical limitations, etc.
Set up times, advancement and process optimization, handling experience, etc.
Available processing slots, lead time, parts per hour, etc.
Customer Request Specification

Figure 31. Insight on requirement categories that are needed to text the functionality

In comparison to that relevant characteristics will be derived based on data and information collected by the pilot partners. As a foundation the IDEF0 diagram (See Deliverable D2.1, pp. 29, or simplified example in Figure 32) is used. It provided a structured break-down of the physical assets that are involved in the different manufacturing process steps that will be essential for providing the MfgS.

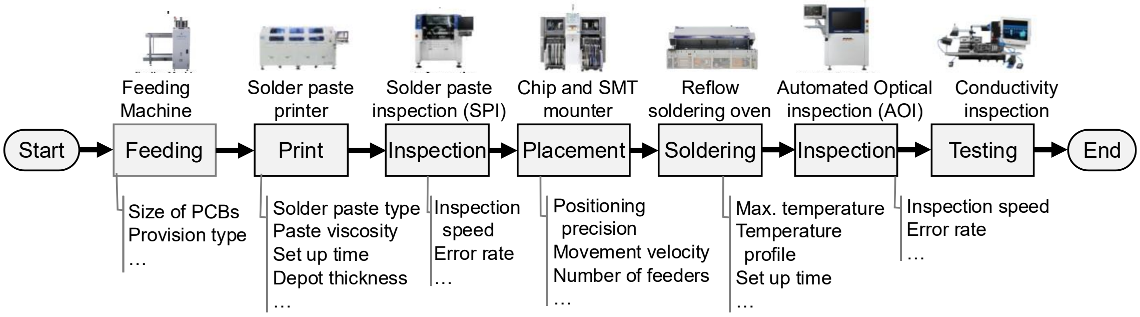
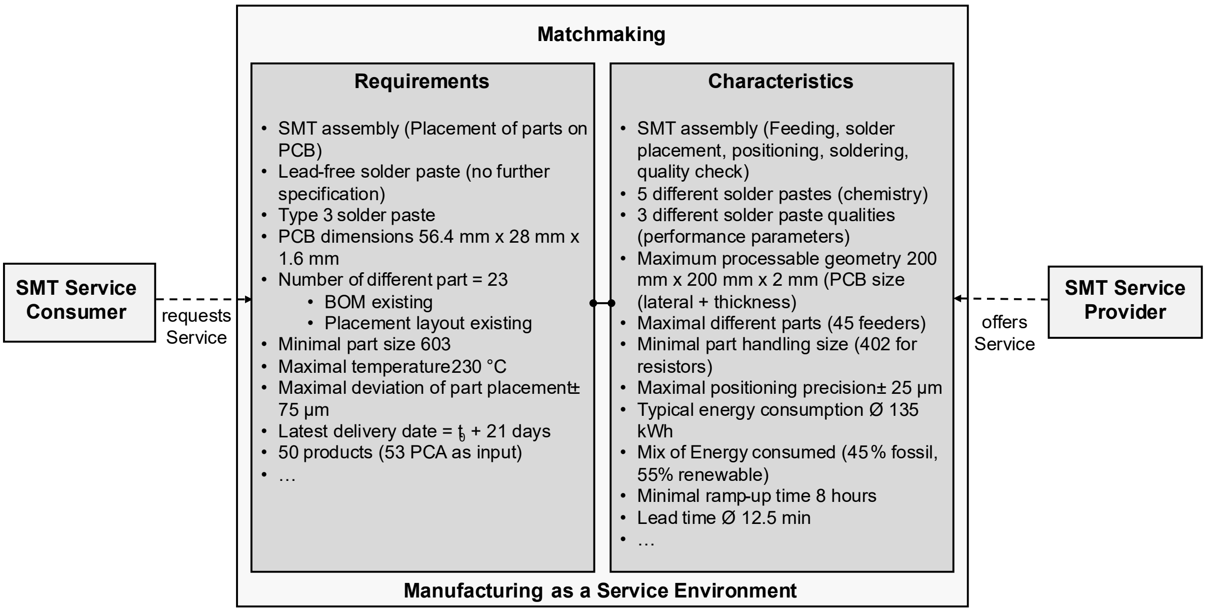


Figure 32. Simplified IDEF0 diagram with specific technical capabilities of involved technical assets

The information collected is used to perform the matchmaking. To check the functionality of the matchmaking functionality a synthetic set of data is derived. Based on information provided by the pilots, the synthetic data set for a SMT assembly service could be presented as an example (Figure 33). It was used for validating matchmaking and finding casualties.



Requirements

- SMT assembly (Placement of parts on PCB)
- Lead-free solder paste (no further specification)
- Type 3 solder paste
- PCB dimensions 56.4 mm x 28 mm x 1.6 mm
- Number of different part = 23
  - BOM existing
  - Placement layout existing
- Minimal part size 603
- Maximal temperature230 °C
- Maximal deviation of part placement± 75 µm
- Latest delivery date = t_0 + 21 days
- 50 products (53 PCA as input)
- …

Characteristics

- SMT assembly (Feeding, solder placement, positioning, soldering, quality check)
- 5 different solder pastes (chemistry)
- 3 different solder paste qualities (performance parameters)
- Maximum processable geometry 200 mm x 200 mm x 2 mm (PCB size (lateral + thickness)
- Maximal different parts (45 feeders)
- Minimal part handling size (402 for resistors)
- Maximal positioning precision± 25 µm
- Typical energy consumption Ø 135 kWh
- Mix of Energy consumed (45% fossil, 55% renewable)
- Minimal ramp-up time 8 hours
- Lead time Ø 12.5 min
- …

Figure 33. Representation of requirements and characteristics to perform matchmaking (matching request & offer)

Of course, the functionalities were validated with synthetic data, it will be transferred to WP7, where pilot data (may be confidential) is used to adjust the Matchmaking service to the needs of the pilots and industrial applications.

3.2 Validation approach for the Digital Twin Registry (DT registry)

In future versions of the Digital Twin Register, access for accredited partners will be validated through the Data-Space infrastructure. This integration will enable fine-grained control over who can view, use, or modify Digital Twins, their information, and metadata, ensuring that sensitive resources remain protected while still being shareable within trusted networks. Partner accreditation data will be stored in the database, allowing the platform to differentiate content and enforce access policies based on user credentials.

4 Conclusion and further steps

This deliverable comprehensively presents the development tasks carried out in WP2. Since Deliverable D2.1 documented the requirements analysis and conceptual aspects in particular, the focus here was more on implementing specific functions that can make a significant contribution to the realisation of Manufacturing as a Service.

The aim of WP2 was to develop a function that allows potential offers (known as manufacturing services) to be matched with requests. The aim is to match technical, organisational, and knowledge-related aspects in terms of compatibility. Comprehensive research into existing concepts and technical fundamentals confirmed the use of semantic modelling. In order to be able to take up and, if necessary, integrate these based on preliminary work and standards, the modular concept outlined in D2.1 was pursued. The already tested capability matching from Jarvenpaa (Figure 7) was further developed (Figure 8). Particularly noteworthy is the fact that not only the characteristics (descriptive properties of the manufacturing service) and requirements (descriptive properties of the request) are taken into account in the model, but also units and other aspects. This allows even inputs of different sizes to be reliably compared with each other. During development, attention was also paid to usability. This is because users of the system are not usually experts in semantic modelling or ontology languages. To use the semantic data model, there is an interface that makes interaction easy. Requests to the semantic data model can be made using either SPARQL or LinkML, with the decision to be made in WP7 during integration.

For validation and initial piloting of the functions, a simple graphical user interface (GUI) was designed, where the matching process can be carried out by the user. The semantic model can be instantiated by importing, for example, a JSON file containing the content.

Besides the Matchmaking approach, the digital twin registry was developed. The objective of the digital twin registry is a collection of descriptive models that might interact with each other. It means that data can be shared after being used or generated by another model. It is also incorporating an orchestration engine that enables the moderation of different data flows through the models. It enables on the one hand the inserting of a mathematical description (as formula for example) or linking digital twins by registration to the registry.

As work package WP2 ends with M22 (30 September 2025), the next step will be to integrate the developed functions (supportive service function) throughout WP6. The ACCURATE project is striving to create a Manufacturing as a Service ecosystem, where manufacturing services can be published and discovered on a marketplace. To make the developed functions utilisable for users in the ecosystem, the functions are to be integrated to also support data exchange between different functions and supportive services. This should create greater added value than individual service function.

In Work Package 7, demonstrations will be implemented for use cases defined and described in Deliverable D7.1 by the pilot partners. The demonstrations are intended to make the functions and the added value they realise understandable. On the one hand, the pilot partners should be able to utilise the added value. In addition, visualising the added value should also generate interest in the functions created among parties outside the project, thereby supporting increased adoption of the Manufacturing as a Service approach. To this end, the outlined validations will be expanded and enriched with demonstration data collected and validated together with pilot partners.

5 References

Ameri, Farhad; Dutta, Debasish (2008): A matchmaking methodology for supply chain deployment in distributed manufacturing environments. In: JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING 8 (1). DOI: 10.1115/1.2830849.

Ameri, Farhad; Patil, Lalit (2012): Digital manufacturing market: A semantic web-based framework for agile supply chain deployment. In: JOURNAL OF INTELLIGENT MANUFACTURING 23 (5), S. 1817–1832. DOI: 10.1007/s10845-010-0495-z.

Balta, Efe C.; Jain, Kshitij; Lin, Yikai; Tilbury, Dawn; Barton, Kira; Mao, Z. Morley: Production as a service: A centralized framework for small batch manufacturing. In: IEEE Conference on Automation Science and Engineering Xi'an Shi (Hg.) – CASE : 2017 13th IEEE, S. 382–389.

Bulut, Serdar; Wende, Martin; Wagner, Christoph; Anderl, Reiner (2021): Impact of Manufacturing-as-a-Service: Business Model Adaption for Enterprises. In: Procedia CIRP 104, S. 1286–1291. DOI: 10.1016/j.procir.2021.11.216.

Cai, M.; Zhang, W. Y.; Zhang, K. (2011): ManuHub: A semantic web system for ontology-based service management in distributed manufacturing environments. In: IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 41 (3), S. 574–582. DOI: 10.1109/TSMCA.2010.2076395.

Delamer, Ivan M.; Lastra, Jose L. Martinez (2006) Ontology modeling of assembly processes and systems using semantic web services. In: 2006 IEEE International Conference on Industrial Informatics, INDIN'06, art. no. 4053458, pp. 611 - 617. DOI: 10.1109/INDIN.2006.275631

Delaram, Jalal; Houshamand, Mahmoud; Ashtiani, Farid; Valilai, Omid Fatahi (2021): A utility-based matching mechanism for stable and optimal resource allocation in cloud manufacturing platforms using deferred acceptance algorithm. In: JOURNAL OF MANUFACTURING SYSTEMS 60, S. 569–584. DOI: 10.1016/j.jmsy.2021.07.012.

Groover, Mikell P. (2020): Fundamentals of modern manufacturing. Materials, processes, and systems. Seventh edition. Hoboken, NJ: John Wiley & Sons, Inc.

Heizer, Jay H.; Render, Barry; Munson, Chuck (2017): Operations management. Sustainability and supply chain management. Twelfth edition, global edition. Boston, Columbus, Indianapolis, New York, San Francisco, Amsterdam, Cape Town, Dubai, London, Madrid: Pearson.

Henzel, Robert; Herzwurm, Georg (2018): Cloud Manufacturing: A state-of-the-art survey of current issues. In: Procedia CIRP 72, S. 947–952. DOI: 10.1016/j.procir.2018.03.055.

Hermann, J.; Rübel, P.; Wagner, A.; Ruskowski, M. (2020): A generic product and resource description to enable capability matchmaking for Production as a service. In: IFAC-PapersOnLine 53. DOI: 10.1016/j.ifacol.2020.12.2826.

Herold DM, Marzantowicz Ł (2023) Supply chain responses to global disruptions and its ripple effects: an institutional complexity perspective. Oper Manag Res 16(4):2213–2224. DOI:10.1007/s12063-023-00404-w

Hertwig, Michael; Schuseil, Frauke; Lentes, Joachim; Borodin, Valeria; Duran-Mateluna, Cristian; Dolgui, Alexandre; Thevenin, Simon (2025) Ontology-based matchmaking and scheduling for Manufacturing as a Service, Procedia CIRP, Volume 134, Pages 372-377. DOI: 10.1016/j.procir.2025.02.144.

Jang, J.; Jeong, B.; Kulvatunyou, B.; Chang, J.; Cho, H. (2008): Discovering and integrating distributed manufacturing services with semantic manufacturing capability profiles. In: INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING 21 (6), S. 631–646. DOI: 10.1080/09511920701350920.

Jarvenpaa, Eeva; Siltala, Niko; Hylli, Otto; Lanz, Minna (2017): Capability matchmaking procedure to support rapid configuration and re-configuration of production systems. In: M. Pellicciari und M. Peruzzini (Hg.): 27TH INTERNATIONAL CONFERENCE ON FLEXIBLE AUTOMATION AND INTELLIGENT MANUFACTURING, FAIM2017, Bd. 11 (Procedia Manufacturing), S. 1053–1060.

Jarvenpaa, Eeva; Siltala, Niko; Hylli, Otto; Lanz, Minna (2018): Product Model ontology and its use in capability-based matchmaking. In: L. Wang (Hg.): 51ST CIRP CONFERENCE ON MANUFACTURING SYSTEMS, Bd. 72. KTH Royal Inst Technol Int Acad Prod Engn (Procedia CIRP), S. 1094–1099.

Jarvenpaa, Eeva; Siltala, Niko; Hylli, Otto; Nylund, Hasse; Lanz, Minna (2023): Semantic rules for capability matchmaking in the context of manufacturing system design and reconfiguration. In: INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING 36 (1), S. 128–154. DOI: 10.1080/0951192X.2022.2081361.

Jirkovský, Václav; Šebek, Ondøej; Kadera, Petr; Burget, Pavel; Knoch, Sönke; Becker, Tilman (2019) Facilitation of domain-specific data models design using semantic Web technologies for manufacturing. In: ACM International Conference Proceeding Series. DOI: 10.1145/3366030.3366111

Katsaliaki K, Galetsi P, Kumar S (2022) Supply chain disruptions and resilience: a major review and future research agenda. Ann Oper Res 319(1):965–1002. DOI:10.1007/s10479-020-03912-1

Klein-Schneider, Hartmut; Beutler, Kai (2013): Werkvertragsunternehmen: Outsourcing auf dem Betriebsgelände. In: WSI 66 (2), S. 144–148. DOI: 10.5771/0342-300x-2013-2-144.

Landolfi, Giuseppe; Bami, Andrea; Izzo, Gabriele; Montini, Elias; Bettoni, Andrea; Vujasinovic, Marko et al. (2018): An Ontology Based Semantic Data Model Supporting A Maas Digital Platform. In: 2018 International Conference on Intelligent Systems (IS): IEEE.

Lartigau, Jorick; Xu, Xiaofei; Nie, Lanshun; Zhan, Dechen (2015): Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm. In: INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 53 (14), S. 4380–4404. DOI: 10.1080/00207543.2015.1005765.

Liu, Min; Shen, Weiming; Hao, Qi; Yan, Junwei; Bai, Li (2012): A fuzzy matchmaking approach for Semantic Web Services with application to collaborative material selection. In: COMPUTERS IN INDUSTRY 63 (3), S. 193–209. DOI: 10.1016/j.compind.2011.10.001.

Liu, Yongkui; Wang, Lihui; Wang, Xi Vincent; Xu, Xun; Jiang, Pingyu (2019): Cloud manufacturing: key issues and future perspectives. In: INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING 32 (9), S. 858–874. DOI: 10.1080/0951192X.2019.1639217.

Nizamis, Alexandros G.; Ioannidis, Dimosthenis K.; Kaklanis, Nikolaos T.; Tzovaras, Dimitrios K. (2018) A Semantic Framework for Agent-based Collaborative Manufacturing Eco-systems. In: IFAC-PapersOnLine 51 (11), pp. 382 - 387. DOI: 10.1016/j.ifacol.2018.08.323

Ozkiziltan D, Hassel A (2020) Humans versus Machines: An Overview of Research on the Effects of Automation of Work. Hertie School of Governance, Berlin

QUDT (2019) QUDT Ontologies Overview. http://www.qudt.org/pages/QUDToverviewPage.html

Ren, Lei; Zhang, Lin; Wang, Lihui; Tao, Fei; Chai, Xudong (2017): Cloud manufacturing: key characteristics and applications. In: INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING 30 (6), S. 501–515. DOI: 10.1080/0951192X.2014.902105.

Schuseil, F., Hertwig, M., Lentes, J., Zimmermann, N., Hölzle, K. (2024). A semantic matchmaking approach to empower human decision-making in Manufacturing-as-a-Service scenarios. In: Tareq Ahram and Waldemar Karwowski (eds) Human Factors in Design, Engineering, and Computing. AHFE (2024) International Conference. AHFE Open Access, vol 159. AHFE International, USA. DOI: 10.54941/ahfe1005751

Shishodia A, Sharma R, Rajesh R, Munim ZH (2023) Supply chain resilience: A review, conceptual framework and future research. IJLM 34(4):879–908. DOI:10.1108/IJLM-03-2021-0169

Siltala, Niko; Jarvenpaa, Eeva; Lanza, Minna (2019): A method to evaluate interface compatibility during production system design and reconfiguration. In: P. Butala, E. Govekar und R. Vrabic (Hg.): 52ND CIRP CONFERENCE ON MANUFACTURING SYSTEMS (CMS), Bd. 81. CIRP (Procedia CIRP), S. 282–287.

Sparr, Kelvin; Drexel, Damian; Hoch, Ralph (2021): Using an Auction-Based System in Cloud Manufacturing for Selecting Manufacturing-as-a-Service Providers. In: A. Dolgui, A. Bernard, D. Lemoine, G. VonCieminski und D. Romero (Hg.): ADVANCES IN PRODUCTION MANAGEMENT SYSTEMS: ARTIFICIAL INTELLIGENCE FOR SUSTAINABLE AND RESILIENT PRODUCTION SYSTEMS, PT V, Bd. 634. Int Fed Informat Proc Working Grp 5 7 Advances Prod Management Syst IMT Atlantique, Campus Nantes Univ Nantes, Centrale Nantes Rennes Business Sch Audecia Business Sch (IFIP Advances in Information and Communication Technology), S. 445–454.

Sun, Zhongliang; Jing, Wuming; Hong, Jun; Huo, Min (2008): Matchmaking of network mold enterprise based on analytical hierarchy process. In: Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University 42 (3), S. 291–294. Online verfügbar unter https://www.scopus.com/inward/record.uri?eid=2-s2.0-41949114309&partnerID=40&md5=cec1b033633cd0d50a38be7eadf64608.

Tao, Fei; Qi, Qinglin (2019): New IT Driven Service-Oriented Smart Manufacturing: Framework and Characteristics. In: IEEE TRANSACTIONS ON SYSTEMS MAN CYBERNETICS-SYSTEMS 49 (1), S. 81–91. DOI: 10.1109/TSMC.2017.2723764.

Tao, Fei; Zhang, Lin; Nee, A.Y.C. (2011): A review of the application of grid technology in manufacturing. In: INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH 49 (13), S. 4119–4155. DOI: 10.1080/00207541003801234.

Tedaldi, G.; Miragliotta, G. (2021): Manufacturing-as-a-Service (MaaS): state-of-the-art of up and running solutions and a framework to assess the level of development of a Cloud Manufacturing platform. In: Summer School "Francesco Turco" – Industrial Systems Engineering 26.

Vennesland, Audun; Man, Johannes Cornelis de; Haro, Peter Halland; Arica, Emrah; Oliveira, Manuel (2019): Towards a semantic matchmaking algorithm for capacity exchange in manufacturing supply chains. In: IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management 2. DOI: 10.5220/0008364404660472.

Wilson HJ, Daugherty PR (2018) Collaborative Intelligence: Humans and AI Are Joining Forces. Humans and machines can enhance each other’s strengths. How Humans and AI Are Working Together in 1,500 Companies. Harvard Business Review (July-August):114–123

Ye, Liu; Ming-Hua, Shi (2010) Deploying networked machining service chain based on semantic web. In: Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application, art. no. 6141556, pp. 141 - 145. DOI: 10.1109/ICCIA.2010.6141556

Yu'an, He; Dongqi, Wu; Tao, Yu (2009): Research on Service Discovery and Matching Based on Ontology and Service Capabilities in Manufacturing Grid. In: 2009 WRI World Congress on Computer Science and Information Engineering: IEEE.

Zhang, Lei; Ma, Jinping; Zhang, Song (2007): Automatic discovery of semantic manufacturing grid services. In: 3rd International Conference on Semantics, Knowledge, and Grid, SKG 2007. DOI: 10.1109/SKG.2007.101.

Zhang, Wenyu; Guo, Shanshan; Zhang, Shuai (2015): Personalized manufacturing service recommendation using semantics-based collaborative filtering. In: CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS 23 (2), S. 166–179. DOI: 10.1177/1063293X15578034.

Zhao, Yuanyuan; Liu, Quan; Xu, Wenjun (2017): Open Industrial Knowledge Graph Development for Intelligent Manufacturing Service Matchmaking. In: Proceedings - 2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, ICIICII 2017 2017-December. DOI: 10.1109/ICIICII.2017.36.