



LeArning and robuSt deciSlon Support systems for agile mANufacTuring environments

Project Acronym:

ASSISTANT

Grant agreement no: 101000165

Deliverable no. and title	D6.1 - Data Fabric Requirements and Specification	
Work package	WP 6	Secure and intelligent data fabric
Task	T 6.1	Data fabric requirements assessment and specification.
Subtasks involved		
Lead contractor	Institut Mines-Telecom (IMT) Alexandre Dolgui, mailto : alexandre.dolgui@imt-atlantique.fr	
Deliverable responsible	Biti Innovations AB (BITI) P-O Östberg, mailto: p-o.ostberg@biti.se	
Version number	v1.3	
Date	24/06/2021	
Status	Final version	
Dissemination level	Public (PU)	

Copyright: ASSISTANT Project Consortium, 2021

Authors

Participant no.	Part. short name	Author name	Chapter(s)
1	IMT	Institut Mines-Telecom	
2	UCC	University College Cork	
3	LMS	University of Patras - Laboratory for Manufacturing Systems and Automation	
4	FLM	Flanders Make vzw	
5	TUM	Technical University of Munich	
6	BITI	Biti Innovations AB	
7	SAG	SIEMENS AG	
8	INTRA	INTRA SOFT	
9	AC	Atlas Copco	
10	SE	SIEMENS Energy	
11	PSA	Groupe PSA	

Document History

Version	Date	Author name	Reason
v0.1	08.02.2021	William Viktorsson	Initial Template
v0.2	22.02.2021	William Viktorsson	Added various notes where they may apply to help future editors. Please remove irrelevant notes, add clarity where needed, etc.
v0.3	2021-03-17	P-O Östberg	Added initial versions of introduction
v0.4	2021-03-27	P-O Östberg	Internal data fabric requirements
v0.5	2021-03-30	P-O Östberg	Executive summary
v0.6	2021-04-12	P-O Östberg	Intro and overview
v0.7	2021-04-14	Thomas Krause	Siemens Energy requirements added
v0.8	2021-04-20	P-O Östberg	Restructured & updated requirements section
v0.9	2021-04-14	Sarah Wagner	Digital twin for process planning requirements added
v1.0	2021-05-21	P-O Östberg	Reorganizations based on first round of internal reviewer feedback
v1.1	2021-06-14	P-O Östberg	Rework based on the second round of internal reviewer feedback
v1.2	2021-06-22	P-O Östberg	Finalization of the document
v1.3	2021-06-22	Félicien BARHEBWA MUSHAMUKA	Final editing and preparation for submission

Publishable Executive Summary

This report, deliverable D6.1 of the ASSISTANT project, outlines the context and requirements of the work of work package 6, and its primary outcome: *the ASSISTANT data fabric*. The ASSISTANT data fabric is a foundational data management system designed to meet the advanced data management and provisioning needs of the digital twins and tools developed in the project. Building on a state-of-the-art service-based architecture, the data fabric is designed as a scalable distributed system capable of self-orchestration, integration with a wide range of tools, and management of multiple types of data in diverse formats.

The data fabric will be delivered in four steps: First, this document positions the data fabric by outlining the project data storage and management needs and deriving functional and non-functional requirements from these. Second, in deliverable D6.2 (due M12), a technical architecture for the data fabric platform and its integration with other tools will be presented. After these, a prototype implementation of the data fabric architecture will be delivered in two steps - a preliminary version in Deliverable D6.3 (due M24) and a refined version in Deliverable D6.4 (due M36). To facilitate adoption of project results, deliverables D6.3 and D6.4 will be delivered complete with software, documentation, data sets, and integration examples, and will also (in particular Deliverable D6.4) report on other related project results, including, e.g., research results, scientific publications, demonstration scenarios, and other project activities.

About the ASSISTANT Project

The ASSISTANT project is a European Horizon 2020 Research and Innovation project developing AI-based models and representations of systems and components for adaptive manufacturing. Funded with a total budget of ca 6 M Euro and a project consortium composed of 12 partners from 7 European countries, the project aims to develop next generation artificial intelligence tools and techniques to improve the cost efficiency, production performance, and flexibility of the European manufacturing industry.

The ASSISTANT work is organized in nine work packages spanning from project management to technical development and integration of project results in real world use cases. This report outlines the context of the work in work package 6, identifies requirements (derived from the project use cases and other technical work, in particular the digital twins of work packages 3 - 5), and leads into key technical design decisions (further described in deliverable D6.2, which details the data fabric architecture).

Table of contents

1. Introduction.....	8
1.1 Objective and scope of the document	8
1.2 The ASSISTANT Data Management Strategy	9
1.3 Digital Twins and AI-based Simulation and Control Functions	9
1.4 Data Fabric Definition	10
1.5 Domain Models and Data Modeling for Integration	10
2. Current State of Data Fabric Development.....	13
2.1 State of the Art in Research and Technology	13
2.1.1 Architecture and Capabilities.....	13
2.1.2 Domain Models and Data Representations	14
2.1.3 Visualization and Platform Interoperability.....	14
2.2 State of Current Practice in Manufacturing Industries.....	17
2.3 Observed Research / Industry Practice Gap.....	17
3. Use Cases.....	18
3.1.1 Atlas Copco Use Cases	18
3.1.2 PSA Use Cases	19
3.1.3 Siemens Energy Use Cases.....	20
4. Requirements	20
4.1 Digital Twin and AI Tool Requirements	21
4.1.1 Process Planning Intelligent Digital Twin.....	21
4.1.2 Production Planning and Scheduling Digital Twin	22
4.1.3 Digital Twin for Reconfigurable Manufacturing Execution	22
4.2 Industrial use cases requirements	23
4.2.1 Atlas Copco	23
4.2.2 PSA.....	25
4.2.3 Siemens Energy	28
4.3 Visualization requirements	31
4.4 Platform interoperability requirements	32
4.4.1 Information interoperability	32
4.4.2 Technical interoperability.....	32
4.5 Summary Data Fabric Requirements	32
4.5.1 Data Storage, Management, and Provisioning	33
4.5.2 Metadata Support.....	33
4.5.3 Services, Interfaces, and Customization Points	34
4.5.4 Monitoring and Logging	34
4.5.5 Security	35
4.5.6 Validation.....	35
5. Data Fabric Usage Scenarios.....	35
5.1 Example Scenario	36
6. Requirements Validation	37
7. Conclusion.....	38
8. References	39

9. Appendix	40
9.1 Abbreviations	40

List of Figures

Figure 1: The data fabric as a communication substrate and integration base for digital twins	9
Figure 2: Overview of the ASSISTANT manufacturing processes and digital twin time scopes	10
Figure 3: Overview of the mapping of digital twins, data models, and data fabric	11
Figure 4: Overview of the mappings of domain models and data fabric	12
Figure 5: Overview of the ASSISTANT domain model element hierarchies	13
Figure 6: MVC pattern	15
Figure 7: Typical GUI application action flow	15
Figure 8: MVC and SOA	16
Figure 9: Life cycle of the AI system in an Atlas Copco use case	24
Figure 10: Example of a machining process and the measurements performed	24
Figure 11: Overview of production cell connectivity	25
Figure 12: Data flow of production with specific solution to collect the data surrounded by red components without data flow	26
Figure 13: A Data models are necessary for each component independently of the provider.	26
Figure 14: Data fabric requirements business process overview for the Siemens Energy use case	28
Figure 15: Simulation inputs and outputs for the Siemens Energy use case	29
Figure 16: Java-based data preprocessing for the Siemens Energy use case	30
Figure 17: Preparation of simulation input data for the Siemens Energy use case	30
Figure 18: Summary of data fabric requirements for the Siemens Energy use case	31
Figure 19: Scenario illustration - AI life cycle integration with the data fabric	36

List of Tables

Table 1: Overview of desired functionality for Stellantis (PSA).....	27
Table 2: Data storage, management, and provisioning	33
Table 3: Metadata support	33
Table 4: Services, interfaces, and customization points	34
Table 5: Monitoring and logging.....	34
Table 6: Security.....	35
Table 7: Validation	35
Table 8: requirements validation	37
Table 9: Abbreviations	40

1. Introduction

1.1 Objective and scope of the document

When developing complex AI-based models and representations of systems and components such as the digital twins targeted in the ASSISTANT project, tools, and mechanisms for simplifying and integrating data management, as well as dealing with complex data requirements, e.g., abstraction of heterogeneous data formats, scaling of storage and processing capabilities, etc., are needed. The ASSISTANT data fabric is an information architecture and data management system designed to meet the advanced data storage and provisioning needs of the tools and twins to be developed and aims to provide a unified architecture for data storage, access, and provisioning in manufacturing environments. To enable seamless integration in complex system environments combining legacy systems with state-of-the-art AI tools, as well as to support performance scaling and deployment in mixed (on-premises, edge, cloud) resource environments, the data fabric is designed as a layered architecture of services to be developed as a containerized distributed system. In summary, the high-level objectives of the data fabric are:

- To establish a unified architecture for data management and provisioning for AI-based digital twins for adaptive manufacturing
- To develop uniform data models and tools for domain-level data exchange and communication among the developed digital twins
- To facilitate seamless integration and flexible deployment of platform-independent data management and provisioning mechanisms

As such, the envisioned end goal of use of data fabric systems is treatment of data as utilities - enabling the developed digital twins to combine real world data with advanced simulations (to, e.g., make predictions and evaluate planning strategies for complex manufacturing operations) without detailed knowledge of the representation, storage, or location of the data. Furthermore, to decouple the digital twins from the underlying infrastructure used to host the data, the data fabric will be developed as a self-managing system capable of abstracting the storage and processing of data on different types of (cloud, edge, and on-premises) computing resources.

The purpose of this document is to position the work within the project and to document the requirements identified for the development of the data fabric. To avoid duplication of information and repetition, requirements related to the digital twins developed in work packages 3 - 5 are documented in their respective requirements deliverables (D3.1, D4.1, and D5.1), this report focuses on the requirements of the data fabric, in particular the data management capabilities identified in the project use cases and the technical work packages (WP 3, 4, 5, and 7).

The remainder of this report is structured as follows: First, the remainder of this section outlines the ASSISTANT data management strategy, positions the data fabric work, and illustrates how the envisioned data fabric architecture contributes towards integration of the digital twins and the AI based tools developed. For clarity: this document focuses on the roles of the data fabric systems in conjunction with other project developed or used tools, a more complete overview of the overall ASSISTANT architecture (including its ethical by design approach and focus) will be presented in Deliverable D2.1, and the technical realization architecture of the data fabric will be further specified in Deliverable D6.2. After this, to position the work, a brief survey of the current state of the art is presented in Section 2

followed by an overview of the ASSISTANT project use cases (from the perspective of the data fabric) in Section 3. This aims to provide context and illustrate the envisioned use of the project data fabric-related results. After this, Section 4 details the project requirements identified for the development of the data fabric, providing formal definitions of required functionality and capabilities that also serve as templates for validation of results, which is then summarized in high-level formal requirements towards the end of the section. To provide context for validation and demonstration, Section 5 then outlines a scenario used for use of project data fabric results, and Section 6 discusses validation of project results in this context. Scenarios are based on workflows, data, and problems from the project use cases, and are designed to be representative of real-world adoption settings. Finally, the report provides conclusions in Section 7 and is supplemented with references and a dictionary of commonly used terms and definitions in an appendix.

1.2 The ASSISTANT Data Management Strategy

The ASSISTANT data management strategy is based on consolidation of the data needs of the project and provisioning of a foundational system that meets most (if not all) data needs of the project - a data fabric. The data fabric constitutes an information architecture and platform for data management and data-level integration, and provides interfaces, APIs, and services for integration and data-level communication of the produced systems. From a systems level perspective, the data fabric can be seen as a communication substrate (e.g., a bus) that provides a unified mechanism for data access and manipulation to all tools in the project. As illustrated in **Erreur ! Source du renvoi introuvable.**, this perspective sees the data fabric as a communication and data management-oriented system, but as outlined below the data fabric is also intended to house additional capabilities for self-management and (limited) contextual data processing.

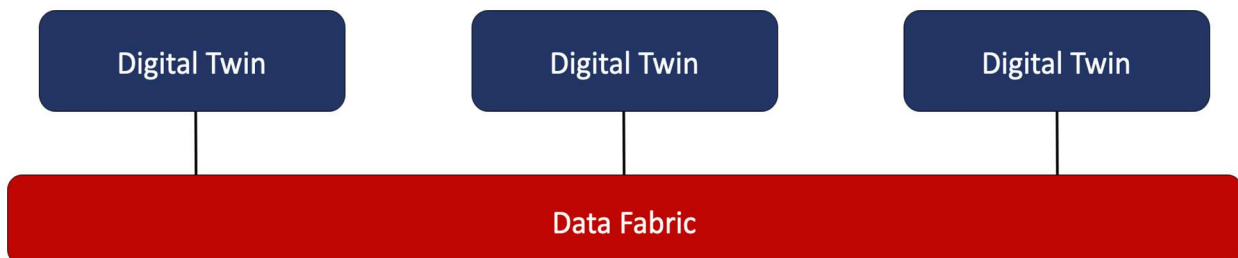


Figure 1: The data fabric as a communication substrate and integration base for digital twins

1.3 Digital Twins and AI-based Simulation and Control Functions

A key research goal of the ASSISTANT project is the production of a set of digital twins: AI and simulation-based systems that use domain knowledge model systems and problems in simulation and use real world data connections and AI tools to provide digital replicas capable of evaluating system configuration strategies and optimizing cyber-physical systems. As the target context is (AI systems for) adaptive manufacturing, the digital twins will (as illustrated in **Erreur ! Source du renvoi introuvable.**) provide capabilities for AI-and simulation-based planning and control functions, forming digital models, digital shadows, and digital twins at different time scales: process planning, production planning, scheduling, and real time control. As these tools operate in different time scales, they also naturally each define their own data requirements and needs, as well as workflows for tools usage and integration.

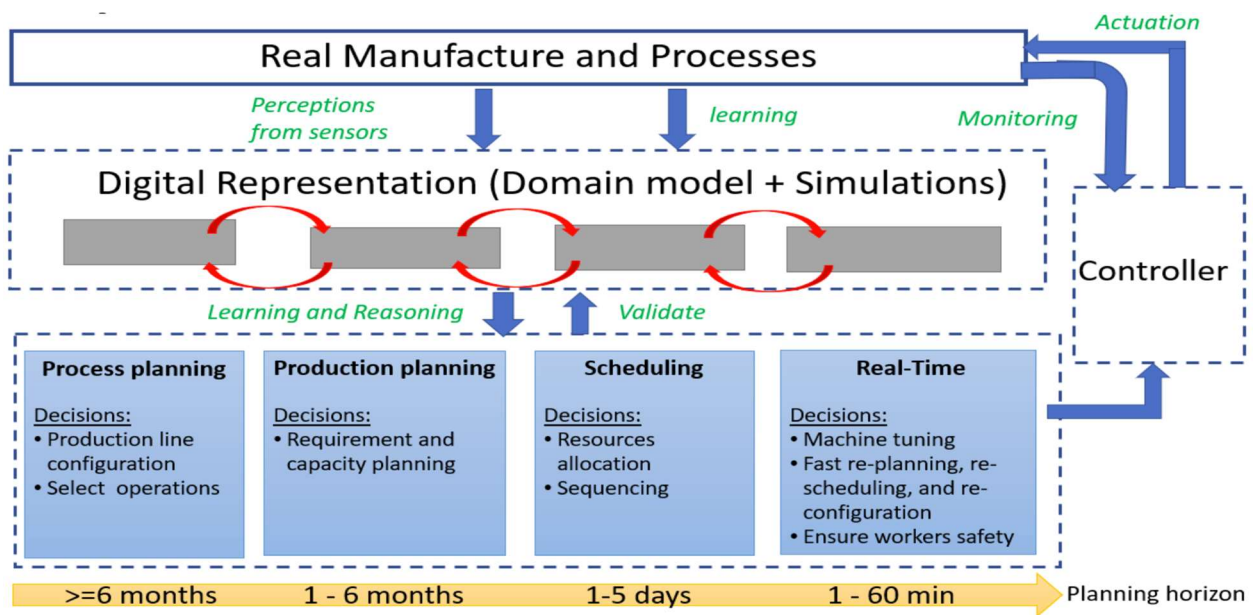


Figure 2: Overview of the ASSISTANT manufacturing processes and digital twin time scopes.

1.4 Data Fabric Definition

A data fabric is a system that provides a unified architecture for management and provisioning of data. To promote flexibility and scalability, data fabrics are typically realized as service-oriented distributed systems where (sets of) services provide consistent interfaces to, and mechanisms for, access to data and storage capabilities. The distributed nature of data fabrics allows scaling, flexible deployment, and adaptation of systems, and is often leveraged to, e.g., facilitate integration of systems across organizational boundaries or combine use of on-premises and cloud-based resources. While primarily designed as substrates for data management and system-to-system communication, data fabrics can also expose interfaces and tools to end-users to facilitate development of mechanisms for convenient management, search, and analysis of data.

The technical realization of the data fabric architecture will be documented in Deliverable D6.2, to provide context for the reader of this document the remainder of this section outlines the data fabrics use and integration with domain models and built-in metadata structures for, e.g., data markup, data curation, search, and data federation.

1.5 Domain Models and Data Modeling for Integration

In ASSISTANT, the domain model (knowledge graph) is the harbor of all knowledge related to your manufacturing process, which is an historical digital twin of your manufacturing system. It is the central point of information for the “historical digital twin” of the production system. The knowledge graph is a conceptual model of a knowledge domain, in this case your product’s design and its manufacturing process. Domain experts use such a knowledge graph to describe and solve problems related to the domain, using its real-world concepts, vocabulary, and relationships between these concepts. These real-world concepts are not directly available from the existing information such as databases and their associated database schema’s. For example, a domain concept “rotor” may have several different representations in different database tables, and all information related to the single concept “rotor” may span multiple tables. Moreover, a lot of information does not necessarily link to data in a database, e.g., experiments, simulation models, obtained insights from a data analysis, uncertainties on these

concepts, etc. For this reason, a domain model must be used as the basis for the knowledge graph, which correctly represents the problem domain, and allows reasoning in term of this problem domain. This domain model must then be linked to the associated data in data Fabric.

The role of domain models and digital twins is shown in the **Erreur ! Source du renvoi introuvable.** below. The figure shows the ASSISTANT digital twin reference architecture and its envisioned application in an industrial context.

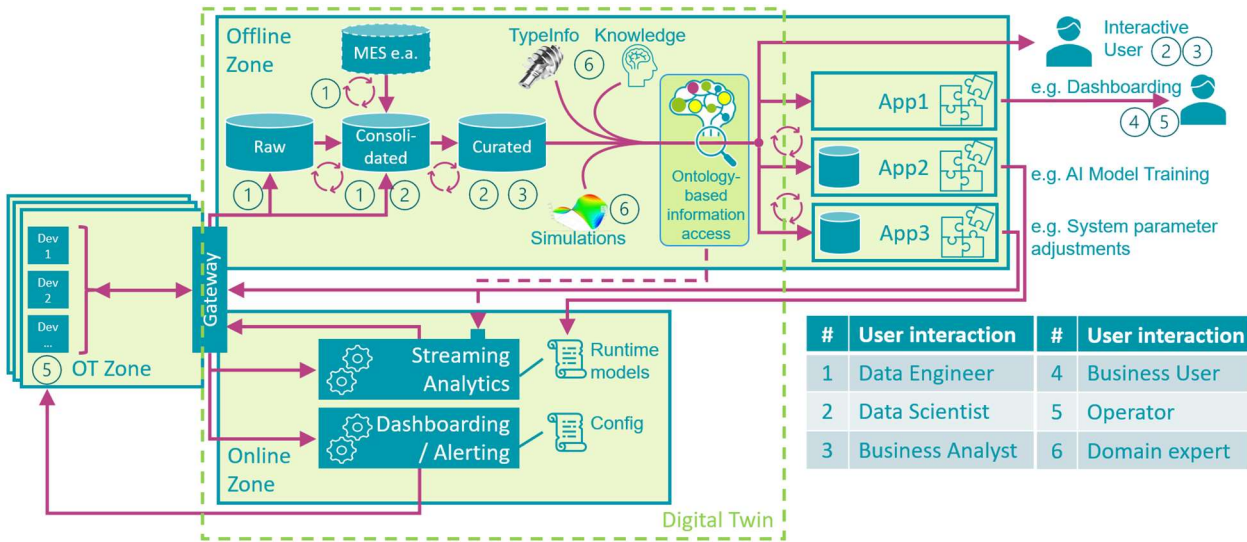


Figure 3: Overview of the mapping of digital twins, data models, and data fabric

The operational technologies (sensors, SCADA, etc.) are shown on the left side. The recorded data (automatically, or by an operator) can be sent over a gateway to an offline zone and/or an online zone. “Offline” refers to the use of historical data, “online” refers to the use of streaming data.

In the offline zone (top), raw data is stored. By means of data warehousing techniques (e.g., Extract-Transform-Load (ETL) procedures), the data is consolidated by data engineers, e.g., to make sure that the data recorded throughout the production process can be traced back to each product. This typically involves information from external systems like a manufacturing execution system (MES). In the curated data zone, the data is organized and stored in such a way that it is optimally usable by applications. The interface to this data is by means of a domain model. In ASSISTANT, the scope of this domain model exceeds data, and includes type information (e.g., dimensions, tolerances, etc. of the products, machines), simulation data (e.g., predicting performance metrics), expert knowledge (e.g., correlations between physical entities, uncertainties that were measured in a data analysis campaign, etc.). Users such as data scientists and business analysts can obtain information by querying the domain model in terms of their knowledge domain, rather than the technical domain. Applications that need to use data and information (e.g., dashboarding, AI model training, etc.) can be created by extracting data and information through the domain model. Results of these applications can be fed back to the domain model or to the OT zone.

In the online zone (bottom), dashboarding or streaming analytics applications can be deployed, that use streaming data during their execution. Such applications may be the result of an AI model training phase in the offline zone.

The domain model requires implementation in such a way that:

- data and information can be retrieved through querying the domain model.
- data and information may be stored in different locations and formats.
- the implementation overhead for adding new information is kept to a minimum.
- evolution of data and information is supported.

- domain knowledge and uncertainty can be modeled (and queried)
- it is modular and can be governed, reused, and extended by multiple stakeholders.

As shown in the **Erreur ! Source du renvoi introuvable.**, the digital twin spans the offline and the online zone. In WP3-4, an offline digital will be used, in WP5, and offline and online digital twin will be used.

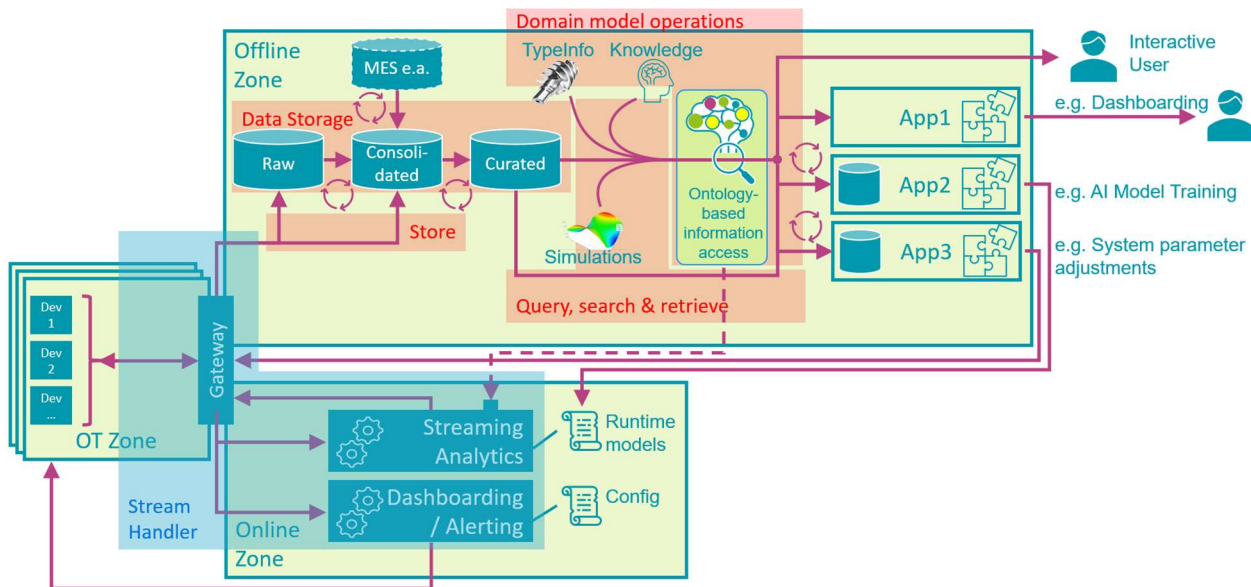


Figure 4: Overview of the mappings of domain models and data fabric

The layered architecture of Section 1.5 is mapped on the **Erreur ! Source du renvoi introuvable.** The red markings represent data fabric functionality, showing how the data fabric will be used to realize this digital twin architecture. The data fabric layers are shown from left to right. Note that the Control Plane layer is not shown on the figure. The intended interface is the domain model, but the lower layer query, search and retrieve can be used as well, thus bypassing the domain model.

The Streamhandler platform (provided by INTRASOFT) is mapped to the architecture as well and serves as (1) a gateway for dealing with streaming data, e.g., monitoring and instrumentation data from production cells and equipment, and (2) integration and deployment of online applications, such as analytics, dashboards, or ERP/MES systems.

Digital twins will be created for each technical work package (WP3 - 5), and for each industrial use case. Nevertheless, they are related through their domain models. A hypothetical illustrative example of such related domain models is shown in the **Erreur ! Source du renvoi introuvable.** The figure provides an overview of all (types of) domain models that will be created.

The domain model represents all concepts and their relationships in the problem domain. In the context of ASSISTANT, these elements can be linked to their data. Some of these concepts (modeled as classes) are generic for the domain of production (grey), and others are specific to process planning (yellow), scheduling (red), real-time control (blue). Furthermore, some concepts are specific to the particular use case (white, purple, green, orange). In the simplified and incomplete domain model above, named relations are defined between some classes (e.g., a Resource produces some Measurements - e.g., by means of a sensor), and is-a (or subtyping, or inheritance) relationships can be defined (e.g., a robot is-a Machine), thus linking together the differently colored subdomains. This modularity is intended to result in:

- reuse of models
- common understanding of concepts
- governance per subdomain

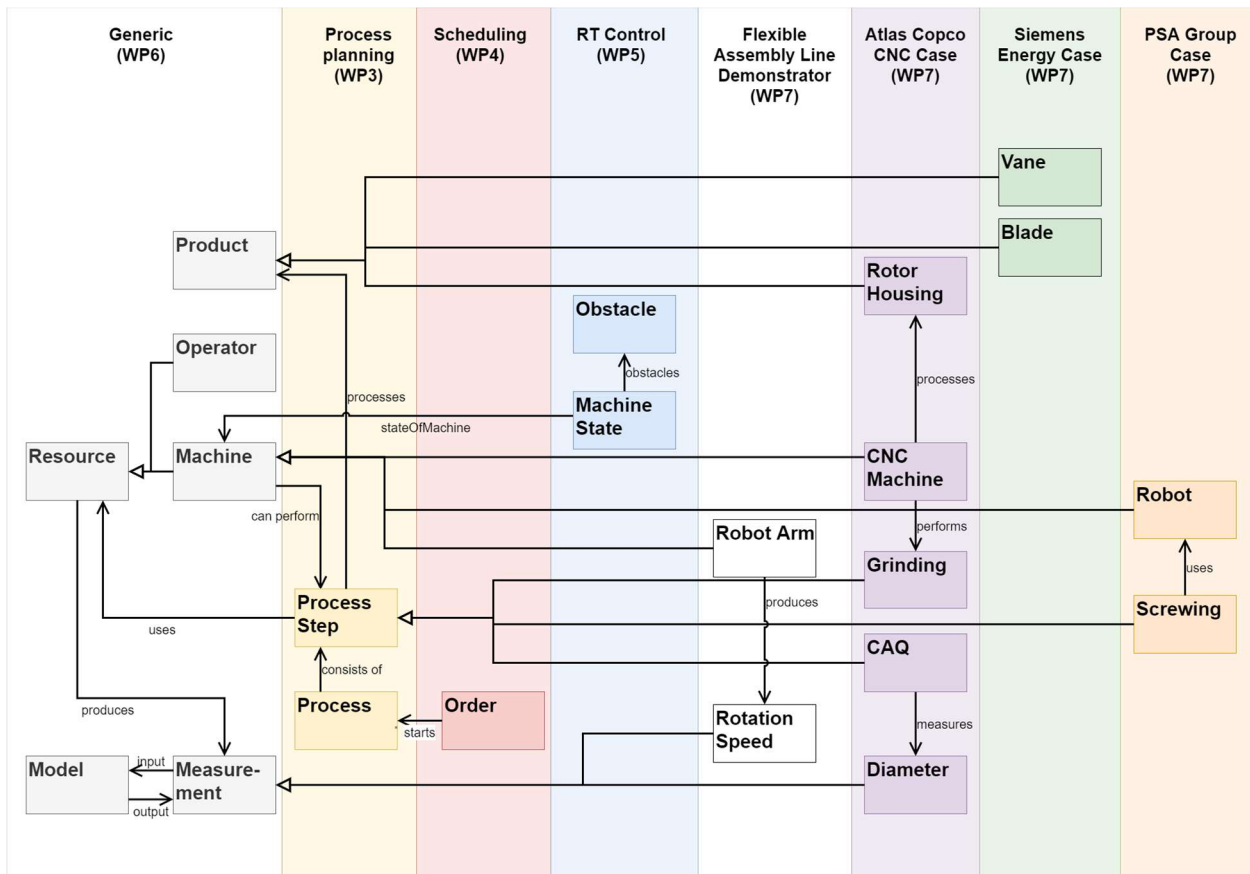


Figure 5: Overview of the ASSISTANT domain model element hierarchies

2. Current State of Data Fabric Development

To introduce readers unfamiliar with the field and provide context for the work, this section outlines a brief survey of the current state of the art for data fabrics and data management technologies in manufacturing. To provide structure, the section is divided into three subsections based on area surveyed: we first look at research initiatives and systems related to the main technologies used, we then discuss current practices in industry, and finally address the gap between these two.

2.1 State of the Art in Research and Technology

2.1.1 Architecture and Capabilities

Within data fabrics, the commercial offerings with large market presences [1] primarily offer the following features:

- Unified platforms for data management. This most importantly include providing a single abstraction for accessing, collecting, and manipulating data without explicit knowledge of data format and storage location [2].
- Mechanisms for data governance, enforcement of security policies and regulations [3].
- Scalable, fault-tolerant, self-managing and automated provisioning of infrastructure in hybrid cloud environments. Generally available as managed services or self-hosted in container orchestrated environments such as Red Hat OpenShift or Kubernetes [4].

- Support of defining data transformation pipelines for cleaning, quality improvements, aggregation etc. of source data [5].
- Various API endpoints for easy self-service for analysts [6].

These base features are generally represented as layers in stratified architectures [7], adopting a Service-Oriented Architecture (SOA) style to leverage the advantages of loose coupling among services, scalability, standardization in communication and integration with various legacy data and heterogenous formats.

In addition to these base features, data fabric providers generally offer optional integrations to further consolidate data management needs into a single product. Most providers supply mechanisms for big data workflow management (often leveraging software from the Hadoop landscape) supporting batch, streaming, real-time or event driven processing of big data jobs. End-to-end data lineage logging is another common feature which enables tracking of data elements back to the original source.

Some providers offer services that implement capabilities represented in more recent research topics such as:

- Autonomous data preparation/curation/quality assurance realized using ML and NLP [8].
- Relationship inference - the analysis of stored data to infer new relationships in the data [9].
- Knowledge graphs and semantic analysis - create meaning from data by mapping entities, their metadata, and their relationship in an evolving information network. This process may perform autonomously or in assistance with AI [10].

2.1.2 Domain Models and Data Representations

A domain model is a conceptual model of knowledge drawn from a specific domain. There are different ways to use such a domain model and to implement it, e.g.,

- Ontologies are widespread [11]. Often used as documentation and common vocabulary, and for classification.
- Metamodeling [12], e.g., the UML [13] provides means to describe a domain as a UML Class Diagram or more specific meta modeling languages such as MOF [14] and ecore [15]. Metamodeling is often used for programming solutions and automation.

The intent of ASSISTANT is to use the domain models as interface to data and information. Several tools try to provide unified access to data via knowledge graphs, including:

- Knowledge graphs, e.g., Anzo [16], Ontotext [17], RML [18]
- Ontology-based data access, like Ontop [19]

In ASSISTANT, we intend to extend such approaches to support the AI lifecycle by:

- representing the information/knowledge that is acquired during an AI lifecycle: potential correlations, data experiments, etc.
- adding different notions of uncertainty
 - Probabilistic ontologies [20]
 - Bayesian approaches [21]
 - Fuzzy logics [22]

2.1.3 Visualization and Platform Interoperability

Traditional visualization architectures involve the pipeline of Input -> Process -> Output. With the input being triggered by a user driven event of the Graphical User Interface (GUI) application (using keyboard/mouse). During development of such approaches, the UI coding,

business logic and applications data domain are often coded into tight-coupled modules that create lack of maintainability, testability as well as scalability of the application. As a result, the well-known visualization software design pattern called Model-View-Controller (MVC) emerged.

The MVC approach (see **Erreur ! Source du renvoi introuvable.**) separates the different aspects of the application modules and more specifically the input data, the business logic, and the visualization of the application, providing loose coupling among these modules. The input data along with the logic of retrieving and modeling belongs to the Model, the visualization belongs to the View and the business logic belongs to the Controller. This separation of concerns enables the development of the different modules in parallel with the utilization of specific interfaces between these modules, thus a simple change on the Model might have minimum or none effect on the View. While this example is derived from the representation of localized GUI systems, the principles extend to implementation in data managing distributed systems (such as the data fabric) and can be used to illustrate requirements for the same.

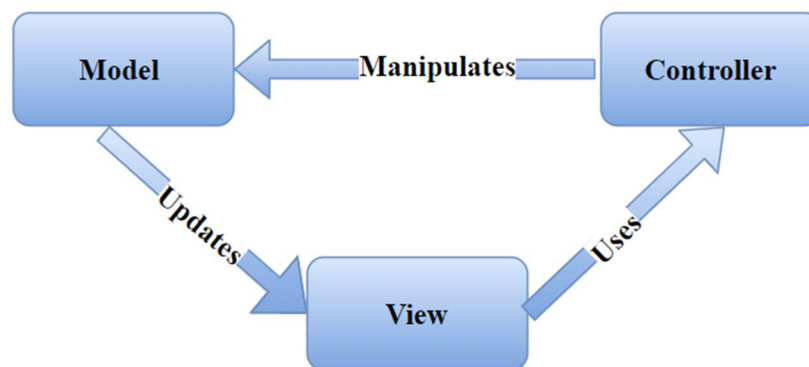


Figure 6: MVC pattern

At the highest level, a typical GUI application (see **Erreur ! Source du renvoi introuvable.**) does four basic things in a specific order:

- Interprets user/client requests.
- Dispatches those requests to business logic.
- Selects the view for display.
- Generates and delivers the next view.

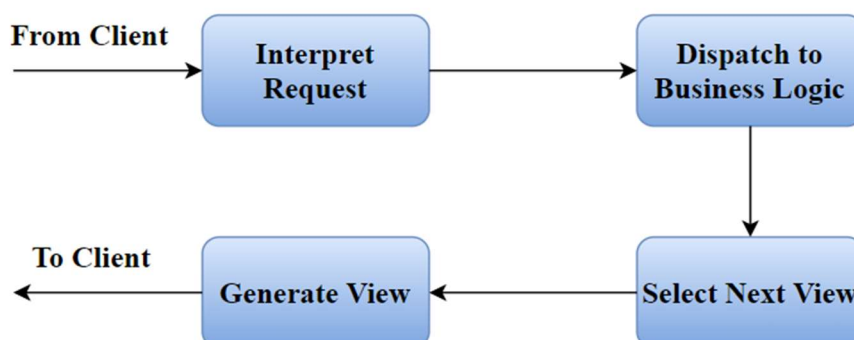


Figure 7: Typical GUI application action flow

The application receives each incoming request generated by the keyboard/mouse or other I/O devices and invokes a requested business logic operation in the application model. Based on the results of the operation and state of the model, the controller then selects the next view to display. Finally, the controller generates the selected view and transmits it to the client for presentation.

A GUI application commonly has the following requirements:

- An application design must have a strategy for serving current and future client types.
- The application controller must be maintainable and extensible. Its tasks include mapping requests to application model operations, selecting and assembling views, and managing screen flow. Good structure can minimize code complexity.
- Application model API design and technology selection have important implications for an application's complexity, scalability, and software quality.
- Choosing an appropriate technology for generating dynamic content improves development and maintenance efficiency.

Model-View-Controller ("MVC") is the blueprint recommended architectural design pattern for interactive applications, but for addressing the above requirements also interoperability should be integrated inside the framework. This is achieved by involving Service Oriented Architectures (SOA) into the design pattern. Interoperability is the most important principle of Service Oriented Architecture (SOA) that can be realized using services (or micro-services). SOA main technologies are aiming on data exchange, data integration and data sharing between system. As such, integration of these technologies (SOA) involves them to the Model component by allowing data acquisition from a diverse set of sources in a standard way. More modern implementations use RESTful services for these data acquisition, which in terms consolidates data from multiple data sources and provide the desired data model to be presented to the end user in the form of a View. **Erreur ! Source du renvoi introuvable.** shows the transition of the MVC architecture incorporating the use of data services.

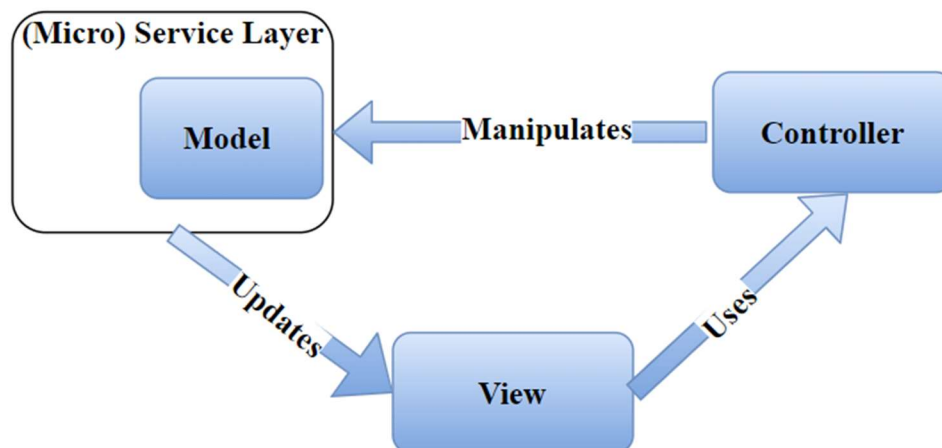


Figure 8: MVC and SOA

From the Data Fabric perspective and to ensure visualization compatibility standard service-based interfaces should be provided for data retrieval and manipulation that are compatible to be used from desktop application up to web application (i.e., RESTful interface).

2.2 State of Current Practice in Manufacturing Industries

With the continuous advances of Industry 4.0, there are more and more opportunities for individual and flexible manufacturing of products. This leads to ever smaller batch sizes and more diverse products with shorter order times and greater changes in order quantity. In current production systems, a large amount of data is usually collected permanently by various sensors and machines. Due to the evolutionary development of most factories - i.e., new machines and new technologies are permanently integrated into the legacy systems or existing structures of the production system - the data landscape in production systems is very heterogeneous and comes from very different sources, such as MES, ERP, SCADA, machine data. In addition, the data can be in different formats, e.g., JSON, CSV, XML, text files.

The data processes are usually executed in the plants close to the machine or specialist system. Industrial edge systems are used, which form the interface between industrial and IT systems. They ensure that analysis and action can occur without data transfer to remote data centers or clouds to provide the highest possible process stability. Various data extraction, data mining, and data cleansing techniques and tools already exist for extracting data from the aforementioned systems, e.g., ETL processes or data virtualization. Nevertheless, data connectors are usually a tool- or system-specific, or even factory-specific task and often must be set up and configured anew in each project. In most cases, this results in several databases, analysis tools, and cross-connected applications via individual interfaces. Only a few manufacturing companies today have an IT infrastructure that enables consistent and straightforward querying of all data sources while providing insight into the meaning and context of the data.

2.3 Observed Research / Industry Practice Gap

To illustrate the gap between current industry practices and state of the art research / data fabric technologies, and map out potential ways to bridge this gap, we here include a brief example and discussion of the experiences of the ASSISTANT partner PSA in their work towards a unified data management platform. Since WP6 is supporting the functional WP 3-5, the aspiration is to implement a state-of-the-art data fabric for ASSISTANT.

In the STELLANTIS project, PSA aims to implement a single data lake for the entire company and all their production plants. For this purpose, PSA has chosen to build on the Amazon S3 systems for on-premises data lake storage using the systems provided by a third-party supplier. To enable extraction of value from data, the data management system is stratified into several layers aimed to refine and derive value from rough / raw data are defined: data structure layer, validation layer, enrichment layer.

On top of this, analysis models are defined that create and define models based on transformed data. These models are then incorporated in data analysis pipelines in trade applications. These systems build on an extensive technology base and include component, service, and system implementations building on / using Kafka, Hadoop, Exadata, Spark, Python/Java programming. In addition, the system also defines specialized tools for management and searching of logs.

When integrating with this complex system, several challenges arise. For example, each individual production plant needs to implement local data collectors with technologies less complex to realize a first filter of the data before transferring them to the central system. In production plants, local data recovery techniques are favored, and systems to aggregate and filter data locally are typically implemented using well understood and generally available / accessible easier technologies such as Python, Node-red, Redis to send data towards the data lake. Tools used in these workflows need to be robust and validated to work in industrial conditions. For this reason, it is generally not possible to use research prototypes that may

impact production requirements and KPIs. Substantial costs are incurred to implement all these structures in terms of hardware, software, and human expert costs.

Regarding evaluation of the possibilities introduced by projects such as ASSISTANT, several options and ways forward are judged as interesting, including for example.

- The concept of research solutions able to detect failures on the equipment level and select functional alternatives in autonomy is seen as innovative and interesting to realize in demonstrators.
- Related data management concepts, e.g., miniature data lakes with various tools capable of performing data-related application processing is also interesting.

For industry actors such as PSA to adopt research results, a deeper understanding of the potential and capabilities of research prototypes are needed. For example, PSA needs to understand how far they can go with a local application as a part of a production line. More flexible solutions able to interact according to recovered data from robotic cells and adapt to circumstances with respect to the resource's location and scheduling. There is currently no automatic feedback loop to select alternative work, neither alert to be considered by a human, which both need to be realized to make full use of the capabilities of the digital twins.

A key consideration in bridging the gap between the current state and the envisioned solutions is likely the accessibility and integration of distributed computing resources, e.g., for storage and processing of data. Hence, a challenge in our project and in research tools would be to package and optimize the tools needed to these treatments to have a real adaptive robotic cell.

3. Use Cases

The ASSISTANT project defines three industrial use cases for context and validation of project results. To familiarize the reader, this section provides brief summaries of each use cases as seen from the perspective of the work of WP6 and the data fabric. Further details on (and a more complete view of) the project use cases are available in the deliverables of WP 3 - 5 and WP 7.

3.1.1 Atlas Copco Use Cases

The Atlas Copco use case is based on the production of a compressor “airend” - the assembly that includes the rotors and the housing they are in - the core element of a compressor. The compressor airend production is a high mix low volume (HMLV) production, and many different variants (up to 30, each consisting of about 20 components) need to be produced in one production area.

The primary use case will focus on a single production step that involves a CNC (computer numerical control) machine. CNC machines can precisely process parts based on computerized instructions, which can be in the form of programming code or CAD models. The CNC machine for this use case machines casings - the outer shell of the compressor element. Typically, this shell consists of 2 parts: a rotor housing and a bearing housing, to enable other components such as rotors, bearings, seals, etc. to be installed inside of the casing in the assembly step.

Atlas Copco has 2 main overarching goals with use cases for the project:

1. Measure as little as possible, while maintaining the same quality for machined parts

2. Make new operators as experienced as employee that has 20 years working experience in the machining area.

These goals will affect KPI's such as

- The overall OEE - mainly the Performance and Quality part, availability will see less effects.
- Production Cost
- Training time needed for new operators/new operations.

Towards these goals, Atlas Copco sees extensive applications of the digital twins for process and production planning and define data management capability requirements on the data fabric indirectly through these tools (as further described in Section 4).

3.1.2 PSA Use Cases

As previously described, PSA defines a data management project where use cases are selected from the engine production lines. These production lines are divided by specification of operations (OPs). The operational sequences (steps that need to be performed, sometimes referred to as routes) of an OP are determined by a range associated to the specificity of the tasks that need to be performed and product (engine) diversity.

In ASSISTANT, the PSA use cases are selected to correspond to specific OPs. In manufacturing and assembly, each OP is managed by a PLC and at the beginning of the production line the motor block is placed on a palette with a RFID tag. The RFID tag contain a specific data mapping which is shared at each OP to the PLC. The PLC manages the configuration of the OP cell(s) (e.g., mechanical, process, robot trajectory, etc.). The RFID also contains a process data list (containing data such as safety screwing, numbers of screwing task done to finalize, cycle time, failures, etc).

Each OP component contains several parts, e.g., robots, LHM, PLC, screwing machines, sensor I/O links, safety components, grippers, conveyor belts, AGVs, etc.; and defines common workflows such as, e.g.,

- The motor arrives on palette placed on the conveyor. The conveyor can have an advance speed from 10 to 14 m/min.
- The motor block will stop on an OP for a specific time, e.g., 30s, to add parts with different component (e.g., robot, specific machine, screwing, etc.).
- At the OP, a designated collaborative robot will select and grip parts and add them monitored by sensitive sensors and with the collaboration of an operator.
- At the end of the OP all tasks performed as well as manufacturing data (e.g., safety screwing data torque and angle, the cycle time, possible issues, etc.) are recorded on a tag to store and retrieve the data of the OP.

As illustrated, there is great diversity in the environment and data recorded, and the production line product and a variability of the numbers of part are needed to be able to be qualified and quantified according to predetermined patterns, e.g., queries on product by diversity and by day according to the customer's needs.

Due to the sequential nature of the process, delays on even a single OP on the line can affect the complete production of the mechanical plant, including the terminal plant waiting to complete the motor block.

To better understand, visualize, and optimize PSA's production systems a wide variety and types of data need to be available to the digital twins. For PSA's use case the twins need to be able to use data from and reason on, e.g.,

- product quality

- cycle times (respected and deviations)
- production costs
- component usability
- maintenance (preventive and predictive)

As a result, the PSA use case places indirect data management and provisioning requirements on the data fabric (via the used digital twins) for capabilities for storage, aggregation, processing, refinement, and provisioning of data. The envisioned goal of the PSA use cases is to formulate a strategy and make advances towards a unified platform for aggregation and management of diverse production data coming from multiple sources and types of provider equipment.

3.1.3 Siemens Energy Use Cases

The Siemens Energy use case is based on production lines in the Siemens facility in Berlin, Germany, where turbine blades and vanes are manufactured. The scope includes a section of the factory which can be simulated on shop floor level, including aggregated information from selected vendors.

The use case targets KPIs such as

- On-Time-Delivery (OTD)
- Machine Utilization
- Lead Time
- Production Cost

which should all be available for simulation and evaluation in the digital models. The data used in the use case is derived from the manufacturing environment (anonymized when appropriate / needed) and is used in the digital models to perform discrete-event simulations for the manufacturing planning of gas turbine blades and vanes. Only pre-selected and anonymized simulation input data from historical planning data (no actual data) is provided by the corresponding Siemens Energy planning department. The simulation output data is further processed in further components for reasoning on optimization measures altering the input data for the next simulation run.

The main system used in the use case is Tecnomatix Plant Simulation, as discrete-event simulation engine produced by Siemens Digital Industries. Goals of the use case include calculation of a two-year planning horizon that involves evaluation of “make or buy” decisions for production item(s). Simulations includes a production time per item(s) of 12 months and consider 100 000 - 200 000-line items in the simulations. About 500 operations (routing steps) are to be processed per day in the simulations.

Like the other industrial use cases, the Siemens use cases impose indirect data management and provisioning requirements on the fabric for storing input and output data through the used digital models (more extensively described in Section 4). Since Siemens Energy does not participate in work package 5, real-time data flows between the virtual and real systems must be compensated for the digital twin by manual interventions of the planner. Still, the results of ASSISTANT on digital twins can be fully exploited later during the exploitation phase.

4. Requirements

To provide a basis for validation and metrics for evaluation of WP6 data fabric results, this section provides requirements for the data fabric based on the needs and intended use cases of the digital twins and industrial use cases. Requirements are gathered by area following an

interview-based requirements gathering methodology (commonly used for the entire project) and classified by category (functional, non-functional) and priority (on a scale from option to showstopper) and presented with descriptions and a motivation (rational). Further information and details on the ASSISTANT requirements elicitation and specification methodology is available in Section 6 and in (more detailed) Deliverable 3.1.

For accessibility, the requirements context and needed functionality are first generally described for the digital twins and other developed tools (Section 4.1) as well as for the project use cases (Section 4.2). After this, more general platform requirements for visualization and platform interoperability are described (sections 4.3 and 4.4). Finally, the previously expressed requirements are summarized and formal descriptions of the high-level data fabric requirements are detailed (Section 4.5).

4.1 Digital Twin and AI Tool Requirements

The respective digital twins from WP3-5 have the following requirements.

4.1.1 Process Planning Intelligent Digital Twin

The data fabric must be able to store data in different formats. In the context of the digital twin for process planning, this is product data, data from the production system, historical changes and historical process plans, user intends/questions and weighted KPIs as input. In addition, the data fabric must save product task and production system models, production graphs, predicted KPIs and optimal process plans as output for stakeholders. This data can be in JSON or XML format, contain textual components or three-dimensional data, e.g., in the Jupiter tessellation format. All this data must be stored. In addition, the data fabric requires methods to access this data, such as SQL interfaces. The data fabric must include extract transform load processes to store data in a format that is easy to understand by users and easy to use by algorithms. This should include a standard for storing historical data, three-dimensional files, or changes. The data must be updated when changes are made to production systems, products or new products, and the digital twin must be informed. Complete and correct data must be ensured, and missing data identified. The digital factory needs central, reusable services to analyze and visualize data e.g., historical process plans.

The data fabric must be able to store and execute methods via a computer connection. In the context of the Process Planner, these are the user interactor, the process planer, the predictor, and the optimizer. Those methods are programmed in C# or in python include AI-functions as well as connections to the data and the knowledge layer, which must be enabled.

The data fabric must allow for different knowledge representations. For the digital twin for process planning, this is knowledge about the domain of process planning as well as about the system of the digital twin. The knowledge representation e.g., as an OWL2 representation must be stored and viewed via the data fabric. Metadata of the generated outputs and the inputs must also be stored. In addition, the data fabric must contain options for querying the knowledge representation that are used by the algorithms. General methods must also be provided to instantiate knowledge representations with the data storage.

Access to algorithms, data and knowledge must be controllable, so that process planners have extended modification rights, whereas production planners only have read rights.

4.1.2 Production Planning and Scheduling Digital Twin

The data fabric must be able to store both raw and processed data. Data may be produced as aggregated report data or by direct monitoring and may be stored in the data fabric asynchronously. The data fabric must provide mechanism for organizing or aggregating raw data to provide meaningful contexts for end-users, and the data must be usable in both testing and training of manufacturing planning and scheduling AI models. For developing the digital models into digital twins, the data fabric shall support a real-time and secure bi-directional data exchange between the virtual systems and components on the one hand, and the production system on the other hand, comprising ERP, MES, SCADA systems. The data fabric must provide metadata defining and explaining raw and processed data, as well as the relationships among different pieces of data (e.g., relation between tables or files). The data fabric should be able to control or recommend the locations where data is stored and provide / have the flexibility needed to allow for storage of data on different types of devices and systems (e.g., on premise, edge, and cloud systems). The data fabric should have the capabilities to deal with streaming data and provide access to data based on time windows incrementally, as well as have the capabilities to perform basic filtering of data, e.g., to detect and filter outliers in data. It should also, support model acquisition and support injection of functionality to organize and find relationships in data (i.e., facilitate basic data processing). The data fabric should support ingestion of data that describe production systems, ideally using something simpler than domain models, e.g., JSON. Finally, the data fabric should support querying processed data based on its original raw data. Should be able to follow a history of changes to the original raw data. A set of separate files that belong together should be easily queried and bundled.

4.1.3 Digital Twin for Reconfigurable Manufacturing Execution

For the monitoring and the controlling of the manufacturing system a third Digital Twin (DT) is deployed. At this level it is important to capture the status of each resource within the system and take decisions for the process execution. The detailed design of this real-time DT alongside with its functionalities is presented in the deliverable D5.1. The connection with the data fabric is very important as it is mainly the backbone that serves the communication between each DT level.

The real-time Digital Twin (DT) interfaces are considered a link between the real and digital world. To effectively represent the whole workstation area this module interacts both directly with the real world through sensors and with the data fabric. The online awareness of the workstation is achieved by gathering data from sensors and machines and this communication is required to be real-time. However, with respect to monitoring and instrumentation data, the communication with the data fabric is not required to be real-time.

The data fabric contains the domain data regarding the manufacturing system if the information that is provided from WP3 and WP4 to WP5. At the beginning, these data need to be provided on the real-time DT for the initialization of the system. Such data could be the process and production details, information on the available resources, the production cell layout. After the succeed initialization, the real-time DT will provide periodically data sets to the data fabric that presents the state work cell. This historic data could be used for offline calculations. The Streamhandler platform (by INTRASOFT) is responsible for the interfacing of the data fabric and the real time DT, e.g., aggregating and integrating monitoring data for storage (and later processing) in the data fabric.

4.2 Industrial use cases requirements

The use cases from the industrial partners have the following requirements. Note that most of the requirements listed here are capabilities for the digital twins, we (briefly) list them here for completeness and providing the reader insight into the origin of some of the data management requirements imposed by the digital twins on the data fabric. For clarity: as the Siemens Energy use case has more direct data management requirements and currently is better understood at detailed level, we here include more details on this use case in this report. This does not in any way reflect a prioritization of the use case, the additional details are included for the benefit of the reader.

4.2.1 Atlas Copco

Atlas Copco has the following use cases for the ASSISTANT project, this does not necessarily translate to direct data fabric needs:

1. Measure as little as possible, while maintaining the same quality as today.
2. Make a new operator as experienced as an employee that has 20 years' experience.

These break down into following sub tasks:

- Determine and explicitly model what correlations exist with the geometric quality of the products. For example, quality may be unstable after starting up the CNC machine or after a changeover but may be stable when after several parts have been produced.
- Use an adaptive measurement strategy to make the number of CNC, CAQ and CMM measurements dynamic when geometric variance is expected to be low, depending on the influential factors. This way, the total number of measurements, and with it, costs, are reduced. After determining these influential factors, AI models that understand the scope and context of products will need to be developed that dynamically suggest the minimal measurement ratio to maintain quality standards. For example, more measurements may be necessary after starting up the CNC machine or after a change over, but measurements may be drastically reduced once the machine is producing stable results.
- (Improve quality and Reduce training period for new employees) Create a virtual operator assistant that suggests machine settings (and additional measurements when needed) to reduce variance when processing a batch.

The desired solution is an implementation of the full life cycle of the AI system, including:

- Learning of the AI models, which includes modeling of existing (possibly implicit) knowledge, finding correlations, learning parameters and uncertainties, etc.
- Online execution on the CNC machine, providing live suggestions to the operator by using predictive models based on AI.

This is shown in the **Erreur ! Source du renvoi introuvable.**: on the top left are some of the potentially interesting data sources that contain data related to production. These sources can be combined with knowledge of domain experts, but also with existing simulations or specific type information. This knowledge graph can then be used to train specific digital twin models, either for analysis purposes (bottom right) or for deployment on the shop floor (bottom left). The deployed model will then provide online feedback to the operator, e.g., should this next part be measured or not.

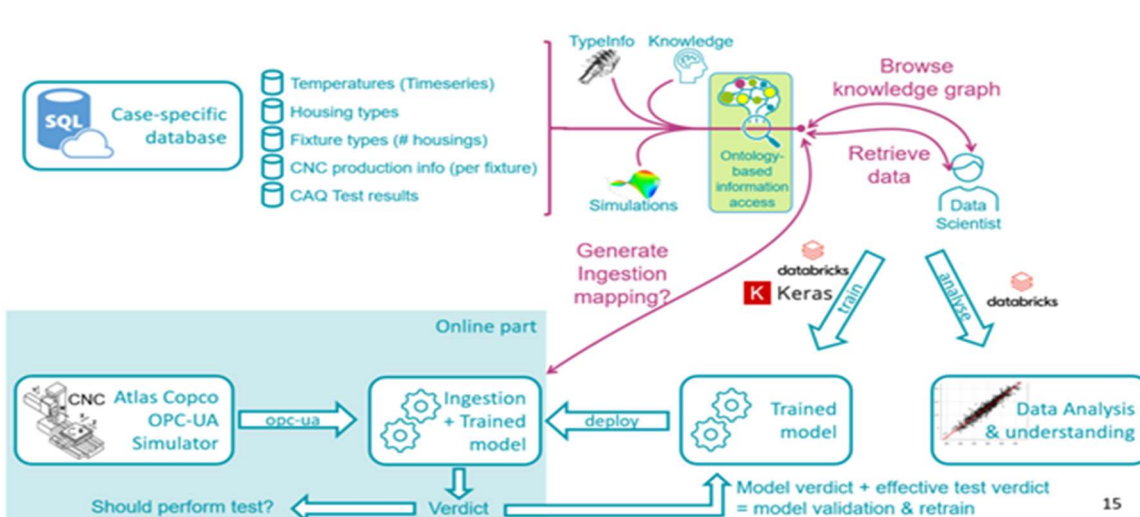


Figure 9: Life cycle of the AI system in an Atlas Copco use case

The Data that needs to be used in the ASSISTANT project for the Atlas Copco use cases, will be coming from multiple different silo's and systems, e.g.,

- ERP data
- MES data
- CAQ hand measurements
- Piweb CMM measurements
- IoT data coming from the machines.

All these data types are generated at different time intervals and frequencies. For example: the processing steps are booked in the ERP system whenever a pallet is moved from one operation to the next, the planning in the MES system is updated twice a day, and the IoT data coming from the CNC is a continuous flow of timeseries data (like temperatures, pressures), and discrete data (like probe measurements performed in the machine).

An overview of an example machine process is illustrated in Figure 10. As illustrated, this data is not synchronized in any way, and not standardized (this is not possible). The data is stored in a variety of on-premises servers and cloud resources, all tailored to that specific use case.

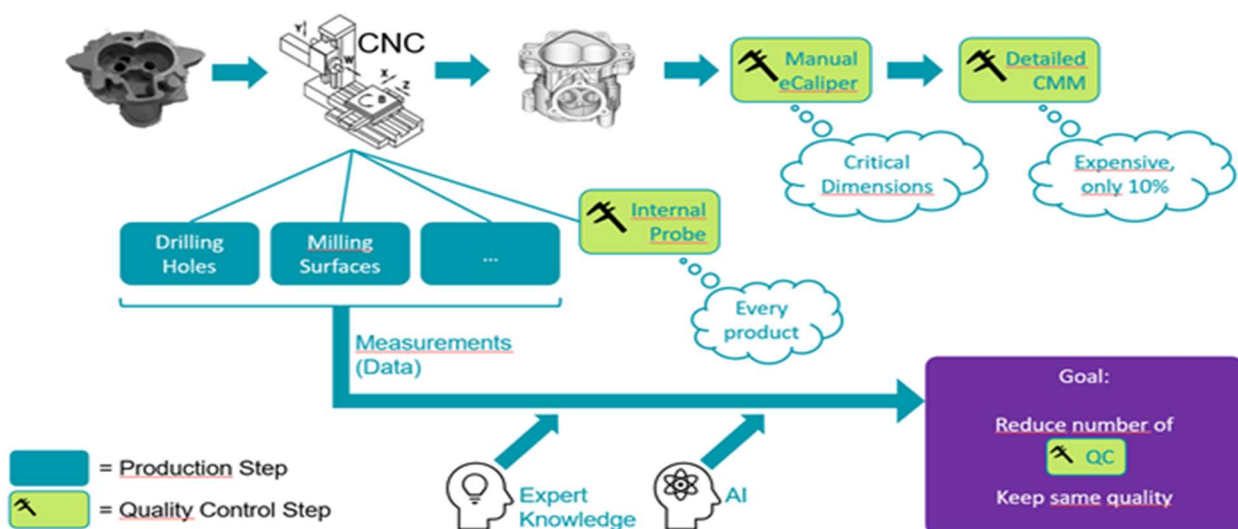


Figure 10: Example of a machining process and the measurements performed

To reach the goals of the project use case, the data fabric must enable the ingestion of all these different data sources, to make data available for further investigation, analysis, and AI model training. Both Batch level and real time data ingestion must be possible.

Batch:

- Data is pulled from source destination and copied.
- Time frame can be defined (minutes to hours to days)
- Typically, easier to configure.

Real time:

- Data is pushed from source destination or requested at real time through an API.
- When data is required from last state, when information needs to arrive as soon as possible to destination application or when unmanageable to copy the data.

From experience, the best approach seems to be an API based architecture: abstraction from applications to data infrastructure to guarantee modularity. Applications using the data fabric are not allowed to make changes to the general data sets, and the data fabric must therefore support “copy-on-write” semantics. If, for whatever reason, a new/modified/altered data set is created, it should become a separate table / file, and the original data should remain untouched to ensure proper functioning of other applications using those data sets. Data quality should be checked and indicated with some sort of quality label, and the data fabric should support injection of functionality for this dynamically. A data catalog function must be available to enable easy discovery of available data sources and avoid duplication of data. A process should be defined to deal with adding new data sources to the data fabric, or adding new structures (e.g., columns) to existing data sources in the data fabric.

4.2.2 PSA

As mentioned, the PSA use case of ASSISTANT selected a use case from an engine production line. The production line is divided in operations and the sequence of the OPs is done by a range associated to engine diversity.

In the use case, as illustrated in Figure 11, the production cell is managed by a PLC connected to the IT. The IT inform the PLC of the production diversity, the PLC have a cycle code by diversity to organize the job of each component.

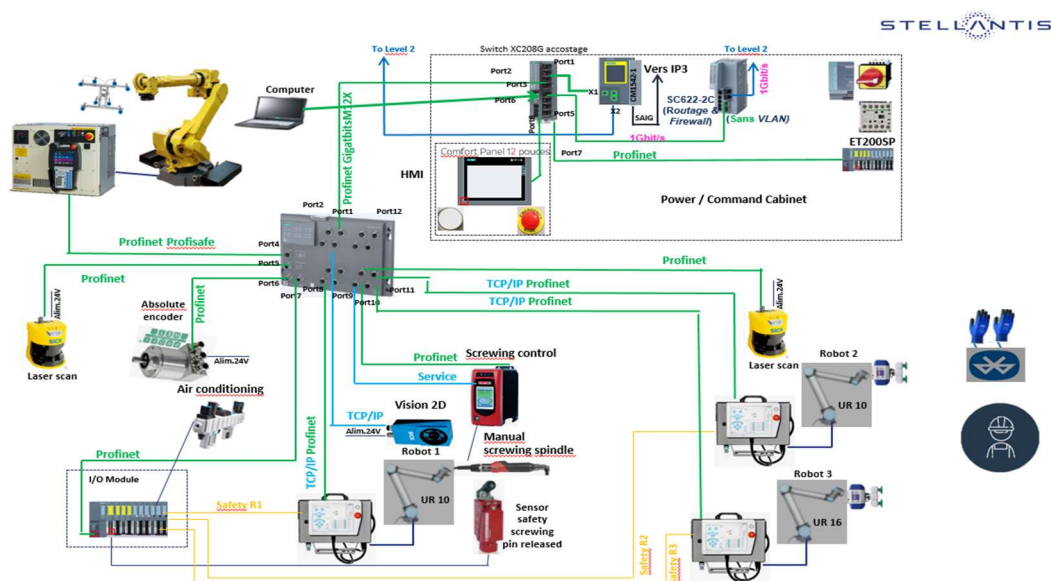


Figure 11: Overview of production cell connectivity

In the actual production line, the data collected are limited to the PLC cycle code, component task, log, status, cycle time. The future goal is to connect all the component and to recover all their data to the DATA FABRIC with the least intrusive hardware solution.

Each component provider has a specific solution to collect the data, with its own software and protocol communication, and combined form a data flow that aggregates and stores data in a common repository as illustrated in Figure 12. The objective is to standardize the data by component and to simplify the connection to make them available on an intelligent Data Fabric.

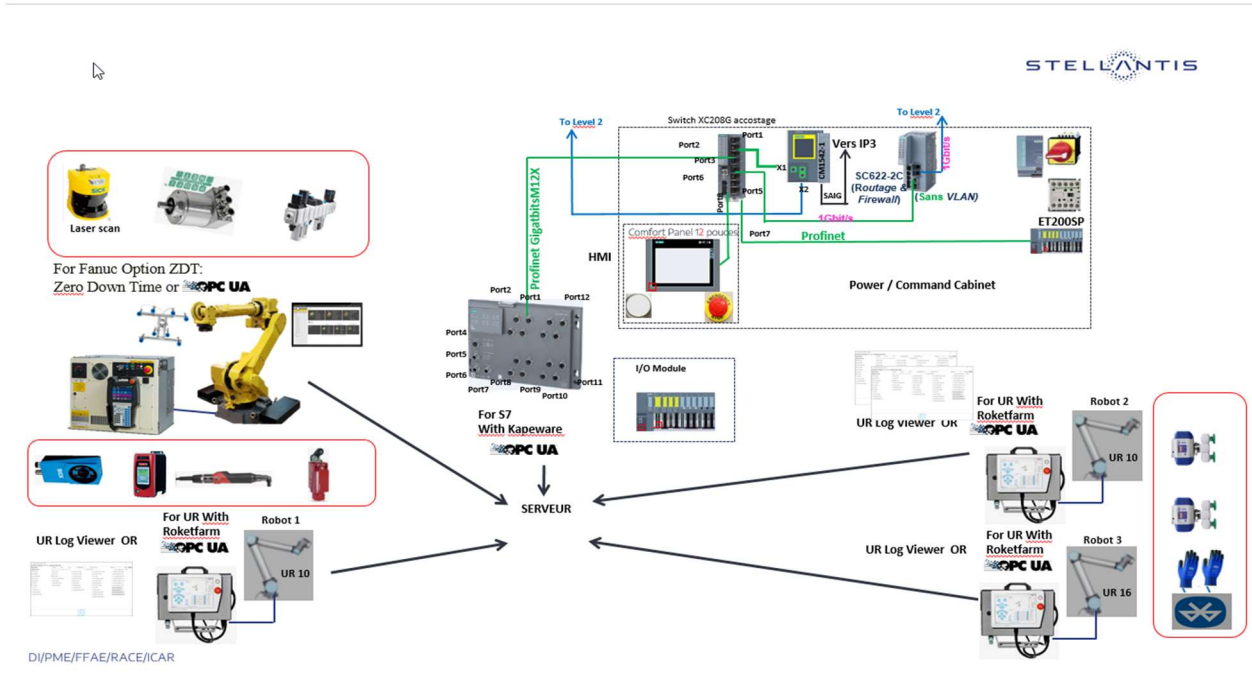


Figure 12: Data flow of production with specific solution to collect the data surrounded by red components without data flow

To support this modeling in the produced digital twins, the data fabric must be capable of providing higher level abstractions for common data sets with different format. This could for example involve robots from different manufacturers producing the same measurements but in different formats. The data fabric much also be able to determinate, identify, or designate the relevant information for the production line so that external systems / clients can find connecting / crossing Data. The data fabric should also support creation of analysis models from transformed data, in external tools.

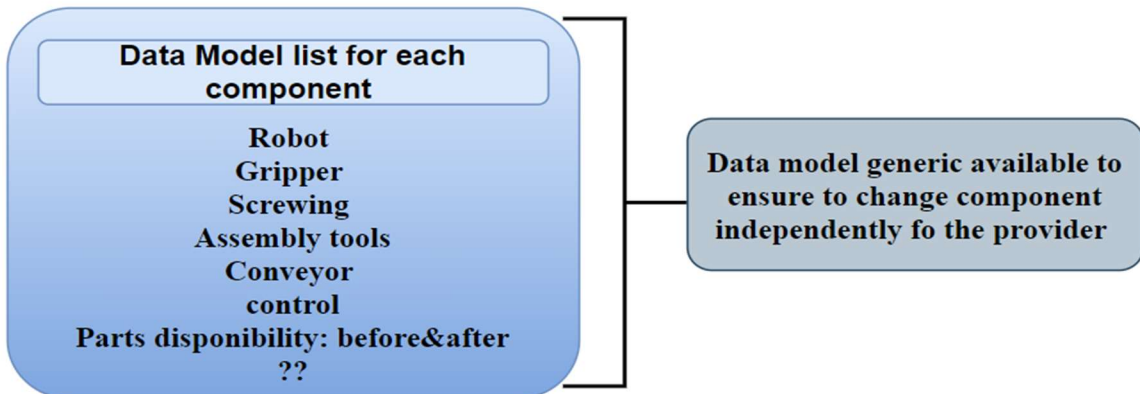


Figure 13: A Data models are necessary for each component independently of the provider.

An overview of the data models involved is provided in Figure 13. As illustrated, individual components have their own data models that provide generic, provider-independent capabilities for exchange and storage of data. The data fabric should also support definition of data mapping models by components independently of the supplier, and potentially also translation of structured models. This could include, e.g., a robot with the main data:


Robot

- Controller
- Software
- Motor axis 1 → 6 (7)
- Position
- Database
- Controller status
- Safety Status

Where the semantic of the measurements is preserved in different representations. It would be desirable to determine the simulation tool for a digital twin and be able to carry out virtual commissioning in industrial condition while connected to the data fabric.

The Data fabric should be able to (via the Streamhandler platform) connect to all the production line component and to the central information systems with ERP software for retrieving information regarding the product supply chain and MES software for a different item that will be produced. All collected data must be organized and validated, the data fabric must contain or expose mechanisms for injecting functionality to inspect or validate the quality of data. Data fabric data must be accessible using easy to use functional tools.

Table 1: Overview of desired functionality for Stellantis (PSA)

Data Fabric Functionality	
Recover data from all the different component of the production cell. → Optimization of the production with the digital twin	Integration of the new model in the digital twin with the KPIs (cycle time, cost, quality, etc). → Adapt the cell to produce the new models
<p>Functionality for the production:</p> <ul style="list-style-type: none"> • KPIs (quality, cost, cycle time by component, tasks done, status of each component) • Cycle code and programme (robot, plc, etc.) running • Axis position (digital twin) • Process Quality Control • Production Management <p>Functionality for the service :</p> <ul style="list-style-type: none"> • Recover log, alarm, default • Predictive maintenance <p style="text-align: center;"></p>	<p>Functionality to add new product in the cell:</p> <ul style="list-style-type: none"> - For the robot: <ul style="list-style-type: none"> • Robot trajectory integration • Program structure (manufacturing language) and robot position • Compliance with the programming standard • Cycle time variance • Geometric difference - PLC: <ul style="list-style-type: none"> • Integrating the standard (PSA) compliant plc program • Check for anticipated malfunctions and failures. • Check the relevance of plc error messages, robot, business.

	- Gap list between virtual and real cell (solution to optimize cycle time)
--	--

4.2.3 Siemens Energy

The data fabric should be flexible to different input data by humans, different algorithms, should be able to communicate with simulation, should be flexible to be updated by actors and AI components. The data fabric should assist in assigning workloads (data) to next available shop floor machine should a machine become overloaded. The data fabric should be able to ingest failures / error rates to aid in simulation and learning and communicate simulation results to human planner.

The Siemens business process is designed to use CSV files as simulation input data which are generated from an ERP system and from end-user data (e. g. SAP) as a first step. Then, the CSV files are converted into a file format which is more appropriate for running quality checks of the simulation input data (e. g. XML or JSON) and uploaded into the simulation application into specific tables as designed for the data fabric for the complex use case of Siemens Energy. “Within the third step, the simulation is run (e.g., in Tecnomatix or SystemC, or a hybrid solution using both commercial and open-source software stacks). Afterwards, the simulation output data shall be generated for end-user self-service to create the KPI Cockpit in EXCEL on a local desktop (Step 3). An overview of the business process of the Siemens Energy use case is provided in Figure 14.

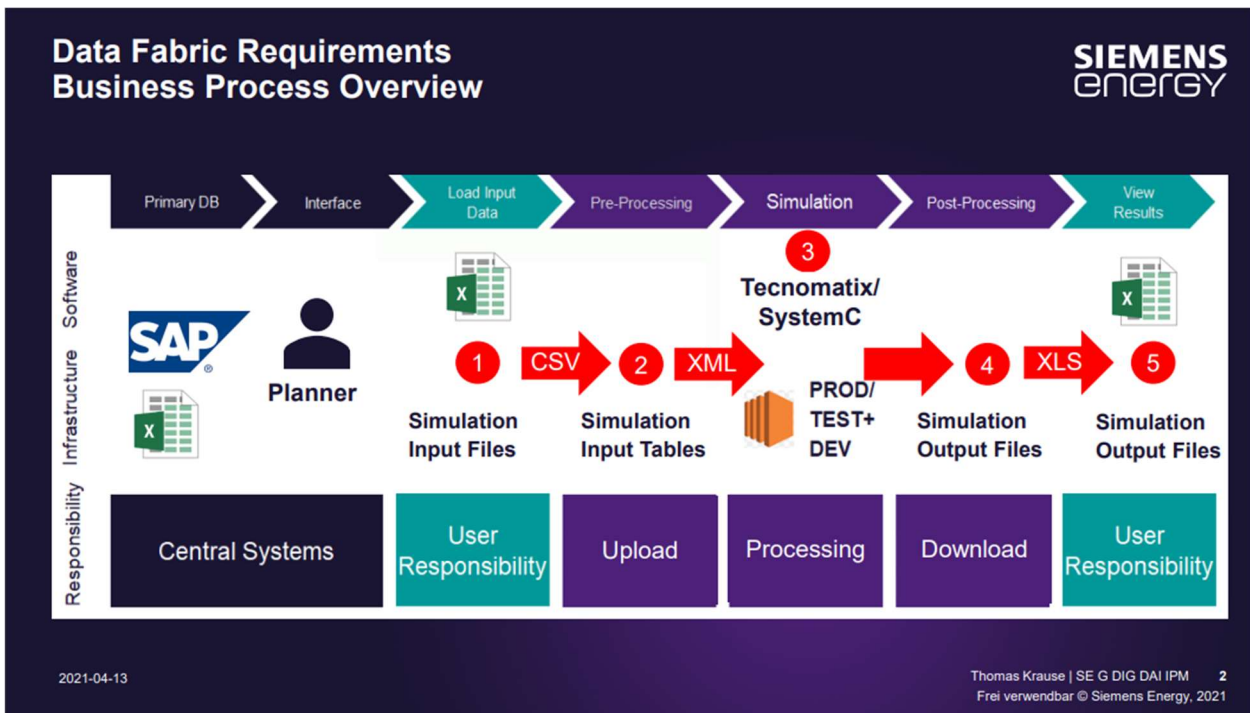


Figure 14: Data fabric requirements business process overview for the Siemens Energy use case

The business requirement is to provide a data fabric solution which supports the defined business process independently from a chosen ERP system, using open-source software to the most possible extend, even including the optimizer/solver which might be needed in addition to the simulation application. The input and output files shall be designed in a way to work with standard tools known by most planners to create KPI metrics on On-Time-Delivery rates of production orders and on utilization rates of resources and workers. In this context, the use of

commercial software (e. g. Matillion, PowerBI, CELONIS, and others) shall be avoided for using the ASSISTANT solution in daily operations.

Summarizing the data fabric requirements for the Siemens Energy use case from a content perspective, Figure 15 outlines the different simulation input and simulation output and provides an overview of the data tables and corresponding data files to be used.

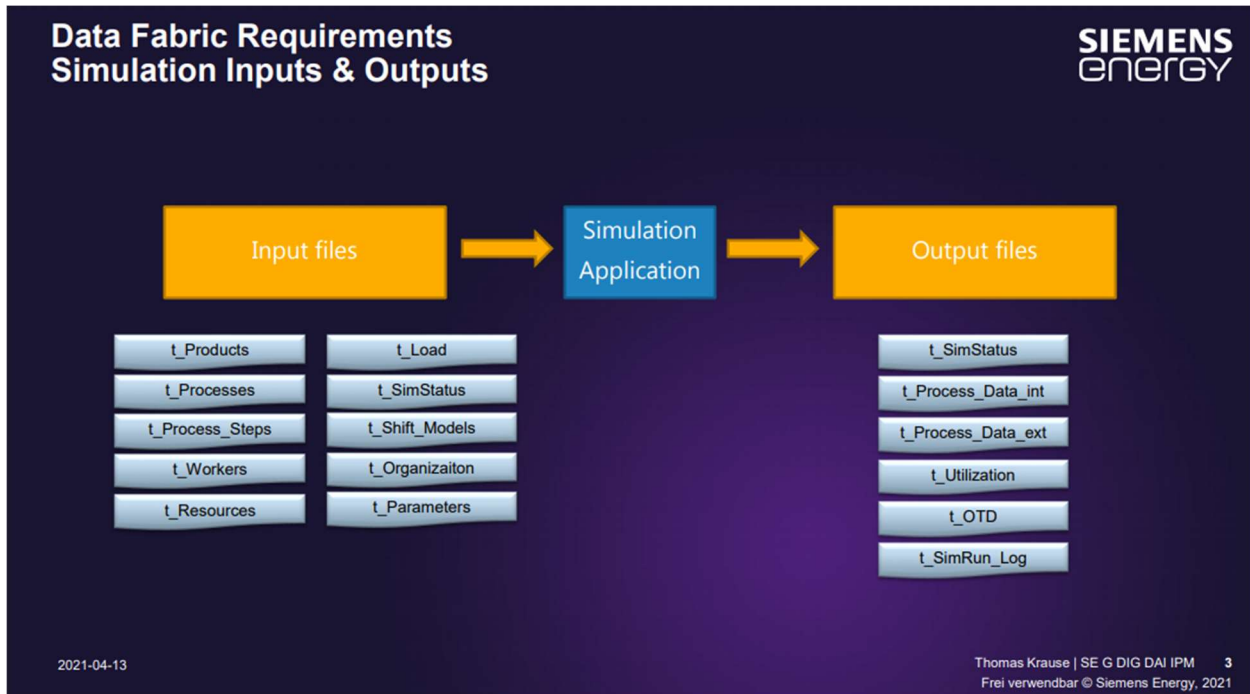


Figure 15: Simulation inputs and outputs for the Siemens Energy use case

In addition, the data fabric must have the capability to store all simulation input data, simulation processing data, and generated simulation output data in a machine-readable format to ease the use of enhanced algorithms, such as dynamic programming methods, simulated annealing methods, stochastic ruler methods, ascend/descend methods, partitioning methods, reinforcement learning methods, e.g., Monte Carlo Tree Search, Monte Carlo Reinforcement Learning, Branch-and-Bound.

As the simulation input data is huge in size (about 100.000 routing steps per fiscal year) and up to 3 fiscal years need to be simulated for the Siemens Energy use case within the long-term planning horizon, the simulator shall be situated in a high-performance environment which supports multi-core processing of 4 to 10 parallel simulations for the generation of one optimized simulation scenario. Due to the fact, that within the mid-term and short-term planning scenarios, the business requires the generation of multiple simulation scenarios per day for more than 30 plants of Siemens Energy in parallel within the deployment phase, each simulation scenario might require up to 100-200 sub-scenarios calculated in the background by the optimizer/solver, the data fabric should be compatible with big data application deployments in professional private cloud environments (e.g., AWS, Azure, SAP Cloud).

To streamline the simulation process, the data fabric shall split the simulation input model and the simulation output model from the simulation data model, which is generating the simulation scenarios, according to the Figure 16.

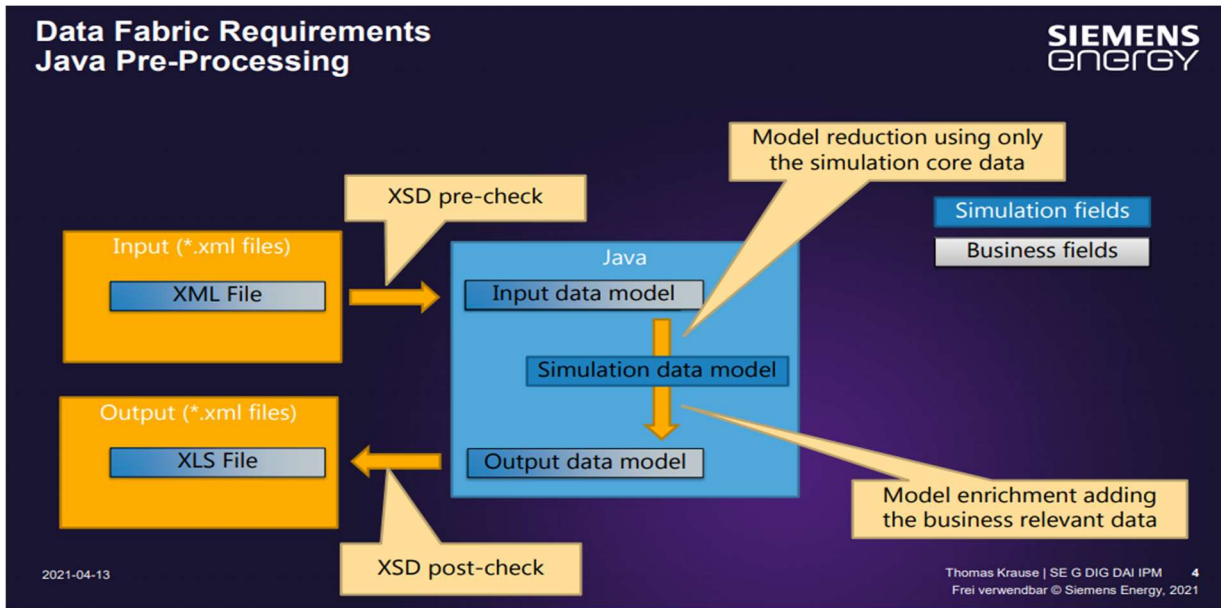


Figure 16: Java-based data preprocessing for the Siemens Energy use case

The Java pre-processing shall deliver a quality report which informs the user on potential data issues in the provided simulation input data, and it shall split the user/location-specific simulation input data (containing the business fields/ extended data) from the generic and light-weight simulation data model (containing the simulation fields/ core data). This pre-processing shall be seamlessly integrated with the Java post processing, where the processed core data is enriched by adding the relevant business fields for the generation of the simulation output data model. Ideally, also the simulation output data shall be checked by an XSC routine to detect quality issues of the simulation results. Therefore, all simulation input data models shall be described as XML Ecore models in EMF for the provision of XSDs.

Further requirements on the data fabric include the data preparation process which requires to follow a defined process to complete the pre-processing within the simulation application. Figure 17 provides an overview of the hierarchy of data used as simulation input for the Siemens Energy use case:

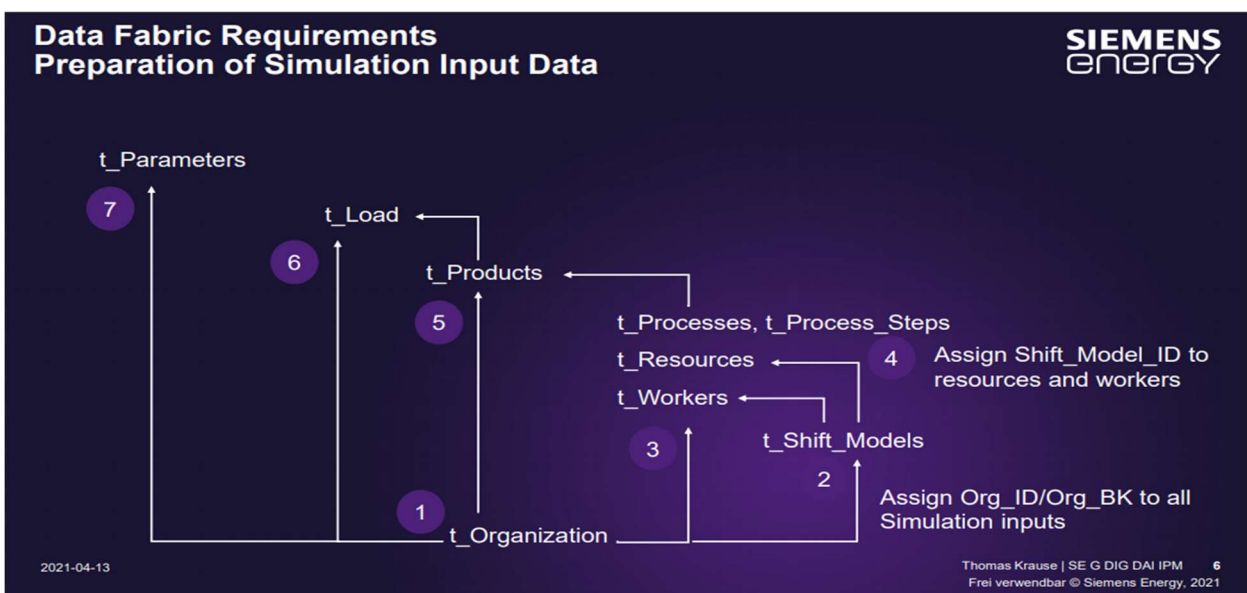


Figure 17: Preparation of simulation input data for the Siemens Energy use case

Summarizing the business requirements on the data fabric for ASSISTANT, Figure 18 can be used. The use case addresses a variation of business targets and uses modeling of simulation parameters and boundary conditions for experimentation in a provided simulation system. Simulation output (estimated objective function values) are used to gauge the validity and utility of experiment configurations, and ultimately provide insight into the effectiveness of the modeled process.

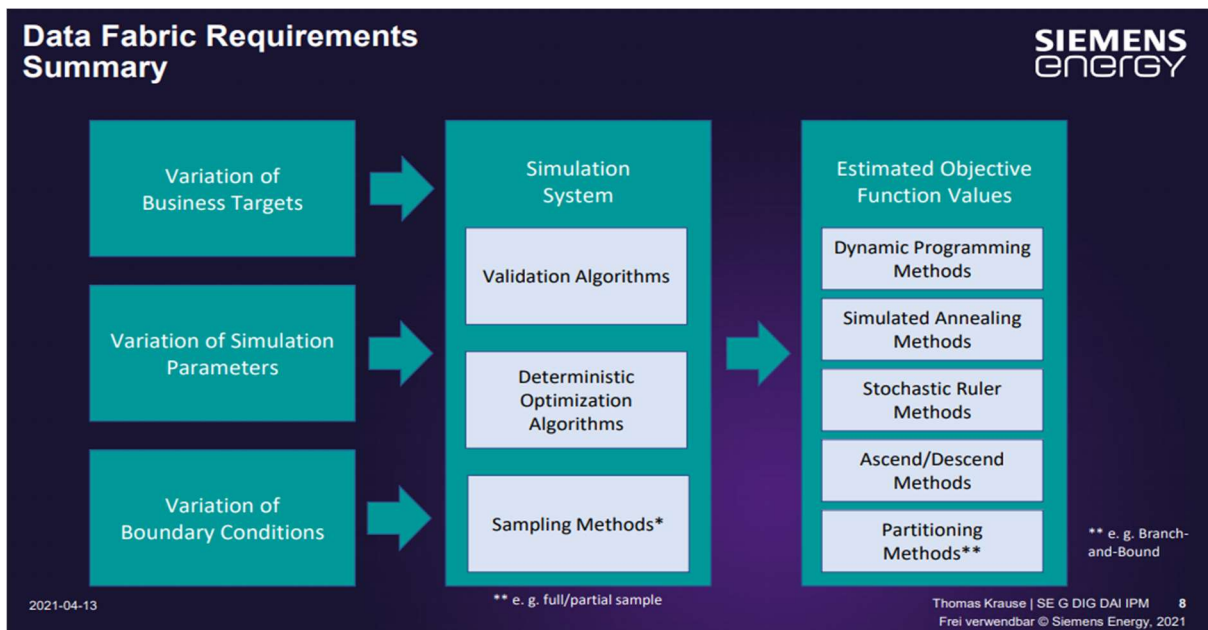


Figure 18: Summary of data fabric requirements for the Siemens Energy use case

It is important to note, that wherever possible, deterministic algorithms shall be used within the data fabric to explore different parameter combinations for Siemens Energy, and to use random/genetic algorithms only in areas, where the space of potential solutions is too huge to be fully explored. AI capabilities shall be only employed, where deterministic methods would not work and where also human planners would not be able to perform the optimization task in a consistent and successful manner.

4.3 Visualization requirements

Through a requirement analysis and filtering procedure, end users still pay attention to the application visualization. These requirements compose different aspect of the visualization development and can be grouped into three main categories:

- Intuitive visual component. This requirement focuses on increasing the awareness of users in respect to their surrounding environment (i.e., awareness of nearby robot behavior PSA use case). Also, in this category belongs the nontraditional visualization technologies like VR/AR glasses etc. These requirements are addressed by providing the most suitable method of visualization per case.
- Data related requirements. These requirements refer to functional requirements to the software system that provides the visualization. There are requirements that the software should be human driven, and these functionalities should be reflected in the visual environment (i.e., human-centric data flow from SE use case).
- Supported Devices. As this category name states these requirements refer to the devices that the visualization should support. Thus, there are requirements for multi-device support in terms of PCs, Tablets and smart phones (derived from the AC use case).

From these visualization requirements the data fabric related requirements come indirectly mainly by providing a common way of data provisioning services supporting the components that will have to address these requirements (see also section 2.1.3). Thus, these services should be used by different hardware platforms, and should allow data manipulation and data grouping functionality for enabling data visualization on different visualization components (i.e., pie-charts, histograms, etc.,)

4.4 Platform interoperability requirements

4.4.1 Information interoperability

From WP3, WP4 and WP5 a diverse set of information are specified. Diverse in the sense of data representation since they involve data sources from ontologies, RDBMS systems, local files, unstructured data and timeseries data. All these data need to be treated in a specific way but also provide a standard as common as possible API for the data manipulation which is mainly a standard Create-Read-Update-Delete (CRUD) interface enhanced with data source specificities (i.e., Ontology reasoning) and business requirement (i.e., support for data processing pipelines) Apart from the handling of these diverse datasets another information interoperability aspect is the data representation. Data representations should be standardized as possible to ensure that Information interoperability will not pose interoperability issues on the other interoperability categories (Visualization and/or Technical/Application). If a global data format cannot be defined for data representation (e.g., XML, JSON, ISO, CSV etc.) then concrete domain specific formats should be used. Here the use of existing industrial standards (e.g., ISO STEP requirement from Siemens Energy pilot case) should be taken into consideration either to be adopted for the common information representation or to ensure compatibility between the common format and the industrial standards (mapping from one to the other to be applicable without information lose).

4.4.2 Technical interoperability

The diversity of applications/modules from the ASSISTANT development work packages are not lacking in diversity either. Systems varying from process planning, production planning, production scheduling, process controllers, digital twins as well as internal components that need to exchange data between them including AI components that are needing training data sets poses a challenge on the technical interoperability to define common application sharing model. Nevertheless, as briefly discussed in section 3.1.3, interoperability can be addressed using SOA principles throughout the architectural decisions. The main requirement here is the use of a common standard communication/integration interface based on SOA principles. This standard interface (SOAP web services, RESTful services etc.) would allow all the components to be loosely coupled and will support the “separation of concerns” concept also ensuring a simpler integration of required components into the pilot cases since not all components will apply to all pilot cases. Regarding the data fabric itself as another software component will address the software independent principles requirement (Atlas Copco case)

4.5 Summary Data Fabric Requirements

Summarizing and complementing the previously described functionality and capability requirements, this section details high-level formal data fabric requirements organized by areas of functionality. For more information on the ASSISTANT requirements gathering process, see

Section 6 and (more detailed) Deliverable 3.1. Further information about the ASSISTANT data fabric architecture will be detailed in Deliverable D6.2, including implementation strategies and design decisions geared towards meeting these requirements.

4.5.1 Data Storage, Management, and Provisioning

The data fabric shall be capable of storing, managing, and provisioning data to meet the needs of the ASSISTANT project and the produced AI tools and digital twins. The data fabric shall be able to deal with both structured and unstructured data, i.e., with or without knowledge of the internal structure of data, integrate with tools and sensors systems for monitoring, instrumentation, and visualization of system data; as well as capable of instrumenting and storing data associated with its internal operations (e.g., performance metrics and KPIs).

Table 2: Data storage, management, and provisioning

R6.1 Data Storage, Management, and Provisioning		Priority SS	M36
Requirement	The data fabric shall be able to store, manage, and provision data of the various formats and types needed by the ASSISTANT tools		
Category	Base requirement		
Description	Storage and provisioning of data is a defining feature of a data fabric		
Rational	Digital twin requirement		

4.5.2 Metadata Support

All data fabric data sets shall be stored along with an accompanying metadata set that annotates and helps track data provenance and versioning of data sets. The data fabric metadata constructs shall also be able to provide non-functional human readable descriptions needed to for use and integration with the data fabric, e.g., API documentation, data schemas, and use guidance descriptions) and tools must be able to use metadata tags to annotate, track, search, and query metadata for, e.g., the construction of (aggregated) virtual data sets and in inference of data structures.

Table 3: Metadata support

R6.2 Metadata Support		Priority HP	M36
Requirement	The data fabric shall be able to store and associate metadata for all data stored in the data fabric		
Category	Functional requirement		
Description	By associating metadata with data sets, the data fabric enables wide ranges of extended capabilities, e.g., identification and documentation of data, for data fabric clients and applications		
Rational	Tool requirement		

4.5.3 Services, Interfaces, and Customization Points

Data fabric functionality shall be published as networked services, accessible via network interfaces and integrable with external tools such as the ASSISTANT digital twins. The data fabric should provide APIs and example clients to integrate with services, and services shall to the extent possible be made location transparent and accessible with comparable qualities of service from different network locations. Service APIs should be provided for major targeted languages along with example clients illustrating the use of the APIs. For adaptation and integration, e.g., application-specific pre- and post-processing of data, the data fabric services should have capabilities to expose customization points for definition of application-specific functionality executing within the data fabric at service level.

Table 4: Services, interfaces, and customization points

R6.3 Services, Interfaces, and Customization Points		Priority HP	M36
Requirement	The data fabric shall expose functionality as customizable, platform independent networked services that can be accessed through APIs		
Category	Base requirement		
Description	The data fabric is a distributed system designed for deployment in heterogeneous network environments, contemporary state of the art distributed systems is developed as service-based systems		
Rational	Digital twin requirement		

4.5.4 Monitoring and Logging

The data fabric shall be able to monitor its own performance and store, retrieve, search and inspect log data detailing the internal operations of the data fabric services. To enable accountability and transparency, this should include, e.g., log records of what user performed what action when, where data originated from and whether it has been pre-/post-processed within the data fabric. Documentation of data placements (and potentially the reasoning behind these) and the use and association of software components (plug-ins or tools) related to data sets may be derived from logs.

Table 5: Monitoring and logging

R6.4 Monitoring and Logging		Priority MP	M36
Requirement	The data fabric should be capable to monitor and log information related to the performance, function, and use of its services		
Category	Functional requirement		
Description	Through continuous inspecting and logging of metrics related to the functionality and performance of the data fabric services the data fabric (as well as external tools) can improve the performance of the system as well as provide auditable information about the use of the system		
Rational	Platform / tool requirement		

4.5.5 Security

Users and tools accessing the data fabric services should be authenticated using strong encryption mechanisms and access to use of the data fabric services should be configurable on role and policy bases for external clients. The data fabric may have the capability to encrypt data transmissions and storage for sensitive data, and the use of such security features shall if so be both configurable and auditable through configuration interfaces. Internal communication within the data fabric (i.e., among data fabric services and / or plug-ins) shall be secured using the same mechanisms and level of security as with external clients and trust zones that span across hosts and networks shall be modeled accordingly.

Table 6: Security

R6.5 Security		Priority LP	M36
Requirement	The data fabric may provide secure and authenticated interfaces to data fabric functionality and audit all interactions with data fabric systems		
Category	Non-Functional requirement		
Description	Through provisioning of system functionality using strong encryption-based mechanisms, the data fabric can protect both the data fabric system itself as well as the data stored in it		
Rational	Being a proof-of-concept implementation, the data fabric may prioritize functional requirements over non-functional in early stages		

4.5.6 Validation

Data fabric functionality shall be validated using concrete scenarios derived from the project use cases, illustrating use of key functionality and major design decisions, and operate on real data from the project use cases whenever possible. Data fabric functionality shall further be demonstrated publicly in a (set of) demonstrator(s) operating on the same or comparable scenarios within the lifetime of the project.

Table 7: Validation

R6.6 Validation		Priority MP	M36
Requirement	The data fabric proof of concept implementation should be validated using realistic scenarios derived from project use cases and tool requirements		
Category	Non-Functional		
Description	Validation of technical and scientific project results in realistic settings		
Rational	Validation of project results showcases project progress and facilitates adoption of project results		

5. Data Fabric Usage Scenarios

To give context and insight into the envisioned use of the data fabric, this section outlines an example scenario that illustrates the role of the system and the interactions among the data

fabric and other tools (in particular the digital twins) produced by the project. The described scenario is intended to be used for demonstration and validation of project results.

5.1 Example Scenario

The described example scenario illustrates the role of the data fabric in the AI life cycle. One iteration of the AI life cycle is shown in the Figure 19. It assumes that data and information is being collected and stored in an intelligent digital twin and can be accessed through the data fabric.

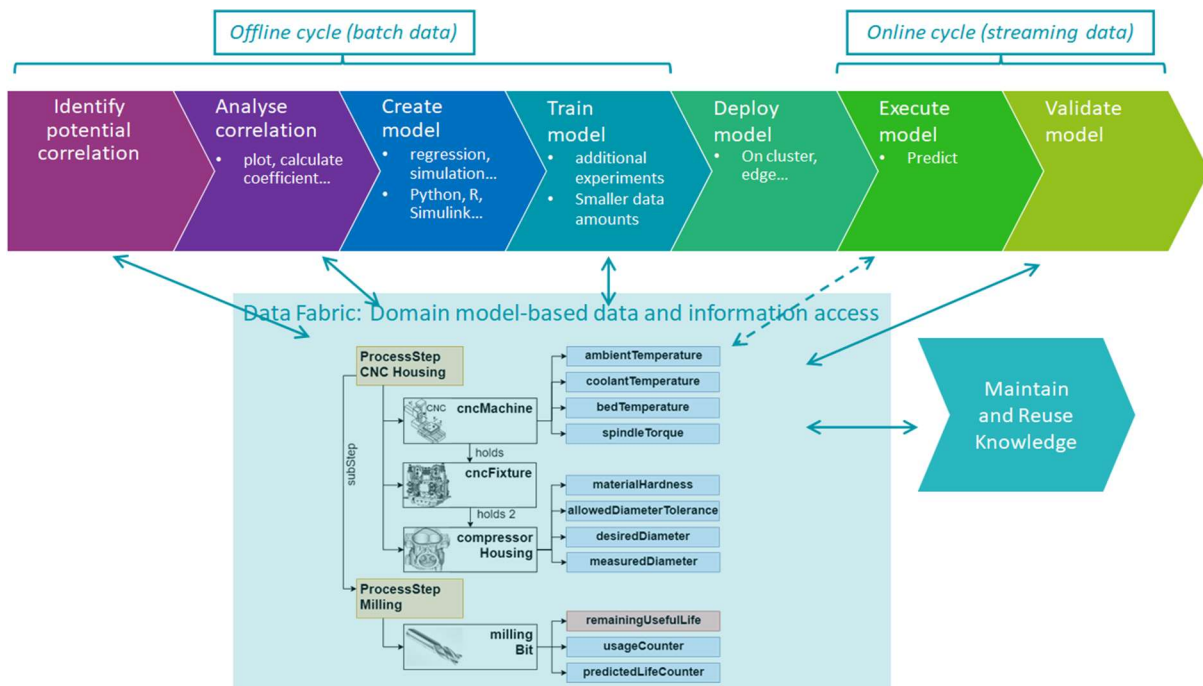


Figure 19: Scenario illustration - AI life cycle integration with the data fabric

1. investigating correlations: potential correlations are identified. This is based on expert knowledge, which is stored in the domain model. For instance, a domain expert may have indicated in the domain model that a correlation between parameters may exist and would be interesting to investigate. Such potential correlations, including uncertainties related to these correlations, can be extracted from the digital twin.
2. analyzing correlations: independents and dependents are analyzed to investigate whether a correlation can be seen in historical data. To do this, historical data needs to be accessible through the data fabric. This data resides in different databases, at different locations, in different formats. The data fabric must unify the data.
3. creating a(n) (AI) model: a data analyst creates a model, e.g., for predictive and prescriptive analysis of quality variance. This can be an AI model, or a simulation. This can be based on prior knowledge, stored in the domain model.
4. training a model: the model is trained using historical data. This may involve deployment to e.g., a Databricks platform. The data fabric must make data available in such data analysis platforms.
5. deploying a model: the model must be deployed in the cloud or on an edge device. The model must be deployed such that it is executed with streaming data and such that its results can be acted upon immediately and stored back into the data fabric.
6. executing a model: the model is executed with streaming data, and the results (e.g., predictions) are visualized to e.g., an operator, and/or used as input (e.g., instructions)

to change production settings) to improve the production process. Results can also be stored back into the data fabric for future use, such as model maintenance.

7. validating a model: the results of the model are validated by comparing the model's outputs to actual measurements. The model can be retrained or adapted as a result.
8. maintenance and evolution: all collected information (correlation coefficients, data analysis experiments, uncertainties, historical data, predicted values, etc.) can be reused. The lifecycle should be reproducible, or reusable with new data, knowledge, inputs etc. The data fabric must support evolution, reuse, and maintenance.

















This scenario will be validated on the Atlas Copco use case and can be iteratively developed to result in Atlas Copco's envisioned Adaptive Measurement Strategy and Virtual Assistant.

In this scenario, the data fabric will collect, link and curate shop floor sensory data in real time and securely store this on infrastructure as required by the use case provider. This could be in the cloud, on premise, etc. For this, a broker (e.g., Apache Kafka) must collect the streaming data and store it in a data lake. For applications on real-time control in WP5, the broker must also use the streaming data as input for deployed models. For these use cases, the broker needs to be deployed on edge devices (as opposed to a broker in the cloud), making real-time interaction possible.

6. Requirements Validation

As already mentioned in the deliverable D3.1 of the ASSISTANT project, there exist well-known procedures to validate requirements. According to GRANDE (2011, p. 83), the results of requirements validation must be recorded using checklists. With the help of the formulation rules, test principles, and questions for requirements validation according to GRANDE (2011, 79 ff.) and KLEUKER (2013, 88 ff.), the following test criteria are derived Table 8 for validating the requirements.

Table 8: requirements validation

Requirements Validation			
Criteria	Fulfillment	Criteria	Fulfillment
Short sentences		Suitable form of documentation	
One requirement per sentences		Construction of development artefacts	
Justification of Requirements		Repeated testing with the help of stakeholders	
Avoid passive		Separation of troubleshooting and correction	
Solution neutral		Different stakeholder perspectives	
Testable and verifiable		Formal examination: spelling and grammar	
Uniqueness of the subject		Bidirectional traceability	
Right stakeholders		Documentation of the core functionality	

While numerous test criteria such as the construction of development artifacts or the justification of requirements are fully covered by the developed requirements in D6.1, not all requirements are formulated in a solution-neutral way. This is due to the iterative approach of requirements and system development as well as the interaction and necessary trade-off between requirements and solution approaches. Formulating holistic solution-neutral requirements leads to statements that offer no added value due to a lack of testability and implementability (LAUENROTH 2015, p. 2). In addition, not all requirements are testable or verifiable, although acceptance and test criteria can be defined based on these requirements

(POHL & RUPP 2011, p. 105). While the right stakeholders are identified during requirements elicitation and both internal and external stakeholder perspectives are considered, an expansion is desirable due to the high relevance of stakeholders. In the context of the present work, numerous scientific and industry publications are used as stakeholders.

To validate the requirements identified, the procedure of existing publications is used (see deliverable D3.1).

7. Conclusion

The ASSISTANT data fabric is designed to meet the requirements regarding data storage, management, and provenance for the project. As the key technical outcomes of the project are the digital twins, many of the technical requirements of these tools indirectly influence the requirements of the underlying foundational data fabric. This document presents an overview of the technical design and context of the data fabric and details the requirements with respect to data management and storage that are deemed essential for the success of the project. This document is presented as a standalone deliverable of the project and used as input to the design process and architecture of the project, in particular the architecture design and prototype implementation of the WP6 data fabric systems to be documented in deliverables D6.2, D6.3, and D6.4.

The purpose of this document is to position the work within the project and to document the requirements identified for the development of the data fabric. To avoid duplication of information and repetition, requirements related to the digital twins developed in work packages 3 - 5 are documented in their respective requirements deliverables (D3.1, D4.1, and D5.1). The data fabric will be delivered in four steps: First, this document positions the data fabric by outlining the project data storage and management needs and deriving functional and non-functional requirements from these. Second, in deliverable D6.2 (due M12), a technical architecture for the data fabric platform and its integration with other tools will be presented. After these, a prototype implementation of the data fabric architecture will be delivered in two steps - a preliminary version in Deliverable D6.3 (due M24) and a refined version in Deliverable D6.4 (due M36).

8. References

- [1] The Forrester Wave™: Enterprise Data Fabric, Q2 2020 - Retrieved 2021-04-12
<https://www.forrester.com/report/The+Forrester+Wave+Enterprise+Data+Fabric+Q2+2020/-/E-RES157288>
- [2] Talend Data Fabric Retrieved 2021-04-12
<https://help.talend.com/r/bRYIPnVJVyUAjbRmezEobA/6Zkus9zg5FPGHJM-AHsL6A>
- [3] Denodo-Platform-Overview Retrieved 2021-04-12
<https://www.denodo.com/en/denodo-platform/overview>
- [4] Architecture for IBM Cloud Pak for Data-Retrieved 2021-04-12
https://www.ibm.com/support/producthub/icpdata/docs/content/SSQNUZ_lates t/cpd/plan/architecture.html
- [5] Cloudera Data Platform - Retrieved 2021-04-12
<https://www.cloudera.com/products/cloudera-data-platform.html>
- [6] Enterprise Data Operations and Orchestration for Cloud and Hybrid Environments - Retrieved 2021-04-14
<https://www.infoworks.io/products/overview/>
- [7] Alvord, M.M., Lu, F., Du, B. and Chen, C.A., Big Data Fabric Architecture: How Big Data and Data Management Frameworks Converge to Bring a New Generation of Competitive Advantage for Enterprises.
- [8] Data Fabric: The Next Generation of Data Management - Retrieved 2021-04-12
<https://info.stardog.com/data-fabric-whitepaper>
- [9] The scalable knowledge graph platform for data integration and analytics. Retrieved 2021-04-12
<https://www.cambridgesemantics.com/anzo-platform/>
- [10] Ghiran, A.M. and Buchmann, R.A., 2019, August. The model-driven enterprise data fabric: A proposal based on conceptual modelling and knowledge graphs. In International conference on knowledge science, engineering, and management (pp. 572-583). Springer, Cham.
- [11] Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho (2004). Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce, and the Semantic Web. Springer, 2004.
- [12] Thomas Kühne: Matters of (Meta-)Modeling. *Softw. Syst. Model.* 5(4): 369-385 (2006)
- [13] Grady Booch, James E. Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide. *J. Database Manag.* 10(4): 51-52 (1999)
- [14] <http://www.omg.org/mof/>
- [15] <http://eclipse.org/emf/>
- [16] <https://www.cambridgesemantics.com/anzo-platform/>
- [17] <https://www.ontotext.com/products/ontotext-platform/>
- [18] <https://rml.io/>
- [19] <https://www.w3.org/2001/sw/wiki/Ontop>
- [20] Emna Hlel, Salma Jamoussi, Abdelmajid Ben Hamadou: A New Method for Building Probabilistic Ontology (Prob-Ont). *Int. J. Inf. Technol. Web Eng.* 12(2): 1-25 (2017)
- [21] Setiawan F.A., Wibowo W.C., Ginting N.B. (2015) Handling Uncertainty in Ontology Construction Based on Bayesian Approaches: A Comparative Study. In: Intan R., Chi CH., Palit H., Santoso L. (eds) *Intelligence in the Era of Big Data. ICSIIT 2015. Communications in Computer and Information Science*, vol 516. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-46742-8_22
- [22] Konstantin Todorov, Peter Geibel, Céline Hudelot: Building A Fuzzy Knowledge Body for Integrating Domain Ontologies. *URSW 2011*: 3-14

9. Appendix

9.1 Abbreviations

Table 9: Abbreviations

Abbreviation	Meaning
ASSISTANT	LeArning and robuSt decision SupportT systems for agile mANufacTuring environments
DT	Digital Twin
ERP	Enterprise Resource Planning
MES	Manufacturing Execution System
ETL	Extract, Transform, Load (data warehousing methodology)
OP	Operation
AI	Artificial Intelligence
UML	Unified Modeling Language
API	Application Programming Interface
ML	Machine Learning
NLP	Natural Language Processing
CNC	Computer Numerical Control
HMLV	High Mix Low Volume
CAD	Computer-Aided Design
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
XML	Extensible Markup Language
RML	RDF Mapping Language
MOF	Meta-Object Facility
GUI	Graphical User Interface
MVC	Model-View-Controller
I/O	Input/Output
SOA	Service Oriented Architecture
IT	Information Technology
OEE	Overall Equipment Effectiveness
RFID	Radio Frequency Identification
OTD	On-Time-Delivery
SQL	Structured Query Language
OWL	Web Ontology Language
SCADA	Supervisory Control And Data Acquisition
CAQ	Computer - Aided Quality Assurance
CMM	Coordinate Measuring Machine
AWS	Amazon Web Services
SAP	Systems, Applications and Products for data processing
EMF	Eclipse Modeling Framework
XSD	XML Schema Definition
VR	Virtual Reality
AR	Augmented Reality
ISO	International Organization for Standardization
SOAP	Simple Object Access Protocol
STEP	STandard for the Exchange of Product model data
KPI	Key Performance Indicator