

Deliverable D1.3

Requirements and base line update 2

Work package leader: Elisa Londero
elisa.londero@eurotech.com

Gianvito Urgese
gianvito.urgese@polito.it

Editor: Elisa Londero
elisa.londero@eurotech.com

Abstract

This document constitutes deliverable D1.3 of the Arrowhead fPVN project.

WP1 objective is to gather requirements regarding the Common Arrowhead Technology, the Microservices paradigm, the Major fPVN data models and the Automated data model translation. WP1 further collects and summarizes common technology and use-case baselines with respect to the project objectives, providing a consolidated way of validating and verifying the project advancement and objectives fulfilment.

Table of contents

1.	Introduction	3
1.1	Overview of WP1 tasks	3
1.2	Task 1.1 role and objective	3
1.3	Task 1.2 role and objective	3
2.	Requirements elicitation.....	5
2.1	The nature and role of requirements	5
2.2	The elicitation process.....	6
2.3	Requirement elicitation and evaluation: second cycle.	7
2.4	Requirements fulfilment final assessment.....	10
2.4.1	Requirement fulfilment summary by use case	11
3.	The baselines	13
3.1	The concept of “baseline”	13
3.2	Building blocks baselines.....	14
3.2.1	Micro-service paradigm	15
3.2.2	Major digital languages.....	22
3.2.3	Autonomous translation	27
3.3	Use case baselines	37
3.3.1	WP6 - Use-case #1.6 – Automotive Battery Innovation fPVN.....	40
3.3.2	WP6 - Use-case #2.6 – Humans in the interoperable System.....	57
3.3.3	WP7 - Use-case #1.7 – Interoperable intelligent management of production lines: Towards Model-based Enterprise	76
3.3.4	WP7 - Use-case #2.7 – Aircraft Health Management System (AHMS) for Trend Monitoring, Predictive Maintenance and Fleet Operations & Maintenance Simulation .	89
3.3.5	WP9 - Use-case #1.9 – Pump Station Engineering.....	104
3.3.6	WP9 - Use-case #2.9 Digital Twins that enable higher performance by interoperability in pulp mills and carton board mills	118
3.3.7	WP9 - Use-case #3.9 – Interoperability for technical information exchange in process industry.....	139
4.	Conclusions	152
5.	Annex	154
6.	Revision history.....	155
6.1	Contributing and reviewing partners.....	155
6.2	Amendments.....	155
6.3	Quality assurance	155

1. Introduction

This deliverable reports the results achieved during the third year of the project regarding the requirements elicitation and their evaluation, the use cases baseline definition, update, refinement, and consolidation. The deliverable describes the activities carried out in Task 1.1 and Task 1.2 during the reporting period M24-M36.

In the first part, the deliverable provides a recap about the concept of requirements, illustrating the process adopted for the requirements elicitation and for the evaluation of their fulfilment, providing a complete analysis of the status at M36.

In the second part, the deliverable provides a recap of the methodology adopted for the analysis of the baselines and reports the results of the third iteration of the baseline definition, for both the three main building blocks of the projects (horizontal WPs) and for the use cases.

The main objective of D1.3 is to provide the final evaluation of requirements fulfilment, the final version of the baselines and the final evaluation of the related KPIs.

1.1 Overview of WP1 tasks

WP1 is composed of two tasks:

- Task 1.1 Requirements
- Task 1.2 Use case baseline

1.2 Task 1.1 role and objective

Task 1.1 is responsible for the elicitation of the project requirements, involving all the project's partners, with the tasks and use cases leaders generally providing the requirements, and horizontal WP/Tasks leaders as potential suppliers of the required technologies/solutions. Requirements are specified defining several information about their nature and relationship with the use case and project objectives and include specific details to allow tracking their fulfilment at the end of each project year.

The task must define a process for the requirements elicitation and fulfilment assessment, which aligns with the project engineering cycles and provides the possibility to update, refine, and add new requirements in every cycle. This flexibility is required by the nature of the research activities carried out in the project, by the complexity of the horizontal technologies and use cases, and to ensure the continuous alignment with the progress and evolution of the project.

1.3 Task 1.2 role and objective

Task 1.2 is responsible for the definition of the use case technology baselines, a set of snapshots that illustrate the status of the architecture, technologies, and solutions adopted in a use case at M0: the baseline is the reference point for the evaluation of project improvements. Initially, the

baseline was focused only on the use cases, but we decided to define a baseline also for horizontal work packages to keep track of the improvements introduced by the technologies/solutions that has been developed in the project when compared to M0.

Considering the complexity and heterogeneity of the baselines and of their analysis, the task adopts a common methodology for the baseline definition and provides surveys to support the analysis and ensure the collection of uniform and coherent information across the different use cases and horizontal WPs. The surveys guide the WP and use case leader in the analysis of the baseline and represent a good exercise intended to support a detailed assessment of the situation at M0.

Task 1.2 is in charge of analysing and aggregating the baselines information, creating a report (D1.1, D1.2 and D1.3) that is updated, refined and potentially extended in every engineering cycle of the project. This flexibility provides the partners with the time required to acquire the right level of expertise in the horizontal technologies and use cases (which is not always available at the beginning of the project), and ensures a high quality of the baselines analysis, which results in a more valuable validation and evaluation of project results at M36.

2. Requirements elicitation

This chapter describes the concept of requirement and illustrates the process for requirements elicitation. During the second year of the project, the preliminary requirements defined in the previous cycle have been revised, updated and refined. In the last part of the year, the fulfilment of requirements has been evaluated. The final list of requirements and their evaluation is available as an annex in Excel format.

2.1 The nature and role of requirements

The requirements describe the relationships between the actual concrete results developed in the work packages and the project objectives they satisfy. In general, they reflect the demand and supply relationship between the use cases and the horizontal WPs (WP2, WP3 and WP4) but are not limited to it: the request for a requirement defined in a use case can be satisfied by the use case itself. Requirements are not a wish list: identifying a requirement means that there is a need for what the requirement defines, and those that provide the requirement should be ready to help assess and explore the potential results intended to fulfil the requirement.

Requirements are defined in the form of a matrix to get a comprehensive and uniform record of the operational goals at the start of the project. The matrix gives, in a compact way, an overview of the expected concrete outcomes to record and improve collaboration between WPs. It has been used as a concrete checklist for achieving the project objectives as results are ticked off.

Requirements definition includes data and metadata related to the nature of the requirement, its description and motivation, and also contains information about how the requirements themselves relate to the project organization and to its objectives. Requirements may be defined at different abstraction levels and may relate to small technology details or larger systems and solutions, but they must be limited in number and must focus on the adoption of the Building Blocks and on the project's objectives. There isn't an absolute limit, but just a soft and reasonable rule to ensure we don't miss the main targets of the projects, avoiding getting lost in an endless list of technical requirements with excessive granularity.

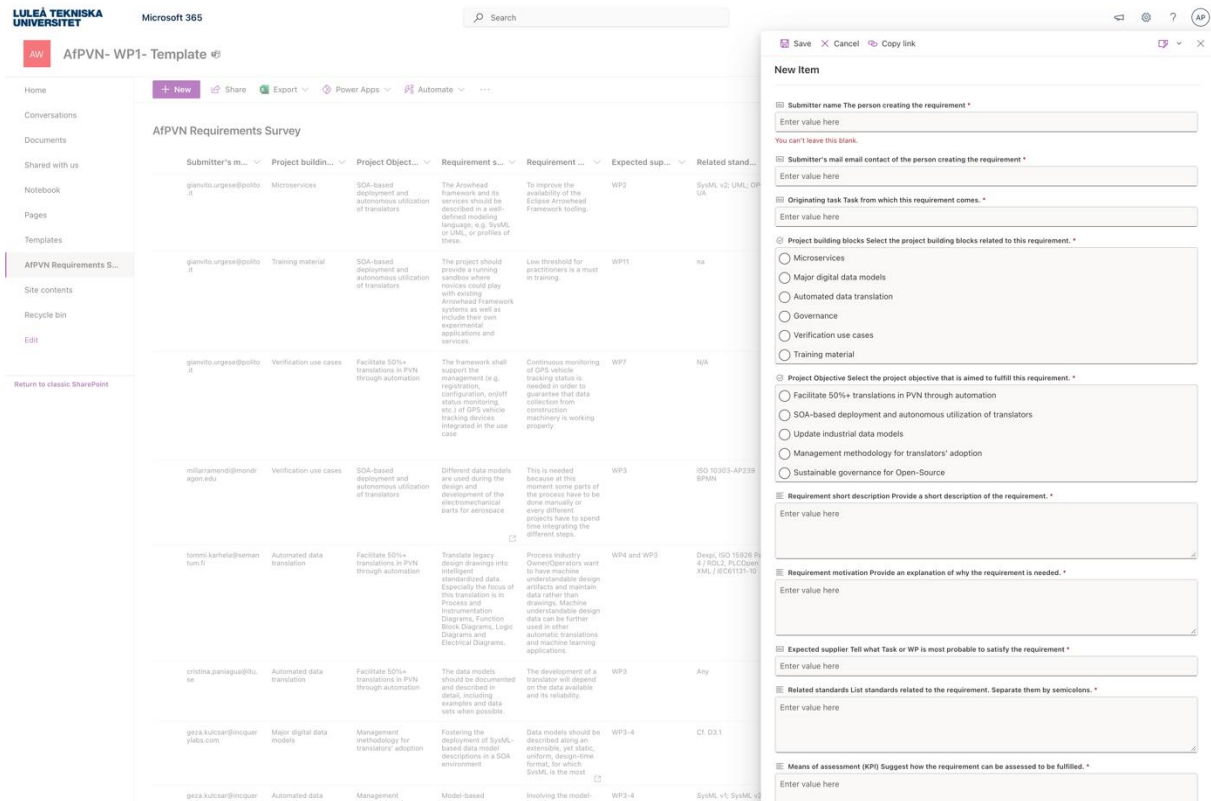
The definition of requirement requires the specification of the following information:

- **ID:** requirement ID, automatically generated.
- **Submitter name:** the person providing the requirement.
- **Submitter's mail:** the email contact of the person providing the requirement.
- **Originating task:** the task from which this requirement originated.
- **Project building blocks:** the project building blocks related to this requirement, namely:
 - Microservices.
 - Major digital models.
 - Automated data translation, Governance.
 - Verification use cases.
 - Training material.
- **Project Objective:** the project objective that is aimed to fulfill this requirement, namely
 - Facilitate 50%+ translations in PVN through automation.
 - SOA-based deployment and autonomous utilization of translators.

- Update industrial data models.
 - Management methodology for translators' adoption.
 - Sustainable governance for Open-Source.
- **Requirement short description:** simply a very short description of the requirement.
- **Requirement motivation:** an explanation of why the requirement is needed.
- **Expected supplier:** the Task or WP that is expected to fulfil the requirement.
- **Related standards:** the list of relevant standards related to the requirement.
- **Means of assessment (KPI):** explain how the requirement fulfilment can be assessed.
- **Validating Use Case:** the use case in which the requirement is validated.
- **Validation Means:** explanation of how the use case has validated the requirement.
- **Y1 Fulfilment (%):**1st year assessment of the fulfilment of the requirement in percentage.
- **Y1 Evaluation:** explanation of the fulfilment percentage achieved in the 1st year.
- **Y2 Fulfilment (%):**2nd year assessment of the fulfilment of the requirement in percentage.
- **Y2 Evaluation:** explanation of the fulfilment percentage achieved in the 2nd year.
- **Y3 Fulfilment (%):**3rd year assessment of the fulfilment of the requirement in percentage.
- **Y3 Evaluation:** explanation of the fulfilment percentage achieved in the 3rd year.

2.2 The elicitation process

To simplify the requirement elicitation process and to make it more efficient, we identified some steps to be followed, and we set up a SharePoint web site providing a form for the definition of the requirement. This approach ensures that the information we collect is uniform and coherent and provides Microsoft Office functionalities to elaborate on the collected data and the export in Excel format (see the following figure).



Submitter's m...	Project buildin...	Project Object...	Requirement s...	Requirement ...	Expected sup...	Related stand...
glenito.urgese@polito.it	Microservices	SOA-based deployment and autonomous utilization of translators	The Arrowhead framework and its services should be described in a well-defined modeling language, e.g. SysML or UML, or profiles of these.	To improve the availability of the Eclipse Arrowhead Framework tooling.	WP3	SysML v2; UML, CP UA
glenito.urgese@polito.it	Training material	SOA-based deployment and autonomous utilization of translators	The project should provide a running sandbox where students could play with existing Arrowhead Framework systems as well as include their own experimental applications and services.	Low threshold for practitioners is a must in training.	WP1	na
glenito.urgese@polito.it	Verification use cases	Facilitate 50%+ translations in PVN through automation	The framework shall support the management (in.g. registration, configuration, on/off status monitoring, etc.) of GPS vehicle tracking devices integrated in the use case.	Continuous monitoring of GPS vehicle tracking status is needed in order to guarantee that data collection from construction machinery is working properly.	WP7	N/A
milarremond@mendi.agri.edu	Verification use cases	SOA-based deployment and autonomous utilization of translators	Different data models are used during the design and development of the electromechanical parts for aerospace	This is needed because at the moment some parts of the process have to be done manually or every different projects have to spend time integrating the different steps.	WP3	ISO 10303-AP239 BPMN
tommi.kahela@seman.tun.fi	Automated data translation	Facilitate 50%+ translations in PVN through automation	Translate legacy design drawings into intelligent, standardized data. Especially the focus of this translation is in Process and Instrumentation Diagrams, Function Block Diagrams, Logic Diagrams and Electrical Diagrams.	Process Industry Owners/Operators want to have machine understandable design artifacts and maintain data rather than drawings. Machine understandable design data can be further used in other automatic translations and machine learning applications.	WP6 and WP3	Devops, ISO 15926 P4 / R02, PCCOpen XML / SC01131-30
crstine.paragade@fu.se	Automated data translation	Facilitate 50%+ translations in PVN through automation	The data models should be documented and described in detail, including examples and data sets when possible.	The development of a translator will depend on the data available and its reliability.	WP3	Any
gera.kulcaran@roqueet.ylaba.com	Major digital data models	Management methodology for translators' adoption	Fostering the deployment of SysML-based data model descriptions in a SOA environment	Data models should be described along an extensible, yet stable, uniform, design-time format, for which SysML is the most	WP3-4	CI, D3.1
gera.kulcaran@roqueet.ylaba.com	Automated data translation	Management methodology for translators' adoption	Model-based architectures	Involving the model-based architectures	WP3-4	SysML, v1, SysML, G2

Figure 1: the Sharepoint web site for requirements elicitation.

The requirements matrix mentioned in the Document of Action is available in the form of an Excel file (see Annex 1).

The link to the SharePoint web site was sent to the WP leaders of WP2, 3 and 4 and to the use cases leaders, but we also informed all the project's partners because, in principle, all the consortium partners could define requirements. As expected, at the beginning there have been some difficulties in the requirement elicitation, due to the complexity of horizontal technologies and of the use cases, but also because of the habit of going into technical details immediately without considering the project objectives as a reference. This problem caused a delay in the elicitation process but gradually resolved, and the first preliminary set of requirements is currently available.

The requirements elicitation process has been cyclical and allowed the project partners to review, refine, update, extend the existing requirements and add new one: the process has been a continuous activity during the year, thanks to the SharePoint web site that allows the online continuous editing of the requirement matrix. On the contrary, at the end of each project year we evaluated the consolidation of requirements and their level of fulfilment.

2.3 Requirement elicitation and evaluation: second cycle.

In the second iteration we ended up **with 50 requirements**, trying to cover uniformly the adoption of the Building Block in the use cases (see Annex 1). The following table reports the requirements distribution.

Originating WP/Task/UC		Number of Requirements
WP2		3
WP3		2
WP4		2
T6.1	UC1.6	6
T6.2	UC2.6	6
T7.1	UC1.7	7
T7.2	UC2.7	7
T9.1	UC1.9	6
T9.2	UC2.9	5
T9.3	UC3.9	6
Total.:		50

Table 1: Requirements distribution.

Table 1 highlights the structured yet differentiated approach adopted across the project in the definition of requirements. This distribution reflects both the role of each task within the project and the complexity of the domain addressed.

The horizontal Work Packages (WP2, WP3, WP4) present a small but carefully scoped number of requirements. Horizontal WPs provide the building blocks and serve as enabling layers, defining interoperability mechanisms, data handling standards, and translator technologies that underpin and support all vertical use cases. Their limited number is intentional, reflecting a focus on *foundational functionality* rather than domain-specific implementation.

Among the use cases, the number of requirements varies (ranging from 3 to 7) and serves as a proxy for the breadth and integration depth of each use case:

- **T7.1 (UC1.7)** and **T7.2 (UC2.7)** are characterized by broad functional coverage involving both technical integration (e.g. translator deployment, MBSE integration) and value-added layers (e.g. decision-support algorithms).
- Use cases **T6.1 (UC1.6)** and **T6.2 (UC2.6)** requirements focus on challenges like full lifecycle traceability and human-in-the-loop interoperability.
- At the other end of the spectrum, **T9.1**, **T9.2** and **T9.3** suggest a focused scope, tailored to specific domains (e.g., technical documentation, digital twins), where interoperability needs are narrower but still significant.

This distribution reflects the project's balanced methodology: while all tasks align under common technological frameworks, the flexibility to define a task-specific number of requirements has allowed each team to shape their implementation roadmap according to real industrial needs. This balance between methodological consistency and domain customization strengthens the overall project coherence.

The diversity in requirement counts reveals how the project has tailored complexity to context, ensuring that both horizontal enablers and vertical demonstrators are adequately addressed without unnecessary redundancy.

Project Objective	Number of Tasks
Facilitate 50%+ translations in PVN through automation	15
Management methodology for translators' adoption	9
SOA-based deployment and autonomous utilization of translators	26

Table 2: requirements and project objectives.

Table 2 illustrates how requirements contribute to the project objectives.

The table confirms that all key technical objectives (Objectives 1–3) of the project are actively addressed by various tasks, demonstrating a strong alignment between project activities and its strategic goals. In particular:

- **Objective 2** (“Microservices/SOA enabling dynamic deployment ...”) emerges as the most addressed objective, with 26 different requirements contributing to its fulfilment. This highlights the project’s emphasis on realizing an open, modular microservices architecture for dynamic and autonomous information translation. This is consistent with the project’s ambition to develop and validate a TRL6 platform based on Eclipse Arrowhead framework, as evidenced by its focus in several use cases such as 1.7, 2.7, and 1.9.
- **Objective 1** (“Facilitate more than 50% of needed translations ...”) is also well covered, with 15 tasks supporting the goal. This indicates a broad commitment across the project to reduce human involvement in translations within production value networks through the development of autonomous translator microservices. The wide task distribution implies a deliberate strategy to develop, test, and adopt translation technology in diverse industrial contexts.
- **Objective 3** (“Update proposals for major digital data model...”) sees 9 tasks linked to it, reflecting the importance of ensuring semantic interoperability and extending existing industrial data standards. This focus supports the broader aim of enabling seamless PVN lifecycle management and is in line with the deliverables and efforts targeting standardization contributions.

While Objectives 1–3 are clearly supported by numerous technical tasks across the work packages and use cases, Objectives 4 and 5 are broader, strategic goals that are not primarily fulfilled through technical implementation, but rather through methodological, organizational, and governance-related activities.

- **Objective 4:** “Develop and validate a digital transformation management methodology...”
This objective focuses on organizational change, process innovation, and stakeholder adoption. Unlike the other objectives, it is less about developing new technologies or systems and more about embedding them within real-world business and operational contexts. As such, it is likely being addressed through dissemination, business modelling, stakeholder engagement, and evaluation activities—not typically captured within individual technical requirements or tasks. For instance, validation of digital transformation approaches may come through pilot-scale deployment feedback, training, or business model adjustments, which are often managed in WP8 or through partner internal processes.

- **Objective 5:** *“Establish and validate a sustainable governance model for Eclipse Arrowhead...”*

This objective relates to the long-term sustainability, community coordination, and governance structure of the Eclipse Arrowhead platform. Rather than being a development target, it is about ensuring that the platform will be maintained, extended, and adopted beyond the project’s lifespan. It involves interactions with the Eclipse Foundation, licensing models, documentation policies, and the engagement of an open-source community. These efforts are inherently transversal and institutional, and while critical to impact, they are not expressed as individual technical requirements or deliverables in the typical WP tasks.

Building Block	Number of Requirements
Automated data translation	12
Major digital data models	9
Microservices	29

Table 3: requirements and Building Blocks

The distribution of requirements across the three horizontal Building Blocks of the project reflects a clear strategic focus. The largest share of requirements (29 in total) relates to the development and use of Microservices, underlining the project’s strong emphasis on a scalable, modular, and service-oriented architecture. This aligns with the project’s overall goal of achieving system interoperability and flexibility across diverse industrial contexts.

Automated Data Translation accounts for 15 requirements, highlighting the importance placed on enabling seamless data exchange between heterogeneous systems. This is essential for the realization of end-to-end interoperability in project use cases, especially where toolchains and industrial platforms vary significantly.

Major Digital Data Models are addressed by 9 requirements. While this represents the smallest share among the three blocks, it still indicates a targeted effort to standardize and structure data representations, which is critical for ensuring long-term compatibility and information consistency across applications.

Overall, the requirements distribution supports the project’s core technological pillars and confirms a well-rounded technical strategy, with a predominance of microservices as the structural backbone for achieving interoperability.

2.4 Requirements fulfilment final assessment

Regarding the evaluation of requirements fulfilment at the end of year 3, the project shows significant achievements. The fulfilment rate of requirements by use cases reached about 88% on average, which confirms the positive validation of the concepts developed in the project for many different industries (batteries' lifecycle management, industry HMI, aerospace manufacturing & maintenance, infrastructure management of cities, and digitalization of the process industry). The highest rates of fulfilment were seen for UC1.7, UC2.7, UC2.6, and UC3.9, with nearly all requirements met through the implementation of autonomous translation systems, service-oriented architectures, standardized models of data, and interoperable digital ecosystems. As regards UC2.9, the highest degree of fulfilment was achieved due to the

deployment of a knowledge graph-based digital twin and semantic interoperability. In turn, UC1.6 and UC1.9 showed lower fulfillment, mainly because of some external factors affecting the process of validation in industry and organizational problems associated with large-scale digital transformation, and not because of any technological limitations or failures. Overall, the project managed to address the vast majority of industrial needs and proved the usability of Arrowhead interoperability principles, autonomous translation, semantic interoperability methods, and SOA.

2.4.1 Requirement fulfilment summary by use case

UC1.6 (WP6) achieved a weighted requirement fulfilment index of **72.1%**. The use case successfully demonstrated the integration of battery lifecycle information through the Digital Twin architecture and achieved the main interoperability and data-sharing objectives. The remaining gap is primarily related to the limited industrial-scale validation caused by the withdrawal of Northvolt from the consortium, which prevented full validation in an operational industrial environment.

UC2.6 (WP6) achieved a weighted requirement fulfilment index of **96.2%**, representing one of the highest fulfilment levels among all use cases. The developed acoustic Human–Machine Interaction framework successfully satisfied almost all requirements, including speech-command recognition, Arrowhead integration, industrial validation, and the creation of the Industrial Noise Corpus. Only minor refinements related to future industrial deployment remain open.

UC1.7 (WP7) achieved a weighted requirement fulfilment index of **97.6%**. The use case successfully fulfilled almost all requirements related to autonomous translation, model-based systems engineering, service-oriented architectures, and logistics optimization. The integration of multiple translators, MBSE methodologies, and Arrowhead services confirms the high maturity reached by the demonstrator.

UC2.7 (WP7) achieved a weighted requirement fulfilment index of **100%**, representing the highest fulfilment level among all use cases. All identified requirements were successfully implemented and validated through the AHMS demonstrator, including AeroBridge translation services, S5000F adoption, Arrowhead orchestration, and automated aircraft maintenance workflows.

UC1.9 (WP9) achieved a weighted requirement fulfilment index of **64.8%**, the lowest among the project use cases. Nevertheless, the use case successfully established standardized information structures, pilot microservice integrations, and a clear roadmap for digital transformation. The lower fulfilment level is mainly related to organizational, governance, and open-source adoption challenges rather than technical shortcomings.

UC2.9 (WP9) achieved a weighted requirement fulfilment index of **91.7%**. The use case successfully implemented the Knowledge Graph-based Digital Twin and demonstrated semantic interoperability across multiple engineering, operational, and maintenance systems. Most requirements were fully addressed, with only limited remaining activities related to long-term governance and deployment aspects.

UC3.9 (WP9) achieved a weighted requirement fulfilment index of **96.3%**. The use case successfully validated semantic interoperability, autonomous translation mechanisms, and runtime integration through Eclipse Arrowhead. The results confirm that the identified requirements for technical information exchange in process-industry value networks were largely fulfilled.

Use Case	Domain	Number of Requirements	Weighted Requirement Fulfilment
UC1.6 (WP6)	EV Battery Digital Twin	6	72.1%
UC2.6 (WP6)	Acoustic Human–Machine Interaction	6	96.2%
UC1.7 (WP7)	Aerospace Manufacturing	7	97.6%
UC2.7 (WP7)	Aircraft Health Management System	7	100.0%
UC1.9 (WP9)	Municipal Wastewater Infrastructure	6	64.8%
UC2.9 (WP9)	Knowledge Graph Digital Twin	5	91.7%
UC3.9 (WP9)	Technical Information Exchange in Process Industry	6	96.3%
Overall Project Average	—	43	88.4%

Table 4: requirement fulfilment summary.

3. The baselines

This chapter introduces the concept of “baseline”, provides a recap of the methodology adopted to analyse the building blocks (horizontal technologies) and use cases to define the correspondent baselines. Finally, it reports the results of the baseline status at the end of the second year.

3.1 The concept of “baseline”

A baseline is a snapshot in time that illustrates the status at M0 of a horizontal technology or a use case. It represents a reference point for the evaluation of the improvements introduced by the technologies developed in the project.

For the horizontal technologies the baseline covers the following aspects:

- The state of the art of specific enabling technologies in the domain of the building block.
- The action plan, starting from M0, to reach WP-specific and project-specific objectives.
- The identification of KPIs and the definition of an evaluation and validation process.

For the use cases the baseline covers the following aspects:

- The architecture of the use case as a sequential list of functional blocks.
- The relation with the project objectives.
- An explanation of how the building blocks will be adopted.
- The description of the engineering process adopted in the use case.
- The analysis of engineering costs.
- The adopted standards.
- An outlook of the previous aspects beyond the baseline, including an action plan to reach the project objective and a final evaluation process.

The WP leaders of the horizontal work packages are responsible for the definition of the baselines of the building blocks, while the use case leaders are responsible for the use case baseline definition. To simplify the baseline definition and to ensure that the analysis generates uniform and coherent information, we have defined two surveys which support and guide WP and use case leaders in this task. The templates of the two surveys are provided as an annex to these deliverables. (see Annexes 2 and 3).

Conceptually, the surveys have been conceived to align and follow the outline of the “project approach and ambitions”, as defined in the project proposal (see next figure and figure 1.13 of the DOA). The surveys are living documents that, starting from the first iteration providing the contents for D1.1, the second iteration for D.1.2 and final one providing the baseline finalization and final assessment in this deliverable (D1.3). The continuous updating of the baselines has allowed us to keep them consistent with the evolution of the project.

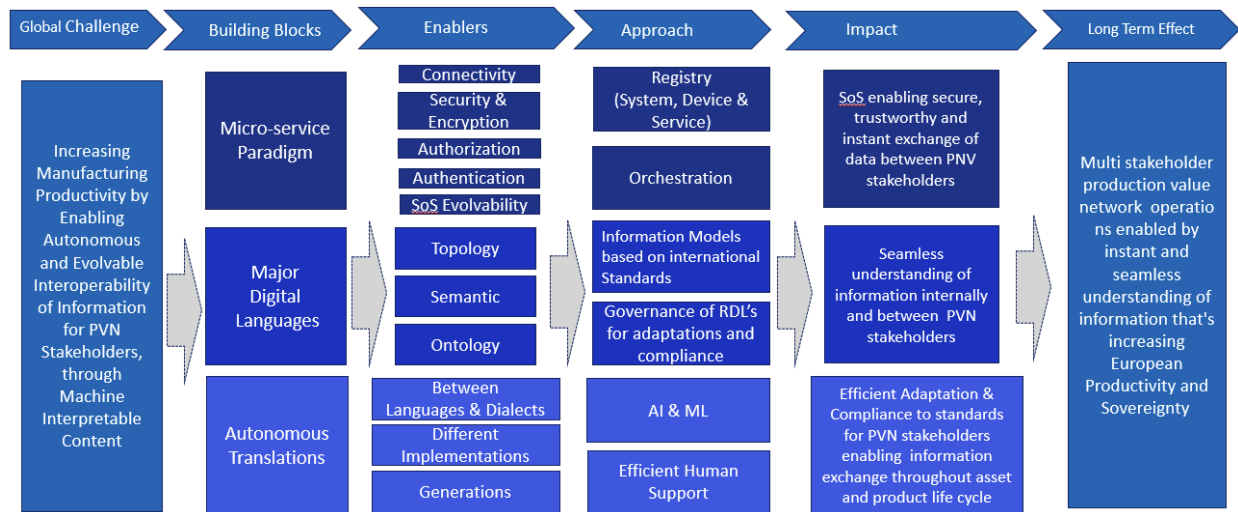


Figure 2: approach and Ambition outline in logical steps (from DOA).

3.2 Building blocks baselines

This section reports the analysis of the baseline of the main building blocks that have been designed and developed in the projects by the horizontal work packages:

- Micro-service paradigm,
- Major digital languages,
- Autonomous translations.

These baselines have been consolidated and finalised during the last year of the project.

At the end of the third year, the overall evaluation of the three building blocks reveals that the project was able to fill all the technological gaps that were observed at baseline analysis stage and make remarkable progress beyond the state of the art in the areas of service interoperability, semantic interoperability, and autonomous translation.

As far as the Micro-service Paradigm (WP2) is concerned, all the relevant KPIs have been reached and exceeded. The project helped greatly mature the Eclipse Arrowhead framework by bringing almost all core systems up to TRL 8 level and adding several new support systems, translators and interoperability mechanisms to the framework. Autonomous mismatch detection and translator allocation capabilities were implemented successfully. These achievements can serve as a strong technological base for constructing federated digital ecosystems and industry-level service-oriented architectures.

The Major Digital Languages building block (WP3) focused on tackling the problem of semantic interoperability between heterogeneous industrial standards. In this context, the project managed to create an alignment framework relying on the concept of IDO and to exceed its initial targets with regard to the recognition of semantic synergies and challenges and standards relations. The developed semantic analyses and alignment mechanisms constitute the foundation necessary for achieving scalable interoperability and translation autonomy.

The Autonomous Translation building block (WP4) managed to deliver on all established key performance indicators (KPIs) and to surpass most of the initially outlined targets. As a result, the WP carried out its research, design and development of ontology-based, AI-based and model-based translation methods, which resulted in the creation of seventeen prototype translations encompassing a variety of industrial standards, data models and communication technologies. Some of the translations were even incorporated into the Eclipse Arrowhead framework and tested in real-world situations. Overall, the average translation accuracy rate (approximately 85%) significantly surpassed the initially expected target.

3.2.1 Micro-service paradigm

3.2.1.1 Building Baseline summary

The grand challenge in the Microservices area is primarily related to the paradox of stagnating productivity despite rapid digitalization. In Arrowhead fPVN, the maturing and extension of the Eclipse Arrowhead architecture and implementation platform will take place. This enables the design and implementation of flexible automation and digitalization solutions based on run-time orchestration and workflow management. This approach puts Eclipse Arrowhead in a leading position as the open-source IoT and SoS platform opening up the IoT and SoS market for open competition and independent development.

3.2.1.2 Baseline analysis

The current industrial production baseline is an emerging use of micro services to enable flexible, dynamic, and scalable production-automation and digitalization solutions. This emerging use starts from handmade integration between multiple monolithic systems addressing one or several functionalities in the ISA-95 production architecture. The introduction of the microservice technology requires a high degree of interoperability at both communication technology and data level to enable integration of current state of the art with the more flexible microservice-based approach. A shift towards Service Oriented Architectures (SoA) has started to evolve for Industry 4.0 – one widely accepted solution is the Eclipse Arrowhead framework. Nevertheless, further expanding the SoA approach with more focus on the microservice paradigm itself and its technology advancements.

The paradigm shift from legacy monolithic systems and closed/proprietary protocol sets to the microservice approach opens for integrating legacy technology with open microservice technology. Protocol translators or adaptors are the primary approach. Using translators and adaptors legacy technology communication can be transformed into microservices. A very concrete example thereof is the OPC-UA to Eclipse Arrowhead adapter. Here OPC-UA to microservice interoperability at the communication and orchestration technology level (hardware and protocols) is demonstrated.

The current state for integrating legacy automation technology with the microservice paradigm consists of several open-source OT/IT integration frameworks like FiWare, Basyx, Eclipse Arrowhead, LWM2M, OCF, IoTivity, Autosar. None of these frameworks address autonomous support of the underlying issues, such as protocol and data model translation, which are clear technology gaps. The comparison indicates that Arrowhead has edge compared to other initiatives regarding Interoperability, Security, Edge computing and Real-time capabilities. Another knowledge and technology gap is autonomous identification of interoperability

mismatch. The ideas of dynamic and autonomous instantiating translators based on needs should be further investigated in the Microservice WP.

For Arrowhead fPVN, the Eclipse Arrowhead framework will be used as the implementation baseline for Arrowhead fPVN advancements beyond SoTA. Protocol and encoding translation use the novel translation and adaptor technology already released by the Eclipse Arrowhead project. This technology results from the Arrowhead Tools, Productive4.0, Arrowhead and other projects which are fundamental steppingstone knowledge to Arrowhead fPVN. For the baseline of the most important available Eclipse Arrowhead core systems the TRL level at M0 is defined in the following table.

At M36 we reached version 5.x for many core systems. This version And TRL M36 is also listed in the following table.

Arrowhead core system	Capability	TRL M0	version M36	TRL M36
ServiceRegistry	Registry for available service within a local cloud	7	v5.2.0	8
Orchestration	Responsible for distribution of orchestration rules to service consumers (Now manifested as separate microsystems)	7	v5.2.0	8
SimpleStore-Orchestration	Separate, new core system implementing the Simple Store Orchestration concept	7	v5.2.0	8
DynamicService Orchestration	Separate, new core system implementing the Dynamic Service Orchestration concept	7	v5.2.0	8
Authorisation	Primary system for checking authentication and authorization rule, ensuring edge security	7	v5.0	8
Authentication	Primary system for separate authentication checking	3	v5.2.0	8
SystemRegistry	Registry for available systems within a local cloud (Now integrated into ServiceRegistry)	6	v5.2.0	8
DeviceRegistry	Registry for available devices within a local cloud (Now integrated into ServiceRegistry)	6	v5.2.0	8
CertificateAutho- -rity	Responsible for distribution of Eclipse Arrowhead compliant X.509 certificates (Now functions included in Authentication.)	6	v5.2.0	8
On-boarding	Managing the on-boarding procedure to ensure only authorized devices, systems, and services will allow into a local cloud.	6	v4.6.2	6
DataManager	Primary data storage from service producers in a local cloud	7	v4.6.2	7
Gatekeeper and Gateway	Secure interoperability in-between Eclipse Arrowhead local clouds – control and data channel	7	v4.6.2	7
PlantDescription	Multi-view descriptor for plant configuration mapping	6	v4.6.2	6
EventHandler	Providing authorized publish-subscribe type, generic event handling capabilities	7	v4.6.2	7
Translation	Translating between HTTP-REST, CoAP, MQTT and Web socket protocols	6	v4.6.2	7
Translation Manager Support System	Including the previous Translator functions and further management functions for the latest translation concepts	5	v5.2.0	8

DeviceQoS-Evaluator Support System	The QoS concept provided by the Arrowhead fPVN project has this new evaluator system	4	v5.2.0	8
FiWare adaptor	Enabling interoperability from/to Arrowhead systems to/from the FiWare broker system.	5	v4.6.2	5
OPC-UA adaptor	Autonomous registration of OPC-UA services with Eclipse Arrowhead	5	v5.0.0	7
Z-wave adaptor	Autonomous registration of Z-wave services with Eclipse Arrowhead	5	v4.6.2	5
Modbus TCP adaptor	Autonomous registration of Modbus-TCP services with Eclipse Arrowhead	5	v4.6.2	5
ISO 10303 adaptor	Arrowhead adaptor enabling integration to ISO 10303 compliant tools like PLM, CAD	7 for spec. tools	v5.0.0	7
WEB of Things adaptor	Adaptor enabling Web of Things to be integrated to Eclipse Arrowhead	5	v5.0.0	7

Table 5: TRL level of Eclipse Arrowhead core systems at M0.

To enable the 10 use cases forecasted impact, a further maturing of a set of the Eclipse Arrowhead core system has been deemed necessary. Thus, maturing a set of Eclipse Arrowhead core systems and related engineering tools and procedures and providing a long-term development and maintenance governance structure is important for both short- and long-term exploitation of the technology.

Thus, the identified knowledge gaps are related to:

- Autonomous detection of non-interoperable properties of protocols and data models.
- Autonomous instantiating of necessary translations.
- Maturing of, and extensions to Eclipse Arrowhead core system and its engineering procedures and tools.
- Long term governance support for Eclipse Arrowhead.

3.2.1.3 Beyond the baseline

Engineering tools and procedures. The planned advancements are – Refinement and working demonstration of dynamic engineering process required to autonomously engineer and deploy identified translators of protocols and data models, targeting minimum TRL 6. The approach depends on the architectural mechanism of Eclipse Arrowhead regarding dynamic instantiating of system and services. This will require extensions to the various relevant modules of Eclipse Arrowhead core system like ServiceRegistry and Orchestration. This topic also covers various levels of governance of microservice architecture.

Within the project, the Arrowhead Engineering Process (Arrowhead-EP) is used as the common mapping model. It partitions a use-case engineering process into eight engineering process phases (EPPs) shown in the following Figure.

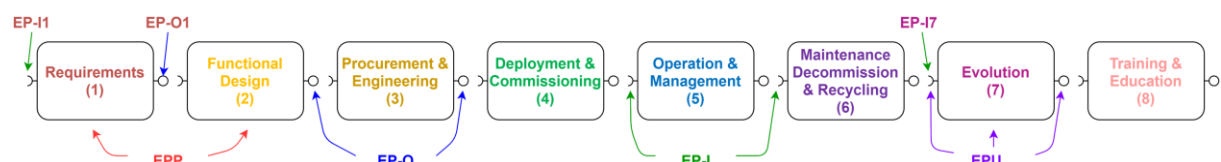


Figure 3: the Arrowhead Engineering Process.

Toolchains and tools are mapped to those phases; feedback and feedforward links are interpreted as service-based information flows between phases and stakeholders.

EPP	Arrowhead-EP phase	Use in this document
EPP1	Requirements elicitation	Elicit operational, business, stakeholder and standardization requirements.
EPP2	Functional design	Define functional architecture, data flows, interfaces, models and system boundaries.
EPP3	Procurement and Engineering	Develop or adapt tools, services, translators, adapters, models and datasets; procure or select needed components.
EPP4	Deployment and Commissioning	Install, integrate, test and validate the solution in a lab, pilot or operational environment.
EPP5	Operation and Management	Operate the system, exchange data, monitor services, execute business or production workflows.
EPP6	Maintenance, Decommissioning and Recycling	Maintain the solution and operational assets; manage repairs, service, updates, end-of-life and recycling where applicable.
EPP7	Evolution	Feed operational and maintenance evidence back to requirements and design; plan future releases and lifecycle improvements.
EPP8	Training and Education	Provide training, documentation, demonstrations and user guidance across the lifecycle.

Table 6: the phases composing the Arrowhead EP.

Maturing core systems of Eclipse Arrowhead. The planned advancements are to mature Eclipse Arrowhead core systems. Beside the main core systems will be made available at TRL 8, several supporting core systems, and adapter systems are going to reach TRL 6-8 according to Table 1.2 of the GA. This widens up the interoperability possibilities for legacy industrial systems and new system components alike, in all targeted application domains. The conceptual extensions as well as the reference implementations for validation and verification are governed through a dedicated road map process execution.

Maturing protocol and encoding interoperability translator and legacy adapters. The planned advancements are - to reach TRL6-8 according to Table 1.2 of the GA. The microservice paradigm will furthermore address autonomous identification of non-interoperability and autonomous instantiating of translation systems to resolve the identified non-interoperability. The approach here is based on analysis of data model metadata like e.g. used ontologies.

Furthermore, Translator-related Providers are also created by M36 of the Arrowhead fPVN:

- **InterfaceTranslatorToGenericHTTP** - v1.1.2 TRL7
- **InterfaceTranslatorToGenericMQTT** - v1.0.0 TRL-7
- **DataModelTranslatorWrapper** - v1.0.0 TRL-7

3.2.1.3.1 Technology action plan

The following actions are planned to be carried out in the Arrowhead-fPVN project to reach the goals in the **Engineering tools and procedures** section:

1. Creation of a cookbook for building Industry5.0-compliant microservice applications.

2. Gathering of architectural and design patterns of microservice based applications and SoS. These patterns should reside in the cookbook.
3. The definition and testing of a process for version and change management of services and systems in the Eclipse Arrowhead framework in some dimension.

The following actions are planned to be carried out in the Arrowhead-fPVN project to reach the goals in the **Maturing core systems of Eclipse Arrowhead** section:

1. Delivery of an Eclipse Arrowhead framework at TRL 7-8, potentially including Open Source and commercial applications.
2. Documentation of interconnection patterns and solutions with other IoT platforms.
3. Implementation of microservices that support interoperability, using automatic translators.
4. Design of means for Autonomous Contracting and Invoicing – to be able to commercialize the domain.
5. Development of third-party access management, in line with contracting and invoicing.
6. Design and develop an AI-based copilot for supporting developers in use case prototyping using the Eclipse Arrowhead technology.

The following actions are planned to be carried out in the Arrowhead-fPVN project to reach the goals in the **Maturing protocol and encoding interoperability translator and legacy adaptors** section:

1. A method or system that detects if non-interoperable interfaces reside in the SoS.
2. Autonomous allocation and assignment of translators when non-interoperable interfaces need to collaborate.
3. Support for security mechanisms in local and global clouds.

3.2.1.3.2 KPIs, Evaluation, and validation process

The Table 7 summarizes the draft KPIs defined for the autonomous translation evaluation and validation.

Key Performance Indicators (KPI)			Expected KPI value	Achieved KPI value
Name	KPI Description	Means of assessment		
2.1.1	Cookbook presence	Review, approve and use the cookbook	5	8
2.1.2	Number of patterns present in the cookbook.	Counting of implemented /verified patterns	3	6
2.1.3	Governance process	Percentage of WP defined services under version control.	3	8
2.2.1	TRL level of Eclipse Arrowhead framework components.	Assess percentage of components that have reached M36 TRL level.	5	8
2.2.2	Number of reports on interconnection analyses.	<TBD>	3	5

2.2.3	Number of available translators in Eclipse Arrowhead.	Counting	4	8
2.2.4	Existence of a process for Contracting and Invoicing	Review, approval and verification of implementation.	3	5
2.2.5	Existence of access control to components in the Eclipse Arrowhead framework.	Verification of a purchase process.	3	5
2.3.1	Presence of detection of non-interoperability.	Demonstration of function.	5	8
2.3.2	Allocation of translators in the Eclipse Arrowhead framework.	Demonstration of function.	5	8
2.3.3	Translators use IA core systems in Eclipse Arrowhead framework	Demonstration of function.	5	8

Table 7: Micro-service paradigm KPIs

3.2.1.4 Final assessment

3.2.1.4.1 KPI assessment

The final evaluation of the Micro-service Paradigm building block clearly shows that all set Key Performance Indicators (KPIs) have been fully accomplished. Eleven KPIs were indentified to measure the performance in terms of the engineering process, governance methods, the maturity of Eclipse Arrowhead framework, and autonomous interoperability capabilities. The expected results for all the eleven KPIs have been achieved at M36.

Notable achievements among them include doubling the number of architecture patterns in the cookbook, setting up governance and version control of the services developed, having more interoperability translators than planned, and implementing a process to detect interoperability mismatches and allocate translators accordingly. The fulfillment of all the KPIs is indicative of success in realizing the intended advancements from the baseline level.

3.2.1.4.2 Assessment against the baseline

The baseline analysis highlighted four key aspects that need improvement: the evolution of Eclipse Arrowhead core systems, the formation of engineering procedures and governance mechanisms, the autonomous detection of interoperability mismatches, and the autonomous allocation of translators to resolve identified interoperability gaps.

All of these were addressed during the course of the project. The Eclipse Arrowhead platform has matured considerably since the start of the project, with the majority of the core systems having reached TRL 8 and been deemed deployable in industry. Various new support systems and interoperability features were added, such as the Translation Manager Support System, DeviceQoS Evaluator Support System, various protocol and information model adaptors supporting interoperability across heterogeneous environments.

A proper governance structure was also implemented by formulating cookbook guidance, reuseable architecture patterns, and versioning strategies. These deliverables establish a

common engineering approach for developing, deploying, and maintaining microservice-based industrial systems.

Regarding interoperability, the project has successfully developed mechanisms for detecting interoperability mismatches and allocating translators in the Eclipse Arrowhead environment. This solves one of the primary technical gaps noted at the time of the baseline assessment.

3.2.1.4.3 Impact and added value

The findings that have been achieved through WP2 form the technology backbone on which other project components and application scenarios have been established and validated. The evolution of the Eclipse Arrowhead framework has enhanced the level of maturity of service-oriented industrial architectures making them available for use in complex industrial environments including automobile, aviation, and manufacturing industries.

The availability of translators and adaptors has enhanced the interoperability capability of the framework, allowing seamless integration of different information systems, industrial protocols, and legacy systems. Also, the establishment of governance processes and reuse of engineering artifacts helps sustain the framework outside the lifespan of the project.

Autonomous interoperability capabilities that have been implemented help decrease integration effort and promote the vision of dynamic and flexible digital ecosystems where services can be interconnected without much manual intervention.

Summarising, the assessment clearly proves that all set goals have been accomplished, thus successfully filling the technology gaps that existed during the baseline analysis phase. All eleven KPIs have been successfully met with results exceeding the set targets, showing considerable success in the maturity of the Eclipse Arrowhead framework, the implementation of governance and engineering processes, and the capability to develop autonomous interoperability.

This resulted in a mature technological framework for implementing federated digital ecosystems that have successfully facilitated the implementation and validation phases of Arrowhead fPVN use cases.

3.2.2 Major digital languages

3.2.2.1 Building Baseline summary

The Grand challenge is information interoperability. According to the European Interoperability Framework, there are several levels of interoperability as shown in Figure 4.

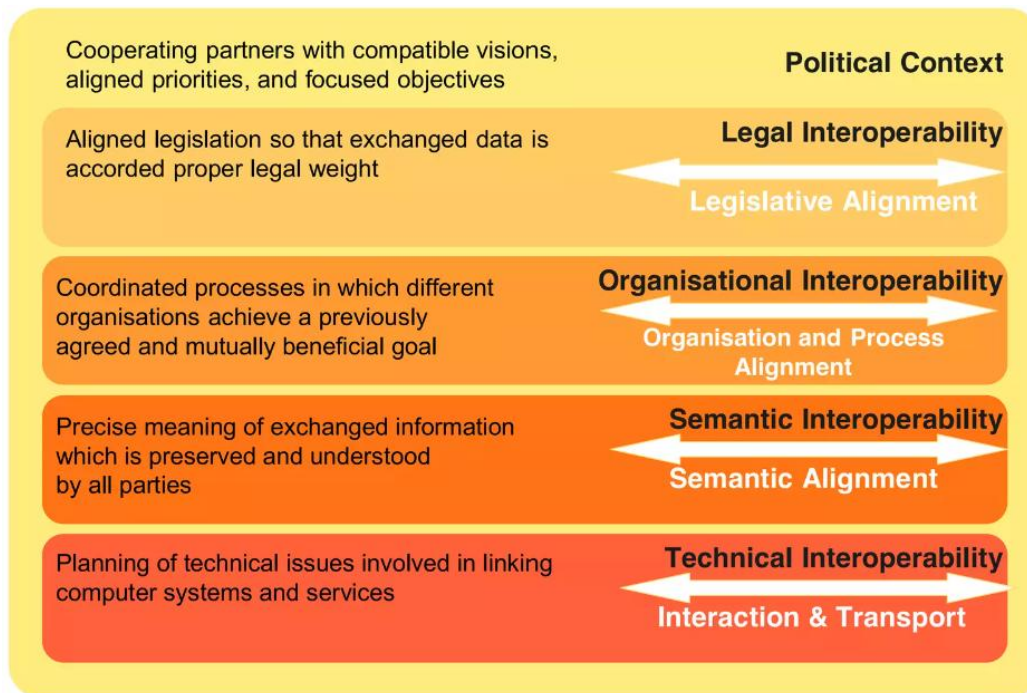


Figure 4: Levels of interoperability described in the European Interoperability Framework

WP3 concentrates its efforts on Semantic Interoperability, especially between defined and in real life used international Information Model standards. Some Management standards addressing Organizational Interoperability include valuable semantics without defining it in a formal way using Information models. The relevant part of these standards needs to be included in a Semantic Alignment effort.

The Grand Challenge on the semantic level is the lack of alignment. Many standards are developed in isolation by a standardization group focusing on a specific part of e.g. a product's life cycle, and the alignment with other standards with different scopes is poor.

To support the specified use-cases the most important information model standards to align are:

- WP6 (Automotive): ISO 10303 STEP, URDF Robot description, (speech and noise standards)
- WP7 (Aerospace): ISO 10303 STEP, ASD-5000F (In service feedback), 2000M Material Management) and 3000L (Logistic support)
- WP9 (Process Plant): ISO 15926 (Oil & Gas) and related industry standards DEXPI, DEXPI+ and CFIHOS, ISO 10303 STEP, ISO 18101 (Interoperability principles), IEC 61987 (CDD), IEC 611311-3 (PLC), ISO 81346-1 Classification and identification), IEC 61406 (GUIDs), ISO 12006 (Classification framework), ISO 19650 (Life cycle phases), ISO 61499 (Distributed automation) and OPAF (Process automation).

3.2.2.2 Baseline analysis

Most of the main standards in scope of WP6, WP7 and WP9 are based on the ISO 10303 and ISO 15926 modelling paradigm. The first generation of these information model standards used the ISO Express language. Later revisions have been using SysML (as a modelling language), XML Schema, and RDF Schema. Both the ISO 10303 family and the ISO 15926 family has been applying methods to achieve internal interoperability but the inter standard capability of interoperability between e.g. ISO 10303 and ISO 15926 has been very limited even though the standard has been developed in the same ISO subcommittee.

All standards in scope for the WP6, WP7 and WP9 pilots carry very valuable semantics, and the possibilities to achieve good interoperability is considerable from a modelling methods point of view. Other standards with a more management/process/organization standard type mentioned by the WP6, WP7 and WP9 require much more basic modelling, to make the semantics in the standard explicit and non-ambiguous. It would be of interest to create a high-quality ontology out of a management standard like ISO 81346.

3.2.2.3 Beyond the baseline

3.2.2.3.1 Technology action plan

WP3 objectives related to the Major digital languages are:

1. Select major standardized data models relevant to the use cases.
2. Identify the foundational properties of the selected data models.
3. Identify similarities and dissimilarities between the standardized data models.
4. In cooperation with WP4 propose updates to the selected standard which will improve data model translation accuracy.

Objective 1 has been achieved by using a survey to ask WP6, WP7 and WP9 which standards they are planning to use in their use-cases.

Objective 2 and 3 will be achieved by expressing the standards in a semantic language, and performing the analyses and comparisons needed to support the use-cases of WP6, WP7 and WP9. ISO 23726 Industrial Data Ontology (IDO) is proposed as the Upper Ontology.

Objective 4 needs more planning in cooperation with WP4.

3.2.2.3.2 KPIs, Evaluation, and validation process

Table 8 summarizes the KPIs defined for the autonomous translation evaluation and validation.

Key Performance Indicators (KPI)			Expected KPI value	Achieved KPI value
Name	KPI Description	Means of assessment		
3.1	Number of collected and analyzed standards regarding Automotive fPVNs	Inventory (Reports and documents in the ownCloud)	2 ASAM-ODS ISO-IEC 25012	0
3.2	Number of collected and analyzed standards	Inventory (Reports and documents in the ownCloud)	5 ISO 10303-239 ISO 10303-242	3 ISO 10303-209 ISO 10303-239

Key Performance Indicators (KPI)			Expected KPI value	Achieved KPI value
Name	KPI Description	Means of assessment		
	regarding Aerospace fPVNs		S5000F S2000N S3000L	ISO 10303-242
3.3	Number of collected and analyzed standards regarding Process Industry fPVNs	Inventory (Reports and documents in the ownCloud)	13 DEXPI, DEXPI+, CFIHOS, ISO 15926-2, -4 ISO 18101 IEC 61987 IEC 611311-3, ISO 81346-1, ISO 10303-239, -242, ISO 12006, ISO 19650, ISO 61499, OPAF	9? ISO 23726 (IDO) (1, 2, 3, 100) ISO 10303 (209, 239, 242) ISO 15926 (1, 2, 4, 6) DEXPI CFIHOS IEC 63278 (AAS) DPP : Digital Product Passport AFO: Arrowhead Framework Ontology IMF: Information Modelling Framework
3.4	Number of collected data model formats regarding use case partners	Inventory (Reports and documents in the ownCloud)	XML JSON SysML	XML JSON SysML (v1, v2) Express RDF/TTL OWL 2 DL
3.5	Number of analyzed data model formats	Inventory (Reports and documents in the ownCloud)	4	6
3.6	Number of analyzed communication protocols	Inventory (Reports and documents in the ownCloud)	5 CoAP MQTT Diameter HTTP HTTP2	OPC UA (AFO) Modbus (AFO) MQTT (AFO)
3.7	Number of analyzed modelling and knowledge representation languages	Inventory (Reports and documents in the ownCloud)	SysML	SysML (v1, v2) OWL 2 DL
3.8	Number of standard comparisons	Inventory (Reports and documents in the ownCloud)	ISO 10628 – DEXPI ISO 15926 – ISO 10303 ISO 15926 – ISO 81346	Star topology, mapping through IDO
3.9	Number of found synergies	Inventory (Reports and documents in the ownCloud)	4	8
3.10	Number of found interoperability problems	Inventory (Reports and documents in the ownCloud)	5	8

Table 8: Major digital languages KPIs

3.2.2.4 Final assessment

3.2.2.4.1 KPI assessment

The assessment of Major Digital Languages building block concluded with significant achievements in regard to its goal of enhancing semantic interoperability between heterogeneous industrial standards and information models. Indeed, KPI metrics indicate success in developing the basis for achieving this goal through conducting analyses of appropriate industrial standards, modelling languages, communication protocols and technologies of knowledge representation.

The evaluation suggests that the majority of the qualitative goals set for the work package have been met or even exceeded. Indeed, such quantitative results as the number of analyzed data model formats, synergies and interoperability problems far surpassed initial expectations. Moreover, WP3 succeeded in establishing an approach for semantic alignment based on Industrial Data Ontology (IDO), which enables comparing and integrating information models generated in various industrial sectors.

Formally, not all KPI metrics were reached completely, but this was due to changes in use-case requirements during the project execution and not by the inefficiency of the methodology applied. In other words, attention was paid to those standards, which were essential for the demonstrators and provided maximum benefit for interoperability and translation.

3.2.2.4.2 Assessment against the baseline

The identified challenge at M0 was the absence of semantic alignment among industrial standards, which limits interoperability potential. Despite being related to the same concepts and lifecycle stages, industrial standards were usually created separately, leading to various terminologies, ontologies, and data models used within the standards.

The task of WP3 to provide semantic interoperability analysis was fully accomplished by designing a framework that enables analysing, comparing, and harmonizing heterogeneous standards. The adoption of ISO 23726 Industrial Data Ontology (IDO) as the upper ontology ensured the presence of a unified semantic model used for the comparative and aligning analysis of standards from different industrial branches.

As a result, the comparison of such standards as ISO 10303, ISO 15926, DEXPI and CFIHOS allowed identifying common concepts and their relations between standards. Consequently, it was shown that much higher levels of interoperability could be achieved using ontology-based approaches instead of pairwise mapping between standards.

Moreover, WP3 broadened the baseline by studying semantic technology solutions like RDF/TTL, and OWL 2 DL along with traditional formats such as XML, JSON, SysML and EXPRESS. Thus, the semantic foundation was extended in order to ensure the interoperability support provided by WP4.

3.2.2.4.3 Impact and added value

The key result in WP3 is the creation of a semantic interoperability basis for the entire Arrowhead fPVN ecosystem and beyond it. The selected standards, ontologies, mappings, and interoperability analyses were vital for the autonomous translations conducted in WP4 and helped conduct integration within the use cases.

One of the most valuable achievements was the discovery of the existing semantic inconsistencies and interoperability challenges across different industrial standards. A wide variety of interoperability challenges were identified, namely different vocabularies, architectural modelling schemes, lifecycle approaches, property definitions, as well as different ontology approaches. This list of challenges will be an invaluable resource for the future standards development and improvement of digital information exchange within industrial ecosystems.

The WP3 research also revealed a high number of semantic synergies in the analysed standards and industrial initiatives. The semantic alignments related to IDO, ISO 10303, ISO 15926, DEXPI, CFIHOS, AFO and IMF proved that there is a possibility of creating a common semantic foundation for multiple industries. Such an outcome supports the project vision of federated industrial digital ecosystems.

3.2.2.4.4 Analysis of unachieved KPIs

Three KPIs were only partially achieved: the number of analysed Automotive standards (KPI 3.1), Aerospace standards (KPI 3.2), and Process Industry standards (KPI 3.3).

Regarding Automotive applications, no standards identified initially needed to be analyzed semantically since the Automotive use cases did not require semantic standardization but focused more on interoperability and data transfer issues. Thus, efforts were shifted to more useful standards, considering the goals of the project.

Regarding Aerospace applications, only three standards out of the five ones originally considered were analyzed due to their relevance concerning demonstrators' information models. Moreover, they provided enough semantic support for the validation exercises carried out within the project.

Concerning Process Industry applications, even though less standards were analyzed than targeted, the ones selected were the most influential in the process industry domain. Furthermore, new semantic assets such as IDO, AFO, and IMF, not included in the KPI definition at the beginning of the project but having a very positive impact, emerged as part of this work package.

Thus, even though not all numeric targets were accomplished, the semantic coverage obtained within WP3 was fully adequate to reach project objectives.

In summary, the evaluation proves that the development of the major building block called 'Major Digital Languages' was very efficient in solving the problems of semantic interoperability defined within the baseline. WP3 exceeded all expectations when it came to semantic modeling, standards mapping, identifying and analysis of interoperability issues, etc. The implementation of ontology-based methods of semantic alignment, along with the detailed study of the issues of interoperability and inter-relationships between the existing industrial standards, provided a strong basis for further research and development of translation technologies and federated digital ecosystems.

3.2.3 Autonomous translation

3.2.3.1 Building Baseline summary

To achieve successful communication, systems require a compatible interface and understanding of the data among them. This includes the communication protocol, encoding, encryption, compression, message structure, payload key values, and semantics. Once the data is successfully received, the data needs to be understood to be properly used.

Nevertheless, in some cases, communication cannot be established due to mismatches between the interfaces or the incompatibility of the data. In this situation, translation techniques can be used to solve the differences between the systems and solve communication problems.

There are several situations where this type of translation is necessary, in some cases as a temporary solution and in others as a permanent solution. Autonomous translation can be used when due to any problem one of the systems stops working and to avoid the pause of the activities, another available system is used instead. The new couple of systems may not have interfaces designed to work together, and translation mechanisms are needed to make it work until the previous system is restored.

Another situation is the new integration of systems that due to their vendor or version are not compatible with the current working systems. In this case, to reduce engineering time, the translation mechanism is put in place. In the same way, when an old system is updated, changes in the function or interface can affect compatibility with others.

Regarding autonomous translation technologies, several grand challenges shape the landscape of this field. Understanding and overcoming these challenges is vital for the successful development of translation services that can seamlessly bridge the gap between diverse data modeling languages.

One of the primary challenges is achieving a deep semantic understanding of data models across different languages. Each data modelling language often carries its own unique semantics, making it necessary to employ advanced techniques and semantic reasoning to accurately map concepts and relationships between them. Moreover, data models are not limited to textual information; they can encompass a wide range of data types, including diagrams, images, and structured data. Handling and translating this multimodal data is a significant challenge.

Ensuring interoperability of translated models across heterogeneous systems is a crucial challenge. Achieving this often involves the use of standardized data exchange formats and well-defined APIs to enable seamless integration into various platforms.

To address these challenges, three approaches are investigated:

- Ontology-based translation
- AI-based translation
- Model-based translation

3.2.3.2 Baseline analysis

To further analyze the state-of-the-art for specific enabling aspects of autonomous translators, this section has been divided into the three main approaches selected for the project.

Ontology-based translation state-of-the-art. Semantic Annotations for Web Services Description Language (SAWSDL) [1] is a W3C recommendation for the semantic annotation of Web Service Descriptions (WSDL) and XML Schema [2]. SAWSDL allows us to include semantic annotations within the schema, enabling the mapping of XML instance documents to

ontologies. SAWSDL model references connect XML Schema elements, types, or attribute declarations with named semantic concepts of some semantic model. To increase the expressiveness of model references, not only to ontology concepts but also to properties and individuals, Annotation Paths [3] were proposed. When annotating a schema with Annotation Paths, path expressions are inserted as values for Model Reference attributes. Later, in [4,5], methods to group semantic annotations and to add complement data values were proposed to solve ambiguities and to provide additional data required by consumers. Additionally, in [5], a tool named TAG-Tool was used to verify XML-based system compatibility as well as to automatically generate translators to support their interaction.

AI-based translation state-of-the-art. Recent advancements in AI- and ML-based methodologies are primarily employed for a range of critical functions. These functions include data analysis, feature extraction, real-time predictive analytics, as well as for addressing security concerns. These security applications encompass threat detection, anomaly detection, and information protection, in addition to classification tasks.

Martin Bauer's [6] work introduced the concept of a virtualized IoT platform, referred to as VirIoT, designed to facilitate information exchange between producers and consumers. This innovation relies on the implementation of a standardized, neutral information model known as NGSI-LD. Furthermore, the paper proposes the integration of Machine Learning methods in the ontology matching process to support developers. The core idea underpinning this solution is the effective utilization of ML algorithms for the automatic extraction and translation of sensor data into a standardized, neutral format. Knowledge infusion techniques were also applied in the matching process.

Neural Machine Translation (NMT) [7] represents a state-of-the-art neural network-based approach for language translation. The training phase leverages dictionaries that are transformed into word pairs following a preprocessing stage. In the realm of NMT, Deep Neural Machine Translation (DNMT) stands out as a variant that operates with multiple neural network layers, distinguishing itself from single-layer NMT models. OpenNMT, on the other hand, is a noteworthy open-source Python framework employed for learning and translating using NMT techniques.

Pasindu et al. [8] contributed to the field by presenting techniques aimed at enhancing the performance of Neural Machine Translation methods in language translation. Their evaluation work utilized the OpenNMT framework, revealing that translating between grammatically similar languages could be notably improved by introducing an intermediary language, such as English, into the translation process.

In addition, continuous learning is a fundamental aspect of autonomous translation technologies. Systems need to adapt and improve continually. Advanced approaches include reinforcement learning techniques, where the system learns from user feedback and interactions, gradually enhancing its translation quality over time.

Model-based translation state-of-the-art. Kleppe et al. [9] provide the following definition of model transformation: **A model transformation** is the automatic generation of a target model from a source model, according to a transformation definition. **A transformation definition** is a set of transformation rules that together describe how a model in the source language can be transformed into a model in the target language. **A transformation rule** is a description of how one or more constructs in the source language can be transformed into one or more constructs in the target language.

This definition is very general and covers a wide range of activities for which model transformation can be used: automatic code generation, model synthesis, model evolution, model simulation, model execution, model quality improvement (e.g., through model refactoring), **model translation**, model-based testing, model checking, model verification, and many more [10].

In the Arrowhead-fPVN project, we will focus on the model-based translation techniques from one data model standard to another.

A model-based translation requires the mastery of a transformation language as well as a sufficient knowledge of the source and the target languages (standards) in which the source and the target models are expressed.

In Model driven Engineering, transformation languages can be classified in three categories according to [11]: The first category represents the declarative transformation languages as Query/View/Transformation language (QVT) [12] and its sublanguage QVT-Relations (QVTr) Triple Graph Grammar (TGG) [13], and the Transformation Nets (TNs) [14].

The second category represents imperative languages such as QVTOperational (QVTo) [15]. While the third category includes hybrid languages such as Atlas Transformation Language (ATL) [16] and Epsilon Transformation Language (ETL) [17].

A Java like transformation languages such as Xtend and Xtend based languages like the VIATRA transformation language [18] are as well used in the model-based translations.

REFERENCES

- [1] Semantic Annotations for WSDL and XML Schema. 2007. Available online: <https://www.w3.org/TR/sawSDL/> (accessed on 20 August 2021).
- [2] Mendelsohn, N.; Thompson, H.; Sperberg-McQueen, M.; Maloney, M.; Gao, S.; Beech, D. W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. W3C Recommendation, W3C. 2012. Available online: <https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/> (accessed on 20 August 2021).
- [3] Köpke, J.; Eder, J. Semantic Annotation of XML-Schema for Document Transformations. In Proceedings of the OTM'10 Workshops; LNCS; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6428, pp. 219–228.
- [4] Moutinho, F.; Paiva, L.; Maló, P.; Gomes, L. Semantic annotation of data in schemas to support data translations. In Proceedings of the IECON (Industrial Electronics Conference), Florence, Italy, 23–26 October 2016; pp. 5283–5288.
- [5] Moutinho, F.; Paiva, L.; Köpke, J.; Malo, P. Extended Semantic Annotations for Generating Translators in the Arrowhead Framework. *IEEE Trans. Ind. Inf.* 2018, 14, 2760–2769.
- [6] M. Bauer, “Iot virtualization with ml-based information extraction,” in 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 915–920.
- [7] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *ArXiv*, vol. 1409, 09 2014.
- [8] P. Tennage, A. Herath, M. Thilakarathne, P. Sandaruwan, and S. Ranathunga, “Transliteration and byte pair encoding to improve tamil to sinhala neural machine translation,” in 2018 Moratuwa Engineering Research Conference (MERCon), 2018, pp. 390–395.
- [9] KLEPPE A, et al MDA Explained, The Model-Driven Architecture: Practice and Promise, Addison Wesley, 2003.
- [10] Tom Mens. 2013. Model Transformation: A Survey of the State of the Art. JohnWiley & Sons, Inc., 1–19.
- [11] K. Berramla, E. A. Deba and D. Benhamamouch, "Model transformation generation a survey of the state-of-the-art," 2016 International Conference on Information Technology for Organizations Development (IT4OD), Fez, Morocco, 2016, pp. 1-6, doi: 10.1109/IT4OD.2016.7479253.

- [12] Qvto an OMG standard specification <https://www.omg.org/spec/QVT/1.3/PDF> 2016
- [13] Klar, F., Königs, A., and Schürr, A. (2007). Model transformation in the large. In Proc. of 6th ESEC-FSE '07, ESEC-FSE '07, pages 285–294
- [15] Schönböck, J. (2012). Testing and Debugging of Model Transformations. PhD thesis, E188 Institut für Softwaretechnik und Interaktive Systeme.
- [16] Jouault, F., Allilaire, F., Bézivin, J., and Kurtev, I. (2008). ATL: A model transformation tool. Sci. Comput. Program., 72(1- 2):31–39
- [17] Czarnecki, K. and Helsen, S. (2006). Feature based survey of model transformation approaches. IBM Systems Journal, 45(3):621–646
- [18] Dániel Varró and András Balogh. 2007. The model transformation language of the VIATRA2 framework

3.2.3.3 Beyond the baseline

The Table 9 summarize the planned actions regarding each WP4 objective.

Objective	Actions Planned
<i>Provision of data set for translation development and early assessment of translation quality.</i>	- Survey WP 1 – Questioner about translation and data sets available. - Connection with the use cases available datasets. - Analysis of the data structure and synthetic creation of data to avoid confidentiality issues. - Test and validation with real data in demonstrators.
<i>Investigating the capabilities and feasibility of the ontology-based approach</i>	- Analysis of data models, datasets, and use cases. - Identification of translation challenges. - Development of a tool prototype (DITAG) for interoperability verification and translation automatic generation. - Make available in open access the tool prototype. - Validation of the tool prototype. - Demonstration about how to use the tool. - Publication about the tool prototype and results.
<i>Investigating the capabilities and feasibility of an ML/AI-based approach</i>	- Research and analysis of the translation gaps that can be addressed by AI. - Investigation of AI techniques to generate translation at service communication. - Development of open-source tools and solutions based on the investigation. - Validation of the tool prototype in demonstrators. - Publication about approach and results.
<i>Investigating the capabilities and feasibility of a model-based approach</i>	- Select at least 3 different data model standards (most used by use cases) and study the commonality and the differences between the standards. - Study the transformation language (select the most convenient one) - Provide an example of a model-based translation in Papyrus. - Study the idea to make SysML a common Intermediate representation for the translations between those standards.

	- Implement other model transformations and study the process to automatically generate them. - Publishing the result in conference papers - Developing the POC and package it to be used by the use cases. - Integration of the tool in the Arrowhead framework
<i>In cooperation with WP3, to provide translation microservices based on the above approaches</i>	- Meetings and technical workshops to integrate the translation solutions into Arrowhead microservices. - Analysis of data model structures and usage in conjunction with the WP3 to further application in the translation solutions - Integration and testing of the solutions in the framework.

Table 9: planned actions regarding WP4 objectives.

3.2.3.3.1 Technology action plan

The following points summarize but are not limited to the research activities planned for the autonomous translation of data models:

- 1) Analysis of the data sets and data models provided by the use cases. This activity includes the identification of the structures and features and the creation of syntactic data for their use in the developments without infringing industrial confidentiality.
- 2) Study of different data models and standards based on the WP3 work, the analysis will be performed from the translation point of view, including commonalities and differences.
- 3) Analysis of the translation scenario, technology gaps, and challenges regarding autonomous translation. This includes mismatch identification, description abstraction, translator generation, and testing and validation among others.
- 4) Investigation of AI and ML techniques to address the gaps and provide new translation solutions.
- 5) Study of the use of SysML V2 textual language as a common intermediate representation for all translations.
- 6) The implementation of prototypes based on the investigated approaches using a set of selected languages for at least three data model standards.
- 7) The Integration of the tool in the Eclipse Arrowhead framework.
- 8) Recompile of the translation solutions and prototypes on an inventory available for the consortia, including technical details and links to the realized code.
- 9) Writing of scientific publications (conferences and journals) and communication of results to a broader scientific audience.

Implementation

The translation solutions will be implemented into tools, prototypes or/and Arrowhead supporting systems, ensuring their practical usability. These tools and prototypes will serve as tangible applications of our translation approaches, allowing for real-world testing and validation. Our objective is to make these integrated tools and prototypes openly available, promoting their accessibility and encouraging broader utilization within the project.

The solutions will follow the microsystem architecture approach and provide accessible APIs. During the project the tools will be integrated with the Arrowhead framework and adapted to the use cases.

Documentation

In the context of our work package's development, a fundamental aspect of ensuring clarity, consistency, and adherence to best practices is the seamless integration of documentation with the implementation of our translation solutions and prototypes. The documentation will follow the Eclipse Arrowhead framework documentation structure and mirror the framework's guidelines, ensuring, as much as possible, that every interaction, component, and feature is comprehensively described, with clear references to the relevant parts of the prototypes implementation. This approach not only enhances the usability of our documentation but also facilitates collaboration among team members, stakeholders, and the broader development community. In the case of a resale tool, this should be documented: how to download, how to install, and how to use with an example provided with the tool.

In addition to the documentation, the results will be also presented as scientific publications in conferences and journals.

Demonstrators

In the forthcoming stages of our project, it is imperative to establish a framework for testing and validating the translation solutions within the context of our demonstrators and use cases. To ensure the effectiveness and reliability of these solutions, a comprehensive testing strategy will be employed. This strategy encompasses both functional and performance testing, wherein the translation solutions will be subjected to a variety of scenarios that mimic real-world use cases. We will assess their accuracy, speed, and scalability under different conditions to validate their practical utility. The demonstrators, which mirror the intended operational environments, will serve as the testing grounds, allowing us to evaluate how well the translation solutions integrate and function within the specific contexts of our use cases. Through these systematic and comprehensive validation procedures, we aim to guarantee that the translation solutions meet the exact standards required for their successful deployment.

Each task will in addition select at least a use case and integrate the tool to the Arrowhead framework to be used in this use case demonstrator.

Timeline

The following diagram in Figure 5 summarizes the expected timeline.

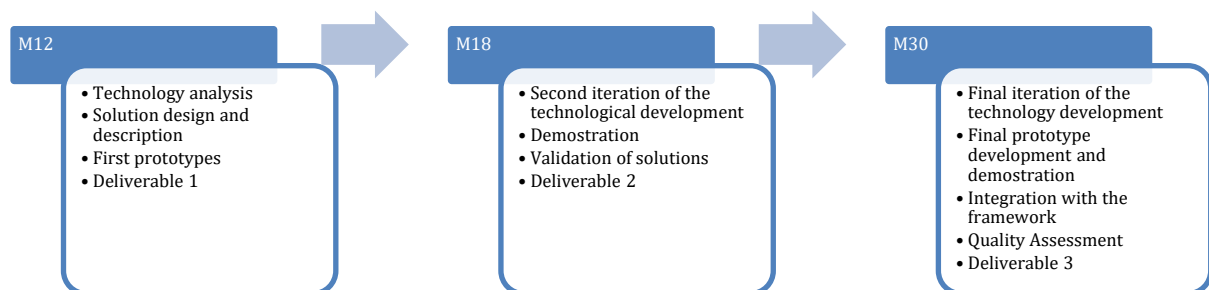


Figure 5: autonomous translation timeline

3.2.3.3.2 KPIs, Evaluation, and validation process

Table 10 summarizes the KPIs defined for the autonomous translation evaluation and validation and provides in the last column the KPI assessment at the end of the third year.

Key Performance Indicators (KPI)			Expected KPI value	Achieved KPI value
Name	KPI Description	Means of assessment		
4.1	Number of translation solutions provided during the project	Inventory (File in the Owncloud)	6	17 prototypes (TRL 3+) ¹
4.2	Range of data-models, standards and protocols translated.	Inventory and demonstrators	5	10 distinct data models ²
4.3	Translation accuracy	Measure the correctness and precision of translations (Metrics)	70%	85% ³
4.4	Number of scientific publications in the work package	Number of register publications.	6	7
4.5	Evolution of the prototypes available	Tools and prototype version (version id)	7	17
4.6	Integration of the solutions on the Eclipse Arrowhead Framework	Number of solutions integrated in the framework	1	4 ⁴

Table 10: autonomous translation KPIs

KPI description:

- 4.1 - Number of translation solutions provided during the project. Throughout the project, a variety of translation solutions based on different approaches will be offered. An inventory will document these solutions, including details about the translated data model, revised code, and integration with the Arrowhead framework. The inventory file is accessible to the consortia via OwnCloud and will be regularly updated as a progress indicator.
- 4.2 - Range of translated data models, standards, and protocols. Based on the data model analysis conducted during the project, several data models and standards will be used in the translation process. The number of pairs translated will serve as an indicator of the progress of the translation solutions.
- 4.3 - Translation accuracy. Translation accuracy refers to the correctness of the translation solutions. Accuracy can be measured using various metrics, including the percentage of messages translated and the correctness of the translated payload when compared to the ideal solution.

¹ See Table 9

² DITAG: IPC2581 - AFO: DEXPI, IDO - AI-Based: EC, IPC2581, ISO 10303 IEEE 1451, ISO 15926, IEC 61499 - ML for Sensor data: ISO 15926, S5000F - Model-based: SysMLv2, Rely

³ The KPI is derived from the quantitative evaluation results reported for the AI-based translation demonstrators. Ontology-based and model-based translators are largely deterministic and therefore are not evaluated through statistical accuracy measures.

⁴ DITAG, AI-based, AFO, Arrowhead DSL

- 4.4 - Number of scientific publications in the work package. The solutions will be published to reach a broader scientific community, and the number of publications will be tracked throughout the project.
- 4.5 - Evolution of the available prototypes. The prototypes will evolve during the project's iterations, with new versions being presented each year.
- 4.6 - Integration of the solutions with the Eclipse Arrowhead Framework. In the first year, the tools and solutions will be developed independently of the Arrowhead Framework. For the remainder of the project, the solutions must be integrated into the framework, and the number of integrated solutions will serve as an indicator of progress.

Note about KPI 4.3:

WP4 demonstrated translation solutions with reported accuracies ranging from approximately 53% to 96%, depending on the translation paradigm and use case. Mature translation approaches, including deep-learning-based meta-modelling and S5000F translation, achieved accuracies around 95–96%, while other experimental AI approaches reported lower values. Considering the heterogeneous nature of the translation technologies and the deterministic behavior of ontology- and model-based translators, the overall translation accuracy objective can be considered achieved with an estimated project-level performance of approximately 85%. This exceeds the expected target of 70%.

Solution	Metric reported
AI-based EC → IPC2581 translation	60% automation rate
RAG-driven XML translation	53.3%
Object detection for P&ID	mAP < 0.5 (≈50%)
Deep-learning-based Meta-modelling	95%
S5000F translation	~96%

Table 11: translators' accuracy.

3.2.3.4 Final assessment

3.2.3.4.1 KPI assessment

The final assessment of the Autonomous Translation building block clearly shows that all the set KPIs have been successfully achieved, and even in most of them, largely exceeded. The results indicate the successful development, validation, and integration of autonomous translation technology, including ontology-based, AI-based and model-based approaches.

Overall, the work package has outperformed the initially set goals concerning the number of developed translations technologies, standards and data model supported, availability of prototypes, and integration of translation services in the Eclipse Arrowhead platform. Altogether, seventeen translation prototypes have been developed, which is much greater than

the initial target of six prototypes, indicating extensive work with alternative translation approaches.

Finally, the obtained results show that technologies of autonomous translation can serve as an effective solution for enabling interoperability within heterogeneous digital environments, thereby validating one of the main assumptions underlying the Arrowhead fPVN project.

3.2.3.4.2 Assessment against the baseline

Some of the critical issues highlighted by the baselining exercise include the problem of understanding semantics in heterogeneous data models, the challenge of translating multiform information, the automation of translators, and integration of translation procedures in an interoperable industrial context.

Three complementary methodologies were considered in order to overcome the aforementioned problems: Ontology driven Translation, Artificial Intelligence based translation, and Model based Translation. In fact, all three methodologies proved their feasibility and applicability in an industrial context.

The ontology-based approach resulted in solutions capable of exploiting semantic annotations and ontological representations to support interoperability verification and translator generation. The development of the DITAG tool kit proved the practicality of ontology based translation techniques.

In the AI-driven approach, the WP4 examined machine learning, deep learning, and large language models for automating translation processes and facilitating the interoperation of heterogeneous information models. Some of the prototypes developed have been able to provide a very high standard of translation, especially in semantic translations.

Through the model-driven technique, WP4 was able to show how model transformation and intermediate representation could be applied in assisting the process of translation among heterogeneous standards. The studies conducted concerning SysML and other modeling technologies have proven that model-driven engineering technology can be the basis for future translation automation.

Another objective successfully achieved by the work package was incorporating some of the translation solutions into the Eclipse Arrowhead environment in order to make them applicable in the project demonstrators and thus creating a link between semantic and service interoperability.

3.2.3.4.3 Impact and added value

The achievements of WP4 can be considered as one of the major technological breakthroughs of the project. The developed translation solutions overcome one of the basic hurdles of digital transformation: the inability of heterogeneous IT-systems, standards, and data models to interact.

The development of seventeen translation prototype systems indicates that the investigated methods have reached a very high level of maturity. They involve the use of various combinations of standards, information models and communication tools and even surpass the initial objectives of the project. This portfolio of technology will allow creating the basis for

future interoperability systems and help to realize the vision of completely autonomous digital ecosystems.

The incorporation of four translation solutions into the Eclipse Arrowhead framework additionally enhances their exploitations opportunities. Not being mere research prototypes anymore, the solutions developed within the project became a part of the functional interoperability system, capable of implementing industrial applications.

3.2.3.4.4 Translation Accuracy Assessment

The KPI related to translation accuracy needs a specific explanation because of the diversity of the technologies that have been considered. Although the machine learning-based translators can be measured by using statistical indicators, ontology-based and model-based translators perform translations by using deterministic rules of transformation, which makes the traditional indicators useless for assessing them.

Translation performances observed during the project varied within the range between 53% and 96% according to the specific application case, the specific data set used and the particular method of translation employed. The matured technologies (AI translators based on deep learning, meta-modelling and S5000F translators) showed the performance of about 95–96%. Others showed lower figures, reflecting their exploratory nature and lower technology readiness levels.

Taking into account the variety of the considered technologies and the determinism of the work of ontology-based and model-based translators, the translation accuracy for the entire project may be estimated to be around 85%, which exceeds the initial target figure (70%) dramatically.

In summary, the final evaluation clearly shows that the Autonomous Translation building block succeeded in all its goals and met all the technical challenges that were pointed out in the baseline study. All KPIs were reached and surpassed their initial expectations showing clear advancements in the realm of autonomous translation technology and semantic interoperability tools.

WP4 successfully delivered a set of ontology based, AI based and model-based translation tools, confirmed their viability in industrial environments and incorporated the findings into the Eclipse Arrowhead platform. This is a very important step in reaching autonomy in federated digital ecosystems and serves as a basis for further research and exploitation endeavors.

3.3 Use case baselines

This section reports results of the baseline analysis at the end of the third year of the project for the following use cases:

- WP6
 - 1_6 - Automotive Battery Innovation fPVN.
 - 2_6 - Humans in the interoperable System.
- WP7
 - 1_7 - Interoperable intelligent management of production lines: Towards Model-based Enterprise.
 - 2_7 - Aircraft Health Management System (AHMS) for Trend Monitoring, Predictive Maintenance and Fleet Operations & Maintenance Simulation.
- WP9
 - 1_9 - Pump Station Engineering.
 - 2_9 - Digital Twins that enable higher performance by interoperability in pulp mills & carton board mills.
 - 3_9 - Interoperability for technical information exchange in process industry.

The list of use cases presented in D1.2 has been confirmed and no changes have to be reported (while in year 2, use 1_8 was cancelled).

At the end of the third year, the overall evaluation of the Arrowhead fPVN use cases proves the successful validation of concepts related to the application of the technologies and methodologies in various fields of industry, such as batteries' lifecycle management, human-machine interactions, aerospace manufacturing and maintenance, municipal infrastructures management, and digitalization in process industries. The seven use cases were able to test 27 Key Performance Indicators (KPIs), of which 26 have been reached fully, whereas only one has been partly achieved. Hence, all results show that the developed technologies and approaches have effectively addressed the challenges of interoperability, autonomous translation, service-oriented architectures, semantic integration, and digital transformation in realistic industrial environments.

In addition, a constant achievement throughout the tested use cases includes the ability to implement and validate Eclipse Arrowhead as an instrument to adopt an interoperability paradigm. Microservice architecture, data model standardization, digital twins, semantic and AI technologies, and automated translation services have been successfully used and proved as efficient solutions allowing integrating different systems, reducing data processing tasks, and creating interoperable digital ecosystems. In addition, the application of major industrial standards and digital languages, such as S5000F, IEC 81346, SysML, DEXPI, ISO 15926, and knowledge graphs technologies confirmed their role as enabler future production value networks.

Furthermore, beyond reaching the KPIs expected results, the use cases have confirmed several improvements in engineering and operations processes, which were made possible through the implementation of innovative approaches and solutions proposed by the project. Indeed, the efforts to integrate different systems decreased, whereas traceability, information reuse, and lifecycle management were significantly improved. Hence, the results also prove that the concept of the Arrowhead fPVN approach is relevant to engineering practices, providing an

appropriate and solid solution to the modern challenges associated with the shift towards the automation of engineering processes.

The following table provides a summary of the KPIs achievements.

Use Case	Number of Defined KPIs	Fully Satisfied	Partially Satisfied	Not Satisfied
UC1.6 (WP6)	3	3	0	0
UC2.6 (WP6)	7	7	0	0
UC1.7 (WP7)	3	3	0	0
UC2.7 (WP7)	4	4	0	0
UC1.9 (WP9)	4	3	1	0
UC2.9 (WP9)	4	4	0	0
UC3.9 (WP9)	2	2	0	0
Total	27	26	1	0

Table 12: summary of KPIs satisfaction.

Notes on the evaluation:

- UC1.6: all three KPIs (DI_V, DI_T, DI_E) achieved the Year-3 targets.
- UC2.6: all seven KPIs achieved according to the final validation results. The cloud integration was still ongoing, but the KPI target was interoperability readiness and integration into Arrowhead infrastructures, which was successfully demonstrated.
- UC1.7: all three KPIs reached 100% achievement.
- UC2.7: all four KPIs fully achieved according to D7.4 validation results.
- UC1.9: Digital Maturity, Standardized Data Infrastructure, and Standardized Data for Pump Station were achieved. The Micro-service Architecture KPI is considered partially satisfied, since only the foundations of the architecture were implemented and only about 10% functional coverage was reached, although the target value reported for Year 3 was attained. This interpretation is consistent with the final assessment narrative, which highlights limited implementation coverage.
- UC2.9: all four KPIs achieved.
- UC3.9: both KPIs achieved at the planned values (80% translation coverage and 30% run-time integration).

This gives an overall KPI fulfilment rate of **96.3% fully satisfied (26/27 KPIs)**, **3.7% partially satisfied (1/27 KPIs)**, and **0% not satisfied**.

Regarding the contribution to the project objectives, the following table provides a high-level summary. For more details refer to the final assessment of the single use case in this document.

Use Case	Autonomous Translation (Obj.1)	Microservices / SOA (Obj.2)	Digital Models & Standards (Obj.3)	Digital Transformation (Obj.4)	Open Ecosystem & Governance (Obj.5)
UC1.6 (WP6)	50% automated translation of battery lifecycle data through Digital Twin integration	Arrowhead-based service architecture connecting battery lifecycle stakeholders	Limited contribution	Smart laboratory management and lifecycle digital thread	Limited contribution
UC2.6 (WP6)	Translation of human speech into machine-executable commands	Arrowhead-integrated acoustic HMI services and APIs	Structured command vocabulary and semantic classification of industrial commands	Human-centred acoustic interaction paradigm for manufacturing environments	DIN database, open datasets and reusable benchmarking framework
UC1.7 (WP7)	Integration of DITAG, AI-based and Rely-to-SysML translators	Full SOA architecture integrating translation and optimization services	Adoption of SysML, MBSE and IPC-2581	MBSE-based lifecycle management methodology	Arrowhead-based open architecture and reusable services
UC2.7 (WP7)	AeroBridge automatic translation of aerospace data into S5000F	Full Arrowhead-based AHMS architecture with dynamic orchestration	Adoption and validation of S5000F; feedback to standard evolution	Automated maintenance and operational workflows	Contribution through adoption of open interoperability technologies
UC1.9 (WP9)	Identification and evaluation of future translation scenarios (e.g. DEXPI)	Microservice interoperability between municipal utility systems	Large-scale adoption of IEC 81346 and related standards	Practical experience and feedback on digital transformation governance	Limited contribution; future opportunities identified
UC2.9 (WP9)	Semantic interoperability through Knowledge Graphs and Digital Twins	Semantic interoperability services and Digital Twin platform	Harmonization of DEXPI, STEP Core, CFIHOS, AFO and related standards	Knowledge-driven digital transformation methodology	Governance of semantic assets, ontologies and Digital Twin ecosystems
UC3.9 (WP9)	80% automated translation coverage for technical information exchange	Runtime integration through Eclipse Arrowhead services	Harmonization of ISO 15926, DEXPI, CFIHOS, IEC 61987 and ISO 10303	Methodology and maturity assessment for interoperable digital processes	Promotion of Eclipse Arrowhead and open interoperability principles
Overall Contribution	Validation of autonomous translation technologies in battery, aerospace and process-industry domains	Demonstration of service-oriented interoperability across all industrial sectors	Adoption, validation and harmonization of major industrial standards and semantic models	Practical methodologies supporting digital transformation and organizational change	Promotion of reusable, open and scalable interoperability ecosystems

Table 13: summary of use cases contribution to project objectives.

3.3.1 WP6 - Use-case #1.6 – Automotive Battery Innovation fPVN

3.3.1.1 Baseline summary

3.3.1.1.1 General description of the baseline

The UC in T6.1 “Interoperable information along the full life cycle of EV batteries” will develop demonstrators for three exemplary stages of the battery life cycle as shown in the Figure 6:

1. operation of battery test labs during the development phase,
2. in-use phase (use in an electric vehicle, EV) and
3. optimized end-of-life processing.

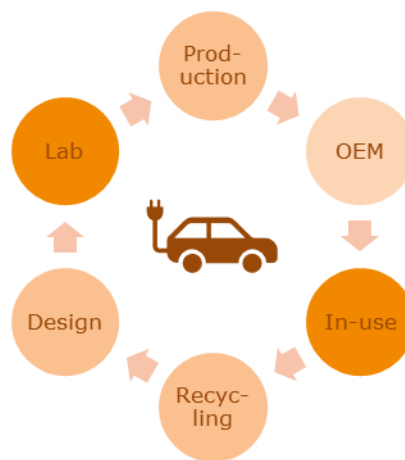


Figure 6: UC-1.6 Full life cycle of EV batteries.

Ad (1): Battery tests are an important part of battery development. Typically, there is a large variety of test devices and systems used. Data produced is sometimes only partly compatible, sometimes incompatible at all. This makes it especially hard to manage large numbers of such devices in test factories. Currently data translators between this plurality of test devices (usually from multiple vendors) on one side and the test factory management systems (TFMS) on the other side have to be developed on a case-by-case bases “per hand”, also called “glue logic coding”. In addition, this might often need reworking upon changes (e.g., updates).

Ad (2): During operation, battery data is typically collected by the battery management systems (BMS) in the vehicle. However, this is often a closed & proprietary system, which makes it hard to learn from real-world experience, and aggregate it across vehicles and fleets, to gain a deeper understanding about battery performance and aging. Relevant data include battery electrical parameters such as time series (current, voltage, state of charge (SoH)) and optionally environmental data (e.g., temperature and vibration/acceleration).

Ad (3): Fast and efficient evaluation of battery health is important for optimal decisions about when to replace it in the vehicle (ending its “first life”) and where to best use it afterwards (“second life” or recycling). This decision is typically based on measurements (charge/discharge tests), taking time and effort.

This estimation could be significantly improved by using detailed battery information (like type, age, materials used, etc.) as well as accessing the conditions during its usage.

3.3.1.1.2 Updated architecture of the use case as a sequential list of functional blocks

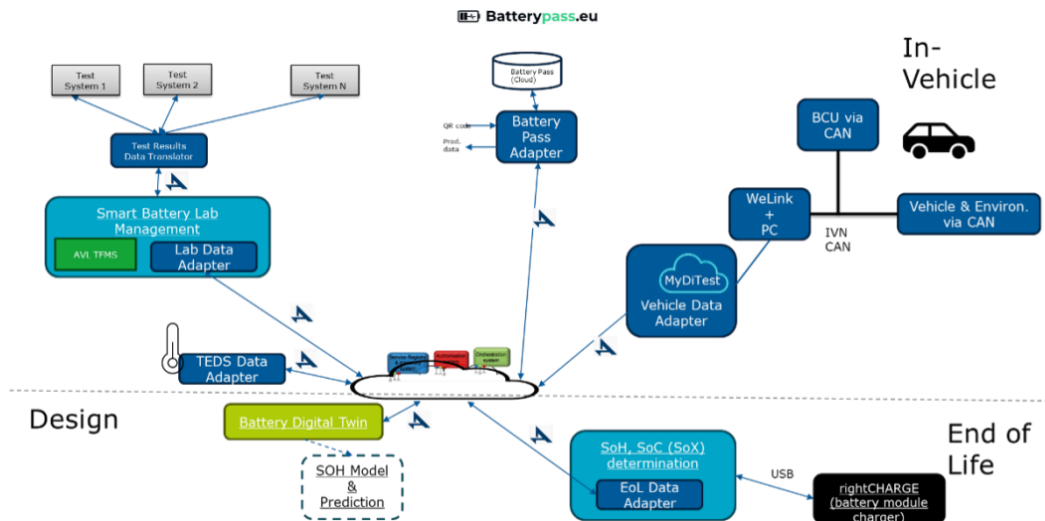


Figure 7: updated UC-1.6 Architecture

Table 14 lists functional blocks planned for prototypical implementation in the project (see Figure 7).

Name/Label	Short description
Test Results Data Translator	Translates result data between different test and measurement systems in the Battery Test Lab
Lab Data Adapter	Arrowhead client enabling data transfer between battery test lab and the Digital Twin
Battery Pass Adapter	Arrowhead client enabling access to descriptive data of specific EV batteries (Battery Pass ⁵)
Vehicle Data Adapter	Arrowhead client enabling data transfer from battery management system (BMS) via battery control unit (BCU). Access via CAN Bus --> diagnostic device --> Cloud
EoL Data Adapter	Arrowhead client access to End of Life (EoL) data about SoH and SOC, for exchange to/from digital twin, in order to improve residual value estimation
TEDS data Adapter	Arrowhead client enabling access to Transducer Electronic Data Sheet (TEDS) data

Table 14: UC 1.6 Functional Blocks planned for prototypical implementation.

⁵⁵ <https://data.consilium.europa.eu/doc/document/PE-2-2023-INIT/en/pdf>

3.3.1.1.3 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	Different Test Results Data Translators will be prototypically implemented in functional blocks as listed in Table 14
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	Will be used to implement and access the functional blocks listed in Table 14
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	Currently n/a
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	Will be demonstrated for optimized lab management in a battery test lab
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	Currently n/a

Table 15: UC 1.6 Arrowhead fPVN objective.

3.3.1.2 Baseline analysis - building blocks & enablers

3.3.1.2.1 Micro-service paradigm

3.3.1.2.1.1 State of the art

Current solutions are either implemented as monolithic applications or based on general SOA (but not micro-services) like AVL's Data.CONNECT™, see <https://www.avl.com/en-at/development-speed-and-methodology/connecting-solutions/data-management-dataconnect>

3.3.1.2.1.2 Enablers

Arrowhead's micro-service approach shall allow a modular, yet flexible orchestration of data exchange between the different "silos" of the phases Test / In-Use / End-of-Life. In addition to direct data exchange, data will be used to feed a digital twin (DT) of the EV battery, to be able to develop data-driven models e.g. for state-of-charge (SoC). The first suggested architecture is depicted in Figure 7.

3.3.1.2.2 Major digital languages

3.3.1.2.2.1 State of the art

- Test Phase: Results of tests are stored in data models typically specific for each operator. The ASAM-ODS standard is often applicable, see:

<https://www.asam.net/standards/detail/ods/>

- In-use Phase: currently no generally accepted standard is used; signals are defined on a quite low level e.g. via CANdb, e.g.
https://cdn.vector.com/cms/content/products/candb/Docs/CANdb_Manual_EN.pdf
- End-of-Life Phase: currently no generally accepted standard is used.

3.3.1.2.2.2 Enablers

- Test Phase: extend to non-ASAM-ODS-compliant systems by supporting also alternative data models/descriptions and developing appropriate translators.
- In-use Phase: the upcoming Vehicle Signal Specification (VSS) by COVESA is a protocol-agnostic approach for describing vehicle data in a standardized and extensible manner, see: https://github.com/COVESA/vehicle_signal_specification. Its adoption across industry is currently not clear.
- End-of-Life Phase: future battery pass extensions might extend on the ISO 15118 defining the communication between EV chargers, network and vehicles (still under discussion), see <https://www.iso.org/standard/69113.html>

3.3.1.2.3 Autonomous translation

3.3.1.2.3.1 State of the art

Typically, only specifically implemented translators exist.

3.3.1.2.3.2 Enablers

This UC shall analyze the potential applicability of autonomous translations, first with a focus on the test phase. Later, additional phases shall be considered (Test / In-Use / End-of-Life).

3.3.1.2.4 Standardization requirements

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	Life Cycle Management, REST APIs, ERP/MES, Function Blocks, ISO 25012	Life Cycle Management, REST APIs, ERP/MES, Function Blocks, ISO 25012
Information and Representation	Product data representation and exchange, ASAM-ODS, CANdb,	Product data representation and exchange, ASAM-ODS, CANdb, EU BatteryPass
Semantic and Language	XML, JSON, UML, Ontologies	XML, JSON, UML, SysML, Ontologies, LDP 1.0
Communication	Fieldbus, OPC-UA, MQTT, Internet Protocols: HTTP, TCP/IP, UDP	CAN, MQTT, HTTP/S, UDP
Cybersecurity and Safety	-	ISO 27001 HTTPS, TLS, UNECE Cybersecurity (UN R155 / R156), ISO 26262

Reference Model	RAMI	RAMI, New automotive EE Architecture (SDV)
Domain-Specific	Automotive	Automotive, Factory Control/SCADA IoT, Digital Twin, Integration Life Cycle
Framework Development and Specific Applications for development	.NET	.NET, Eclipse, Python, Docker

Table 16: UC 1.6 Standardization Requirements

3.3.1.2.5 Use Case Engineering Process

3.3.1.2.5.1 State of the art

AVL uses a specifically tailored agile process called ALAKSA, which is based on SCRUM (Figure 8). ALASKA is the acronym for ALASKA = AVL Lean Agile Software Development Process with Kaizen.

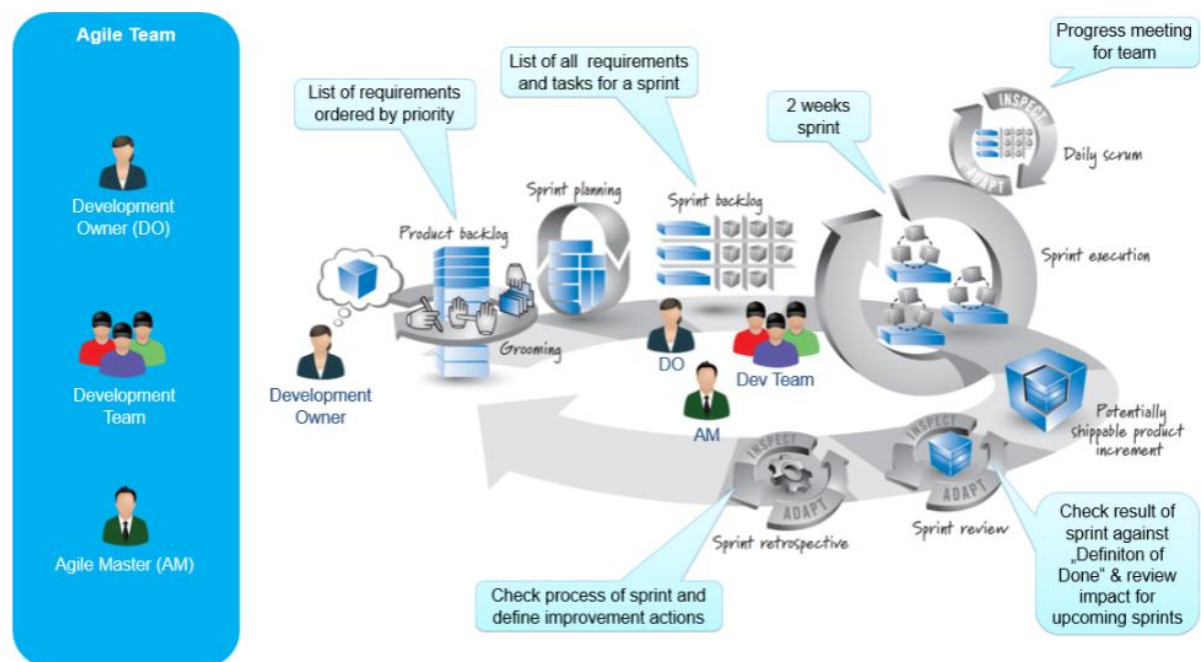


Figure 8: the AVL "ALASKA" Engineering process

ALASKA is Lean + Agile + Kaizen integration, therefore continuous improvement is a core built-in loop, not an add-on. It has a multi-level structure, from Portfolio / Business (Epics) to Program (Features, PI) to Team (Stories, Tasks).

ALASKA is agile at scale (similar to SAFe but customized) with a structured release model: (i) Team Increment, (ii) System Increment, (iii) Release.

The UC1.6 engineering process combines AVL's tailored agile ALASKA process with the Arrowhead Engineering Process view of the full EV battery lifecycle. ALASKA, the AVL Lean Agile Software Development Process with Kaizen, is based on SCRUM and integrates Lean, Agile and Kaizen principles. Continuous improvement is therefore a built-in process loop. The process is structured across portfolio/business level (Epics), program level (Features and Program Increments) and team level (Stories and Tasks), and follows a structured release model based on Team Increment, System Increment and Release.

Within UC1.6, this execution framework is applied to a lifecycle-oriented engineering context: design, battery lab testing, in-use operation, SoH assessment, second life and recycling are connected through a service-based digital thread. The practical Y3 emphasis is optimized battery test-lab management, where test orders for battery cells are scheduled against lab resources such as cyclers, chambers and slots while lifecycle data is exploited to improve utilization and energy efficiency. In addition to ML-assisted data-translation demonstrators, the engineering process includes AI-supported scheduling research based on reinforcement learning and a predictive robustness concept for anticipating premature battery test failures. Validation remains mainly prototype- and concept-based; full industrial validation still requires further integration work and larger operational deployments.

3.3.1.2.5.2 Enablers

ALASKA is formally governed through ISO 9001-compliant process management via QESMS, with quality management as a core driver and additions from ISO 14001 for environmental management and ISO 45001 for safety. It is also influenced by Automotive SPICE through the requirements hierarchy, controlled increments, release staging, formal quality assurance and risk management updates. The software lifecycle structure is aligned with ISO/IEC 12207, while IEC 15288 lifecycle thinking is partially implemented and adapted to an agile setting.

ALASKA element	ISO 9001 concept
DoR / DoD	Quality gates / acceptance criteria
Risk management updates	Risk-based thinking
Kaizen	Continuous improvement
Formal procedure	Documented process control
ASPICE area	ALASKA realization
SYS.2 / SWE.1 Requirements	Epics / Features
SWE.2-SWE.4 Design and Coding	Team iterations
SWE.5 Integration	System Increment
SWE.6 Testing	DoD and System Demo
SUP.1 Quality Assurance	QESMS and governance
MAN.3 Project management	PI planning
ISO/IEC 12207 process	ALASKA equivalent
Requirements	Epics / Features
Design	Backlog refinement
Implementation	Sprints
Integration	Increments
Verification	DoD / testing

Release	Final release stage
IEC 15288 concept	ALASKA realization
Stakeholder needs	Epics
System requirements	Features
System architecture	Program level
Integration	System increment
Validation	System demo

Table 17: the software lifecycle.

UC1.6 is further enabled by the Eclipse Arrowhead framework for service-based integration, ML-assisted result-data translation, BatteryPass-style descriptive data services, digital-twin data handling, and prototype data exchange between laboratory, in-use and end-of-life contexts. Additional Y3 enablers include reinforcement-learning scheduling research using Proximal Policy Optimization and predictive approaches for premature test-failure detection.

3.3.1.2.5.3 Arrowhead-EP of the Use Case

ALASKA maps strongly to Arrowhead-EP phases EPP1-EPP4 and EPP6, covering requirements through deployment and maintenance. Operations are covered through feedback into the backlog, while decommissioning and recycling remain largely outside the explicit ALASKA tooling scope and are handled at portfolio or lifecycle-management level. The EV battery use-case adds the product lifecycle dimension that completes the Arrowhead-EP view. EPP1 battery lifecycle, ALASKA epics/features and lab-optimization requirements -> EPP2 feature refinement, architecture, PI planning, digital-thread and service architecture -> EPP3 iterations, development, CI/CD, ML translators, TFMS interfaces, digital-twin/SoH components, PPO scheduling model and predictive failure concept -> EPP4 system increment, release, demo, prototype and lab-oriented validation -> EPP5 runtime battery test-lab operation, resource allocation and lifecycle data consumption with feedback into the backlog -> EPP6 software maintenance, SoH/EoL support, rescheduling after test failures and backlog updates -> EPP7 portfolio-level evolution and exploitation of battery lifecycle optimization applications -> EPP8 training, workshops, thesis work, documentation and process learning.

Arrowhead-EP	ALASKA and UC1.6 realization
EPP1 Requirements	Epics, Features, DoR, lifecycle scenarios, lab-management, SoH and BatteryPass needs
EPP2 Functional design	Feature refinement, architecture, PI planning, digital thread, BatteryPass module, full-life-cycle Digital Twin and lab scheduling architecture
EPP3 Procurement and Engineering	Iterations, development, CI/CD, ML translators, AVL TFMS, AVL DiTEST SoH framework, sensor framework, PPO scheduling model and predictive failure concept
EPP4 Deployment and Commissioning	System Increment, Release, Demo, Arrowhead services, AVL lab prototype and prototype-based validation
EPP5 Operation and Management	Runtime lab operation, scheduling of cells to cyclers/chambers/slots, lifecycle data consumption and operations feedback into backlog
EPP6 Maintenance / Decommissioning / Recycling	Software maintenance, backlog updates, SoH assessment, EoL data exchange, failure prediction, rescheduling, reuse, repurposing and recycling support
EPP7 Evolution	Portfolio-level planning, exploitation analysis and lifecycle optimization roadmap
EPP8 Training and Education	Workshops, partner documentation, master thesis, demonstrator knowledge transfer and Kaizen learning loops

Aspect	Arrowhead-EP	ALASKA
Lifecycle coverage	Full eight-phase lifecycle model	Partial and development-centric execution framework
Process type	Lifecycle model for product and SoS engineering	Agile execution framework for software/system development
Scope	Product, service and SoS lifecycle	Software and system development with release governance

Table 18: ALASKA mapping on the Arrowhead EP

Scalability level: medium. The process can be recursively applied to battery cells, test orders, instruments, laboratory resources and lifecycle data services, but the current evidence remains limited to prototypes and ongoing research rather than completed multi-site industrial deployment.

3.3.1.2.5.4 Mapping of tools in the Arrowhead-EP of the Use Case

ALASKA, the EP used in AVL, clusters tools into backlog, design, development, release, operations, and maintenance domains that map strongly to Arrowhead-EP phases EPP1–EPP4 and EPP6, with governance spanning all phases, while lifecycle end-of-life phases remain largely outside its explicit tooling scope.

Arrowhead-EP Phase	ALASKA Tool/Method Groups	Tool / method groups	Role in the use case
EPP1 Requirements	G1 (Backlog, Epics, DoR)	JIRA backlog, Epics, Features, DoR, requirement matrix, lifecycle scenarios	Defines lifecycle, lab-resource, EoL, scheduling and data-interoperability requirements.
EPP2 Functional Design	G2 (Architecture, PI Planning)	Architecture, PI planning, backlog refinement, UC1.6 architecture, BatteryPass module, Digital Twin, lab scheduling architecture	Defines data silos, services, scheduling boundaries and service interactions.
EPP3 Procurement & Engineering	G3 (Development, CI/CD)	IDEs, VS Code, Git, CI/CD, automated tests, ML translation, TFMS, SoH framework, PPO scheduling model	Develops translators, digital twin/SoH services, resource-allocation logic and predictive scheduling robustness.
EPP4 Deployment & Commissioning	G4 (Release, system increment)	Build/deployment tools, configuration management, System Increment, Release, Demo, Arrowhead services	Deploys and validates prototype services and lab-optimization concepts.
EPP5 Operations & Management	G5 (Monitoring, feedback)	Monitoring/logging tools, issue feedback, battery lab operation, scheduling of cells to resources	Uses services and AI-supported scheduling concepts in operational lab-management scenarios.
EPP6 Maintenance	G6 (Bug fixing, change mgmt)	Issue tracking, change management, bug fixing, SoH assessment, EoL data exchange, failure prediction	Supports maintenance, reuse, repurposing, recycling and recovery from premature test failures.
EPP7 Decommissioning	(No direct ALASKA group; partially G7 governance)	Governance, portfolio decisions, exploitation roadmap	Evolves the use case toward exploitable battery-lab and lifecycle optimization applications.
EPP8 Recycling / Reuse	(Implicit in G3/G7 via reuse; no explicit group)	Kaizen, retrospectives, web-based training, onboarding, workshops, master thesis	Supports continuous learning and adoption of ML translation, scheduling and digital-thread methods.

Table 19: tools mapping on the Arrowhead EP.

3.3.1.2.5.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the ALASKA EP of the Use Case	Contribution of the integrated UC1.6 engineering process
Obj. 1 - The change from design time to run time engineering	Strong enablers: G3 (Development & Integration (CI/CD, integration)) enables continuous delivery and rapid transition from design artifacts to executable systems G4 (Release & Deployment) → structured pipeline (Team → System → Release) bridges design → runtime G5 (Operations & Feedback) introduces runtime feedback into design loop Supporting: G1 / G2 (Requirements + Design) define traceable artifacts reused at runtime (partial continuity) G6 (Maintenance) closes loop via runtime-driven changes Cross-cutting: G7 (Governance (Kaizen, QESMS)) ensures continuous improvement loop (design ↔ runtime)	CI/CD, increments and release pipelines bridge design artifacts to executable systems. The digital thread links battery design/test data with lab runtime scheduling, SoC/SoH operation data and EoL decisions. Runtime feedback from operations and failures is routed back into backlog and maintenance loops.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Strong enablers: G3 (Development & Integration) integration of supplier components, external development G4 (Release & Deployment) system-level integration (System Increment) Supporting: G1 (Requirements) Epics can represent multi-stakeholder needs G2 (Design) architecture alignment across teams Cross-cutting: G7 (Governance (ASPICE, QESMS)) enforces standardized interfaces and processes	ALASKA supports integration of supplier and external-development contributions, while Arrowhead services align AVL, AVL DiTEST, SYN, UNI-KLU, LTU, AITIA and Sensicore-related activities through common lifecycle services and shared data/analytics concepts.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Primary contributors: G3 (Development & Integration) scalable CI/CD pipelines → modular development (microservices, components) G2 (Design) architecture definition supports system decomposition Supporting: G5 (Operations) monitoring large data streams G6 (Maintenance) handling faults in complex systems Cross-cutting: G7 (Governance) structure via ASPICE-like rigor	Modular development, CI/CD and service-based access to test devices, BatteryPass data, vehicle data, SoH tools, resource slots, work orders and predictive failure signals enable the EP to manage an increased number of inputs and outputs.
Obj. 4 - Address digital learning and	Explicit enablers: G7 (Governance (Kaizen, continuous improvement)) retrospectives, process learning loops;	Kaizen, retrospectives, QESMS governance, LTU thesis work, demonstrations, workshops and exploitation documentation create reusable learning

training activities as an integral part of the engineering cycle	Supporting: G1 (Requirements / backlog) can include training-related requirements G5 (Operations feedback) real-world learning input	loops for lab scheduling, ML translation and lifecycle optimization.
--	--	--

Table 20: mapping to Arrowhead EP phases.

3.3.1.2.5.6 Adopted toolchain

ALASKA employs a modern DevOps-oriented Agile toolchain, integrating backlog management, CI/CD, and quality governance tools to support continuous, traceable, and incremental engineering across the lifecycle.

- Requirements & backlog tools: JIRA (Epics, Features, Stories) and Requirements repository
- Development toolchain: IDEs and programming environments like VS.Code and Version control (e.g. Git)
- Integration & quality: Continuous Integration / Continuous Delivery (CI/CD) pipelines and Automated testing frameworks
- Release & configuration Build and deployment tools and Configuration management systems
- Maintenance & feedback: Issue tracking / ticketing systems; Monitoring/logging tools feeding back into backlog

3.3.1.2.5.7 Toolchain automation

The ALASKA toolchain shows a medium-to-high level of automation in development, integration and release through CI/CD pipelines and structured workflows. The UC1.6 technical automation level is medium: attribute mapping and test-result translation are partially automated, service discovery/consumption has been prototyped, and AI-supported scheduling is under development. Remaining automation gaps include full deployment across all lifecycle silos, operational scheduling integration with lab systems, automatic handling of premature test failures, and validation in an industrial battery-lab environment.

3.3.1.2.5.8 Toolchain limitations

End-to-end tool interoperability is not fully automated; several processes remain governance-driven and loosely coupled rather than service-integrated.

Runtime feedback integration, multi-stakeholder system orchestration and full lifecycle automation still require further engineering.

Legacy test and measurement systems still use vendor-specific structures and naming, so adapters and translation services remain necessary.

Resource constraints reduced access to full industrial battery-production validation environments.

The PPO-based scheduling work is ongoing and still needs operational validation in a battery lab.

The predictive failure concept defined initial requirements and collected sample data, but an integrated solution was not completed within the reported period.

3.3.1.2.5.9 Use case and engineering costs

At M0 (current state), the cost of ALASKA-based use case engineering can be described across the Arrowhead-EP lifecycle as a medium–high, development-centric cost profile dominated by human effort and partially automated toolchains.

In EPP1–EPP2 (requirements & design), costs are mainly people-driven (≈ 30 – 40% of total effort), involving 3–6 roles (product owners, architects, stakeholders) over several months (e.g., ~ 10 – 20 man-months), including backlog management, PI planning, and training on Agile/ASPICE standards.

In EPP3–EPP4 (engineering, integration, deployment), the largest share arises (≈ 40 – 50% , ~ 20 – 40 man-months), covering development teams, CI/CD infrastructure, testing, and release pipelines; tool licenses (JIRA, CI systems), integration effort, and partial automation reduce but do not eliminate manual work.

EPP5–EPP6 (operation & maintenance) account for ≈ 20 – 25% ongoing costs (~ 5 – 15 man-months/year), driven by monitoring, bug fixing, and change requests, with feedback loops requiring manual interpretation.

EPP7–EPP8 (decommissioning, reuse) incur low but non-negligible costs ($< 10\%$), mostly handled via governance without dedicated tooling. Additional overhead comes from training (Web based training, onboarding), process compliance (QESMS/ASPICE), and toolchain integration gaps, where lack of full automation and interoperability leads to extra manual coordination effort (~ 10 – 20% overhead across phases).

Overall, M0 reflects a well-structured but not fully automated lifecycle, where costs are still largely driven by engineering manpower rather than autonomous toolchain execution.

Arrowhead-EP phase	Relative engineering cost/effort	Indicative effort / cost driver
EPP1-EPP2 Requirements and design	Medium-high	About 30-40% of total engineering effort; typically 3-6 roles over several months, including product owners, architects, stakeholders, backlog management, PI planning and Agile/ASPICE training.
EPP3-EPP4 Engineering, integration and deployment	High	About 40-50% of effort; roughly 20-40 man-months for development teams, CI/CD infrastructure, testing, release pipelines, tool licenses, integration effort and prototype deployment.
EPP5-EPP6 Operations and maintenance	Medium-high	About 20-25% ongoing effort, or approximately 5-15 man-months/year, driven by monitoring, bug fixing, change requests, runtime feedback interpretation, SoH/EoL concepts and rescheduling logic.
EPP7-EPP8 Evolution, reuse and training	Low to medium	Less than 10% direct effort for decommissioning/reuse governance, plus training, onboarding and exploitation activities. Toolchain integration gaps can create an additional 10-20% coordination overhead across phases.

Table 21: engineering process costs.

For battery-lab operation, even moderate improvements in utilization have a strong economic impact. A medium lab with 1000 channels provides approximately 720000 potential channel-hours per month. Assuming operational costs of about EUR 12 per channel-hour, moving from 70% utilization to 85% utilization would recover roughly 108000 productive cyclers-hours per month, corresponding to approximately EUR 1.296 million monthly savings or more than EUR 15 million annually, before considering additional energy and planning-efficiency benefits. These estimates demonstrate the relevance of AI-driven scheduling and predictive optimization, although the optimization approach still requires operational validation and maturation.

3.3.1.2.5.10 Key Performances Indicators

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation (expected year results)		
Name	KPI Description	Means of assessment	Year 1	Year 2	Year 3
DI_V	Number of data items from the vehicle (in-use phase) available in EV-Battery-Digital Twin	Count	in-vehicle data items: 2	in-vehicle data items: 5	in-vehicle data items: 6
DI_T	Number of data items from test labs (development phase) available in EV-Battery-Digital Twin	Count	Test labs data items: 0	Test labs data items: 1	Test labs data items: 2
DI_E	Number of data items for determining residual value (end-of-first-life) available in EV-Battery-Digital Twin	Count	Residual value data items:0	Residual value data items: 0	Residual value data items: 3

Table 22: UC 1.6 KPIs.

3.3.1.3 Beyond the baseline

3.3.1.3.1 Use case action plan

As outlined in the DoA, the use case will follow these steps:

- Analyse use case, scenarios, data standards, models and structures, as well as access and security requirements
- Define requirements for the 3 planned demonstrators (see below)
- Develop architecture for a unified digital twin, allowing data exchange and use across phases (based on
- Figure 7)
- Implement prototype of digital twin based on Arrowhead framework
- Implement prototype data transfer for selected data sources
- Integration of developed technologies into three demonstrators described below
- Development of applications for realistic real-world validation and demonstration
- Validation against defined KPIs and project goals

Demonstrator development

- demonstrate an intelligent battery lab-management system. The goal is to optimize the allocation of lab resources (cyclers, channels) to units-under-test (UUT) (e.g., sample battery cells) by enabling interoperability in the PVN. In the R&D chain, additional information becomes available for consideration, most prominently results from other test systems (“previous experience”), and business objectives. This use-case will optimize the utilization of a battery lab significantly, with the aim of a considerable reduction in cost, throughput time, and energy consumption.
- demonstrate in-vehicle/in-use monitoring to support model extraction and validation for improved SoC/SoH calculation
- demonstrate efficient re-purpose/recycle decision support by improved determination of state of health (SoH) of batteries. This will be supported by combining inspection (observing performance in specific test scenarios) with estimation of wear out based on degradation data from the in-use phase (e.g., usage profiles).

3.3.1.3.2 Contribution to project objectives

Project Objective	Status at M0	Expected improvement	Planned actions	Expected Status at M12	Expected Status at M24	Expected Status at M36	Achieved value M36
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	0%	Focus on in-vehicle data and test lab data	Requirement specification, design, Prototype implementation	0%	25%	50%	50
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	Not available	Focus on test-lab data	As above	Design spec available	Early prototype	TRL6 prototype	TRL 5 (prototype)
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	n/a	-	-	-	-	-	
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	simple digital solution available	Vision: Smart lab management	Implement optimized allocation of lab resources exploiting data from other phases	Design spec available	Early prototype	TRL6 prototype	TRL 5 (prototype)
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	n/a	-	-	-	-	-	

Table 23: UC 1.6 - Contribution to project objectives

3.3.1.3.3 Use case and engineering costs

To estimate the potential impact of the proposed approaches, a simplified but realistic scenario can be considered. Assume a medium battery test laboratory operating 1000 channels resulting in a potential of 720000 testing hours per month.

For this calculation we estimate operational costs (including depreciation, energy, maintenance, and overhead) of ~€12 per channel-hour.1000

With conventional (rule-based/manual) scheduling, a typical utilization of ~70% can be assumed, meaning that ~216000 hours per month are effectively lost due to idle time, suboptimal sequencing, early test aborts, or reconfiguration delays.

By applying AI-driven scheduling and predictive optimization (1.6.1.2.1), supported by improved data availability and semantic integration (1.6.1.2.2), a conservative utilization increases to ~85% is realistic for early-stage deployment. This reduces idle or non-productive time to ~108000 hours per month, effectively recovering the other 108000 cycler-hours. At €12 per hour, this corresponds to ~€1.296.000 monthly savings, or more than 15M€ annually, without requiring additional hardware investment.

Additional benefits, such as reduced energy consumption through fewer unnecessary test restarts, improved planning accuracy from better test duration prediction, and lower manual engineering effort, can reasonably add a further 5–10% operational efficiency gain.

Although both approaches are currently at prototype stage, this estimate demonstrates that even moderate improvements in utilization can translate into substantial economic impact for large-scale battery testing facilities, where long test durations, high equipment costs, and tight scheduling constraints make efficiency gains particularly valuable.

3.3.1.3.3.1 Long-term prospects

With more data for Lab operations, maintenance events, battery testing and more becoming available over time, the RL-based optimizer will become better.

We are therefore confident that the optimization approach will become more effective as well.

3.3.1.3.4 Micro-service paradigm

3.3.1.3.4.1 Approach

The application of Arrowhead-based service-oriented architectures combined with AI-supported data translation enables access to heterogeneous data sources across laboratory systems, production, and lifecycle phases. This reduces manual data integration effort and makes previously siloed data usable for advanced analytics and optimization.

3.3.1.3.4.2 Impact

Summarized in the estimation under section 3.3.1.3.3.

3.3.1.3.4.3 Long-term effect

Summarized in the estimation under section 3.3.1.3.3.

3.3.1.3.5 Major digital languages

No direct contribution in UC1.6

3.3.1.3.5.1 Approach

n/a

3.3.1.3.5.2 Impact

n/a

3.3.1.3.5.3 Long-term effect

n/a

3.3.1.3.6 Autonomous translation

3.3.1.3.6.1 Approach

Initial work on reinforcement learning–based scheduling and predictive approaches for battery testing laboratories showed strong potential to improve resource utilization, reduce idle time, and minimize costly reconfigurations. This directly addresses inefficiencies in current lab operations and can lead to significant reductions in testing costs and energy consumption.

3.3.1.3.6.2 Impact

Summarized in the estimation under section 3.3.1.3.3.

3.3.1.3.6.3 Long-term effect

Summarized in the estimation under section 3.3.1.3.3.

3.3.1.3.7 KPIs, Evaluation, and validation process

See Table 22.

3.3.1.4 Final assessment

3.3.1.4.1 KPIs assessment

The final assessment of UC1.6 shows that all KPIs set for the use case were met. The EV Battery Digital Twin that was created effectively integrated data from all three lifecycle phases, the battery's development and testing, its operation, and the end-of-life assessment. The goals

for Year 3 were met, making sure that the Digital Twin had access to the expected nine data items: six from vehicle operations, two from tests, and three from end-of-life analyses.

Reaching the KPIs means that the necessary tech required to connect battery lifecycle data is now available: info generated at any stage, from development to end-of-life, is collected and shared through systems that can interact, letting analysts across different phases use the same data for deeper insights.

The project's goals for autonomous translation and interoperability were met too, having machines handle 50% of needed data translations and showed off working models that used service-oriented interoperability ideas. Some demos stayed at the prototype stage because of problems with the industrial testing space.

A limitation affecting the final validation activities was the withdrawal of Northvolt from the consortium, which prevented validation in a large-scale industrial environment. Tests had to be done using only prototypes in AVL settings instead. Still, what was developed proved the concepts worked, and the interoperability stuff needed for our task checked out successfully.

3.3.1.4.2 Contribution to project objectives

UC1.6 contributed directly to some key objectives of the Arrowhead fPVN project. It showed how data that machines can understand and use can be swapped between different EV battery lifecycle stages. This backs the project's aim for connected digital worlds and easy info sharing: indeed, the use case set up a system to blend data from battery testing labs, management systems, and other sources into one Digital Twin setup.

Moreover, UC1.6 adopted WP2 micro-service architecture. It presented battery-related information through service-style links and fitted data providers and consumers using Arrowhead services. In addition, with WP4 explored ways to use machine learning to change mixed-up data into shareable forms. The trial of automapping between various data models worked well too.

Looking at the bigger picture, UC1.6 highlighted what linking up can mean during a battery's lifespan. The digital thread it created helps to improve tests, predict health states better, and decide end-of-life worth accurately. But it also paves the way for smart reuse and recycling down the line. These achievements push toward greener battery handling and aid the tech revamp in the car biz.

3.3.1.4.3 Improvement in the engineering process

UC1.6 showed some improvements in the engineering process from adopting interoperable architectures, a solution based on a Digital Twin, and service-oriented integration methods. Traditionally, data needs to be exchanged between battery test systems, lab management platforms, battery management systems, and end-of-life assessment tools. For this, dedicated point-to-point integrations and custom translation logic have to be developed. But the suggested setup simplifies things by adding standardized service interfaces and machine-interpretable data exchange methods.

Introducing a Digital Twin that collects lifecycle info lets different stages of the engineering process share data smoothly. Data made in one phase can be used again in later phases without being manually reprocessed. This builds a continuous digital chain that helps with battery development, testing, operation, and end-of-life stuff. Plus, the work done showed how AI-assisted scheduling and predictive analytics could make battery labs run better with better resource allocation and usage.

3.3.2 WP6 - Use-case #2.6 – Humans in the interoperable System

3.3.2.1 Baseline summary

The future of industrial production depends not only on the interoperability of machines, but also on the interoperability between humans and machines. While considerable progress has been made in connecting industrial systems through digital communication infrastructures, human-machine interaction (HMI) often remains a bottleneck in production environments.

Current industrial communication is frequently based on proprietary interfaces, push-button controls, touchscreens, or highly specialized command systems. Such solutions often require extensive training, limit flexibility, and become difficult to operate in dynamic production environments. The challenge becomes even more pronounced in modern automotive manufacturing, where workers increasingly collaborate with robots and intelligent systems across a wide range of production tasks.

Three major challenges characterize the current baseline.

First, existing interaction concepts are often not sufficiently adapted to the needs of workers. Many systems require complex procedures that increase cognitive load, reduce efficiency, and can lead to operational errors. More intuitive interaction methods are therefore required to support productive collaboration between humans and machines.

Second, industrial environments differ substantially in their communication requirements. Depending on whether robots are used in logistics, assembly, inspection, maintenance, or material handling, different command repertoires and interaction intensities are needed. Existing solutions rarely provide a flexible framework that can be adapted across such scenarios.

Third, European production environments are increasingly multilingual and acoustically challenging. Workers often communicate in different languages and with different accents while operating in noisy industrial halls. Conventional speech-based interfaces typically show limited robustness under such conditions and frequently require users to remain close to the machine, thereby reducing flexibility and limiting practical deployment.

The baseline situation is therefore characterized by a lack of intuitive, noise-robust, and interoperable communication mechanisms between humans and machines. The objective of Use Case 2.6 is to address these limitations by developing a language-independent and acoustically optimized command vocabulary, combined with intelligent speech processing and interoperability services that enable reliable communication between human operators and industrial systems under realistic production conditions.

3.3.2.1.1 Updated architecture of the use case as a sequential list of functional blocks

The updated architecture of UC-2.6 is centred on a modular acoustic HMI pipeline for robust human-machine communication in noisy industrial environments. The use case connects three main layers: (i) the human operator and the acoustic production environment, (ii) the signal-

processing and command-translation components, and (iii) the Arrowhead-based microservice and machine-control infrastructure.

The architecture starts with a human operator issuing a spoken command in an industrial environment. The acoustic scene is characterized by machine-specific noise, reverberation, distance between operator and machine, and potentially multilingual or accented speech. The command is captured by a directional microphone system with steerable focus direction and distance adjustment. This enables spatial targeting of the desired speaker signal while reducing interfering industrial noise.

The captured signal is then processed by the acoustic HMI layer. This layer includes speech enhancement, noise suppression, and automatic speech recognition. The recognized command is evaluated against a noise-resilient and largely language-independent HMI command inventory developed by SDU. In addition, the command can be assessed by a microservice that determines the suitability of existing or new command items for a given industrial noise context. The translated command is then passed to the Arrowhead-based interoperability layer. Here, the acoustic command representation is converted into machine-readable control logic and made available as a modular microservice. This enables integration with different industrial systems and supports dynamic deployment in production value networks.

Finally, the machine or robot executes the command and provides feedback to the human operator. This feedback may be speech-based, sound-based, or a combination of both, depending on the complexity and sequential structure of the required feedback items. The feedback channel is supported by the same acoustic infrastructure and can use directional loudspeaker technology to provide targeted machine-to-human communication.

Table 24 lists the functional blocks planned for prototypical implementation in UC-2.6.

Name/Label	Short description
Industrial Acoustic Environment	Realistic production-hall soundscape including machine noise, reverberation, distance effects, and multi-source interference. The environment is represented through field-recorded industrial noise data and laboratory-based soundscape simulations.
Human Operator Command Input	Spoken machine commands produced by operators in realistic working conditions. The command input accounts for multilingual speakers, accent variability, and different interaction distances.
Directional Microphone Array	Hardware component for capturing the operator's speech signal with steerable focus direction and distance adjustment. The system aims to reduce pickup of unwanted industrial noise from other directions.
Speech Enhancement / Noise Suppression Module	Signal-processing component that improves the signal-to-noise ratio of the captured command signal before recognition and translation.
Automatic Speech Recognition Module	Recognition component that converts the enhanced speech signal into a symbolic command representation. Different ASR systems can be evaluated or integrated depending on the deployment context.
Noise-Resilient HMI Command Inventory	A largely language-independent and acoustically optimized command vocabulary for basic machine operation under industrial noise conditions.
Command Suitability Assessment Microservice	Microservice for evaluating whether existing or newly proposed command items are suitable for reliable recognition in a given noise context.
Command Translation Layer	Translation component that converts recognized acoustic command structures into machine-readable and machine-executable control formats.
Arrowhead Microservice / Local Cloud Layer	Interoperability layer enabling modular deployment, orchestration, and access to the command-processing and translation services within the Arrowhead framework.

Machine / Robot Control Interface	Interface through which translated commands are transferred to industrial machines, robots, or control systems for execution.
Machine Feedback Module	Component for generating machine-to-human feedback, either as speech, non-speech sounds, or a combined feedback strategy, depending on the required complexity.
Directional Feedback Output	Loudspeaker-based output component for spatially targeted feedback from the machine or robot to the human operator.
Industrial Noise Corpus / DIN Database	Open-access database of authentic industrial noise recordings used for development, benchmarking, validation, and future standardisation of acoustic HMI systems.
Validation and Pilot Deployment Layer	Real-world and laboratory-based validation infrastructure, including industrial pilot environments and benchmarking under realistic noise conditions.

Table 24: UC 2.6 – Sequentially ordered list of architectural blocks

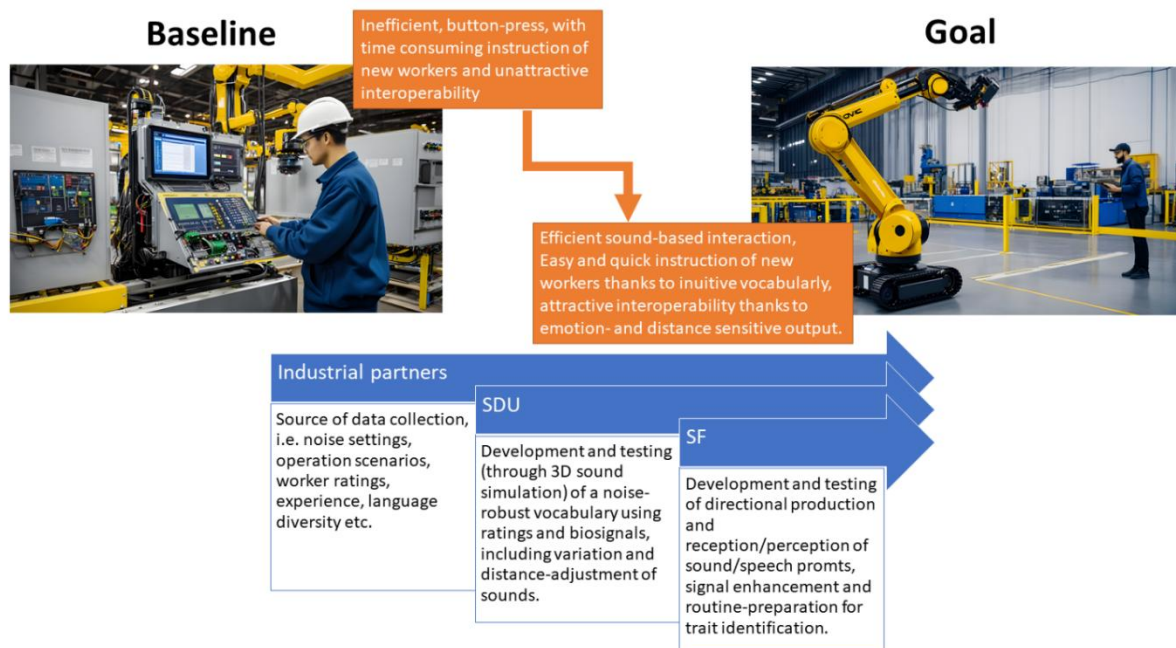


Figure 9: use case architecture.

3.3.2.1.2 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC 2.6 Contribution
1. Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services, thus significantly reducing the need for human support.	UC 2.6 contributes to this objective by developing an intuitive and noise-robust interoperability vocabulary for acoustic Human–Machine Interaction. The vocabulary is designed to remain functional under realistic industrial noise conditions and at interaction distances where traditional button-based operation is impractical. Based on these framework conditions, translation micro-services are developed that establish a bridge between voice commands and machine-control logic. Spoken commands, including command combinations and command sequences, are mapped onto existing machine-operation codes, while machine outputs can be converted into suitable acoustic feedback signals where appropriate.
2. Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs.	UC 2.6 contributes to this objective through the development of a microservice platform that enables stakeholders to evaluate the suitability and robustness of acoustic command structures under realistic industrial noise conditions. Through browser-based APIs, users can upload recordings of their own industrial sound environments and test existing or newly proposed voice commands. Commands can be submitted either as audio or text files. The system automatically evaluates command suitability and robustness and provides recommendations as well as potential alternatives where applicable.

3. Update proposals for major digital data models for industrial production between which autonomous and seamless data model understanding is enabled.	—
4. Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN.	WP6 contributes to this objective through its interaction with WP10 and through the development of practical industrial deployment concepts for interoperable acoustic Human–Machine Interaction technologies.
5. Established a sustainable governance of open-source architecture and implementation platform.	UC 2.6 contributes to this objective through (1) the open dissemination and publication of noise-robust Human–Machine Interaction command and feedback structures, including documentation of their robustness across different industrial noise levels and noise types, and (2) the establishment of the open-access Industrial Noise Corpus (DIN), which provides authentic industrial noise recordings for benchmarking, research, hearing-protection applications, well-being studies, and future industrial developments.

Table 25: UC-2.6 Arrowhead fPVN objective

3.3.2.2 Baseline analysis - building blocks & enablers

3.3.2.2.1 Micro-Service Paradigm

3.3.2.2.1.1 State of the Art

Current Human–Machine Interaction (HMI) systems in industrial environments are typically based on proprietary machine interfaces, dedicated control panels, touchscreens, or application-specific software solutions. While many of these systems support remote monitoring and machine control, they rarely provide interoperable mechanisms for evaluating, deploying, and adapting speech-based interaction across different industrial environments.

Existing speech-based HMI solutions are often tightly coupled to specific hardware platforms, machine types, languages, or acoustic conditions. As a result, adapting a speech interface to a new industrial environment usually requires extensive manual configuration and testing. In particular, there is currently no widely available infrastructure that allows industrial stakeholders to assess whether specific voice commands remain robust under realistic noise conditions before deployment.

Furthermore, the suitability of voice commands depends strongly on environmental characteristics such as machine noise, reverberation, speaker distance, and competing sound sources. Existing systems generally provide little support for systematically evaluating these factors in a scalable and reusable manner.

3.3.2.2.1.2 Enablers

The Arrowhead micro-service paradigm provides the foundation for overcoming these limitations through modular and interoperable service architectures. Within UC 2.6, this approach is implemented through the development of a browser-based API and micro-service platform dedicated to acoustic Human–Machine Interaction.

The platform enables industrial stakeholders to upload recordings of their own production environments and evaluate the robustness of existing or newly proposed speech commands under realistic noise conditions. Commands can be submitted either as audio recordings or text-based command specifications. Based on acoustic analyses and benchmarking procedures, the system automatically assesses command suitability and provides recommendations for optimisation or alternative command structures where appropriate.

The micro-service architecture allows individual functionalities such as industrial-noise simulation, command evaluation, speech-processing, and command benchmarking to be deployed independently while remaining interoperable within the Arrowhead ecosystem. This modular approach supports flexible deployment, scalability across industrial domains, and future integration with additional machine-control and interoperability services.

By transforming acoustic HMI evaluation into a reusable service rather than a machine-specific implementation, UC 2.6 extends the Arrowhead micro-service concept into the domain of human-machine communication and provides a scalable framework for future industrial deployment.

3.3.2.2.2 Major Digital Languages

3.3.2.2.2.1 State of the Art

Current industrial communication systems rely on a variety of machine-oriented data models, communication protocols, and control languages. While these frameworks enable interoperability between machines and software systems, considerably less attention has been devoted to interoperability between humans and machines. Human operators typically interact with machines through proprietary interfaces, predefined control panels, or machine-specific command structures that vary across manufacturers, industries, and deployment environments.

Speech-based interaction systems face an additional challenge. Human language is inherently variable across speakers, accents, languages, and communication contexts. As a result, speech commands that perform well in one environment may become unreliable in another. Existing industrial voice interfaces therefore often depend on highly constrained command sets that remain closely tied to specific applications and are difficult to transfer across operational contexts.

Consequently, there is currently no broadly accepted semantic framework for structuring machine-oriented speech commands in a manner that simultaneously supports human usability, multilingual adaptation, interoperability, and robust deployment under realistic industrial conditions.

3.3.2.2.2.2 Enablers

UC 2.6 addresses this gap through the development of a language-adaptable and semantically structured command inventory for acoustic Human–Machine Interaction. Rather than treating speech commands as isolated words or phrases, the project organises commands according to their operational meaning and functional role within industrial workflows.

The resulting framework represents an intermediate communication layer between human language and machine-control logic. Commands are designed to remain semantically stable while allowing adaptation to different languages, cultures, accents, and industrial environments. This approach enables interoperability not at the level of linguistic form, but at the level of communicative function and operational intent.

The command inventory is continuously evaluated under authentic industrial noise conditions using the Industrial Noise Corpus (DIN). Through this process, semantic command structures

can be refined, benchmarked, and validated across multiple acoustic environments. The resulting framework provides a basis for future standardisation efforts concerning machine-oriented speech commands and acoustic interoperability.

Within the Arrowhead ecosystem, these semantically structured command representations constitute a form of machine-interpretable communication layer that supports translation between human speech input and machine-executable operational logic. In this sense, UC 2.6 extends the concept of “Major Digital Languages” beyond traditional machine-to-machine communication towards interoperable human-machine communication.

3.3.2.2.3 Expected Impact

The proposed approach contributes to the creation of reusable and transferable command structures that can be deployed across different industrial domains without requiring complete redesign of the interaction framework. By separating semantic intent from linguistic realisation, the framework facilitates multilingual deployment, improves scalability, and reduces adaptation effort for future industrial implementations.

Furthermore, the developed command inventory constitutes an important step towards future standardisation of acoustic Human–Machine Interaction and supports the broader Arrowhead vision of seamless interoperability within distributed production value networks.

3.3.2.2.3 Autonomous Translation

3.3.2.2.3.1 State of the Art

Current industrial Human–Machine Interaction systems generally rely on fixed and deployment-specific mappings between user actions and machine operations. In most production environments, machine commands are executed through physical interfaces such as buttons, switches, touchscreens, or dedicated control panels. Where speech interfaces exist, they are typically implemented as isolated solutions with predefined command vocabularies and limited adaptability across machines, languages, or industrial environments.

Consequently, the translation between human intent and machine-executable actions remains largely manual. Operators must learn machine-specific command structures, while system integrators must manually configure how human inputs correspond to underlying machine-control logic. This process is often time-consuming, difficult to scale, and highly dependent on individual deployment scenarios.

Furthermore, existing systems rarely consider the acoustic characteristics of industrial environments. Commands that may function reliably in one context often become ineffective in another due to differences in noise levels, noise types, reverberation, communication distance, or language use.

3.3.2.2.3.2 Enablers

UC 2.6 addresses these limitations through the development of an interoperable translation layer connecting human speech commands with machine-executable operational logic.

The approach does not seek to replace existing machine-control systems. Instead, it creates an intermediate translation framework capable of mapping spoken commands onto already existing machine-control processes. Physical interactions such as button presses, menu selections, or predefined machine operations are analysed and represented through corresponding speech-command structures.

The translation process consists of several stages:

1. Human communicative intent is expressed through a spoken command.
2. The command is recognised and mapped onto a semantically structured command inventory.
3. The semantic command representation is translated into machine-oriented operational logic.
4. The resulting machine command activates the corresponding operational process within the existing machine-control system.
5. Where appropriate, machine states and operational outcomes are translated back into acoustic feedback signals directed towards the human operator.

This framework transforms speech commands into interoperable operational instructions while preserving compatibility with existing industrial infrastructures.

3.3.2.2.3.3 Expected Impact

The autonomous translation framework significantly reduces the effort required to establish speech-based interaction in industrial environments. Instead of redesigning machine-control architectures, existing operational procedures can be accessed through a reusable translation layer that links human communication and machine logic.

The approach furthermore supports interoperability across different industrial domains by separating three layers that are traditionally tightly coupled:

- linguistic realisation,
- semantic intent,
- machine execution logic.

As a result, command structures can be adapted to different languages, acoustic environments, and industrial contexts while preserving their operational meaning.

In the long term, this approach enables the creation of reusable translation templates and standardised command structures that support scalable deployment of acoustic Human–Machine Interaction systems across production value networks.

3.3.2.2.3.4 Long-Term Vision

From an Arrowhead perspective, autonomous translation extends beyond the translation of technical data models and communication protocols. UC 2.6 demonstrates how translation can also occur between human communication systems and machine-control systems.

The developed framework therefore establishes a bridge between human language and machine logic, enabling future industrial environments in which workers interact with interoperable machines through intuitive, robust, and context-aware acoustic communication. In this sense, UC 2.6 broadens the concept of autonomous translation from machine-to-machine interoperability towards human-machine interoperability.

3.3.2.2.4 Use Case Engineering Process

3.3.2.2.4.1 State of the Art

Engineering processes for industrial Human–Machine Interaction systems are traditionally highly application-specific. Development activities are often conducted within a single industrial environment and tailored to a predefined hardware configuration, machine type, and user group. While this approach may lead to successful local deployments, it frequently limits scalability, transferability, and interoperability across production value networks.

Furthermore, the development of speech-based industrial interaction systems is often complicated by the need to simultaneously consider acoustic conditions, machine-control requirements, human usability, operational workflows, and interoperability constraints. These aspects are commonly addressed separately rather than through an integrated engineering process.

The UC2.6 engineering process has changed substantially from the initial single-site automotive HRI demonstrator concept. Following the withdrawal of the original industrial partner, the process was strategically adapted into a distributed and generalisable industrial interoperability framework for acoustic Human-Machine Interaction. The EP now combines requirements for robust speech-command recognition, semantic command structuring, microservice-based command evaluation, beamforming-based communication, simulation-based benchmarking using authentic industrial noise, and pilot deployment preparation. The technical evaluation uses the Industrial Noise Corpus (DIN), with more than 100 recorded industrial machines and environmental noise sources. Reported speech-command recognition rates are 81-94%, with an average accuracy of 90.7%, and the practical reduction of repetitive manual interaction effort is estimated at 40-55% for realistic operation because a limited set of highly frequent functions is suitable for speech-based execution.

3.3.2.2.4.2 Enablers

UC 2.6 adopted an iterative engineering approach combining laboratory research, simulation-based validation, industrial stakeholder involvement, and pilot deployment activities.

The engineering process initially focused on a dedicated automotive demonstrator involving collaborative Human–Machine Interaction within a specific industrial production environment. Following the withdrawal of the original industrial partner during the early phase of the project, the engineering strategy was revised and expanded. Rather than focusing on a single deployment scenario, the project evolved towards a more generalisable framework for acoustic Human–Machine Interaction under realistic industrial conditions.

The revised engineering process consisted of five major phases:

1. Collection and analysis of industrial communication requirements through collaboration with industrial stakeholders.
2. Development of a language-adaptable and noise-robust command inventory together with corresponding acoustic feedback concepts.
3. Creation of the Industrial Noise Corpus (DIN) and implementation of immersive simulation environments for controlled testing and benchmarking.

4. Development of the required hardware and signal-processing infrastructure, including directional microphone and loudspeaker systems, speech-enhancement procedures, and microservice-based interoperability components.
5. Validation through laboratory experiments, benchmarking activities, and industrial pilot deployments in authentic production environments.

This engineering process enabled continuous refinement of the interaction framework while maintaining compatibility with the evolving Arrowhead architecture and interoperability objectives.

3.3.2.2.4.3 Arrowhead-EP of the Use Case

EPP1 revised acoustic HMI and industrial-noise requirements -> EPP2 generalisable command-framework and distributed acoustic-HMI architecture -> EPP3 DIN corpus, semantic command inventory, API/browser platform, ASR workflows, beamforming microphones/loudspeakers and translation layer -> EPP4 laboratory validation, 3D acoustic simulation and pilot deployment preparation at Anthon and Arla, with final Arrowhead cloud integration still ongoing -> EPP5 runtime speech-command evaluation, acoustic feedback and selected machine-control execution -> EPP6 command robustness improvement, hardware-position optimization, translation-template refinement and deployment maintenance -> EPP7 evolution toward standardised, multilingual, multimodal and cross-domain acoustic HMI services -> EPP8 user onboarding, deployment guidelines, manuals and dissemination. Scalability level: high. The revised EP is no longer bound to one robot or one factory; it can be recursively applied to command sets, noise environments, machine types and industrial sites, although final cloud integration and pilot completion remain ongoing.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	Revised HMI requirements; industrial noise requirements; interoperability and usability requirements	Captures robust speech-command, distributed infrastructure, usability, multilingual and deployment constraints.
EPP2	Generalisable acoustic-HMI framework; semantic command structure; distributed deployment concept	Defines the target architecture after the shift from a single-site demonstrator to a reusable framework.
EPP3	DIN corpus; 3D acoustic simulations; SDU vocabulary and API; browser platform; SoundFocus beamforming hardware; translation layer	Develops the acoustic HMI, benchmarking, command evaluation, directional communication and command-to-control technologies.
EPP4	Laboratory validation; API/interoperability tests; pilot preparation at Anthon GmbH and Arla A/S; Arrowhead cloud integration	Deploys and validates services and hardware in lab and pilot-oriented environments.
EPP5	Speech-command recognition; command robustness evaluation; selected runtime machine-control execution	Operates the HMI services and evaluates command suitability under realistic noise conditions.
EPP6	Command robustness refinement; hardware positioning; translation-template improvement; deployment maintenance	Maintains and improves acoustic performance and machine-control mapping during deployment.
EPP7	Standardisation roadmap; multilingual/multimodal extension; cross-domain exploitation	Evolves the framework beyond automotive production toward general industrial acoustic HMI.
EPP8	User onboarding; deployment guidelines; manuals; open-access corpus and dissemination	Supports learning, transfer and repeatable deployment across industrial sites.

Table 26: use case Arrowhead EP.

3.3.2.2.4.4 Mapping of tools in the Arrowhead-EP of the Use Case

The updated mapping table above represents the UC2.6 decomposition after incorporating the final WP6 update. The main engineering-process change is the shift from a local HRI demonstrator toward a reusable acoustic-HMI service stack: industrial noise data, command semantics, API services, browser access, beamforming hardware, simulation validation and pilot deployment are now explicitly mapped across the EPPs. EPP8 is significant because user onboarding, manuals and deployment guidelines are still required to convert the technology into repeatable industrial practice.

3.3.2.2.4.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - change from design time to run time engineering	The command framework is evaluated under runtime-like industrial noise using DIN recordings, 3D simulation and pilot deployment workflows; feedback refines commands, hardware positioning and translation logic.
Obj. 2 - move from single to integrated multi stakeholder automation and digitalization	The revised EP integrates SDU, SoundFocus and pilot stakeholders through a distributed acoustic-HMI platform, API services, semantic commands and machine-control translation.
Obj. 3 - handling increased number of I/Os	The EP manages many acoustic inputs and outputs: noise recordings, speech/audio uploads, command text files, microphone-array signals, loudspeaker feedback, recommendation outputs and machine-control mappings.
Obj. 4 - digital learning and training	The open DIN corpus, browser platform, onboarding activities, user manuals and deployment guidelines become integral engineering assets for training and replication.

Table 27: Arrowhead EP objectives.

3.3.2.2.4.6 Adopted toolchain

Industrial Noise Corpus (DIN) with authentic field-recorded noise environments, immersive 3D acoustic simulation workflow, SDU command-vocabulary development and semantic structuring, ASR and speech-processing workflows, SoundFocus beamforming microphone and loudspeaker arrays, modular API and browser-based command-evaluation platform, Arrowhead ecosystem integration, translation layer between speech commands and machine-executable logic, pilot deployment workflows for Anthon GmbH and Arla A/S, and onboarding/deployment documentation.

3.3.2.2.4.7 Toolchain automation

Medium-high for laboratory and prototype-service validation, and medium for industrial deployment. functional validation of the API and interoperability concepts, reliable browser/API operation during validation activities, and 81-94% speech-command recognition rates with a 90.7% average under realistic noise conditions. Remaining automation gaps are final Arrowhead cloud integration, completion of pilot deployments, higher automation of speech-command to machine-control translation, multilingual and multimodal extension, and standardized deployment guidance.

3.3.2.2.4.8 Toolchain limitations

Final Arrowhead cloud integration and the completion of industrial pilot deployments remain ongoing activities.

The translation layer from speech-command structures to machine-control systems is highly deployment-specific and requires reusable templates.

Multilingual, multimodal and cross-domain command standardisation require further extension beyond the validated prototype scope.

User onboarding, hardware-position optimisation, deployment manuals and operational guidance must still be consolidated for repeatable industrial rollout.

3.3.2.2.4.9 Expected Impact

The adopted engineering process provides a scalable methodology for the development of future acoustic Human–Machine Interaction systems. By combining simulation-based development, open benchmarking resources, industrial stakeholder involvement, and iterative deployment activities, the approach reduces dependency on individual industrial partners and facilitates transferability across different industrial sectors.

The process furthermore demonstrates how human-centred interaction technologies can be developed within the broader context of interoperable production value networks, thereby supporting future deployment, standardisation, and cross-domain adoption.

3.3.2.2.5 Use Case and Engineering Costs

3.3.2.2.5.1 State of the Art

The development of industrial Human–Machine Interaction (HMI) systems typically requires substantial engineering effort due to the need to integrate hardware, software, signal processing, machine-control interfaces, and user-centred interaction concepts. Additional costs arise from deployment-specific adaptation, testing under realistic environmental conditions, and the validation of operational robustness.

In the case of speech-based industrial interaction, engineering costs are further increased by the need to evaluate system performance under varying acoustic conditions, languages, speaker characteristics, and machine environments. As a result, many existing solutions remain highly customised and difficult to transfer across industrial domains.

3.3.2.2.5.2 Baseline Costs and Challenges

At project start, the primary engineering costs were expected to arise from:

- development of acoustic Human–Machine Interaction concepts,
- acquisition and processing of industrial noise data,
- design and implementation of directional communication hardware,
- development of speech-processing and interoperability services,
- validation under realistic industrial conditions, and
- integration with existing machine-control infrastructures.

The withdrawal of the original industrial partner required a restructuring of the deployment strategy. However, this restructuring also created opportunities for developing a more generalisable and scalable framework that is applicable across multiple industrial environments rather than a single deployment site.

3.3.2.2.5.3 Enablers and Expected Cost Reductions

Several project outcomes are expected to reduce future engineering effort and deployment costs.

The Industrial Noise Corpus (DIN) provides a reusable benchmarking resource that reduces the need for repeated data collection activities. The modular microservice architecture enables re-use of core functionalities across deployment scenarios, reducing implementation effort for future industrial users. Furthermore, the development of a standardised command inventory and evaluation framework decreases the amount of manual optimisation required when adapting speech-based interaction systems to new environments.

To summarize, these developments support a transition from highly customised HMI solutions towards reusable and scalable interoperability components, thereby reducing engineering effort and facilitating broader industrial adoption.

A qualitative relative cost profile is used, based on the observed engineering activities. For UC2.6 the Y3 update increases the relative effort of EPP3-EPP5 and EPP8 because the revised process includes DIN corpus creation, acoustic simulation, API/browser platform development, beamforming hardware/software integration, pilot preparation and deployment documentation. The cost profile remains qualitative because the available inputs do not provide man-month or EUR data.

Arrowhead-EP phase	Relative engineering cost/effort	Comment
EPP1 - Requirements	Medium-high	Requirements were redefined after the partner withdrawal and expanded to cross-domain acoustic HMI.
EPP2 - Functional design	High	Framework redesign, semantic command structuring and distributed deployment architecture required substantial effort.
EPP3 - Procurement and Engineering	Very high	DIN corpus creation, acoustic simulation, API/browser platform, beamforming hardware and translation layer dominate engineering effort.
EPP4 - Deployment and Commissioning	High	Laboratory validation, API/interoperability testing and pilots at Anthon and Arla are effort-intensive.
EPP5 - Operations and Management	Medium-high	Runtime command evaluation and selected machine-control execution require realistic noise and workflow testing.
EPP6 - Maintenance / Decommissioning / Recycling	Medium	Command refinement, hardware positioning and translation-template maintenance are deployment-specific.
EPP7 - Evolution	Medium-high	Cross-domain, multilingual, multimodal and standardisation extensions are central to the exploitation path.
EPP8 - Training and Education	Medium-high	User onboarding, manuals, deployment guidelines and open dissemination are necessary for repeatable adoption.

Table 28 – engineering costs.z

3.3.2.2.6 Standardization Requirements

	Baseline	Expected Final Status
--	----------	-----------------------

Categories of Standards		
Human–Machine Interaction	Proprietary machine-specific command structures and interaction concepts. Limited interoperability across systems and deployment environments.	Language-adaptable and semantically structured command inventory supporting interoperable acoustic Human–Machine Interaction across industrial environments.
Information and Representation	Machine-specific command definitions and operational procedures. Limited standardisation of speech-command structures.	Structured command inventory linking semantic intent, acoustic realisation, and machine-control logic.
Semantic and Language	Natural language commands without standardised representation or interoperability framework.	Semantically organised command structures supporting multilingual adaptation and future standardisation efforts for acoustic industrial communication.
Communication	Conventional machine interfaces such as buttons, touch panels, and proprietary control systems.	Integration of acoustic communication channels through speech commands, acoustic feedback, browser-based APIs, and interoperable microservices.
Acoustic Infrastructure	No standardised benchmarking resources for industrial acoustic Human–Machine Interaction.	Open-access Industrial Noise Corpus (DIN) enabling benchmarking, evaluation, simulation, and future standardisation activities.
Cybersecurity and Safety	Existing industrial machine-control procedures and safety protocols.	Integration of acoustic interaction concepts into existing industrial safety and machine-control frameworks without modification of core machine-control architectures.
Reference Models	Application-specific Human–Machine Interaction concepts.	Interoperable Human–Machine Interaction framework integrated into the Arrowhead ecosystem and production value network architecture.
Development and Specific Applications	Isolated deployment-specific speech interfaces.	Reusable API-based evaluation platform and modular microservice architecture supporting deployment across multiple industrial domains.

Table 29: UC 2.6 Standardization Requirements

3.3.2.2.7 Key Performance Indicators

Key Performance Indicator (KPI)	KPI Description	Means of Assessment	Baseline	Expected Final Status	Achieved at year 3
Command Recognition Accuracy	Correct identification of speech commands under realistic industrial noise conditions.	Recognition accuracy in laboratory simulations and industrial pilot environments.	No dedicated framework available.	Recognition rates between 81% and 94%, with an average accuracy of 90.7%.	Recognition rates between 81% and 94%, average 90.7% obtained through laboratory validation and immersive 3D acoustic simulations using authentic industrial noise recordings.
Industrial Noise Coverage	Number and diversity of authentic industrial noise environments available for testing and benchmarking.	Number of recorded and annotated industrial noise sources.	No dedicated corpus available.	Industrial Noise Corpus (DIN) comprising recordings from more than 100 industrial machines and environmental noise sources.	More than 100 industrial machines and environmental noise sources recorded, annotated and used for validation, benchmarking and simulation activities.
Speech-Command Robustness	Stability of command recognition across varying noise conditions and	Comparative evaluation across multiple industrial soundscapes.	Deployment-specific testing only.	Systematic benchmarking across diverse industrial noise environments.	Command vocabulary validated across multiple authentic industrial soundscapes using the DIN corpus and immersive acoustic simulations.

	acoustic environments.				Recognition performance remained robust under varying acoustic conditions.
Human–Machine Interaction Efficiency	Reduction of repetitive manual machine-control actions through speech-based interaction.	Workflow analyses and pilot deployment evaluations.	Manual interaction required.	Approximately 20–30% of machine-control functions directly addressable by speech commands, corresponding to an estimated 40–55% reduction of repetitive manual interaction effort.	Validation demonstrated that approximately 20–30% of machine-control functions can be replaced by speech interaction, corresponding to an estimated reduction of repetitive manual interaction effort between 40% and 55%.
Interoperability Readiness	Ability to integrate acoustic HMI services into distributed industrial infrastructures.	Validation of API functionality, service deployment, and interoperability testing.	No interoperable acoustic framework available.	Functional integration of acoustic HMI services into the Arrowhead ecosystem.	API and interoperability concepts successfully validated; acoustic HMI services integrated into Arrowhead-based microservice infrastructures. Final cloud integration still ongoing at the time of reporting.
Industrial Validation	Number of industrial pilot environments used for deployment and validation.	Pilot deployment activities and industrial stakeholder evaluations.	No pilot deployments.	Validation activities conducted in manufacturing and production environments in Denmark and Germany.	Pilot deployment activities initiated and executed at Anthon GmbH (Germany) and Arla A/S (Denmark), including hardware optimization, command integration and workflow adaptation.
Open Access Dissemination	Availability of reusable resources for industry and research communities.	Public release of datasets, command inventories, and methodologies.	No common resource available.	Open-access Industrial Noise Corpus (DIN) and openly documented command-evaluation methodologies.	Industrial Noise Corpus (DIN) established; open-access datasets, benchmarking methodologies, publications and dissemination assets generated for future reuse.

Table 30: KPIs in UC 2.6.

3.3.2.3 Beyond the Baseline

3.3.2.3.1 Use Case Action Plan

Following the refinement of the use case and the restructuring of the original deployment strategy, UC 2.6 will proceed through the following major activities:

- Completion and deployment of the browser-based microservice platform for acoustic command evaluation and benchmarking.
- Expansion and dissemination of the Industrial Noise Corpus (DIN) as an open-access benchmarking resource.
- Finalisation of the noise-resilient and language-adaptable command inventory for industrial Human–Machine Interaction.
- Completion of the directional microphone and loudspeaker communication system developed together with SoundFocus ApS.

- Integration of the developed services into the Arrowhead ecosystem and completion of cloud-based deployment activities.
- Validation and optimisation of speech-command translation procedures between human operators and machine-control systems.
- Pilot deployment and evaluation in industrial production environments in Denmark and Germany.
- Development of deployment guidelines, user manuals, and exploitation concepts.
- Validation against the defined KPIs and Arrowhead project objectives.

3.3.2.3.2 Contribution to Project Objectives

Project Objective	Status at M0	Expected Improvement	Planned Actions	Expected Status at M36	Achieved value M36
Obj 1 – Autonomous translation microservices	No speech-based translation framework available	Establish translation layer between human speech and machine-control logic	Command inventory, translation services, API integration	Functional translation framework validated in industrial environments	UC 2.6 developed and validated a noise-robust command vocabulary, automatic speech-recognition pipelines, and services for assessing and translating human commands into machine actions. The translation concept was demonstrated and prepared for deployment, although validation remained limited to project demonstrators and selected industrial scenarios rather than broad industrial adoption.
Obj 2 – Microservices/SOA	No dedicated acoustic HMI microservices	Development of reusable acoustic evaluation and translation services	Browser-based API, command evaluation services, Arrowhead integration	Operational microservice platform available	This is one of the strongest contributions of UC 2.6. D6.3 reports the development of Arrowhead-integrated microservices that allow companies to upload noise samples or command candidates and receive automated suitability assessments via service interfaces. A local Arrowhead cloud was also successfully deployed.
Obj 3 – Major digital data models	Not applicable	Limited contribution	Support semantic structuring of machine-oriented speech commands	Conceptual contribution to future standardisation	The contribution is indirect. UC 2.6 produced a typological and terminological framework for industrial noise and a structured command vocabulary, but it did not directly create or revise major industrial data-model standards. The impact is therefore moderate and mostly conceptual.
Obj 4 – Digital transformation methodology	Traditional machine-operation concepts	Introduction of human-centred acoustic interaction concepts	Industrial pilots and deployment guidelines	Contribution to WP10 and industrial deployment methodology	The use case introduced a novel human-centred interaction paradigm based on speech and sound instead of buttons and touch screens, developed evaluation methods, industrial testing procedures, and deployment concepts. These results contribute significantly to digital transformation practices in manufacturing environments.
Obj 5 – Sustainable governance and implementation platform	No shared infrastructure available	Creation of reusable datasets and benchmarking resources	DIN database, publications, open dissemination activities	Open-access resources and reusable evaluation framework established	UC 2.6 generated reusable assets including the industrial-noise corpus, open-access datasets, publications, open standards selections (e.g., OPUS), and plans for public dissemination through scientific platforms. These outputs provide a strong basis for long-term reuse beyond the project.

Table 31: outline of how updated UC 2.6 Actions relate to project objectives.

3.3.2.3.3 Use Case and Engineering Costs

The technologies developed within UC 2.6 are expected to reduce future engineering costs by replacing deployment-specific development activities with reusable interoperability components.

The Industrial Noise Corpus (DIN) reduces the need for repeated industrial noise recordings during future deployments. The command-evaluation platform allows speech commands to be assessed before implementation, thereby reducing testing effort. Furthermore, the modular microservice architecture enables re-use of functionality across industrial domains and deployment scenarios.

In the long term, the developed framework is expected to reduce engineering effort, deployment complexity, and adaptation costs associated with industrial speech-based Human–Machine Interaction systems.

3.3.2.3.4 Micro-Service Paradigm

3.3.2.3.4.1 Approach

The project adopts a service-oriented architecture in which individual functionalities are encapsulated as interoperable microservices. Core services include industrial-noise benchmarking, command evaluation, command optimisation, speech-processing, and machine-command translation.

Users access these services through browser-based APIs, enabling deployment-independent evaluation of acoustic Human–Machine Interaction concepts.

3.3.2.3.4.2 Impact

The microservice architecture transforms acoustic HMI assessment from a deployment-specific engineering activity into a reusable service. This reduces implementation effort and enables consistent benchmarking across industrial environments.

The approach also facilitates integration with existing Arrowhead infrastructures and supports future extension through additional services and interoperability layers.

3.3.2.3.4.3 Long-Term Effect

The long-term vision is an ecosystem of reusable acoustic interoperability services that can be deployed across multiple production value networks. Industrial stakeholders will be able to evaluate, optimise, and deploy speech-based interaction concepts without requiring extensive redevelopment for each new environment.

3.3.2.3.5 Major Digital Languages

3.3.2.3.5.1 Approach

UC 2.6 introduces semantically structured command inventories as an intermediate representation between human language and machine-control logic. Commands are organised according to communicative intent and operational function rather than linguistic form.

This enables adaptation across languages and industrial contexts while preserving semantic consistency.

3.3.2.3.5.2 Impact

The resulting framework provides a basis for interoperable machine-oriented speech communication. It supports multilingual deployment, improves transferability across industries, and reduces adaptation effort when introducing acoustic interaction technologies into new environments.

3.3.2.3.5.3 Long-Term Effect

The developed command structures may contribute to future standardisation activities concerning industrial speech commands and acoustic interoperability. The framework provides a foundation for reusable command inventories that remain stable across deployment scenarios while allowing local linguistic adaptation.

3.3.2.3.6 Autonomous Translation

3.3.2.3.6.1 Approach

UC 2.6 implements a translation layer connecting human speech commands and machine-executable operational logic. Rather than replacing existing machine-control systems, the project maps acoustic command structures onto already established operational procedures. Speech commands are recognised, semantically interpreted, translated into machine-oriented instructions, and executed through existing control architectures. Machine states and operational outcomes can subsequently be transformed into acoustic feedback directed towards the operator.

3.3.2.3.6.2 Impact

The translation layer significantly lowers barriers for introducing speech-based interaction into industrial environments. Existing machine infrastructures remain unchanged while gaining access to an additional, intuitive communication channel.

The approach enables workers to interact with machines through natural communication processes while maintaining compatibility with established operational procedures.

3.3.2.3.6.3 Long-Term Effect

The long-term objective is the establishment of reusable translation templates capable of connecting human communication systems with machine-control architectures across industrial domains.

This extends the Arrowhead concept of translation beyond machine-to-machine communication and introduces a scalable framework for human-machine interoperability.

3.3.2.3.6.4 Other Aspects

The translation process involves three principal entities:

- Human operators issuing speech commands.
- Translation services performing semantic interpretation and command mapping.
- Industrial machines executing translated operational logic.

The primary translation challenge concerns the mapping between speech-command structures and machine-control procedures. The Industrial Noise Corpus (DIN) provides the acoustic data foundation for testing and validating these translation mechanisms under realistic industrial conditions.

3.3.2.3.7 KPIs, Evaluation and Validation Process

Validation activities are conducted through a combination of laboratory-based experimentation, immersive acoustic simulation, benchmarking using the Industrial Noise Corpus (DIN), and industrial pilot deployments.

The evaluation process focuses on:

- speech-command recognition accuracy,
- robustness across industrial noise environments,
- interoperability within Arrowhead infrastructures,
- reduction of repetitive manual machine interactions,
- usability and operator acceptance,
- scalability across industrial domains.

The developed framework has demonstrated recognition rates between 81% and 94%, with an average performance of 90.7% under realistic industrial conditions. Validation activities furthermore confirmed the feasibility of integrating acoustic Human–Machine Interaction services into distributed industrial infrastructures and demonstrated substantial reductions in repetitive manual interaction effort.

The KPI framework will continue to support future deployment activities, benchmarking procedures, and standardisation efforts beyond the duration of the project.

3.3.2.4 Final assessment

3.3.2.4.1 KPI assessment

The final assessment of UC2.6 shows that all seven defined KPIs were met. The acoustic HMI framework hit speech-command recognition rates between 81% and 94%, with an average accuracy of 90.7% in real industrial noise. To validate this, tests were run with both simulated and real industrial sounds.

One major success was creating the DIN, the Industrial Noise Corpus. It has over 100 industrial machine and environmental noise recordings. This collection served as the foundation for evaluations across different industrial settings.

Also, around 20–30% of control tasks could be handled by voice, cutting down the need for manual actions by about 40–55%. The API and microservice framework seamlessly integrated

into the Arrowhead ecosystem, showing its interoperability. Plus, pilot tests in Germany and Denmark confirmed the system worked well in actual production settings.

Overall, these results prove that acoustic HMI systems based on Arrowhead principles are technically viable, relevant to industry, and have good interoperability prospects.

3.3.2.4.2 Contribution to project objectives

UC2.6 contributed to Arrowhead fPVN's objectives on self-guided translations and service-oriented interoperability. It created a system that turns people's voice commands into actions machines can follow. This expanded how machines talk to humans based on existing tech for machines to talk to each other.

The use case obtained also very good results using the microservice approach and creating services that fit into the Arrowhead framework. This lets them check how well speech commands work in noisy factories. Plus, they made a structured list of commands and set up some useful tools, like the DIN database and ways to test its performance. These contributions give a solid base for future standards and more uses down the line.

3.3.2.4.3 Improvement in the engineering process

UC2.6 demonstrated a more scalable engineering approach for industrial Human–Machine Interaction systems by replacing deployment-specific developments with reusable parts, benchmarking tools, and service-oriented solutions. The Industrial Noise Corpus lets acoustic interaction systems be tested systematically, cutting down work for later testing and setup.

With its modular microservices and organized command system, integrating speech-based interactions in industrial settings gets easier. This reduces the adaptation effort and helps with deploying across different industries. Thus, the use case sets up a reusable engineering guide for future acoustic HMIs.

3.3.3 WP7 - Use-case #1.7 – Interoperable intelligent management of production lines: Towards Model-based Enterprise

3.3.3.1 Baseline summary

3.3.3.1.1 General description of the baseline

The products related to the use-case, in most cases, are electromechanical products to be produced where in their design, different departments (electronics and mechanics) must work in parallel to reach a single product. The integration and efficient management of all this development up to production is still not fully automated. Once the designs are ready (after the pertinent revisions required also by the aerospace domain regulations), they are launched to the factory and nowadays, both worlds, OT and IT are not completely connected, and it is not evident to keep the traceability of the product. Furthermore, once the product reaches the customer, it is currently not possible to track the operation of these products. There is a need to link the products with the interoperable operational data to improve future designs.

The goal is to deploy an intelligent method system that can effectively communicate and exchange data between different systems, resulting in a more efficient and error-free production process.

This use case is part of the larger context of intelligent manufacturing, which covers the entire life cycle of a product, from design to maintenance. The implementation of interoperable intelligent management of production lines in aerospace requires a collaborative and interdisciplinary approach to system development, as well as the use of advanced tools and techniques such as MBSE.

3.3.3.1.2 Architecture of the use case as a sequential list of functional blocks

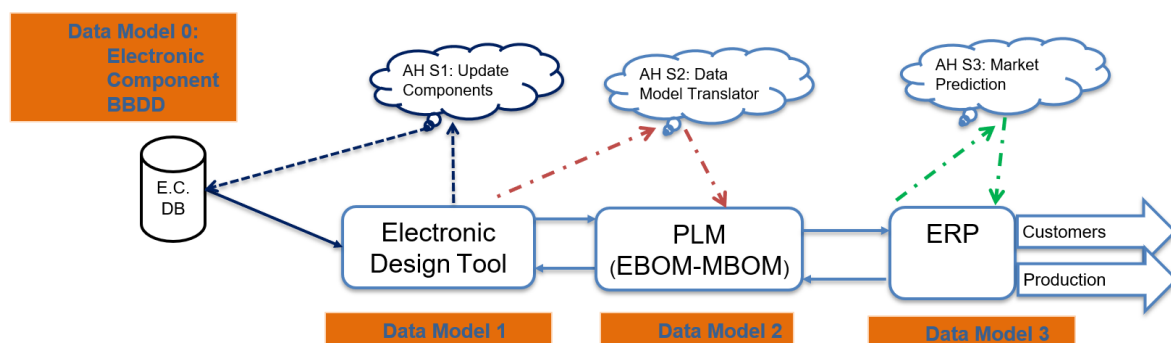


Figure 10: UC 1.7 architecture.

The use case represented in Figure 10 involves several components that need to be interoperable in order to manage the lifecycle of electromechanical products. The architecture of this use case can be broken down into the following components:

- Database: This component stores information about electronic components that will be used by the electronic designer by the electronic designing tool.

- Electronic Design Tool: This tool needs inputs from the database to design electronic components and circuit boards.
- PLM Tool: This tool manages the EBOMs (Engineering Bill of Materials) and generates the MBOM (Manufacturing Bill of Materials). It needs to be interoperable with the electronic design tool to ensure that the MBOM is accurate and up-to-date.
- ERP: This component receives the MBOM from the PLM tool and manages the production, customers, and logistics. It ensures that the right components are ordered and delivered to the right place at the right time.

In order to make it work, during the use case three Arrowhead compliant services will be developed.

On the one hand, there will be a service that will ensure that the information stored in the DDBB is synchronized with the information that is in the PLM. Both these tools are able to update information, but it is not possible to have different information/not synchronized information in both components. Thus, this service is going to be developed since the tools themselves, considering just commercial ones, do not provide this communication and traceability in both directions.

It will be also developed a second service to try how it is possible to translate the different data models that are using the DDBB module and the PLM module.

Lastly, the ambition is to integrate in the process an AI-based service that will help personnel from customer sales when buying the parts. This AI service will be based on historical data of the companies' purchases (delays, quantities bought, providers, ...) to make a market prediction.

3.3.3.1.3 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	The use case will need to translate data models in the different steps of the product lifecycle and these translations will be automated by using micro services that can make this work
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	The different steps of the process, will need to publish and consume services that will translate the data models and to do that a microservices oriented solution has been defined
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	SysML for modelling the process to develop will be considered and used by Developair to translate from requirements defined by a DSL to SysML. In addition, the information of the electronic circuits may follow IPC-2581. Translation of this std also has been considered.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	The methodology to be used during the product lifecycle will follow the MBSE methodology based on Arrowhead services
Obj 5 - Established a	The translation services and intelligent services applied for optimising the sales department

sustainable governance of open-source architecture and implementation platform	decisions will be implemented in the Arrowhead platform
--	---

Table 32: UC 1.7 Arrowhead fPVN objective

3.3.3.2 Baseline analysis - building blocks & enablers

3.3.3.2.1 Micro-service paradigm

3.3.3.2.1.1 State of the art

The current solution and methodology are not based on microservices paradigm. The current solution is a close solution, and any update require to develop every step according to the new requirements. It is not a flexible and open solution.

3.3.3.2.1.2 Enablers

In order to develop a microservice oriented solution, the use case will use the core services of the Arrowhead Framework (Registry, Orchestration and Authorization Services).

3.3.3.2.2 Major digital languages

3.3.3.2.2.1 State of the art

The digital languages or data models used in the use case are the ones to represent electromechanical components and the data models that represent EBOMs and MBOMs needed in the PLM and automatic production lines.

To represent electronic circuits, IPC-2581 standard has been also considered.

3.3.3.2.2.2 Enablers

The ontologies that are able to represent the electromechanical elements to be designed and developed in the production lines will be used.

3.3.3.2.3 Autonomous translation

3.3.3.2.3.1 State of the art

Current status: during the process (starting from design to production and sales), different tools and data models are used. At this moment, the process is semiautomatic but close and specific translators has to be developed to automatize the process.

3.3.3.2.3.2 Enablers

DITAG tool provided by UNINOVA has been identified to use as autonomic translator and in the use case this possibility has been identified as a possible solution to translate different data models that are in the process.

3.3.3.3 Use Case Engineering Process

3.3.3.3.1 State of the art

The use case follows the Arrowhead-EP from requirements through functional design, engineering, deployment and operation of production-line interoperability services. The baseline stage focused on requirements and functional design, while final demonstrations and validation have shown services ready for Arrowhead integration. Decommissioning/recycling is not central to this software/process use case but maintenance and evolution of services are relevant.

3.3.3.3.2 Enablers

MBSE, SysML, IPC-2581, Arrowhead services and service-based translators form the engineering-process framework. The use case demonstrates how an Arrowhead-EP can connect design-time artifacts, such as Rely and electronic design models, with production, logistics and runtime optimization services.

3.3.3.3.3 Arrowhead-EP of the Use Case

EPP1 production-line and traceability requirements -> EPP2 MBSE and translation service architecture -> EPP3 DITAG, AI translator, Rely2SysML and logistics service engineering -> EPP4 Arrowhead microservice deployment -> EPP5 production/PLM/ERP/logistics operation -> EPP6 service maintenance and translator updates -> EPP7 data-model and service evolution; EPP8 demonstrations and expert review support all phases.

Scalability level: High for software/service replication across product lines and projects. Scalability is strongest when data models are standardized and services are registered in Arrowhead; it is lower for proprietary tools with limited APIs.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	Requirement definition, aerospace production-line objectives	Defines traceability, interoperability and optimization needs.
EPP2	MBSE methodology, service architecture, IPC-2581 and SENER EC mapping	Defines data model and service design.
EPP3	DITAG, AITIA AI translator, Rely2SysML, Multiverse optimizer	Develops translators and optimization service.
EPP4	Arrowhead ServiceRegistry, Orchestration, Authorisation, local deployment	Deploys services as Arrowhead microservices.
EPP5	Electronic design tool, database, PLM, ERP, logistics service	Runs the interoperable production-line workflow.
EPP6	Translator updates, mapping correction, service monitoring	Maintains and adjusts services.
EPP7	Improved mappings, new data model coverage, optimization improvements	Evolves service ecosystem.
EPP8	Functional demos, expert review, documentation	Supports adoption and validation.

Table 33: Arrowhead EP mapping.

3.3.3.3.4 Mapping of tools in the Arrowhead-EP of the Use Case

The mapping table above represents the current UC-EP decomposition. Tools and methodologies are grouped by Arrowhead-EP phase, with EPP8 treated as a cross-cutting training and adoption phase.

3.3.3.3.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - The change from design time to run time engineering	Rely/SysML and electronic design data are connected to production, PLM, ERP and logistics runtime services.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The use case defines a process/methodology starting from design to production of electromechanical components. The automatization and digitalization of the process will be the contribution. SENER, UNINOVA, AITIA, Developair, CEA-LIST and Multiverse provide interoperable services in one Arrowhead ecosystem.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	The service architecture handles multiple PCB/project files, PLM/ERP exchanges and logistics inputs. This automation will contribute in avoiding human errors and at the end more efficient and correct electromechanical parts will be developed. As a result, less defects and parts to be recycled will be produced and this will impact in sustainability.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Mondragon University will be helping in the development and integration of the Arrowhead services and the knowledge gained in this process will be also transferred to the future engineers. Demos, V&V workflows and expert review generate reusable knowledge for service adoption.

Table 34: Arrowhead EP objectives.

3.3.3.3.6 Adopted toolchain

The use case is using commercial close tools at different steps of the process (Altium for the electronic design, Dassault Systems PLM, Microsoft Navision, ...) and lifecycle. DITAG deterministic translator, AITIA AI-based data model translator, Rely2SysML model translator, Papyrus/SysML workflow, SENEC EC Data Model, IPC-2581, electronic design tool, PLM, ERP, Multiverse quantum-inspired logistics optimization, Arrowhead core services.

3.3.3.3.7 Toolchain automation

High for demonstrated translation and optimization scenarios. Automation gaps remain in generalized mapping coverage, expert validation of semantic equivalence, and integration with all proprietary lifecycle tools.

At this point, in some cases there are close connectors that link the different tools (Altium+Dassault Systems) but they are close solutions. The use case wants to work with an open alternative that enables the automation of the toolchain between them.

3.3.3.3.8 Toolchain limitations

Limitations: close solutions. Specific developments are needed everytime that an evolution of one of the tools is launched or a new data or new requirement is defined in the overall process. Initial data model mapping effort remains significant.

Some consuming systems are proprietary and require adapters.

Decommissioning/recycling phase is not meaningful for the main software workflow.

Semantic quality and edge cases still need expert validation beyond functional demos.

3.3.3.3.9 Use case and engineering costs

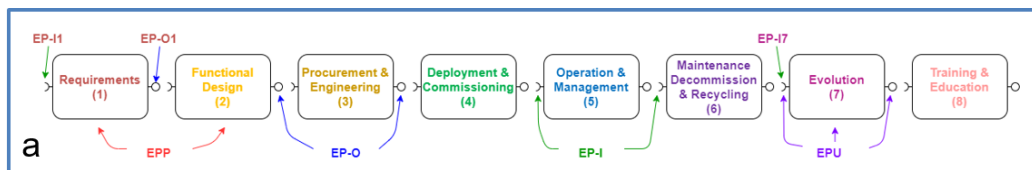


Figure 11: the Arrowhead EP.

A qualitative relative cost profile is used, based on the observed engineering activities, toolchain integration needs and validation burden.

Arrowhead-EP phase	Relative engineering cost/effort	Comment
EPP1 - Requirements	Medium	3 normalized man-month
EPP2 - Functional design	Medium	7 normalized man-month
EPP3 - Procurement and Engineering	High	20 normalized man-month
EPP4 - Deployment and Commissioning	High	10 normalized man-month
EPP5 - Operation and Management	Medium	10 normalized man-month
EPP6 - Maintenance / Decommissioning / Recycling	Low	5 normalized man-month
EPP7 - Evolution	Medium	5 normalized man-month
EPP8 - Training and Education	Medium	5 normalized man-month

Table 35: engineering process costs.

3.3.3.3.10 Standardization requirements

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	Life Cycle Management, Software life cycle processes, REST APIs, ERP/MES, Function Blocks	Life Cycle Management, MBSE, Software life cycle processes, REST APIs, ERP/MES, Function Blocks

Information and Representation	Product data representation and exchange,	Product data representation and exchange , Framework for object-oriented information exchange Worksite data exchange,
Semantic and Language	XML, JSON,	XML, JSON, SysML, UML,
Communication	Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP	Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP
Cybersecurity and Safety	Authentication,	Authentication, Safety for Electrical/programmable electronic ??
Reference Model		RAMI, Digital Factory
Domain-Specific	Integration Life Cycle	Integration Life Cycle,
Framework Development and Specific Applications for development	SQL Server, Java	JAVA, Eclipse, Python, Hadoop, Spark, SQL Server, SQL Lite

Table 36: UC 1.7 Standardization Requirements

3.3.3.3.11 Key Performances Indicators

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation (expected year results)			Achieved value Year 3
Name	KPI Description	Means of assessment	Year 1	Year 2	Year 3	
Interoperable	Automatization of the lifecycle workflow	Number of steps automatized: data models translated automatically	20%	70%	100%	100%
Logistic Routes Optimization	Quantum Inspired Logistic Route's Optimization service	Intelligent service able to optimize logistics routes depending on the needs of the companies and the context	10%	60%	100%	100%
SOA Model	Service Oriented Architecture	The use case will be deployed in a SOA architecture	10%	40%	100%	100%

Table 37: UC 1.7 KPIs.

3.3.3.4 Beyond the baseline

3.3.3.4.1 Use case action plan

In this use-case the aim is to work on three aspects that is believed that will improve the productivity of the electromechanical products:

- Prepare the PLM system so that electronic and mechanical designs are automatically integrated, thus generating manufacturing orders automatically and intelligently in a more efficient way.

- Automate the planning and management of plant production through intelligent agents based on customer characteristics/requirements and the status of the designs (electronic and mechanical).
- Enable a system to have interoperable and up to date product information in operation so that this information can be used to improve future designs increasing efficiency and in addition also improving the carbon footprint achieving better sustainability results

The specific requirements and the plan that have been defined for the use case in order to achieve the aforementioned topics, are the following:

Design phase: M0-M9

- Analyse the data models used during the different steps of the MBSE methodology deployed in the use case
- Define the transformations that are needed in each step of the MBSE methodology deployed in the use case.
- Analyse the market information of the Electronic Components in order to develop an intelligent algorithm to predict the most cost-effective components when buying the components for each project.

Development phase: M9-M30

- Develop the automatic translator able to solve the interoperability issues when linking the different tools.
- Develop an intelligent algorithm that enables the market prediction of electronic components.

Deployment phase: M26-M32

- Deploy the translator in the use case.
- Deploy the intelligent algorithm that will be linked to the MBSE methodology to help making decisions to the sales department.

Validation phase: M32-M36

- Validate the automatic integration of the PVN in the use case.
- Validate the intelligent algorithm that will be linked to the MBSE methodology to help making decisions to the sales department.

3.3.3.4.2 Contribution to project objectives

Project Objective	Status at M0	Expected improvement	Planned actions	Expected at M12	Expected at M24	Expected at M36	Achieved at M36
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based	Not automatic translation	Automatize process by using automatic translators	Develop or use services/tools able to translate data	Translators Tools/Services defined	Translators Developed	Translators Integrated in the use case	Demo prepared with 3 translators

translation micro-services thus significantly reducing the need for human support.			models such as DITAG				
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	No SOA architecture	More flexible solution that will enable the evolution of the solutions or integration of new stakeholders in an easy way	Every step, any translation/integration or communication with other tools/frameworks will be done by services	Architecture and Services defined	First services developed	All services developed and integrated. A SOA solution available	Translator and optimization service integrated.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	No contributions	Use of standards to represent data of the domain of the use case	Identify the digital data model used by the use case and consider them in the use case	Std. IPC-2581 identified as a possible standard to represent the electronic circuits projects	If IPC-2581 has been decided to be used: implement its translator	If IPC-2581 has been decided to be used: integrate its translator	IPC-2581 translator developed and integrated in the architecture.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	No autonomous methodology in operation	Define and launch a MBSE methodology for electromechanical components in aerospace	The product workflow and lifecycle will be traceable automatically	First steps of the process implemented	More than two steps/actors of the process integrated using the MBSE methodology	MBSE methodology launched in the company	MBSE methodology integration modules developed and Sysml translator working.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	No open source solution/platform	Use of open source and flexible solution to automatize the electromechanical components for aerospace	Arrowhead based solution will be developed and deployed in validation	First service implemented using the Arrowhead framework and its core services	Eclipse Arrowhead platform based solutions development for new services (following the Model Based translators using Papyrus)	All the development finished using the Arrowhead Eclipse tooling from the design to deployment.	Demos available. DITAG, AITIA's IA based translator, Rely2Sysml and Logistic optimization services available

Table 38: UC 1.7 - Contribution to project objectives.

3.3.3.4.3 Use case and engineering costs

See previous section.

3.3.3.4.4 Micro-service paradigm

3.3.3.4.4.1 Approach

This use case will use the Arrowhead framework to implement the micro-service paradigm. To do that, the core services of the Arrowhead platform will be used (registry, authorization, orchestration) and service providers such as data model translators, or market predictors will be developed. In order to link all the development process of the electro-mechanical components, consumers of those services will also be developed and deployed in the use case.

3.3.3.4.4.2 Impact

The main impact of having implemented this approach, will be the automatic integration of the different steps of the process and the capability to have the traceability of the products. Instant exchange of data between the PNV stakeholders will be also enabled.

3.3.3.4.4.3 Long-term effect

As long-term event, the use of micro-services paradigm will provide a more flexible approach to the lifecycle of the products enabling the incorporation of new stakeholders or the evolution of new products or processes.

3.3.3.4.5 Major digital languages

3.3.3.4.5.1 Approach

The digital languages to be used will be SysML to model the lifecycle of the products and the processes involved in and also data models that define the electronic components that will be integrated into the electromechanical components/products for the aerospace industry.

In addition, IPC-2581 standard has been identified to be used as a reference to represent part of the electronic circuits' information.

3.3.3.4.5.2 Impact

The impact of adopting digital languages will be that the development of the solution will be automatized because the tooling that enables the development of the services and the logics to integrate the different stakeholders will allow first to design/model the lifecycle model and then generate automatically the skeleton of all the services that will link the different steps of the process automatically.

3.3.3.4.5.3 Long-term effect

The long-term effect will be that any evolution or new services that will be integrated into the process can be developed and deployed with less development effort.

3.3.3.4.6 Autonomous translation

3.3.3.4.6.1 Approach

The use case needs to translate the data models to be used in the different steps of the life cycle of the electromechanical products. To do that the use case intends to use automatic translators and to start this work, the idea is to use the DITAG tool developed by UNINOVA.

3.3.3.4.6.2 Impact

The use of this type of technology will enable the efficient adaptation and compliance to standards and automatic information throughout assets and product life cycle.

3.3.3.4.6.3 Long-term effect

As long-term effect, we expect to have a more standardized and automatized PVN in the development of electromechanical components. This will impact also in having less defects in the process because of the automatization (humans' errors and delays in the process will be avoided).

3.3.3.4.6.4 Other aspects

The data models that will be used in the use case will be the ones that represent the electronic components and EBOMs and MBOMs of the production assets.

3.3.3.5 KPIs, Evaluation, and validation process

A validation and verification plan is strategic to be used for testing a product, service, or system to ensure that it meets the requirements and specifications and that it satisfies its intended purpose. Based on the development plan defined in the previous sections, the validation and verification plan of this use case can be broken down into three phases: initial validation, mid-term validation, and final validation.

The initial validation phase, which occurs at M15, involves testing the first version of the services in a lab environment. This phase is critical to ensure that the services are functioning as intended and to identify any issues that need to be addressed before moving on to the next phase.

The mid-term validation phase, which occurs at M27, involves testing the second version of the services in a lab environment and integrating at least one service and proving it by using a consumer in a real environment. This phase is important to ensure that the services are working together seamlessly and that they are meeting the needs of the consumer.

The final validation phase, which occurs at M33, involves conducting a final demo to ensure that the services are fully functional and meeting all the requirements and specifications. This phase is critical to ensure that the services are ready for production and can be used by customers and logistics.

In summary, this will be the plan for the use case:

- Initial validation (M15)
 - First version of the services in lab environment
- Midterm validation (M27)
 - Second version of the services in lab environment
 - At least one service integrated and proved by using a consumer in real environment
- Final validation (M33)
 - Final Demo.

3.3.3.6 Final assessment

3.3.3.6.1 KPI Assessment

UC1.7 successfully achieved all the KPIs defined for the use case. The objective of establishing an interoperable and intelligent management framework for aerospace production lines was validated through the implementation of translation services, optimization services, and a service-oriented architecture connecting the different stages of the product lifecycle.

The first KPI, related to the automation of the lifecycle workflow, achieved the target value of 100%, demonstrating that the identified data-model translations between the different stages of the process can be automatically performed through the developed translation services. This represents a significant improvement compared to the baseline situation, where data exchanges relied on closed integrations and specific developments.

The second KPI, concerning Logistic Routes Optimization, also achieved 100%. The quantum-inspired optimization service developed within the use case successfully demonstrated its capability to support logistics planning and decision-making based on company requirements and operational constraints. The service was integrated into the overall architecture and validated through demonstrator activities.

The third KPI, related to the adoption of a Service-Oriented Architecture (SOA), reached 100%, confirming the successful deployment of the use case using Arrowhead-compliant services and microservice-based integration mechanisms. The final demonstrator included operational translation services and optimization services deployed within the Arrowhead ecosystem.

3.3.3.6.2 Contribution to Project Objectives

UC1.7 contributed directly to the Arrowhead FPVN objectives by demonstrating how autonomous translation services and microservice-based architectures can support the complete lifecycle of electromechanical products, from design to production and logistics. The use case successfully integrated three translation solutions, including DITAG, an AI-based translator, and a Rely-to-SysML translator, thereby supporting the project objective of reducing manual intervention in data-model translations.

The use case also validated the application of Eclipse Arrowhead as a service-oriented interoperability framework. Translation services and optimization services were integrated into a common SOA architecture, demonstrating the feasibility of dynamic service deployment and utilization within production value networks.

Furthermore, UC1.7 contributed to the adoption of major digital languages and model-based engineering approaches through the use of SysML, MBSE methodologies, and the IPC-2581 standard for electronic design information. The successful implementation of the IPC-2581 translator and the integration of MBSE modules demonstrate the applicability of standardized digital models for improving interoperability throughout the product lifecycle.

3.3.3.6.3 Improvement in the Engineering Process

UC1.7 demonstrated a clear evolution from a fragmented and tool-specific engineering process towards an integrated model-based engineering environment. At the beginning of the project, interoperability between design tools, PLM systems, ERP systems, and logistics processes

relied on proprietary integrations and required significant manual effort whenever modifications or extensions were introduced.

The developed solution introduced automated translation services, MBSE-based workflows, and Arrowhead-enabled interoperability mechanisms that allow information to flow automatically across the different lifecycle stages. This improves traceability, reduces manual intervention, and simplifies the integration of new stakeholders, tools, and services.

In addition, the adoption of standardized digital models and service-oriented architectures creates a more flexible engineering environment, reducing the effort required to evolve the system and supporting future scalability of aerospace production networks. The integration of intelligent optimization services further contributes to more efficient planning and decision-making processes across the value chain.

3.3.4 WP7 - Use-case #2.7 – Aircraft Health Management System (AHMS) for Trend Monitoring, Predictive Maintenance and Fleet Operations & Maintenance Simulation

3.3.4.1 Baseline summary

3.3.4.1.1 General description of the baseline

Current embedded hardware platforms are not capable of delivering the required processing power to manage the amount of data generated by the aircraft. Digital twin technologies have to be used to design aircraft and their required support, enabling assessment of alternatives through simulations.

Since these activities comprehend all the life cycle phases of an aircraft, there are several involved stakeholders: Customers, Industry, and Suppliers. Some proof of concepts of Aircraft Health Management System (AHMS) have been developed to test the possible benefits in terms of data processing, maintenance procedures optimization and effort reduction.

A potential improvement of the current AHMS, powered by Digital Twins, high performance embedded computers and interoperability between the three stakeholders is envisaged to offer new services aimed at increasing aircraft availability.

3.3.4.1.2 Initial architecture of the use case as a sequential list of functional blocks

The AHMS is aimed at gathering, collecting, and analysing data concerning aircraft fleet maintenance.

A first prototype has been developed through the Cyber Physical Systems For Europe Project (CPS4EU), as depicted in the figure below:

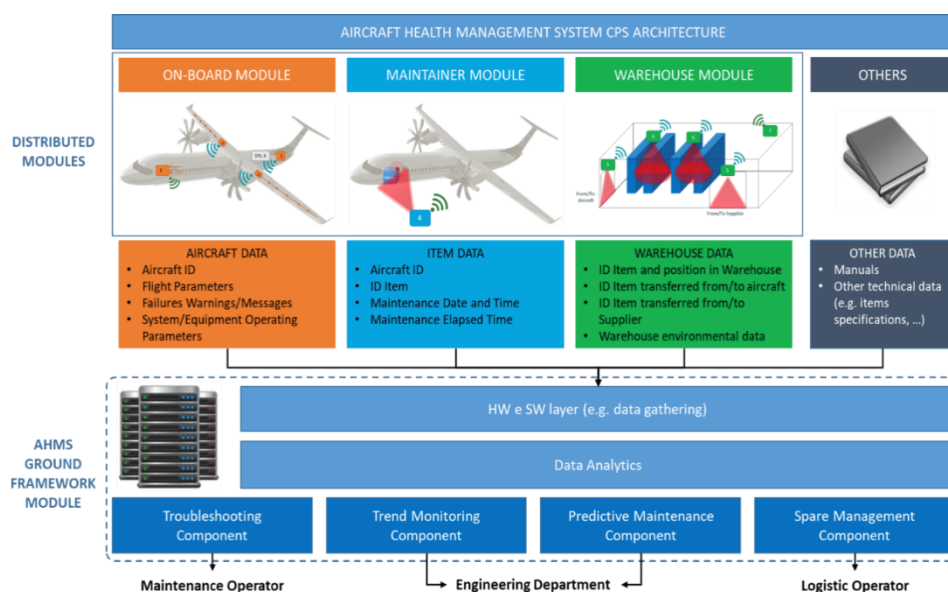


Figure 12 - The use case architecture.

Data coming from aircraft belong to two main categories: failures (i.e. events having a possible impact on aircraft availability) and performances (to be used to monitor aircraft systems health status). The first has to be fixed as soon as possible, the latter has to be used to anticipate future possible failures.

Regarding warehouse data, they are related to equipment/components removed from the aircraft to be repaired (at Customer or Supplier premises) and equipment/components available as spare in the warehouse.

Other data (e.g. maintenance data, manuals) are obtained from other sources/external systems that this framework is supposed to be connected with.

AHMS Ground Framework (GF) is a module on ground where the above data are collected and analysed to:

- monitor aircraft performance
- support the decisions of several operators concerning the troubleshooting of failures, the setting of maintenance actions and the spare parts supply.

The output are several digital services, two of which already fully developed during the CPS4EU project:

- **Troubleshooting Component:** This area gives Maintenance Operator decision support on troubleshooting operations, optimizing duration and tools/facilities availability
- **Spare Management Component:** This section permits, through a series of indicators with predictive features, to optimize spare quantification in order to maximize fleet availability, reducing Aircraft On Ground events due to missing parts.

3.3.4.1.3 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	The proposed solution is intended to increase interoperability in translating and exchanging data from the three different sources using S5000F. The subset of S5000F relevant for this use case will be fully translated.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	The AHMS Services should follow a SOA modular architecture and adopt the Eclipse Arrowhead framework as a service bus to manage the services offered and consumed by the various components of the AHMS architecture.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	Data shall be exchanged basing on S5000F and S-SERIES data model
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	The Orchestration module of AHMS allows the automation of repeated data processing operations, encapsulated in workflows, that transform source data, move data between multiple sources and sinks, load the processed data into an analytical data store, or push the

	results straight to a report or dashboard.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	Currently n/a

Table 39: UC 2.7 Arrowhead fPVN objective.

3.3.4.2 Baseline analysis - building blocks & enablers

3.3.4.2.1 Micro-service paradigm

3.3.4.2.1.1 State of the art

We are not aware of SOA platforms adopted for predictive maintenance and digital twins for the vertical domain of this use case.

The Eclipse Arrowhead framework, a microservice based architecture, represents the current state of the art and will be used together with the S5000F standard to reach interoperability inside and between fPVNs. The framework will provide an exchange of information through the routing of messages and data to the base modules that contribute to the development of the AHMS. A High-Performance Embedded Computer will be used to process a simulated data stream.

3.3.4.2.1.2 Enablers

The core services of the Eclipse Arrowhead framework represent the state of the art to enable and improve interoperability between the AHMS systems, adopting a SOA paradigm and micro-service architecture.

3.3.4.2.2 Major digital languages

3.3.4.2.2.1 State of the art

At the moment no standard is adopted for exchanging data between aircraft and its maintenance and supply chain and the ground storage and processing architecture.

The data generated by the aircraft have been defined during the aircraft design phase for engineering and maintenance purposes and are based on custom Company data models.

The same method adopted for maintenance and supply chain data, designed for logistic and contractual purposes.

The transfer of data from the aircraft to the ground architecture for digital services is done manually and on demand, based on Customer requests for specific in-service technical issues.

3.3.4.2.2.2 Enablers

Two main enablers are related to digital languages, that for us are represented by a standardized models for data exchange/sharing, the S-SERIES standard, a common denominator for a set of specifications associated to different integrated logistics support aspects including aeronautics, and the translators developed in WP4 and intended to convert proprietary data format into S-SERIES standard.

3.3.4.2.3 Autonomous translation

3.3.4.2.3.1 State of the art

As anticipated, currently the transfer of data from the aircraft to the ground architecture for digital services is done manually and on demand, based on customer requests for specific in-service technical issues.

The conversion from the aircraft data to information that can be used for digital services does not follow a specific standard, therefore has to be re-assessed each time a new type of aircraft is added to the fleet or if a customer wants to share only a subset of data. Same for maintenance and supply chain data, which can change depending on the Customers software tools (e.g., info-logistic tools, SAP, ...).

The validation of the data received is done only after its reception, and there are no business rules, valid values, or orchestration of the information.

3.3.4.2.3.2 Enablers

The translators developed in WP4 are the enablers, allowing to automatically convert custom data format into a standard (the S-SERIES), and thus improving interoperability, simplifying system integration, improving efficiency of quality of service, etc.

3.3.4.3 Use Case Engineering Process

3.3.4.3.1 State of the art

The engineering process covers requirements for AHMS interoperability, functional architecture for three scenarios (in-flight, landed, simulated), engineering of gateway, edge, translator, digital twin and analytics services, deployment in Arrowhead, operation through dashboards and simulations, and maintenance/evolution through predictive models and S5000F updates.

3.3.4.3.2 Enablers

The Arrowhead-EP is enabled by the Arrowhead local cloud, AeroBridge, modular gateway, edge computing stack, S5000F/S-Series standards, digital twin model and predictive maintenance/fleet simulation toolchains. The process links design-time data model definition with runtime aircraft health management.

3.3.4.3.3 Arrowhead-EP of the Use Case

EPP1 AHMS and S5000F interoperability requirements and Modular Gateway requirements -> EPP2 SOA and scenario architecture -> EPP3 Development of AeroBridge, Modular Gateway Edge Computer, Benchmarking tool for Modular Gateway, Digital Twin and algorithms -> EPP4 Arrowhead local cloud and pipeline deployment -> EPP5 Trend Monitoring, Predictive Maintenance and Fleet Simulation operation -> EPP6 maintenance planning and model updates -> EPP7 S5000F and service evolution; EPP8 dashboard, S-Series and user training.

Scalability level: High for extending AHMS services across aircraft, fleets and maintenance

stakeholders, provided data sources can be mapped to S5000F and the edge/cloud infrastructure can be deployed consistently.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	AHMS objectives, S5000F use-case analysis	Defines interoperability and digital service needs.
EPP2	Provider/Consumer/Control Station architecture, scenarios	Defines SOA and data exchange design.
EPP3	AeroBridge, Modular Gateway, Edge Computer, Digital Twin, predictive algorithms	Develops translation, computing and analytics services.
EPP4	Arrowhead local cloud, data ingestion pipeline, functional tests	Deploys services and validates integration.
EPP5	Trend Monitoring dashboard, Predictive Maintenance webapp, Fleet O&M Simulation	Operates AHMS digital services.
EPP6	Maintenance reports, RUL estimation, model recalibration	Supports maintenance and service continuity.
EPP7	S5000F proposals, additional use cases, model extension	Evolves data models and services.
EPP8	S-Series documentation, dashboards, user workflows	Supports training and adoption.

Table 40: Arrowhead EP mapping.

3.3.4.3.4 Mapping of tools in the Arrowhead-EP of the Use Case

The mapping table above represents the current UC-EP decomposition. Tools and methodologies are grouped by Arrowhead-EP phase, with EPP8 treated as a cross-cutting training and adoption phase.

3.3.4.3.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - change from design time to run time engineering	S5000F data models and AHMS algorithms are connected to live/simulated aircraft health data.
Obj. 2 - move from single to integrated multi stakeholder automation and digitalization	Customers, industry, suppliers and maintenance actors exchange standardized services through the AHMS architecture.
Obj. 3 - handling increased number of I/Os	Flight parameters, fault messages, removal reports, test-bench sensor data and simulation outputs are ingested through services.
Obj. 4 - digital learning and training	Dashboards, mockups, S-Series workflows and simulation services support learning for maintainers and planners.

Table 41: Arrowhead EP objectives.

3.3.4.3.6 Adopted toolchain

AeroBridge translator, S5000F/S-Series modelling, Arrowhead local cloud, data ingestion pipeline, Modular Gateway Edge Computer test bench, digital twin of hydraulic system/test bench, Trend Monitoring dashboard, predictive maintenance algorithms and webapp, Fleet Operations & Maintenance Simulation.

3.3.4.3.7 Toolchain automation

High in selected workflows. Translation through AeroBridge, data ingestion, dashboards and analytics are automated for defined scenarios. Gaps remain in broader coverage of aircraft components, supplier data heterogeneity and validation of predictions across fleets.

3.3.4.3.8 Toolchain limitations

Translation depends on quality and availability of source data.
S5000F coverage is still evolving for health data monitoring.
Predictive models need enough historical removals and operational data for each component.
Full fleet-scale validation requires broader operational datasets.

3.3.4.3.9 Use case and engineering costs

A qualitative relative cost profile is used, based on the observed engineering activities, toolchain integration needs and validation burden.

Arrowhead-EP phase	Relative engineering cost/effort	Comment
EPP1 - Requirements	Medium	5 normalized man-month
EPP2 - Functional design	Medium	7 normalized man-month
EPP3 - Procurement and Engineering	High	30 normalized man-month
EPP4 - Deployment and Commissioning	High	15 normalized man-month
EPP5 - Operation and Management	Medium	15 normalized man-month
EPP6 - Maintenance / Decommissioning / Recycling	Low	5 normalized man-month
EPP7 - Evolution	Low	2 normalized man-month
EPP8 - Training and Education	Medium	5 normalized man-month

Table 42: engineering process costs.

3.3.4.3.10 Standardization requirements

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	Microsoft Azure, PowerBI, Parquet	Microsoft Azure, PowerBI (to be confirmed), Parquet
Information and Representation	No standards adopted	S-SERIES, with particular focus on S5000F
Semantic and Language	XML, JSON, RDF, SQL, Python, UML, HTML/CSS3	XML, JSON, RDF, SQL, Python, UML, HTML/CSS3
Communication	MQTT, WIFI, ARINC Internet Protocols: HTTP, FTP, SSH, SFTP, TCP/IP, IPSEC, SMTP, UDP	MQTT, WIFI, ARINC Internet Protocols: HTTP, FTP, SSH, SFTP, TCP/IP, IPSEC, SMTP, UDP

Cybersecurity and Safety	IEC 62443, ISO 27001, Dir. 94/9 ISO/IEC 80079-34, ISO/IEC 20000-1:2011, ISO 22301:2012 Societal security - Business continuity management systems, ISO/IEC 27001:2013 information security management	IEC 62443, ISO 27001, Dir. 94/9 ISO/IEC 80079-34, ISO/IEC 20000-1:2011, ISO 22301:2012 Societal security - Business continuity management systems, ISO/IEC 27001:2013 information security management
Reference Model	RAMI (IEC 62890), IEC 81346	RAMI (IEC 62890), IEC 81346
Domain-Specific		S-SERIES, with particular focus on S5000F
Framework Development and Specific Applications for development	JAVA, Javascript, .NET, Python, Jupiter	JAVA, Javascript, .NET, Python, Jupiter

Table 43: UC 2.7 Standardization Requirements

3.3.4.3.11 Key Performances Indicators

Key Performance Indicators (KPI)			Achieved in year 3
Name	KPI Description	Means of assessment	
Data exchange capability	- Data continuously collected during flight - Data collected after aircraft landing - Data collected from simulated flights [Time required to retrieve, process and share data between the AHMS subsystems - M0: 1-20 hours - M36: 5 min.]	- Exchange between flying aircraft and ground station will be simulated only, therefore will be tested the capability of a ground system to handle telemetries and exchange them with the AHMS - Post flight data will be provided as input. Will be tested the capability of the Consumer Service to exchange them with AHMS GF - Simulation results will be provided as input. Will be tested the capability of the Consumer Service to exchange them with AHMS GF.	As reported in D7.4, the end-to-end execution of the AeroBridge translator, integration with the Modular Gateway, and validation of data exchange workflows for Flight Parameters, Fault Codes, Item Removals and Fleet Availability have been successful. Translation of datasets up to 10,000 records required between a few seconds and about 2 minutes including XML validation. The overall process is therefore lower than the maximum expected value (5 min.).
Data exchange model in accordance with S5000F	Compliance between data exchanged through S5000F message and defined business rules [Y/N - M0: no - M36: yes]	The compliance of the exchange data with a defined set business rules will be verified	AeroBridge was completed and tested. XML messages were generated according to S5000F, validated against XSD schemas, and reviewed by Leonardo experts on S5000F. All tested use cases produced valid XML outputs.
Data integrity	Compliance between data exchanged through S5000F message and raw data [Y/N - M0: no - M36: yes]	The data exchanged through S5000F messages will be tested with respect to raw data	As reported in D7.4 reports model consistency testing, semantic integrity validation, preservation of relationships through uid/uidRef mappings, validation against expected outputs, and comparison with original legacy data have been verified. The

			generated XML was verified to preserve the semantics of the original datasets.
Orchestration of the 3 service providers and 1 service consumer	Adoption of SOA oriented solution [Y/N - M0: no - M36: yes]	The architecture of the adopted solution must be based by design on the SOA paradigm and microservice architecture	Verification and validation activities demonstrated successful deployment of the Eclipse Arrowhead microservice architecture. Producer services (AeroBridge on Modular Gateways) and consumer services (AHMS application) were authenticated, registered, discovered and orchestrated automatically. Tests confirmed dynamic producer-consumer matching and secure service consumption.

Table 44: UC 2.7 KPIs

3.3.4.4 Beyond the baseline

3.3.4.4.1 Use case action plan

The use-case will address the development of an interoperable AHMS to support the aircraft life cycle, in particular the Customer Support and Engineering processes. There are three different scenarios:

- Real-time Data collection and transfer to AHMS via satellite. Note: the whole scenario will be simulated only in terms of data collection and data transfer.
- Collected data will be used for the needs of customer support, engineering, and maintenance.
- Data models will be used to understand the impact on operations and logistics.
- The overall use-case has schematized in figure 1 that mainly consists of three macro-areas:
- Cohesive suite for physical and digital systems.
- Component function interrelationships for advanced analysis.
- Diagnostic highways for users of the AHMS.

In the next figure, the systems, the capabilities and the fPVN are interconnected, and the data exchange is interoperable. Some components had been partly developed in CPS4EU project. Three types of data generation/utilization will be considered:

- Real-time flight data, produced by the aircraft (simulated only).
- Post flight data, acquired when the aircraft is on the ground after landing (stored during the flight).
- Simulated flight scenarios, based on aircraft mission planning.

The results of the use-case will be used by industry to improve the design and logistic support of aircraft, and by the customer to increase the flight-ready aircraft availability through interoperable maintenance and operations planning activities during and post flights.

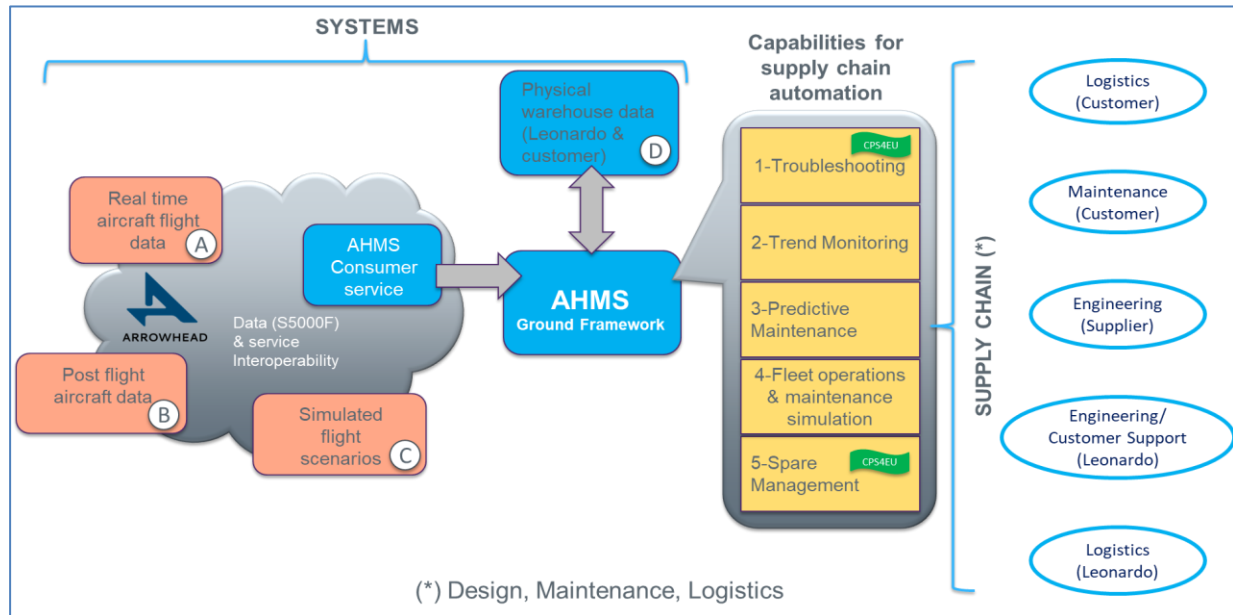


Figure 13 - The Arrowhead local cloud, the interconnected systems, and their capabilities.

In the overall context of the S-Series Specification (an international standard not exclusively used in the Aerospace domain) the use-case will leverage the S5000F (International specification for in-service data feedback). The scope of the S5000F is to handle information from in-service operation of a product, in this case an aircraft system. However, the specification can be used for data exchange at any moment of the life cycle of the aircraft. The S5000F is focusing on information exchange, therefore the system used for originating and receiving the data will be unaffected by the specification. S3000L (International procedure specification for Logistic Support Analysis) and S2000M (International Specification for material management) will also be used as a reference for the Integrated Logistic Support processes involved in the use-case scenario. The use-case will start at TRL4 and end at TRL6.

3.3.4.4.2 Contribution to project objectives

Project Objective	Status at M0	Expected improvement	Planned actions	Achieved results at year 3
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	Automatic translation not available.	The subset of S5000F relevant for this use case will be fully translated.	Integrate into the AHMS an automatic translator from WP4.	The AeroBridge translator was fully developed and integrated into the AHMS architecture. Translation of selected aerospace datasets into S5000F was successfully completed and validated for multiple use cases (Fault Codes, Flight Parameters, Item Removals and Fleet Availability). Automated generation and validation of S5000F-compliant XML files significantly reduced manual translation activities and demonstrated the feasibility of autonomous translation services.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	Solution based simply on a modular system of systems	AHMS will be based on the SOA paradigm and microservice	Adoption of Eclipse Arrowhead Framework	The AHMS architecture was successfully implemented using Eclipse Arrowhead. Modular Gateways, AeroBridge services and AHMS applications were integrated through Arrowhead Service/System Registry, Orchestration and Authorization services.

		architecture		Verification and validation activities demonstrated dynamic service registration, discovery, orchestration and secure service consumption between producers and consumers.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	No standard adopted	Data shall be exchanged based on S5000F and S-SERIES data model. We don't plan to update the standard.	Adoption of S5000F	S5000F was successfully adopted as the common exchange standard. During the development of AeroBridge, several limitations and improvement opportunities were identified, particularly in the Health Monitoring domain. Leonardo contributed feedback and proposals through its participation in the S5000F Steering Committee, supporting future evolution of the standard.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	Low level of automation, mainly based on a human operator.	The Orchestration module of AHMS allows the automation of repeated data processing operations, encapsulated in workflows, that transform source data, move data between multiple sources and sinks, load the processed data into an analytical data store, or push the results straight to a report or dashboard.	Development of an orchestration module for the automation of maintenance process operations.	The project established a complete workflow for collecting, translating, validating and exploiting aircraft operational data through digital services. The combination of AeroBridge, Arrowhead orchestration and the AHMS digital services demonstrated a practical methodology for introducing automated data translation and processing into aerospace maintenance and operational workflows.
Obj 5 - Establish a sustainable governance of open-source architecture and implementation platform	N.A.			

Table 45: UC 2.7 - Contribution to project objectives

3.3.4.4.3 Micro-service paradigm

3.3.4.4.3.1 Approach

In this use case, the approach to adopt a Service Oriented Architecture is based on the Eclipse Arrowhead framework.

3.3.4.4.3.2 Impact

The impact generated from the adoption of a Service Oriented Architecture includes:

- Modularity, allowing to easily customize the solution for the customer
- Seamless update/substitution of services without impacting on the architecture
- Possibility to customize the services portfolio depending on customer needs
- Improved scalability of the entire solution
- Security improvement based on the adoption of the Arrowhead local cloud

3.3.4.4.3.3 Long-term effect

The effects of the adoption of a Service Oriented Architecture includes:

- Evolvability of the solution, which will be open
- Increased interoperability among the stakeholders involved in the value chain
- Near real-time information sharing between the stakeholders involved in the value chain, with an impact on costs, efficiency, quality of offered service, and fleet availability.

3.3.4.4.4 Major digital languages

3.3.4.4.4.1 Approach

S5000F provides a data model that allows the exchange of in-service feedback. It can be tailored to the specific use case and progressively expanded.

The approach will be the following:

- The use cases (within AHMS Use Case) that will require exchange of in-service data, will be identified
- Business rules about the data exchange will be defined for the selected use cases
- A data mapping between the raw data and the S5000F will be performed
- An XML that identifies the exchange message will be tailored starting from the specification
- An export capability that generates the XML data to be transmitted will be developed
- An import capability that allows to import the XML data received will be developed.

If necessary, additional elements taken from the S-SERIES model (S3000L, S2000M...) will be taken into account.

3.3.4.4.4.2 Impact

The selected use cases within AHMS Use Case selected for the mapping versus S5000F are actual industrial examples that are used to improve Customers support services, product, and supply chain design.

Using this standard enables aircraft post-delivery analysis and in-service optimization that have impacts both on aircraft availability and life cycle costs.

Since the standard is tool agnostic, it can be integrated within the AHMS architecture without designing new or modifying legacy databases to export and import information. A translator should be necessary.

3.3.4.4.4.3 Long-term effect

Compliance with the standard allows the adoption of a common model for exchanging information between Customers, Industry and Suppliers. That information, validated through business rules, can be used to analyze performance, and calculate KPI.

Moreover, whenever a new product (a new aircraft equipment) or a new Customer enters the network, the common model can be used in order to integrate their information with a minimum effort within the AHMS architecture.

3.3.4.4.5 Autonomous translation

3.3.4.4.5.1 Approach

The autonomous translators will allow to automate the following step of the aircraft maintenance process and related logistics:

- A data mapping between the raw data and the S5000F will be performed
- An XML that identifies the exchange message will be tailored starting from the specification
- An export capability that generates the XML data to be transmitted will be developed
- An import capability that allows to import the XML data received will be developed

Currently this process requires a human operator and will be fully automated with the proposed solution.

3.3.4.4.5.2 Impact

A solution based on autonomous translators will significantly boost the interoperability of the AHMS and will bring the company to the state of art in terms of data exchange standards, enabling future commercial opportunities.

3.3.4.4.5.3 Long-term effect

With the proposed solution we can integrate potentially any kind of aircraft used by different customers, significantly extending the potential share of the market accessible to the company.

3.3.4.4.6 KPIs, Evaluation, and validation process

In accordance with D7.1 section 2.2.7, a high-level plan for validation and verification for scenarios of input data feeding has been defined.

In general, several elements have to be verified:

- Data exchange capability
- Data exchange model in accordance with S5000F
- Data integrity
- Orchestration of the 3 service providers and 1 service consumer

Actual data will be provided for testing. The aim is to determine if within the AHMS platform (Consumer service + GF) the data are correctly exchanged and stored for the successive analyses.

In detail, for data exchange capability:

- Data continuously collected during flight: exchange between flying aircraft and ground station will be simulated only, therefore will be tested the capability of a ground system to handle telemetries and exchange them with the AHMS
- Data collected after aircraft landing: post flight data will be provided as input. Will be tested the capability of the Consumer Service to exchange them with AHMS GF

- Data collected from simulated flights: simulation results will be provided as input. Will be tested the capability of the Consumer Service to exchange them with AHMS GF.

For data exchange model in accordance with S5000F, the compliance of the exchange data with a defined set business rules will be verified.

For data integrity, the data exchanged through S5000F messages will be tested with respect to raw data.

3.3.4.5 Final assessment

3.3.4.5.1 KPI Assessment

UC2.7 successfully achieved all the KPIs defined at project start, demonstrating the feasibility and effectiveness of an interoperable Aircraft Health Management System (AHMS) based on service-oriented principles, autonomous data translation, and standardized information exchange. The achieved results show a significant advancement with respect to the baseline situation, where data exchange was largely manual, non-standardized, and dependent on proprietary data models.

The Data Exchange Capability KPI was fully achieved. The initial objective was to reduce the time required to retrieve, process, and share aircraft operational data between AHMS subsystems from a baseline ranging between one and twenty hours to less than five minutes. Validation activities demonstrated successful end-to-end execution of the AeroBridge translator, integration with the Modular Gateway, and processing of multiple operational datasets, including Flight Parameters, Fault Codes, Item Removals, and Fleet Availability information. Translation and validation of datasets containing up to 10,000 records were completed in a timeframe ranging from a few seconds to approximately two minutes, significantly outperforming the target value.

The Data Exchange Model in accordance with S5000F KPI was also successfully achieved. At project start no standard model was adopted for data exchange between aircraft systems, maintenance stakeholders, and supply chain actors. During the project, AeroBridge was developed and validated to automatically generate S5000F-compliant XML messages. The generated messages were successfully validated against official XSD schemas and reviewed by Leonardo experts and members of the S5000F Steering Committee, confirming compliance with the specification and applicable business rules.

The Data Integrity KPI was fully satisfied through extensive verification activities. Model consistency testing, semantic validation procedures, preservation of relationships through uid/uidRef mappings, validation against expected outputs, and comparison with original source datasets demonstrated that the generated S5000F messages preserve the semantics and information content of the original proprietary data. This represents a substantial improvement over the baseline situation, where no formal validation or standardization mechanisms were available.

Finally, the Orchestration of Service Providers and Consumers KPI confirmed the successful adoption of a Service-Oriented Architecture based on Eclipse Arrowhead. Verification and validation activities demonstrated automatic authentication, registration, discovery, and

orchestration of AeroBridge services deployed on Modular Gateways and consumed by the AHMS applications. The successful implementation of dynamic service orchestration validated the use of Arrowhead as the interoperability framework supporting the AHMS ecosystem.

3.3.4.5.2 Contribution to Project Objectives

UC2.7 provided a strong contribution to the main objectives of the Arrowhead fPVN project by demonstrating the practical application of autonomous translation services, service-oriented architectures, and standardized data models in the aerospace maintenance domain.

For Project Objective 1, the use case successfully integrated the AeroBridge translator into the AHMS architecture, enabling the automated translation of multiple aerospace datasets into S5000F-compliant messages. Before the project, automatic translation capabilities were not available, and data exchange required significant manual intervention. The successful validation of the translator for Fault Codes, Flight Parameters, Item Removals, and Fleet Availability demonstrated the feasibility of autonomous machine-based translation services capable of reducing human effort while improving interoperability.

Regarding Project Objective 2, UC2.7 successfully transformed the original modular system-of-systems architecture into a fully service-oriented environment based on Eclipse Arrowhead. Modular Gateways, AeroBridge services, and AHMS applications were integrated through Arrowhead Service Registry, Authorization, and Orchestration services, enabling dynamic service deployment, discovery, and utilization. This validated the adoption of microservice-based architectures in realistic aerospace production value networks.

Concerning Project Objective 3, the use case adopted S5000F as the common information exchange standard for aircraft health management and maintenance-related information. The implementation activities not only demonstrated the applicability of S5000F in operational scenarios but also generated valuable feedback regarding the applicability of the specification to health monitoring use cases. Through Leonardo's participation in the S5000F Steering Committee, the project contributed to the future evolution of the standard and promoted broader interoperability within the aerospace ecosystem.

For Project Objective 4, the project established a practical methodology for introducing seamless and autonomous translation into aerospace maintenance workflows. The integration of AeroBridge, Arrowhead orchestration services, and AHMS digital services enabled the creation of automated workflows capable of collecting, translating, validating, processing, and exploiting aircraft operational data. This demonstrated a concrete digital transformation pathway for maintenance and operational support processes.

3.3.4.5.3 Improvement in the Engineering Process

UC2.7 produced a substantial improvement in the engineering and operational processes supporting aircraft health management, predictive maintenance, and fleet operation planning. The baseline situation was characterized by heterogeneous proprietary data formats, limited interoperability between stakeholders, manual information exchanges, and ad hoc integration procedures whenever new aircraft platforms, customers, or maintenance systems had to be incorporated into the ecosystem.

The introduction of AeroBridge and the adoption of S5000F fundamentally transformed the data management process. Information originating from aircraft systems, maintenance databases, logistics systems, and simulation environments can now be mapped into a common standardized model and exchanged automatically through validated XML messages. This significantly reduces integration effort, improves consistency and traceability, and facilitates interoperability among customers, aircraft manufacturers, suppliers, and maintenance organizations.

The adoption of Eclipse Arrowhead introduced a further improvement by replacing static integrations with dynamically orchestrated service interactions. New services can be deployed, discovered, and consumed without modifying the overall architecture, enabling greater flexibility, scalability, and maintainability. This represents a major advancement compared with the baseline architecture, where interoperability was largely achieved through dedicated integrations and manual coordination between stakeholders.

From an engineering perspective, the project also established a complete Arrowhead Engineering Process covering requirements definition, service-oriented architecture design, development of AeroBridge and Modular Gateway technologies, deployment within the Arrowhead local cloud, operation of trend monitoring and predictive maintenance services, and continuous evolution of data models and analytics capabilities. This process links design-time definition of data models and interoperability requirements with runtime operation of aircraft health management services, creating a structured and scalable engineering framework for future deployments.

Finally, the implementation of trend monitoring, predictive maintenance, digital twin technologies, and fleet operations and maintenance simulation services demonstrated the transition from reactive maintenance practices toward a data-driven and predictive operational paradigm. The resulting engineering process enables faster information exchange, more informed maintenance decisions, improved fleet availability, and increased readiness for future integration of additional aircraft platforms, stakeholders, and digital services.

3.3.5 WP9 - Use-case #1.9 – Pump Station Engineering

3.3.5.1 Baseline summary

The baseline for the Pump Station Engineering use case reveals a traditional and fragmented approach. Engineering work and project information have historically been decided on a project-to-project basis, influenced by the current selection of suppliers and the customer representatives ordering the project.



Figure 14: the initial engineering process

The initial overview of the engineering process (Figure 14) has been detailed further from the project proposal and the following eight life-cycle phases have been proposed for further work with this use-case. These phases involve multiple steps and necessitate collaboration across various companies and departments. Key roles and responsibilities have been mapped out, with a focus on Owner-operators (e.g., Roslagsvatten) and EPCs (e.g., Granitor).

Currently, most engineering tasks are executed in isolated systems, with limited data exchange. The exchange that does occur is largely confined to text-based documents (e.g., PDFs) and tabular data formats (e.g., Excel, XML, or CSV), necessitating manual interventions for integration.

As a first outline for the systems that are operating (or planned for the near future) at Roslagsvatten, the following illustration (Figure 15) has been put together.

For each key role, that has been identified, persons from the participating partners will also be contacted and interviewed about their current work process in the different steps of the life cycle of a pump station, to provide a baseline from real life operations. As part of the interviews, one of the objectives is to chart how each role utilizes different systems and engineering tools. Initially it is clear that some roles use some systems in a more isolated manner, e.g. the Operators will use the OT-environment (Operational Technology) a lot, and the Maintenance Staff will use the Maintenance system, while others such as Environmental Scientists need to combine information from the Geographic Information System (GIS) with data from the Historian database and external information from municipal development plans.

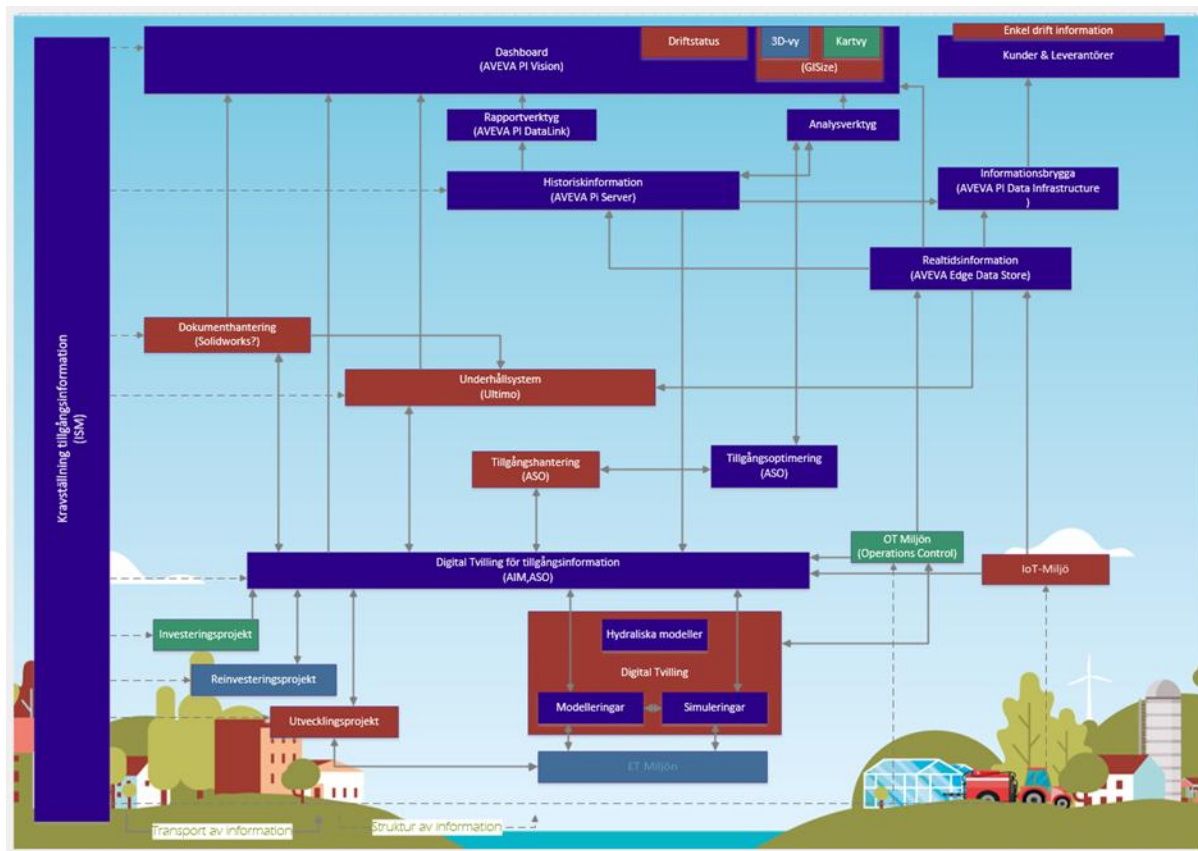


Figure 15: outline for the systems that are operating (or planned for the near future) at Roslagsvatten.

3.3.5.1.1 Architecture of the use case as a sequential list of functional blocks

The proposed architecture emphasizes seamless and standardized data exchange between contractors and facility owners. Leveraging Project Information Models (PIM) and Asset Information Models (AIM), the architecture aims to enable technology-agnostic communication and integration across diverse systems. Functional diagrams and block models will be used to illustrate this architecture, highlighting tool roles and functions.

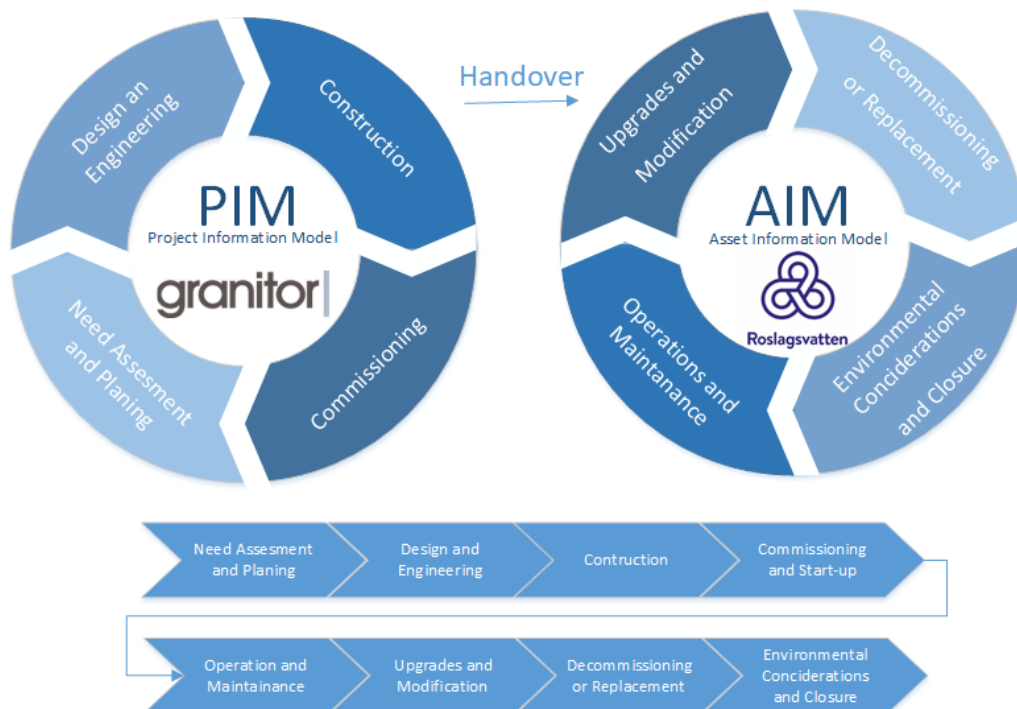


Figure 16: use case architecture.

3.3.5.1.2 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	A significant part of this UC is focused on identifying standards, data models and how they are used in different parts of the production value network of a process industry in general, and a pump station in particular. Through this work, the idea is to provide concrete examples of combinations of standards and data models that could use translation or harmonization in order to reduce the need for human support in large parts of the process industry.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	Once interactions have been properly analysed and harmonious communication, through use of common standards/data models or translations, has been tested, there will be candidates for new services and to propose in cooperation with WP2. In identifying most suitable candidates for service proposals, the UC is intending to coordinate with UC 3.9 that is operating in a similar scenario and shares many challenges with this UC.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	The UC is actively working with their data models and needs for the near and far future. It focuses on a digital twin and real time transfer of information between systems and solutions from different vendors. RSV is working with IEC 81346 / IEC 61175 and work is being done on building a common data model for an asset that lives in several systems and can be moved from system to system when needed. RSV intends for the data models to be open source and public.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	RSV had a digital transformation program dedicated to digital transformation. Experiences can be shared from the insights that have been discovered during the project today and in the future.
Obj 5 - Established a sustainable governance of open-source architecture	Sustainable governance of open-source resources is of particular interest to non-profit actors such as Roslagsvatten, where shared utilization of common digital resources between municipal organizations is central to the vision for more efficient utilities operation.

and implementation platform	
-----------------------------	--

Table 46: UC 1.9 - Contribution to project objectives.

3.3.5.1.3 Micro-service paradigm

3.3.5.1.3.1 State of the art

At M0 no micro-service architecture is implemented in any part of the systems involved in the use case. There some systems providing service interfaces via API but compatibility has not been investigated and implementation of a modern micro-service architecture is preferred.

3.3.5.1.3.2 Enablers

RSV is actively seeking to implement a micro-service architecture based on established standards and proven technologies. The focus is on interoperability and scalability. To achieve this efficient and successfully, the choice of standards to use and exact solutions for micro-service architecture is under evaluation

3.3.5.1.4 Major digital languages

3.3.5.1.4.1 State of the art

At M0 no major digital languages are in use. Some standards may be in use within individual domains or systems at RSV, but there is no framework or master system that can correlate data across systems.

3.3.5.1.4.2 Enablers

The project will utilize standards such as IEC 81346 and IEC 61175. Efforts will focus on defining a standard from OT layer 0 to business level 4.

3.3.5.1.5 Autonomous translation

3.3.5.1.5.1 State of the art

RSV is planning to work mainly with a few selected standards, which are hopefully compatible without translation. RSV and Granitor see the benefit of translation between RSV picked standards and standards in use at other companies within the value network, and there can be scenarios where automatic translation between them would be beneficial. Currently, there are no plans to include this in the use case.

3.3.5.1.5.2 Enablers

No clear enablers have been identified at this point.

3.3.5.2 Use Case Engineering Process

3.3.5.2.1 State of the art

The process is explicitly lifecycle-oriented: need assessment and planning, design and engineering, construction, commissioning and start-up, operation and maintenance, upgrades and modifications, decommissioning or replacement, and environmental considerations and closure. The baseline process is fragmented and tool-specific, but it maps clearly to Arrowhead-EP phases.

3.3.5.2.2 Enablers

IEC 81346 and RAMI offer a structured engineering-process framework. Arrowhead-EP adds explicit service-based data flow, feedback between engineering and operation, and support for multi-stakeholder PIM/AIM handover. The process includes detailed identification of key tools and a pilot area named Rydbo.

3.3.5.2.3 Arrowhead-EP of the Use Case

EPP1 need assessment and planning -> EPP2 pump-station design and asset modelling -> EPP3 procurement, engineering and construction data -> EPP4 commissioning and start-up -> EPP5 operation and maintenance with AIM/EAM/GIS/SCADA -> EPP6 upgrades, replacement, closure -> EPP7 investment and standards evolution; EPP8 user and supplier training.

Scalability level: Medium. The model can scale from one pilot pump station to municipal utility portfolios, but organizational support, procurement delays and tool constraints limit near-term full lifecycle execution.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	Need assessment, stakeholder interviews, PIM/AIM scope	Defines requirements and roles.
EPP2	IEC/ISO 81346, DEXPI, engineering drawings, PIM	Defines plant objects and design information.
EPP3	Granitor engineering tools, supplier data, IEC 61499 software interfaces	Executes procurement and engineering information flow.
EPP4	Commissioning/start-up process and standard data exchange	Validates engineered assets before operation.
EPP5	AIM, EAM, GIS, SCADA platform	Operates and maintains the pump station.
EPP6	Upgrade/modification and replacement workflows	Manages asset changes and closure.
EPP7	Standards roadmap, service candidates, supplier integration	Evolves architecture and practices.
EPP8	Training material, user interfaces, standardized work instructions	Supports adoption by stakeholders.

Table 47: Arrowhead EP mapping.

3.3.5.2.4 Mapping of tools in the Arrowhead-EP of the Use Case

The mapping table above represents the current UC-EP decomposition. Tools and methodologies are grouped by Arrowhead-EP phase, with EPP8 treated as a cross-cutting training and adoption phase.

3.3.5.2.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - The change from design time to run time engineering	The use case integrates real-time data exchange between engineering tools and operational environments, enabling dynamic adjustments to engineering designs based on live system data. PIM-to-AIM handover and live operational data enable operational feedback into engineering.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The engineering process includes a framework for multi-stakeholder collaboration, integrated workflows to harmonize efforts across departments and organizations. Owner-operator, EPC, suppliers and technical partners coordinate around shared standards and services.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	By adopting a standardized data model and microservice architecture, the use case is equipped to manage an expanded number of input and output points effectively. A standardized asset model and microservice architecture can manage many plant objects and data points.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Training materials and user interfaces are being designed to integrate with the engineering process, supporting continuous learning and adaptation for all stakeholders involved.

Table 48: Arrowhead EP objectives.

3.3.5.2.6 Adopted toolchain

PIM/AIM handover workflow, IEC/ISO 81346 asset structuring, DEXPI translation evaluation, IEC 61499 software interfaces, IEC 61175 signal/reference designation, RSV AIM/EAM/GIS/SCADA, Granitor engineering tools, candidate Arrowhead Configuration and Plant Description services.

3.3.5.2.7 Toolchain automation

Low to medium. Standards-based structuring and pilot data exchange are emerging, but much engineering data exchange remains manual or file-based. DEXPI translation is under evaluation rather than fully automated.

3.3.5.2.8 Toolchain limitations

No microservice architecture at M0.

Fragmented engineering tools and manual document exchange.

Less organizational support and public procurement delays slowed progress.

Scope reduced from full pump-station construction to digital process and information exchange.

3.3.5.3 Use case and engineering costs

Several of the main objectives for the use case are related to engineering effort and efficiency. The overall engineering cost for an actual project resembling the use case is hard to estimate and depends a lot on the size of pump station and which steps are included in the project. Looking at the electrical and industrial automation design disciplines, the engineering effort is usually in the range of 1-2 man months. Depending on level of standardization and similarity between pump stations, subsequent deliveries to the same facility owner would often come at an effort of about half that of the initial delivery.

An analysis of engineering costs at M0 consider:

- Training and documentation preparation.
- Toolchain integration and automation.
- Lifecycle costs, including design, deployment, and operation. Normalized metrics such as man-months and resource allocations will provide a comprehensive view.

A qualitative relative cost profile is used, based on the observed engineering activities, toolchain integration needs and validation burden.

Arrowhead-EP phase	Relative engineering cost/effort	Comment
EPP1 - Requirements	Medium	10 normalized man-month
EPP2 - Functional design	Hgh	18 normalized man-month
EPP3 - Procurement and Engineering	High	20 normalized man-month
EPP4 - Deployment and Commissioning	Medium	12 normalized man-month
EPP5 - Operation and Management	Medium	15 normalized man-month
EPP6 - Maintenance / Decommissioning / Recycling	Medium	10 normalized man-month
EPP7 - Evolution	Low	2 normalized man-month
EPP8 - Training and Education	Medium	8 normalized man-month

Table 49: engineering costs.

3.3.5.4 Standards

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	ERP/MES, Function Blocks	REST APIs, ERP/MES, Function Blocks
Information and Representation	-	IEC 81346, IEC 61175
Semantic and Language	JSON	JSON
Communication	Fieldbus, WIFI, Internet Protocols: HTTP, FTP, TCP/IP	Fieldbus, OPC-UA, WIFI, Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC
Cybersecurity and Safety	ISO 27001, Firewall	IEC 62443, ISO 27001, NIST SP 800-82, Authentication, Firewall,

Reference Model	-	RAMI
Domain-Specific	-	IEC 61499
Framework Development and Specific Applications for development	-	-

Table 50: adopted standards.

3.3.5.5 Key Performances Indicators

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation			Achieved at 3 rd year
Name	KPI Description	Means of assessment	Year 1	Year 2	Year 3	
Digital maturity	Level of digital maturity among RSV Stakeholders	Involvement in discussions and decision from system owners with regards to creating a coherent digital infrastructure at RSV	10%	25%	60%	Digital maturity has increased immensely but also exposed significant challenges in management and coordination between different efforts.
Standardized data infrastructure	Number of systems related to the use-case that operate with standardized data	Counting the systems where IEC 81346 is used to identify assets	0	1	4	Infrastructure is now at a very good level as a foundation to continue integrating more systems, but digital maturity and management challenges becomes clearer now that the foundation is in place.
Standardized data for pump station	Aspects of pump station data that is structured according to established standards	Estimating the amount of information about a pump station that has been standardized according to compatible standards	10%	30%	80%	Standardization of pump stations have been a success, and are currently used for commercial procurement of new pump stations. The results of these procurements have not been evaluated yet.
Micro-service architecture	Level of complete coverage provided by the implemented architecture	Estimating the level of useful functionality that is provided at the current level of implementation of the selected architecture	0%	0%	10%	The very foundation of the micro-service architecture is now in place both at RSV and GRA, through the expertise of SIN, but only a very minor part of the available functionality has been implemented so far. Further implementation is being planned, but the timeline is not fixed yet.

Table 51: KPIs.

3.3.5.6 Beyond the baseline

3.3.5.6.1 Contribution to project objectives.

Project Objective	Status at M0	Expected improvement	Planned actions	Status at M12	Status at M24	Status at M36	Achieved at year 3
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	No translations	Translations tested experimentally	Identify standards that require autonomous translations for this use-case	-	Translation of P&ID to DEXPI as shown in UC3.9 is of interest to UC1.9	Actual translations have not been included in the use-case.	Investigated candidates for translation, and found the case tested in UC3.9 (P&ID to DEXPI) to be the best candidate for the use case. Time and effort did not allow a separate implementation to evaluate it in UC1.9.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	No microservices	Microservice architecture implemented for PVN interoperability	Identify important services, implement existing parts and contribute to further development if needed for the use case	Vision established and key systems identified	Pilot integration and information structure and governance models established	Integration via microservices between four systems at RSV and one system at GRA established.	Microservices integration at RSV between four live production systems, implemented with expertise from SIN. Demonstration interoperability with GRA systems set up.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	No major digital data models	Evaluation, selection and implementation of data models, to provide feedback for update proposals.	Identify and evaluate suitable standards and data models. Implement selected ones and provide feedback for updates.	First data models and standards identified, evaluation ongoing.	Main structure for data model selected and implementation ongoing.	Main data model implemented. Some feedback provided in terms of difficulty to ensure coherent interpretation of standards across domains and stakeholders within a PVN.	IEC 81346 for overall information structure applied at RSV, facilitating interoperability with GRA demonstration systems. Feedback gathering in the broader PVN (outside of project) started, discussing the interpretation issues noticed when applying standards within the project.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	Ambitious and detailed vision for digitalisation.	Combining the vision with expertise from other partners to provide strong cases of best practice	Active part in the work on digital transformation management	Difficulty in applying the digital transformation management within the use-case	More focus on establishing basic standards and data models appear to be necessary before further steps in digital transformation	A few system integrations between live systems provide a starting point for larger digital transformation. Management of culture related to digitalization proven difficult.	Digital transformation has been ongoing before and will continue after the project. Some significant challenges encountered during demonstration implementation has been included as feedback for further development of the methodology.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	-	Unclear how this will fit into the use case	-	-	Some experiments with making standardised solutions for similar usage available as open-source.	Open-source experiments halted after difficulties with strategy and final goals.	Current implementations are not planned for open-source publication, partly due to lack of internal governance expertise, but there are candidates that will be

							considered for the future if the expertise can be found.
--	--	--	--	--	--	--	--

Table 52: contribution to project objectives.

3.3.5.6.2 Use case and engineering costs

The use case very clearly illustrates how the technologies promoted within the project, and in particular by WP2, WP3 and WP4 can significantly reduce the engineering costs. Looking at the electrical and industrial automation design disciplines, the engineering effort for a pump station is estimated to be lowered significantly. Looking at how much of the engineering effort is spent on interpreting documentation and understanding how data from one system correlates to data from a different system, it is likely that the reduction in engineering effort is in the range of 25-40%, meaning that an estimated 2 person month project could be reduced to 1,2 – 1,5 person months.

Outside of the direct savings shown in the demonstration, there is additional potential for automation as increased interoperability creates further incentives for standardization. This would likely lead to a decrease in engineering effort to below half of the original effort required.

3.3.5.6.3 Micro-service paradigm

3.3.5.6.3.1 Approach

The use of services will be essential to ensure the interoperability within the use case, especially between different partners.

To ensure that the interoperability is possible to implement also in a larger scope, with many suppliers, rather than a fixed connection between a small group of partners, as exemplified in the use case, it is essential that the interfaces are thought through and well designed for efficient implementation, but also that the interfaces can be managed.

Therefore, the main services that are envisioned are the core services, especially the service registry, to show how suppliers can be registered at once, and in addition the IDD registry developed by Sinetiq.

3.3.5.6.3.2 Impact

Full adaptation of the technologies proposed in the use case could significantly reduce the engineering effort for projects and other types of changes. Provided that the information is also well structured according to interoperable standards, i.e. this is also dependent on Major digital languages, the engineers would be able to focus on the factors where there are design choices to be made, rather than straightforward movement of data between systems.

3.3.5.6.3.3 Long-term effect

Potentially, the improvements in engineering efficiency could result in more efficient implementation of important technologies. Enabling solutions that improve all aspects of society at a lower cost, meaning improvements not only to productivity of companies but also enabling environmental, infrastructure and health improvements that would previously have been too expensive. This means that although each engineer will be more efficient, there is still likely to be a lot of work to do for skilled engineers.

3.3.5.6.4 Major digital languages

3.3.5.6.4.1 Approach

RSV has identified and partly demonstrated the value of applying IEC/ISO 81346 across several engineering disciplines, project phases, and organizational units. The work has shown that a common reference designation and information structure can support more consistent information management throughout the lifecycle of municipal utility assets.

3.3.5.6.4.2 Impact

The application of standardized data models are critical for the success of the use-case.

3.3.5.6.5 Autonomous translation

3.3.5.6.5.1 Approach

Given the immature usage of standards within Roslagsvatten especially, it has been difficult to see exactly how translation will fit into the use case and even more so to specify exactly which standards the translation would work between.

However, if the digital maturity at Roslagsvatten can be improved, a parallel implementation of the P&ID to DEXPI translation performed by Semantum in UC3.9 could potentially fit well to UC1.9 as well.

3.3.5.6.5.2 Impact

It was estimated that the identified translations are not required for the successful implementation of the use-case, however they will most likely be important for extending the demonstration outside of the project.

3.3.5.6.5.3 Long-term effect

Translations will likely be very important to extend the usefulness of the demonstrated solutions outside of the actual PVN involved in the project.

3.3.5.6.5.4 Other aspects

Given that RSV is a Swedish municipal utility company, much of the information within RSV is available through the public disclosure regulations that apply to all Swedish state and municipal organisations. However, given that some of the services provided by RSV are considered critical infrastructure, some information is also classified.

Given these considerations, there are specific concerns to take into account when providing data from RSV for further research and development activities.

3.3.5.7 Final assessment

3.3.5.7.1 KPI Assessment

UC1.9 successfully demonstrated a significant improvement in the digitalization and interoperability of engineering processes for municipal wastewater pump stations. The baseline situation was characterized by fragmented engineering workflows, isolated systems, manual information exchange through documents and spreadsheets, and the absence of common standards for asset identification and lifecycle information management.

The Digital Maturity KPI showed substantial progress during the project. Stakeholder involvement in discussions, decision-making, and implementation of a coherent digital infrastructure increased considerably, resulting in a significantly improved awareness of digitalization opportunities and challenges. While the project exposed organizational and governance difficulties associated with large-scale digital transformation, it successfully established a common vision and practical implementation roadmap for future developments.

The Standardized Data Infrastructure KPI was achieved through the implementation of IEC 81346-based asset identification across multiple systems. Four systems were successfully aligned using standardized asset structures, providing a solid foundation for future interoperability and lifecycle information management. The resulting infrastructure established a common information structure across asset management, maintenance, documentation, GIS, and operational systems.

The Standardized Data for Pump Stations KPI achieved its target with approximately 80% standardization of pump station information structures and engineering concepts. The project delivered standardized pump station designs, standardized automation cabinet concepts, standardized P&ID structures, and harmonized information deliveries from suppliers based on IEC 81346 and IEC 61175. These results are already being applied in procurement activities and provide a strong basis for future engineering automation and information reuse.

The Micro-service Architecture KPI demonstrated the successful establishment of the foundations of a service-oriented architecture. Although the implementation coverage remains limited compared to the long-term vision, production-level integrations were successfully deployed between asset management, maintenance management, document management, and GIS systems. Furthermore, interoperability demonstrations between Roslagsvatten and Granitor validated the applicability of Arrowhead principles and microservice-based integration for municipal utility environments.

3.3.5.7.2 Contribution to Project Objectives

UC1.9 contributed to the Arrowhead fPVN project primarily through the establishment of standardized information structures, service-oriented integration mechanisms, and lifecycle-oriented engineering practices within a municipal utility context.

For Project Objective 1, the contribution was mainly preparatory rather than implementation-oriented. The use case investigated candidate translation scenarios and evaluated the potential benefits of autonomous translation technologies for process industry engineering workflows. The translation of P&ID information into DEXPI was identified as a particularly relevant future application. Although no dedicated translation demonstrator was implemented within UC1.9, the use case generated valuable requirements and practical feedback for future translation solutions.

The strongest contribution was made to Project Objective 2, where the use case successfully established microservice-based interoperability between multiple production systems at Roslagsvatten and demonstrated interoperability with external engineering partners. Integration between asset management, maintenance management, GIS, and engineering environments

validated the applicability of service-oriented architectures and Arrowhead principles within municipal infrastructure management.

Regarding Project Objective 3, UC1.9 provided substantial contributions through the practical implementation of IEC 81346 as a common information structure across multiple domains and systems. The project demonstrated both the benefits and challenges associated with standard adoption and generated valuable feedback concerning interpretation consistency, governance, and cross-organizational harmonization. These experiences contribute directly to future standardization activities and digital data model evolution.

For Project Objective 4, the use case generated significant practical experience regarding digital transformation management in organizations with limited digital maturity. The project highlighted the importance of governance structures, organizational alignment, information ownership, and long-term information management strategies. Several lessons learned and implementation challenges have been documented and incorporated into the project's digital transformation methodology.

The contribution to Project Objective 5 was more limited. While the use case explored opportunities for open-source governance and reusable solutions, organizational and governance constraints prevented concrete implementation during the project. Nevertheless, the work identified future opportunities for shared municipal digital infrastructure and common information models that could support open and reusable solutions in the future.

3.3.5.7.3 Improvement in the Engineering Process

UC1.9 delivered a substantial improvement in the engineering process for municipal wastewater infrastructure by replacing fragmented and supplier-specific information management practices with standardized, lifecycle-oriented, and service-enabled engineering workflows. At project start, engineering information was exchanged primarily through documents, spreadsheets, and manual processes, with little consistency between systems or stakeholders. Asset information was structured differently across departments, suppliers, and lifecycle phases, creating significant inefficiencies and risks for long-term asset management.

The project introduced a structured engineering methodology based on IEC 81346, IEC 61175, RAMI principles, and Arrowhead-inspired service-oriented integration. Standardized identification and classification of assets now allow information to be consistently represented across engineering, maintenance, operational, GIS, and documentation systems. This significantly improves traceability, reduces ambiguity, and facilitates information reuse throughout the asset lifecycle.

A major improvement was achieved in the handover process between Project Information Management (PIM) and Asset Information Management (AIM). Through standardized information structures and service-based integration mechanisms, information generated during engineering activities can be transferred more efficiently into operational systems while maintaining consistency and traceability. The demonstration involving Roslagsvatten and Granitor showed how engineering modifications can be initiated, executed, reviewed, and integrated through structured workflows supported by interoperable services.

The implementation of pilot integrations between GIS systems, asset registers, maintenance management systems, and document management environments demonstrated the practical feasibility of service-oriented interoperability within real production environments. These integrations represent the first operational examples of a future architecture where engineering, operational, and maintenance information can be exchanged automatically through standardized interfaces rather than manual document transfers.

Perhaps the most important outcome from an engineering perspective is the establishment of a long-term roadmap for digital transformation. The project demonstrated that successful interoperability requires not only technical integration but also governance models, organizational alignment, standardized information definitions, and lifecycle-oriented information management practices. The resulting engineering process therefore provides a scalable framework that can support future expansion toward fully interoperable, digitally managed municipal infrastructure systems.

3.3.6 WP9 - Use-case #2.9 Digital Twins that enable higher performance by interoperability in pulp mills and carton board mills

3.3.6.1 Baseline summary

The following baseline for this use case has been identified:

- Incompatible data in different system making data integration hard and costly.
- Today the situation is vendor lock-in with tech suppliers. Very hard to mix and match between vendors and use best of breed technology. Very costly also over time.
- Operation is today performed with high dependency of a few individuals and much customization.
- The need for long term resilience and risk management is hard to meet.
- Pragmatic solutions are used with lack of traceability in software and data deliverables with liability risks as results.

3.3.6.1.1 General description of the baseline

What is presented is a general description of what it typically looks like at the mills within Stora Enso's Packaging Materials Division. Examples are taken from different mills.

Use Case UC2.9, described in this section, is closely related to Use Case UC3.9, which is described in a separate section of this document. Together, these use cases support digital information exchange across industrial value chains, including investment project data exchange, dependability and maintenance information sharing, and sustainability traceability throughout production and service lifecycles.

UC2.9 focuses on the development of a Knowledge Graph-based Digital Twin of the production system. The Digital Twin provides a semantically integrated representation of production assets, processes, operational data, and maintenance-related information, creating a unified and machine-readable view of the production environment.

Building on this foundation, UC3.9 extends the Digital Twin beyond the boundaries of the production facility by enabling digital interaction with external suppliers of maintenance services, spare parts, and equipment. Maintenance needs identified, predicted, or analyzed through the Digital Twin can be shared with relevant suppliers, supporting more efficient maintenance planning, improved service coordination, enhanced supply chain collaboration, and increased operational reliability.

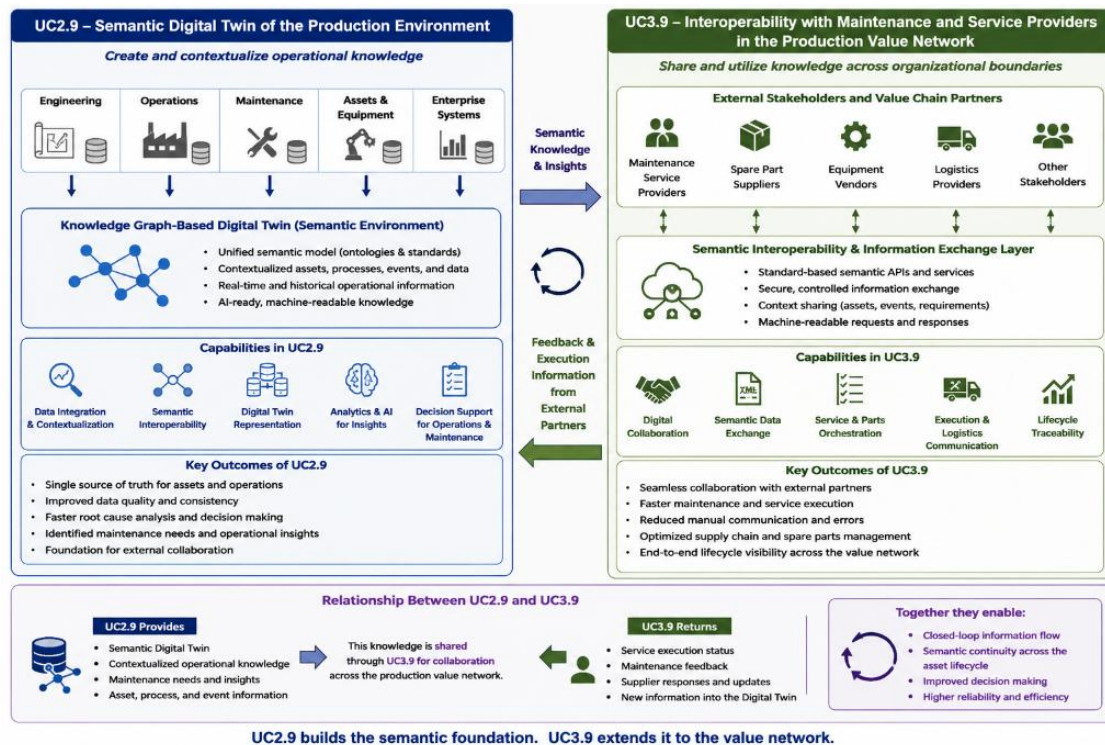
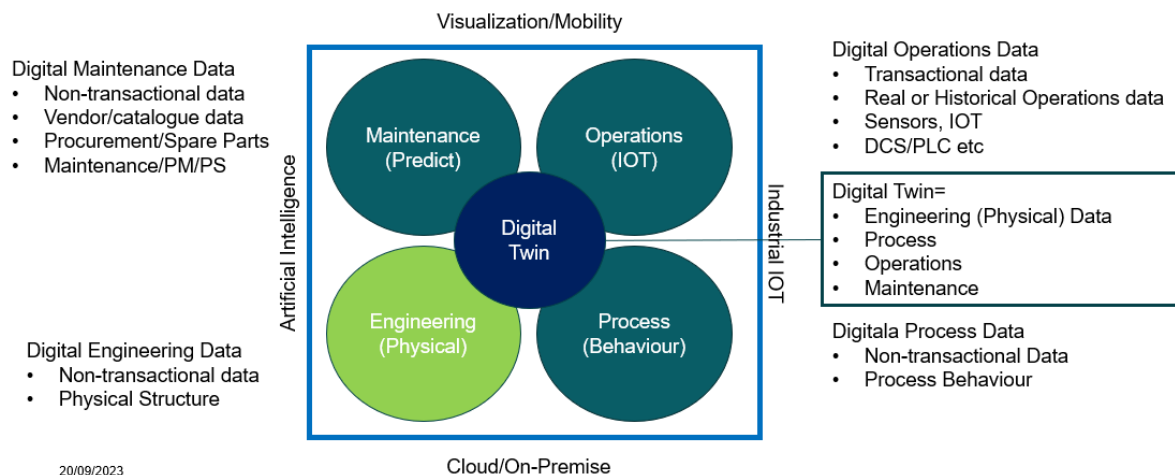


Figure 17: UC2.9 and UC3.9 architectures.

3.3.6.1.2 Initial architecture of the use case

This use case contains two separate use-case parts from task 9.2:

- Autonomous Mills & digitalization** use-case in digital value chains
 - Reliable and secure solution for a digital twin containing data from IT (such as Engineering, Maintenance, IoT and Conditioning monitoring data etc) and OT (such as DCS, PLC, MES, IIoT and Conditioning monitoring etc) in order to improve internal and external collaboration.
- Remote Operations and support**
 - Access to necessary information (IT and OT) in order to fulfil collaboration
 - Dynamic state Simulation of process based on digital twin P&ID



20/09/2023

Figure 18: UC2.9 conceptual diagram.

3.3.6.1.3 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	UC 2.9 does not use translations, but instead utilizes semantic interoperability based on W3C technology. UC2.9 establishes a Knowledge Graph-based Digital Twin that maintains semantic continuity between engineering, operations, and maintenance information. This enables engineering knowledge and standards to be reused throughout the operational lifecycle.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	UC2.9 enables semantic interoperability across engineering systems, operational technologies, enterprise systems, maintenance systems, and external stakeholders. The shared semantic environment provides a common understanding of assets, processes, and information.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	The ontology-based Knowledge Graph architecture is designed to manage large numbers of interconnected assets, sensors, observations, events, and relationships. Semantic abstraction and contextualization enable scalable handling of increasing information volumes.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	UC2.9 captures engineering, operational, and maintenance knowledge in a machine-readable semantic form. This supports knowledge preservation, training, AI-assisted information retrieval, decision support, and continuous organizational learning.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	Based on the results of the UC 2.9 implementation, a discussion has been initiated regarding whether the outcome can be jointly maintained and further developed within, for example, the Eclipse framework. UC2.9 reduces the need for traditional translations by establishing a shared semantic understanding based on ontologies and knowledge graphs. Existing industrial standards such as DEXPI, STEP Core, CFIHOS, and AFO are reused and semantically integrated within a common interoperability framework.

Table 53: contribution to project objectives.

3.3.6.2 Baseline analysis - building blocks & enablers

3.3.6.2.1 Micro-service paradigm

3.3.6.2.1.1 State of the art

Microservices are currently not the de facto integration model, although they are used in some applications. A standardized integration approach would accelerate implementation, improve security, and make it easier to develop future integrations, replacing traditional point-to-point connections.

3.3.6.2.1.2 Enablers

The Arrowhead Framework has been used as the interoperability layer connecting the time-series database containing sensor measurements, the fault analysis and evaluation mechanisms, and the SAP S/4HANA Mockup. Sensor data is continuously collected and analyzed to identify potential faults and maintenance needs.

The SAP S/4HANA Mockup serves as the maintenance management system, handling maintenance work requests and work orders throughout their lifecycle, including planning, scheduling, execution, and completion. This integration demonstrates how insights generated by the Digital Twin can trigger maintenance processes and support digital collaboration between operational systems and maintenance service providers.

3.3.6.2.2 Major digital languages

3.3.6.2.2.1 State of the art

Industrial ecosystems currently rely on a wide range of heterogeneous standards, proprietary data models, and communication technologies. Consequently, interoperability across systems and organizational boundaries is generally low, requiring costly point-to-point integrations and manual data transformations. This challenge highlights the need for semantic interoperability approaches that can bridge differences between standards while preserving the meaning and context of exchanged information.

3.3.6.2.2.2 Enablers

UC2.9 has adopted and applied WP3 technologies by representing DEXPI, STEP Core, CFIHOS, and AFO as ontologies and knowledge graphs. This approach has successfully validated the reuse of existing industrial standards within a semantic interoperability framework. The results demonstrate that standards originating from different industrial domains can be semantically aligned and integrated, providing a common knowledge foundation for Digital Twins, cross-system interoperability, and information exchange throughout the industrial value chain. UC2.9 has utilized WP3 technology with semantic representation of DEXPI, STEP Core, CFIHOS and AFO as ontologies and knowledge graphs, thereby validating the reuse of existing industrial standards within a semantic environment.

3.3.6.2.3 Autonomous translation

3.3.6.2.3.1 State of the art

See State of the art of UC 3.9 regarding translation services

3.3.6.2.3.2 Enablers

For details of the process flow used to transform document-oriented P&IDs into an equivalent RDF-based representation, please refer to the State-of-the-Art section of UC3.9. This process describes how information contained in traditional Piping and Instrumentation Diagrams (P&IDs) is extracted, semantically interpreted, and represented as RDF knowledge graphs, enabling machine-readable interoperability and integration within the Digital Twin environment.

3.3.6.3 Use Case Engineering Process

3.3.6.3.1 State of the art

State of the art is a document driven engineering process with a lot of shortcomings.

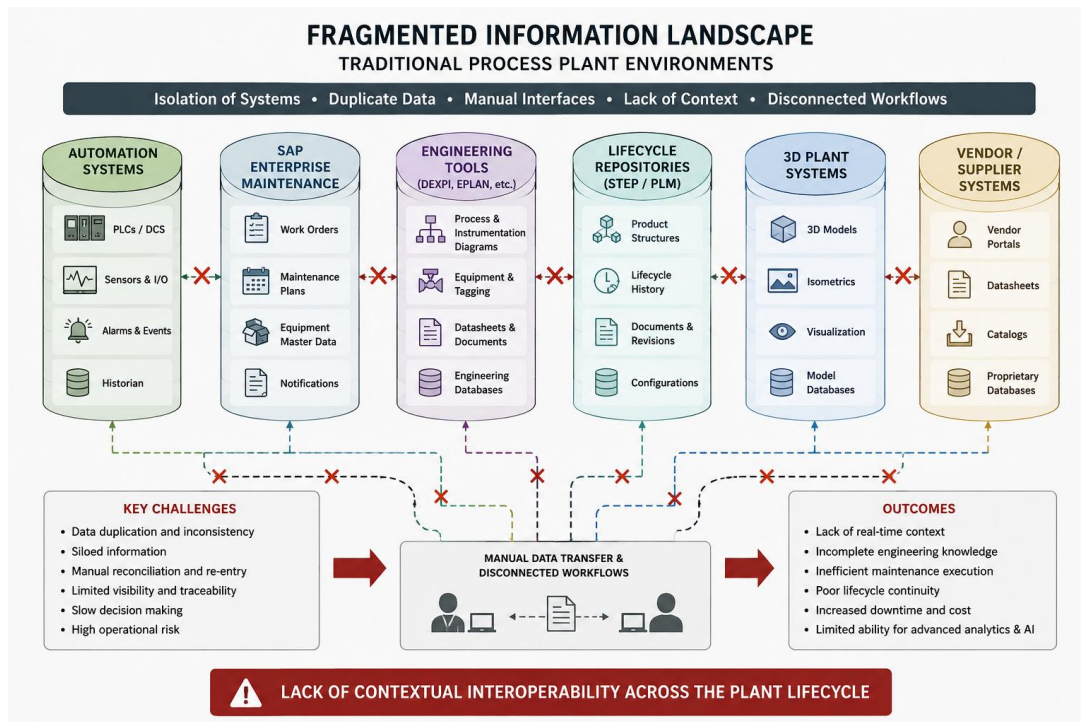


Figure 19: UC 2.9 state of the art.

The engineering process covers requirements from Future Mill 2030, digital twin architecture definition, engineering of KG/GraphDB/SPARQL/Cadmatic/simulation components, deployment through lab equipment POC and mill pilots, operational monitoring and simulation, maintenance through SAP work orders and evolution toward advanced predictions and learning environments.

3.3.6.3.2 Enablers

For UC2.9, the **process characteristics** of using a Knowledge Graph (KG)-based Digital Twin are significantly different from those of traditional data integration and monitoring solutions. The KG-based Digital Twin is not merely a data repository but a semantic operational layer that continuously contextualizes information across systems, domains, and lifecycle stages.

Process Characteristics of a KG-Based Digital Twin

1. Semantic Integration

Information from multiple heterogeneous sources is integrated through shared semantic models rather than custom point-to-point mappings. Data from engineering systems, IoT platforms, maintenance systems, ERP systems, and operational databases is linked through common concepts and relationships.

2. Context-Driven Information Management

The Digital Twin maintains contextual relationships between assets, processes, measurements, maintenance activities, documents, and organizational entities. Information is interpreted based on its meaning rather than its original source system.

3. Continuous Data Acquisition

Operational data, sensor measurements, events, and system updates are continuously ingested and linked to existing knowledge structures, creating an up-to-date representation of the production environment.

4. Lifecycle Information Continuity

Information remains connected throughout the asset lifecycle, from engineering and commissioning through operation, maintenance, and decommissioning. This eliminates information silos between lifecycle phases.

5. Knowledge-Centric Rather than Data-Centric

The Digital Twin focuses on managing knowledge and relationships rather than simply storing data. Assets, processes, dependencies, and business rules are represented explicitly within the knowledge graph.

Event-Driven Analysis

Operational events, alarms, anomalies, and maintenance needs can trigger automated analysis and workflows. The Digital Twin can identify dependencies and potential impacts across connected assets and processes.

7. Traceability and Provenance

All information can be traced back to its source, including engineering documents, sensor systems, ERP transactions, maintenance records, and external data sources. This improves auditability and trustworthiness.

8. Dynamic Querying and Exploration

Users and applications can retrieve information through semantic queries rather than predefined database structures. This enables flexible analysis and discovery of relationships that may not have been anticipated during system design.

9. Decision Support

The Digital Twin provides a contextualized view of operational conditions, supporting maintenance planning, fault diagnosis, production optimization, asset management, and sustainability reporting.

10. AI and Automation Enablement

Because information is represented semantically and machine-readably, the Digital Twin provides a strong foundation for AI applications, predictive maintenance, automated reasoning, and autonomous operational support.

Typical UC2.9 Process Flow

1. Engineering and operational information is acquired from source systems.
2. Information is transformed into semantic representations using ontologies and knowledge graphs.
3. Data is linked to existing assets, processes, and contextual information.
4. Sensor data and operational events continuously update the Digital Twin.
5. Analytics and fault detection mechanisms evaluate operational conditions.
6. Maintenance needs and operational insights are identified.
7. Relevant information is made available to enterprise systems and external stakeholders.
8. In UC3.9, maintenance requests, spare-part requirements, and service needs can be communicated to suppliers and service providers.
9. Feedback from maintenance execution and operational changes is reintegrated into the Digital Twin.

Summary

The key characteristic of a KG-based Digital Twin is that it transforms isolated data into connected operational knowledge. Instead of merely exchanging information between systems, it creates a shared semantic understanding of assets, processes, events, and relationships, enabling interoperability, traceability, intelligent analysis, and lifecycle-wide decision support. For UC2.9, this semantic and knowledge-centric approach is the foundation for both internal operational optimization and external value-chain collaboration as demonstrated in UC3.9.

3.3.6.3.3 Arrowhead-EP of the Use Case

The process we have used is dependent on the technology used.

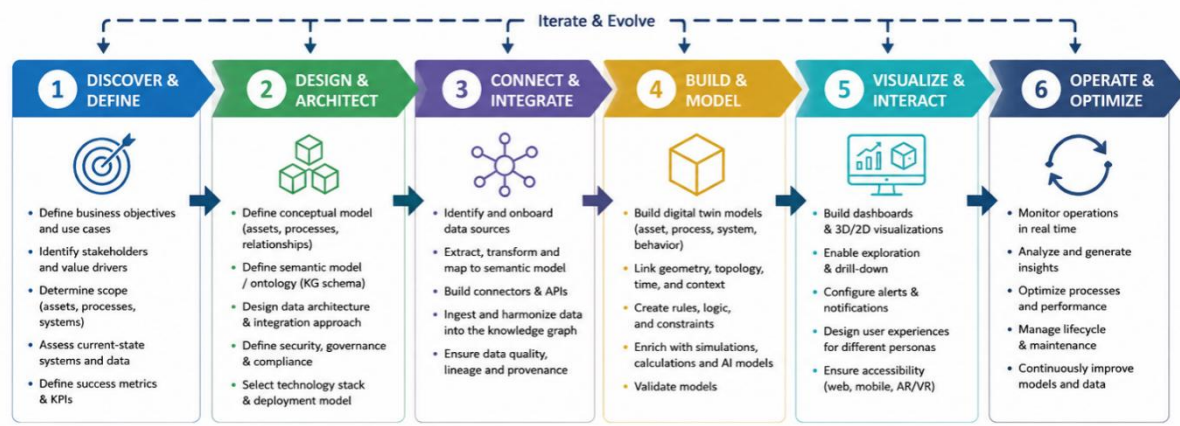


Figure 20: UC 2.9 engineering process phases.

In the following an example: EPP1 Future Mill 2030 and collaboration requirements -> EPP2 Cognitwin/Knowledge Graph architecture -> EPP3 ontology, GraphDB, OPC-UA, Cadmatic and simulation engineering -> EPP4 lab-equipment POC and pilot deployment -> EPP5 mill monitoring, remote operations and simulation -> EPP6 SAP work orders and digital twin maintenance -> EPP7 predictions, what-if and learning evolution; EPP8 simulation-based training and user guidance.

Scalability level: High in concept, because the KG/service architecture can handle many systems and mills. Practical scalability depends on data quality, standards alignment, IT access and availability of internal resources.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	Future Mill 2030 baseline, stakeholder workshops	Defines digitalization and interoperability gaps.
EPP2	Cognitwin architecture, DEXPI/STEP/IDO ontology design	Defines semantic digital twin structure.
EPP3	GraphDB, SPARQL, FluxDB, OPC-UA, Cadmatic eShare, Semantum/VT simulation	Builds digital twin, visualization and simulation services.
EPP4	LTU lab equipment POC, pilot mill datasets	Deploys and validates architecture.
EPP5	3D visualization, monitoring, remote support, simulation results	Operates digital twin and decision support.
EPP6	SAP work requests/work orders, digital twin update loop	Maintains physical and digital twin alignment.
EPP7	Predictions, what-if studies, model extension, additional data sources	Evolves capabilities and coverage.

EPP8	Simulation-based learning environment, user training	Supports operational learning.
------	--	--------------------------------

Table 54: Arrowhead EP mapping.

3.3.6.3.4 Mapping of tools in the Arrowhead-EP of the Use Case

In UC2.9, existing systems are integrated using a combination of:

1. Standard-to-ontology transformation (DEXPI, STEP Core, CFIHOS, AFO).
2. Ontology-based semantic mappings.
3. Semantic wrappers around operational systems.
4. Event-driven integration of sensor and maintenance data.
5. Knowledge graph-based Digital Twin representation.
6. Federated interoperability through W3C-aligned RDF and OWL standards.

This allows legacy systems, enterprise applications, engineering tools, IoT platforms, and maintenance systems to participate in a common semantic environment without requiring replacement of the original systems.

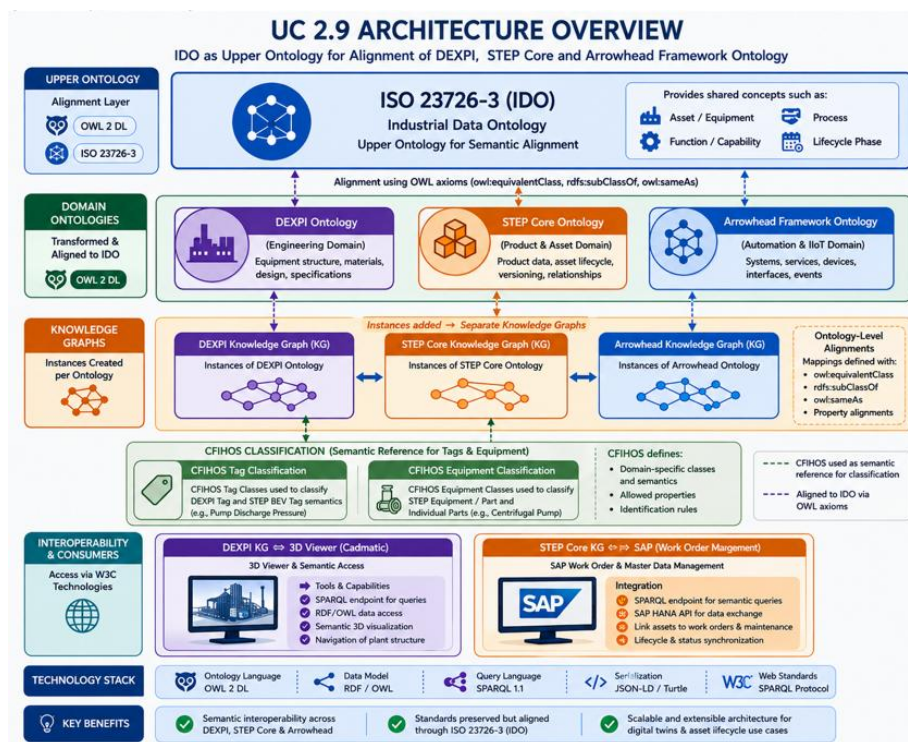


Figure 21: mapping tools in the engineering process.

3.3.6.3.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - The change from design time to run time engineering	UC2.9 establishes semantic continuity between engineering information and operational information through a Knowledge Graph-based Digital Twin. Engineering models, standards, and asset information remain accessible and reusable throughout operations and maintenance, reducing the gap between design-time and run-time engineering.

Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	UC2.9 enables semantic interoperability across engineering systems, operational systems, maintenance systems, and external stakeholders. The shared semantic environment allows multiple organizations and applications to collaborate using a common understanding of information and assets.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	The ontology-based information model and Knowledge Graph architecture are designed to manage large numbers of interconnected assets, sensors, events, and observations. Semantic abstraction reduces complexity while preserving detailed operational information and supporting scalable integration of new data sources.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The semantic technology supports the knowledge management and distribution in a superior way. The Knowledge Graph-based Digital Twin serves as a persistent organizational knowledge repository that supports AI-assisted learning, semantic search, decision support, and the continuous transfer of knowledge between engineering, operations, and maintenance personnel.

Table 55: Arrowhead EP objectives.

3.3.6.3.6 Adopted toolchain

Traditional industrial digitalization architectures are often built around tool chains, where information is passed sequentially from one application to another through a series of point-to-point integrations. In such environments, each tool maintains its own data model, and information must be repeatedly transformed, translated, and synchronized as it moves between engineering systems, asset management systems, maintenance platforms, operational technologies, and enterprise applications. This approach creates complexity, data duplication, semantic inconsistencies, and significant maintenance overhead.

The UC2.9 architecture adopts a fundamentally different approach. Instead of connecting tools directly to one another, all participating systems interact with a shared Knowledge Graph-based Digital Twin that serves as the common semantic information backbone. Information from engineering models, operational systems, IoT platforms, maintenance systems, enterprise applications, and external sources is integrated into a common knowledge representation based on shared ontologies and semantic standards.

The Knowledge Graph becomes the authoritative source of contextualized information, where assets, processes, documents, events, measurements, maintenance activities, and organizational entities are represented as interconnected semantic objects. Rather than exchanging information through multiple tool-specific interfaces, applications access the information they need directly from the Knowledge Graph through standardized semantic interfaces.

This architecture significantly reduces integration complexity. New tools do not require custom integrations with every existing system; instead, they only need to establish semantic alignment with the shared Knowledge Graph. As a result, integration complexity grows approximately linearly rather than exponentially as additional systems are introduced.

The common Knowledge Graph also ensures that all tools operate on a shared understanding of the underlying assets, processes, and relationships. Changes made by one application become immediately available to all other authorized applications through the shared semantic environment, eliminating the need for multiple data copies and synchronization mechanisms.

Furthermore, the architecture enables different tools to coexist and evolve independently. Existing systems can remain unchanged while exposing or consuming information through semantic interfaces. New applications, analytics services, AI agents, and Digital Twin services can be introduced without disrupting existing integrations.

In summary, the UC2.9 architecture avoids traditional tool chains by replacing sequential system-to-system information flows with a shared Knowledge Graph-based Digital Twin. Rather than moving information between tools, the architecture provides a common semantic environment from which multiple tools, services, and stakeholders can access, contribute, and reuse the same contextualized knowledge. This creates a more scalable, interoperable, and future-proof digital ecosystem.

3.3.6.3.7 Toolchain automation

Medium to high in the POC. Ontology transformations and SPARQL access are automated for selected datasets; operational automation still depends on reliable adapters, data quality and access to mill IT/OT resources.

3.3.6.3.8 Toolchain limitations

Data is fragmented and definitions vary across systems.

No mill-wide Reference Data Library at baseline.

Delay from limited internal IT resources and dependency analysis not delivered.

Translation of tag discrepancies remains a key risk.

3.3.6.3.9 Use case and engineering costs

A Knowledge Graph (KG)-based architecture provides significant cost advantages over traditional tool-chain architectures by reducing integration complexity, improving information reuse, simplifying governance, and accelerating the introduction of new applications and services. These benefits can be categorized into procurement and implementation savings, adoption savings, and governance and operational savings.

1. Procurement and Implementation Savings

Reduced Integration Costs

In a traditional tool-chain architecture, each system typically requires multiple point-to-point integrations with other systems. As the number of systems grows, integration complexity increases exponentially, resulting in higher implementation costs, longer project durations, and greater dependency on specialized integration expertise.

A Knowledge Graph architecture replaces multiple pairwise integrations with a shared semantic layer. Each system only needs to establish a semantic connection to the Knowledge Graph, significantly reducing the number of interfaces that must be developed and maintained.

Benefits:

- Fewer custom integrations
- Reduced middleware requirements

- Lower development effort
- Faster deployment of new systems

Reduced Vendor Lock-In

Traditional architectures often rely on proprietary integration technologies and vendor-specific data models. This can increase licensing costs and make future technology changes expensive. Knowledge Graph architectures are based on open standards such as RDF, OWL, SPARQL, and SHACL, reducing dependency on specific software vendors and increasing procurement flexibility.

Benefits:

- Greater supplier competition
- Reduced licensing costs
- Lower migration costs
- Increased technology flexibility

Reuse of Existing Standards

Semantic architectures can directly reuse standards such as IFC, DEXPI, STEP Core, CFIHOS, and AFO by representing them as ontologies and knowledge graphs.

Benefits:

- Reduced modeling effort
- Reduced custom schema development
- Faster interoperability implementation
- Preservation of previous investments in standards

2. Adoption and Business Change Savings

Faster Onboarding of New Systems

Traditional tool chains require extensive integration projects whenever new systems are introduced.

In a KG-based architecture, new systems only need to align with the common semantic model.

Benefits:

- Faster onboarding
- Lower implementation risk
- Reduced integration project scope
- Accelerated innovation

Simplified User Adoption

Users often struggle with multiple disconnected systems presenting inconsistent information. The Knowledge Graph provides a shared semantic context across systems.

Benefits:

- Reduced training effort
- Improved user productivity
- Consistent terminology and data interpretation
- Better cross-functional collaboration

Faster Development of New Applications

Because the Knowledge Graph provides a common information backbone, new applications, dashboards, AI services, and Digital Twin capabilities can be developed without creating new integrations for each source system.

Benefits:

- Reduced application development costs
- Faster time-to-market
- Improved reuse of existing information assets

3. Governance and Operational Savings

Reduced Data Duplication

Traditional architectures often create multiple copies of the same information across systems and data warehouses.

Knowledge Graphs support federated and linked information models.

Benefits:

- Lower storage costs
- Reduced synchronization effort
- Improved data consistency
- Reduced maintenance overhead

Simplified Change Management

Schema changes in one system frequently trigger modifications across many integrations in traditional environments.

In a semantic architecture, systems are decoupled through shared ontologies.

Benefits:

- Reduced impact of system upgrades
- Lower regression testing effort
- Reduced maintenance costs
- Improved architectural resilience

Improved Data Governance

Knowledge Graphs make semantics explicit through ontologies, constraints, and validation rules.

Benefits:

- Improved data quality
- Reduced data reconciliation effort
- Better compliance support
- More efficient governance processes

Lifecycle Information Continuity

Traditional tool chains often create information silos between engineering, operations, maintenance, and supply chain functions.

Knowledge Graphs preserve relationships across the entire asset lifecycle.

Benefits:

- Reduced information loss
- Improved traceability
- Better decision support
- Lower operational risk

4. AI and Analytics Savings

Reduced Data Preparation Costs

A significant portion of AI and analytics projects is typically spent locating, cleaning, and integrating data.

Knowledge Graphs provide contextualized and semantically integrated information.

Benefits:

- Faster AI project delivery
- Reduced data engineering effort
- Improved model quality
- Increased reuse of analytics assets

Enhanced Automation

The machine-readable semantics of a Knowledge Graph enable automated reasoning, intelligent search, and workflow automation.

Benefits:

- Reduced manual processing
- Faster decision-making
- Lower operational costs
- Increased productivity

5. Comparative Cost Characteristics

Cost Category	Traditional Tool Chain	KG-Based Architecture
Integration Development	High	Low
Integration Maintenance	High	Low
New System Onboarding	High	Low
Data Duplication	High	Low
Change Management	Complex	Simplified
Governance Effort	Fragmented	Centralized
Vendor Dependency	High	Low
AI Readiness	Limited	High
Standards Reuse	Limited	High
Lifecycle Traceability	Low	High

Table 56: comparative costs.

In conclusion, the primary source of cost savings in a Knowledge Graph-based architecture is the replacement of multiple tool-to-tool integrations with a shared semantic information

backbone. This reduces implementation effort, lowers maintenance costs, accelerates adoption of new technologies, simplifies governance, improves data quality, and enables greater reuse of information across the organization.

While the initial development of shared ontologies and semantic models requires investment, the long-term benefits typically result in substantially lower total cost of ownership compared to traditional tool-chain architectures. As the number of participating systems, stakeholders, and data sources grows, the economic advantages of the Knowledge Graph approach become increasingly significant.

3.3.6.4 Standards

3.3.6.4.1 Standardisation requirements

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	ERP/MES, JSM	Life Cycle Management, Software life cycle processes , SQuaRE, REST APIs, ERP/MES, Function Blocks UC 2.9 do not have a System approach but a Knowledge sharing approach
Information and Representation	The principals of 81346, UNSPC	IDO, STEP Core, DEXPI, CFHIOS, Arrowhead Framework Ontology,
Semantic and Language	XML, JSON,	XML, JSON, RDF, SPARQL, OWL2DL
Communication	MT Connect, ETHERCAT, Sercos, Fieldbus, OPC-UA, UMATI, MQTT, M2M, WIFI, RDF and NFC, EAP, COAP, Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP	MT Connect, ETHERCAT, Sercos, Fieldbus, OPC-UA, UMATI, MQTT, M2M, WIFI, RDF and NFC, EAP, COAP, Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP
Cybersecurity and Safety	Standard technology in MS-Azure	Standard technology in MS-Azure, Firewall
Reference Model	???	W3C Reference model
Domain-Specific	Robotics, IoT	Robotics, Artificial Intelligence, IoT, Semantic Digital Twin, Arrowhead framework
Framework Development and Specific Applications for development	Node.JS, Angular JS, JAVA, Javascript, .NET, Node Red, PHP, Python,	Node.JS, Angular JS, JAVA, Javascript, .NET, Eclipse, Python, GrapfDB

Table 57: standards adoption.

3.3.6.4.2 Key Performances Indicators

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation			Final Year 3 evaluation
Name	KPI Description	Means of assessment	Year 1	Year 2	Year 3	
Autonomous Mills & Digital Twin (Digital Value Chain)	Data quality & integration	Number of integrated data sources Number of connected systems (DCS, PLC, MES, IoT, etc.)	0		9	9
Autonomous Mills & Digital Twin (Digital Value Chain)	Digital twin usage	Adoption rate (% of relevant processes) Share of operations using the digital twin	0%		100% 10 of 10	100%
Remote Operations & Support	Information access & collaboration	Time to access information (seconds/minutes)	hours		From hours to a few minutes	From hours to a few minutes
Dynamic Simulation (Digital Twin P&ID)	Model quality	Model update effort	40 hours per A1 format		20 hours per A1 format	20 hours per A1 format (P&ID)

Table 58: UC 2.9 Key Performance Indicators.

3.3.6.5 Beyond the baseline

The next phase in the development of the semantic environment is to expand the Knowledge Graph-based Digital Twin from a production-centric solution into a broader ecosystem platform that supports interoperability across the entire value chain. This includes integrating additional industrial standards, onboarding new data sources and enterprise systems, and extending semantic interoperability to suppliers, service providers, and external stakeholders. Future developments will focus on automated semantic integration, AI-assisted reasoning, predictive maintenance, digital thread enablement, and support for federated data spaces. By strengthening governance, standardization, and ontology reuse, the semantic environment will evolve into a scalable and trusted foundation for Digital Twins, autonomous decision support, and cross-organizational collaboration.

3.3.6.5.1 Use case action plan.

3.3.6.5.1.1 Year 1 – Foundation, Digitalization and AI Readiness

Semantic Foundation

- Consolidate the Knowledge Graph-based Digital Twin architecture.

- Establish governance processes for ontologies, metadata, and semantic models.
- Expand support for IFC, DEXPI, STEP Core, CFIHOS, AFO, and other relevant standards.
- Integrate ERP, CMMS, MES, SCADA, Historian, and engineering systems.
- Implement semantic validation and automated data quality monitoring.

Process Digitalization

- Identify and document manual and paper-based processes.
- Establish digital workflows for maintenance, inspections, and engineering change management.
- Create a unified semantic asset model across engineering, operations, and maintenance.

AI Introduction

- Deploy AI-powered semantic search across documents, assets, and operational information.
- Implement AI assistants for engineering and maintenance knowledge retrieval.
- Introduce AI-supported anomaly detection using sensor and operational data.
- Deploy AI tools for automatic classification, tagging, and contextualization of engineering documents.
- Establish a secure AI environment connected to the Knowledge Graph.

Expected Outcome

- Reduction of manual information searches.
- Improved information accessibility.
- Foundation established for enterprise-wide AI adoption.

3.3.6.5.1.2 Year 2 – Process Transformation and AI-Assisted Operations

Semantic Expansion

- Extend the Digital Twin to additional production assets and business processes.
- Integrate suppliers, maintenance service providers, and external stakeholders.
- Support federated information exchange and participation in industrial data spaces.

Process Transformation

- Replace manual information handovers with semantic information exchange.
- Automate maintenance planning and work-order generation based on Digital Twin insights.
- Enable digital thread traceability across engineering, operations, maintenance, and supply chain processes.

AI-Assisted Decision Making

- Deploy AI agents for maintenance planning and scheduling support.
- Implement AI-driven root-cause analysis using the Knowledge Graph.
- Introduce predictive maintenance models connected to Digital Twin data.
- Deploy AI-powered engineering assistants capable of answering operational and asset-related questions.
- Automate document generation, reporting, and compliance documentation.
- Introduce AI-assisted supplier interaction and procurement recommendations.

Expected Outcome

- Significant reduction of manual administrative work.
- Faster maintenance planning and issue resolution.
- Improved operational efficiency and knowledge reuse.

3.3.6.5.1.3 Year 3 – Autonomous Operations and Intelligent Ecosystem

Semantic Ecosystem

- Establish the Knowledge Graph as the primary interoperability backbone across the organization.
- Enable cross-company Digital Twin collaboration with suppliers, customers, and service providers.
- Scale the semantic environment across multiple facilities and production sites.

Autonomous Process Transformation

- Replace traditional workflow-driven processes with event-driven and knowledge-driven processes.
- Enable automatic generation and orchestration of maintenance workflows.
- Implement closed-loop feedback between engineering, operations, maintenance, and suppliers.
- Support autonomous lifecycle management of assets and equipment.

Advanced AI and Agentic Systems

- Deploy AI agents capable of independently monitoring operational performance.
- Enable autonomous fault diagnosis and recommendation generation.
- Implement AI-supported optimization of production, energy usage, and maintenance activities.
- Introduce multi-agent collaboration where engineering, maintenance, procurement, and sustainability agents work together using the shared Knowledge Graph.
- Enable conversational interaction with the Digital Twin through natural language interfaces.
- Deploy AI-driven decision support for investment planning, risk management, and sustainability optimization.

Expected Outcome

- Transformation of traditionally manual processes into intelligent, knowledge-driven workflows.
- Significant reduction in operational and maintenance costs.
- Improved asset availability and production performance.
- AI-enabled decision support across the entire asset lifecycle.
- Establishment of a scalable and future-proof industrial semantic ecosystem.

3.3.6.5.2 Strategic Vision After Three Years

- Knowledge Graph serving as the organization's semantic operating system.
- AI agents becoming active participants in engineering, maintenance, and operational processes.
- Most information-intensive manual processes partially or fully automated.
- Digital Twin supporting predictive and autonomous operations.
- Semantic interoperability extending across the entire value chain.
- Human experts focusing on decision-making and innovation rather than information gathering and administration.
- A transition from document-centric and system-centric processes to knowledge-centric and AI-enabled operations.

3.3.6.5.3 Contribution to project objectives.

Project Objective	Status at M0	Expected improvement	Improvement using UC2.9 Semantic Technology
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	No translations	Have translations between a few, most relevant technical data format	Develop semantic representations of DEXPI, STEP Core, CFIHOS, AFO, and other relevant standards. Integrate these standards into the Knowledge Graph-based Digital Twin and demonstrate automated semantic mapping and information exchange between heterogeneous systems.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	No autonomous utilization of translations	Ability to inject translations into existing processes, thus improving interoperability	Establish the Knowledge Graph as a semantic interoperability backbone. Expose information through semantic APIs and services that can be consumed by enterprise systems, maintenance applications, Digital Twins, AI agents, and supplier ecosystems.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	Several different standards exist and are in use	Select and harmonize the use of different standards	Develop ontology mappings and semantic alignments between DEXPI, STEP Core, CFIHOS, AFO, IFC, and related standards. Demonstrate how existing standards can coexist and interoperate through shared semantic concepts rather than direct schema translations.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a	Low capability for transformation management	Digital transformation management effects to ensure	Develop guidelines, governance processes, onboarding procedures, and best practices for introducing Knowledge Graphs and semantic interoperability into engineering,

PVN		smooth transition in work practices.	operations, maintenance, and supply chain processes. Support organizational change through AI-assisted and knowledge-driven workflows.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform			Establish governance for ontologies, semantic models, Digital Twin assets, metadata, and interoperability services. Promote reuse of open standards, open-source semantic technologies, and shared Knowledge Graph infrastructure to support long-term scalability and ecosystem growth.

Table 59 – Contribution to project objectives.

While the original project objectives emphasize autonomous translation services, UC2.9 addresses the underlying interoperability challenges through semantic interoperability and Knowledge Graph technologies. Rather than continuously translating between heterogeneous data models, UC2.9 establishes a shared semantic understanding that significantly reduces the need for translations while improving scalability, maintainability, and interoperability across production value networks.

3.3.6.6 Final assessment

3.3.6.6.1 KPI Assessment

UC2.9 met all the KPIs initially set. The Knowledge Graph-based Digital Twin showed its ability to merge different types of data, like engineering, operations, maintenance, and sensor info, into one neat framework. They successfully combined nine data sources and systems, hitting their goal and showing that semantic interoperability can actually work across separate IT and OT spaces.

For the adopted Digital Twin, 100% of the planned processes got the upgrade. This means every considered scenario can indeed take advantage of the semantic Digital Twin setup. This strong performance suggests the created system works well for all the needed engineering and operation tasks, not just a few tests here and there.

Significant improvements were also achieved in information accessibility and collaboration. Using semantic search and organized data handling cut down time spent looking for tech details from hours to minutes. This result directly addresses one of the main baseline challenges, namely the fragmentation of information across multiple systems and the dependence on expert knowledge to identify where relevant information is stored.

Furthermore, the effort required to update and maintain process models was reduced from approximately 40 hours to 20 hours per A1-format P&ID. This 50% reduction demonstrates the benefits of the semantic Digital Twin approach in maintaining consistency between engineering information, operational data and associated documentation, while reducing manual engineering effort.

Overall, the KPI evaluation confirms that the use case objectives related to interoperability, information accessibility, Digital Twin utilization and engineering efficiency were fully achieved. The results demonstrate that semantic technologies and Knowledge Graph-based Digital Twins can significantly improve the management, accessibility and reuse of information across industrial production value networks.

3.3.6.6.2 Contribution to project objectives

UC2.9 contributed to the Arrowhead fPVN project objectives by demonstrating semantic interoperability as an alternative and complementary approach to traditional information translation. While the use case did not focus on direct format-to-format translation services, it established a shared semantic environment based on Knowledge Graph technologies, enabling information originating from different engineering, operational, and maintenance standards to coexist and interoperate within a common framework. This directly supports the project objective of reducing the need for manual interpretation and enabling machine-understandable information exchange across production value networks.

The use case validated the project vision of service-oriented and microservice-based interoperability by integrating Digital Twin services, maintenance management functions, operational data sources, and simulation capabilities through Arrowhead-inspired architectures. The semantic Digital Twin acted as a common interoperability layer capable of serving multiple applications and stakeholders without requiring dedicated point-to-point integrations. This demonstrates how dynamic deployment and autonomous utilization of information services can be achieved in realistic industrial environments.

3.3.6.6.3 Improvement in the engineering process

The most significant engineering-process improvement introduced by UC2.9 is the transition from a traditional document-centric and tool-centric engineering environment toward a knowledge-centric engineering paradigm based on a shared Knowledge Graph Digital Twin. In the baseline situation, engineering, maintenance, operational, and enterprise systems managed information independently, resulting in duplicated data, manual reconciliation efforts, fragmented knowledge, and limited traceability across lifecycle phases. Information exchange largely relied on static documents and custom integrations, making change management and cross-disciplinary collaboration difficult.

The implemented Digital Twin architecture fundamentally changes this approach by establishing semantic continuity between engineering design information, operational data, maintenance activities, simulation models, and enterprise processes. Information is no longer exchanged through sequential tool chains but is instead represented as shared semantic knowledge accessible by multiple stakeholders and applications simultaneously. Engineering information generated during design remains connected to operational and maintenance activities throughout the entire lifecycle of the asset. This eliminates many traditional information handover problems and reduces the loss of engineering knowledge after commissioning.

The use case further demonstrated how semantic technologies enable dynamic querying, contextual information retrieval, lifecycle traceability, and automated reasoning capabilities that are not achievable with conventional engineering databases. The Digital Twin supports event-driven workflows, automated fault analysis, maintenance planning, and integration with simulation models, thereby transforming engineering activities from isolated design tasks into continuous lifecycle engineering processes. The resulting approach provides a scalable foundation for future autonomous operations, AI-assisted decision support, predictive maintenance, and collaborative engineering across production value networks.

3.3.7 WP9 - Use-case #3.9 – Interoperability for technical information exchange in process industry

3.3.7.1 Baseline summary

3.3.7.1.1 General description of the baseline

Information is currently shared between companies manually based on point-to-point and case-by-case specifications. Technical solutions and standards for interoperable digital information exchange are not available. Emerging technology from DEXPI is available only for a small part of this, e.g., P&I diagrams.

Another important emerging technology is ISO15926, based on Semantic Web technologies, that can act as a glue to integrate different data and concepts throughout the lifecycle. The challenge with applying ISO15926 is that it does not yet cover all necessary application domains, such as pulp&paper or processing in general. Also, a lot of attributes and properties, defined in other standards, are needed but not yet included.

Mappings to different systems and between different standard data models are proprietary and implemented manually and often developed case by case. OPC UA is a de facto standard in process automation integration but despite its extensive information modeling features it primarily provides solutions for runtime systems integration and not for the whole lifecycle including engineering, operation and maintenance needs.

Information cannot be aggregated from multiple sources for many operation and maintenance purposes or process analytics. In such integrations, implementations are based on manually setting up connections to necessary proprietary systems and defining how information is to be used and integrated case-by-case.

3.3.7.1.2 Initial architecture of the use case as a sequential list of functional blocks

This use case contains the three separate use-case parts from task 9.3:

- Digital process for investment project and supplier service data exchange - technical information exchange in fPVNs throughout the lifecycle from investment project needs to service operations:
 - Translation of plant legacy information into machine understandable standard format (links to WP3 and WP4).
 - Enriching the machine understandable data from multiple sources from the plant, EPCs and equipment manufactures.
 - Transformation of engineering data into value adding machine understandable data for other models like steady state and dynamic process simulation models.
- Dependability of data in digital value chains: a combination of technical asset information with operational data to improve dependability.
- Traceability and documentation of sustainability in production value chains - gathering, and use of sustainability data through network LCA.

All in all, in an investment project, information is exchanged within and between project phases with numerous different parties. The most important providers of information are external suppliers, machine and equipment manufacturers, providers of design and installation work and other subcontractors.

In the investment project, the plant is preparing for the continuous operation and maintenance phase, which is described by a dash-dotted purple box. The supplier service use-case part belongs to the O&M phase. However, since its basic features are very similar to an investment project, it is described in the same context as the use-case of the investment project.

The dependability use-case part is targeted also to O&M phase (indicated by a purple box). Dependability, however, needs extensive information from engineering and design phase.

The sustainability use-case part is usually not included in the plant lifecycle scope. It primarily focuses on calculating the environmental impact of production, where raw materials, chemicals and energy play the main role. But the components of production assets during the life cycle also influence it, which in the future can be taken into account considered, for example, through a digital product passport⁶ (DPP).

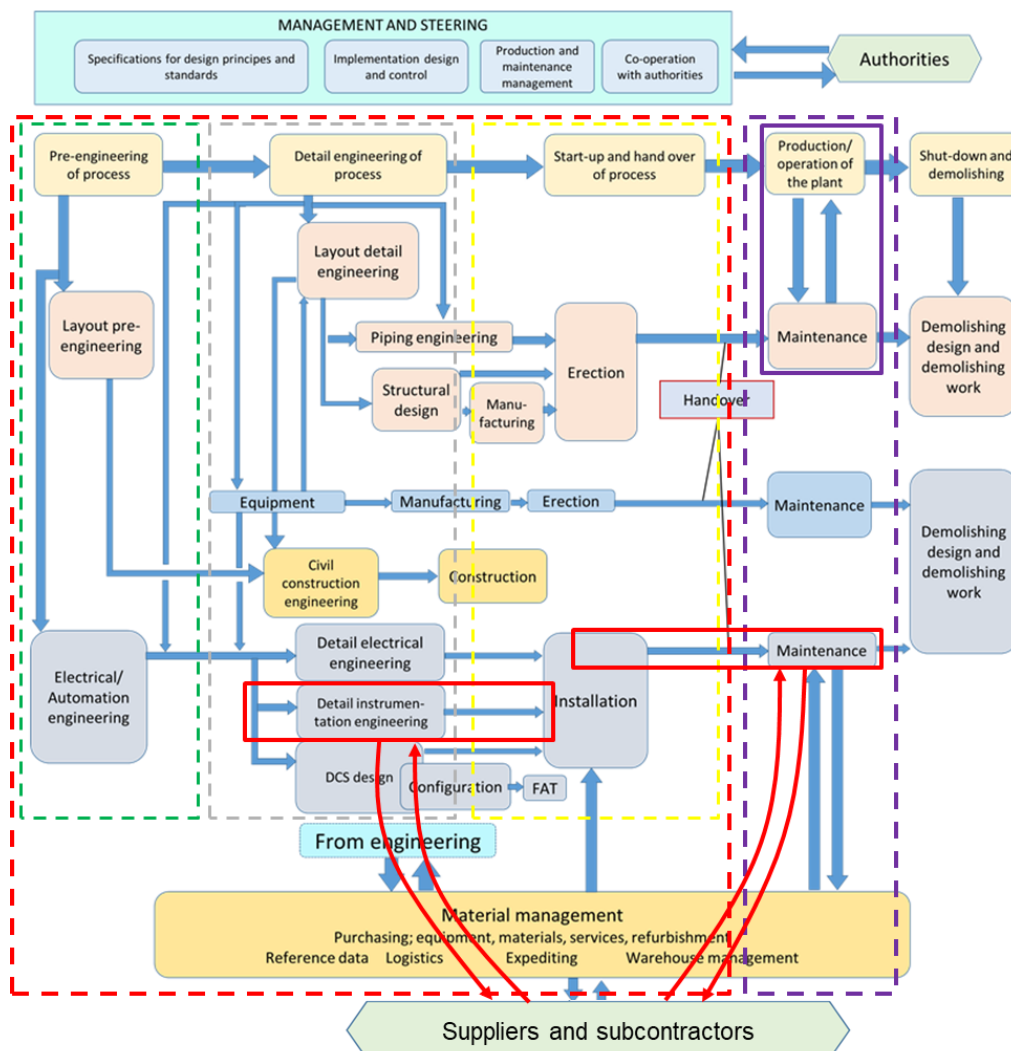


Figure 22 - General information flow diagram for plant lifecycle data from an investment project (red dashed line box) to plant demolition (source PSK 5990 standard). Thick blue lines describe information flows. The investment project and

⁶ The New Digital Product Passport, European Commission https://commission.europa.eu/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/sustainable-products/ecodesign-sustainable-products-regulation_en#the-new-digital-product-passport

supplier service pilots for valves and instrumentation are described by solid line red boxes (from engineering to installation and O&M). The dependability use-case is described by solid line purple box inside the O&M phase box (dash-dotted purple box). (Source the national PSK5990 standard).

3.3.7.1.3 Objectives that are linked to the Arrowhead fPVN objective

Project Objective	UC Contribution
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	The definitions have contributed to identifying different formats and information models used in fPVNs. These include model and data brokering of asset information systems, engineering systems, operational systems, and digital twins.
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	Interoperable infrastructure is needed for composition of services as well as authentication and authorization in changing production value networks. Special needs concerning security and integrity, and data access management and data sharing concept have been identified. Use of Eclipse Arrowhead service infrastructure to manage integrations and authorizations in these environments is being designed.
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	Several important standards have been identified in use for some aspects on the baseline level implementation of the use cases, and several transformations and translations between such models are being implemented.
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	Digital transformation management effects and concrete actions to ensure smooth transition in work practices is being identified. This is done in close collaboration with WP10. The maturity and goal assessment developed in WP10 supports clear decision-making and progress tracking across the project lifecycle.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	-

Table 60 – Arrowhead EP objectives.

3.3.7.2 Baseline analysis - building blocks & enablers

3.3.7.2.1 Micro-service paradigm

3.3.7.2.1.1 State of the art

Currently, micro-services are not the de facto integration model although used in some applications. A standardized method would speed up integration, improve security, and facilitate building new integrations in the future.

3.3.7.2.1.2 Enablers

Micro-services paradigm could help with services offered and consumed, and to provide interoperable infrastructure regarding discovery and composition of services as well as authentication and authorization as part of the production value networks.

3.3.7.2.2 Major digital languages

3.3.7.2.2.1 State of the art

Currently, a huge number of different and heterogeneous standards are used. Interoperability to and between existing systems is therefore low in general.

3.3.7.2.2.2 Enablers

Guidelines for standards and information models to promote when developing integrations and future implementations would facilitate interoperability but also foster managing and having information available that is needed in various operations.

3.3.7.2.3 Autonomous translation

3.3.7.2.3.1 State of the art

Currently, translations are done manually which is laborious and prone to error.

3.3.7.2.3.2 Enablers

Translation capabilities between different information models would allow integrating processes much more efficient. Automatic and autonomous translations would facilitate use of information in various new data use cases, also between different organisations. Availability of translations for certain data models and standards would promote their use further fostering interoperability in the value chains.

3.3.7.3 Use Case Engineering Process

3.3.7.3.1 State of the art

Currently, there is no standard or defined engineering process of the integrations associated in with technical information exchange in process industry. The practices follow typical software engineering processes for system integration that are adapted case by case and include typically requirements specification, design and implementation, testing and validation, user testing, piloting and acceptance into use.

The process is multi-stream. Investment project data exchange covers concept/pre-feasibility, design, quotation, order, change management, delivery, installation and handover. Supplier service covers ordering and delivery of equipment/refurbishment data. Dependability spans early design and operational stages. Sustainability focuses on value-chain traceability and LCA documentation. Arrowhead-EP is used as a unifying lifecycle lens across these streams.

3.3.7.3.2 Enablers

Agreed data models and templates for information exchange, agreed interfaces and defined business processes with data exchange sequences constitute the building blocks for managing and versioning interoperability throughout the lifecycle.

The engineering process is enabled by the IEP/Sharecat concept, Peppol, DEXPI, PCA Library Service, Model Broker, AFRY ProElina APIs, Balas Web API/Arrowhead adapter, NetLCA/SMPC, BPMN process modelling and Arrowhead framework integrations.

3.3.7.3.3 Arrowhead-EP of the Use Case

EPP1 lifecycle, investment, supplier, dependability and sustainability requirements -> EPP2 technical/business exchange architecture -> EPP3 DEXPI, IEP, Peppol, Balas, ProElina and NetLCA engineering -> EPP4 pilot service deployment -> EPP5 procurement, supplier service, dependability and LCA operation -> EPP6 service, maintenance and lifecycle documentation -> EPP7 standards, workflow and data-space evolution; EPP8 Model Broker and partner training.

Scalability level: High but complex. The architecture targets distributed value networks and multiple organizations, but requires governance of standards, business rules, access rights and data confidentiality.

EPP	Mapped methodology/toolchain element	Role in the use case
EPP1	Investment project, supplier service, dependability and sustainability requirements	Defines multi-stream scope and KPIs.
EPP2	BPMN, Peppol + Arrowhead architecture, IEP/Sharecat concept, PCA Library Service	Defines business/technical exchange architecture.
EPP3	Model Broker, DEXPI conversion/enrichment, Balas, ProElina API, NetLCA/SMPC	Engineers translators, adapters and data platforms.
EPP4	IEP pilot, Balas Web API adapter, sustainability PoC, supplier service pilot	Deploys pilots and adapters.
EPP5	Purchase-to-Pay, supplier service, dependability analytics, LCA exchange	Operates digital workflows.
EPP6	Installed-base updates, RAMS/dependability monitoring, sustainability traceability	Maintains lifecycle data and operational evidence.
EPP7	Standardization proposals, workflow refinement, data-space evolution	Evolves network architecture and models.
EPP8	Model Broker training, guidelines, documentation	Supports partner adoption.

Table 61: Arrowhead EP mapping.

3.3.7.3.4 Mapping of tools in the Arrowhead-EP of the Use Case

The mapping table above represents the current UC-EP decomposition. Tools and methodologies are grouped by Arrowhead-EP phase, with EPP8 treated as a cross-cutting training and adoption phase.

3.3.7.3.5 How the Arrowhead-EP allows matching the WP2 objectives

Arrowhead-EP is not directly applied in the use case, but objectives similar to the ones considered in the following table are valid.

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - The change from design time to run time engineering	Integration and binding of information systems at run-time. Engineering/design data, procurement data, service records, dependability and sustainability data are connected through lifecycle workflows.

Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Agreed interfaces and practices improve the interoperability. Plant owner, EPCs, suppliers, service providers, universities and tool vendors exchange data through common standards and services.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Improved automation possibilities are expected to increase the handling of more devices, I/Os, and associated attributes and data. The use case manages many technical objects, documents, business messages, sustainability data points and dependability events.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Training, including simulation, and evaluating integrations are expected to benefit from the standardised model. Model Broker training, standards guidance and pilot feedback create reusable learning for the ecosystem.

Table 62: Arrowhead EP objectives.

3.3.7.3.6 Adopted toolchain

Sharecat/IEP concept, Peppol/UBL/ISO 19845, BPMN, DEXPI Plant and Process, PCA Library Service, Model Broker, AFRY ProElina API, Balas simulator and Web API/Arrowhead adapter, NetLCA/SMPC platform, SAP/MES data integration for dependability, Arrowhead framework services.

3.3.7.3.7 Toolchain automation

Medium. Several workflows are specified and some pilot implementations have started; automated Purchase-to-Pay, DEXPI transformations and sustainability data exchange are being implemented/tested. Manual work remains in standards mapping, data enrichment and quality assurance.

3.3.7.3.8 Toolchain limitations

Baseline processes are document-centric and manually managed.

Pre-award business process definitions and data models are incomplete.

SAP/MES data quality limits dependability analysis.

Value-chain sustainability data exchange must preserve confidentiality and trade secrets.

Integration of Peppol commercial interfaces with Arrowhead technical data services adds architectural complexity.

3.3.7.4 Standards

3.3.7.4.1 Standardisation requirements

Standardisation Requirements		
Categories of Standards	Baseline	Final
System and Software	Life Cycle Management, Software life cycle processes , REST APIs, ERP/MES	Life Cycle Management, Software life cycle processes , SQuaRE, REST APIs, ERP/MES, Function Blocks

Information and Representation	Product data representation and exchange, eCI@ss	Product data representation and exchange, eCI@ss, RDL2 (Ed.3), ISO 15926-4, IEC 61987, CFIHOS, ISO 23726-3 ontology, ISO 10303, DEXPI Plant, DEXPI Process, EN ISO 10628
Semantic and Language	XML, JSON, CSV, Excel	XML, JSON, CSV, Excel, RDF, SPARQL
Communication	Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP	Internet Protocols: HTTP, SSH, FTP, TCP/IP, IPSEC, SMTP, UDP
Cybersecurity and Safety	IEC 62443, ISO 27001, NIST SP 800-82, NIS2	IEC 62443, ISO 27001, NIST SP 800-82, NIS2
Reference Model	RAMI, IIRA, ASS, NIST, Digital Factory	RAMI, IIRA, ASS, NIST, Digital Factory
Domain-Specific		IEC 61406, ISO/IEC 81346-1; ISO/IEC 61355, IEC 82079; ISO 14040 and ISO 14044
Framework Development and Specific Applications for development		PEPPOL (UBL, ISO 19845)

Table 63: adopted standards.

3.3.7.5 Key Performances Indicators

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation			Achieved in year 3
Name	KPI Description	Means of assessment	Year 1	Year 2	Year 3	
Automated data translation	Translation coverage of technical information	Selected demonstration use case, PCA RDL data	0	0	80 %	80% - the demonstrator provides operational semantic translation and harmonization of the selected engineering, maintenance and operational datasets, covering the technical information required by the industrial use case. No significant gaps affecting the intended workflows are reported in the validation results.
Run-time connection of systems	The share of system components integrated at run-time	Selected demonstrations use case evaluation, Eclipse Arrowhead integration	0	10 %	30 %	30% - The demonstrator successfully establishes operational run-time connectivity between the selected engineering, maintenance and operational information systems through the semantic Digital Twin platform. The integrated environment supports live access to distributed information sources and enables cross-system information exchange during

						operation, fulfilling the scope defined for the KPI.
--	--	--	--	--	--	--

Table 64: KPIs.

3.3.7.6 Beyond the baseline

The use-case objectives aim to improve process industry information-exchange through interoperability for data exchanged, digital processes and system integrations. This is targeted especially for distributed value networks, for which understanding the data, preferably using machine interpretable content, would foster digitalization along supply chain during the lifecycle. The use case basically consists of the four streams “Investment project” which is the base case, “Supplier service”, “Dependability” and “Sustainability”.

3.3.7.6.1 Use case action plan.

The use cases for the investment project and the supplier service for Stora Enso have been mostly defined and implementation already began in the summer. The key new information system component to be tested for both is the so-called IEP concept (tested via the Norwegian Sharecat SaaS service). The network architecture principles for collaboration in pilot use-cases has been defined.

In the fall of 2024, the focus on dependability has centered around SAP data. This data, sourced from Stora Enso, spans eight years of history. A preliminary analysis has been conducted to assess the quality and coverage of the most critical data fields. The initial findings indicate that the main challenges with SAP data are related to event consequences, recovery time duration, and precise event targeting within the device hierarchy. Without additional data, there remains significant uncertainty in the analysis results.

The next step involves supplementing the SAP event history with MES data, which may address issues related to event consequences and recovery time duration. Although obtaining MES data has been challenging, a solution has been found that allows for the continuation of integration tests between SAP and MES data.

Regarding sustainability data, a Proof of concept (PoC) based on data space and secure multi-party computing (SMPC) is being developed and tested in iterative manner, which will facilitate privacy and confidentiality-preserving approach to exchange data between value-chain companies to enable online / dynamic life cycle assessment (LCA) and enable more accurate data for value-chain LCA calculation purpose to be available. This PoC development is motivated by the requirements of the Stora Enso’s use case where there is a challenge in obtaining data from their suppliers for LCA purpose due to concern on revealing trade secrets. This data exchange architecture will possess the features which address this challenge.

3.3.7.6.2 Contribution to project objectives.

Project Objective	Status at M0	Expected improvement	Planned actions
Obj 1 - Facilitate more than 50% of needed translations in realistic production value networks by autonomous machine-based translation micro-services thus significantly reducing the need for human support.	No translations	Have translations between a few, most relevant technical data format	Identify important translation needs, implement and test them in practice
Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs	No autonomous utilization of translations	Ability to inject translations into existing processes, thus improving interoperability	Design and prototype a technical solution
Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	Several different standards exist and are in use	Select and harmonize the use of different standards	Provide guidelines which standard data models to use and how to use them
Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN	Low capability for transformation management	Digital transformation management effects to ensure smooth transition in work practices.	collaboration with WP10. The maturity and goal assessment developed in WP10 supports clear decision-making and progress tracking across the project lifecycle.
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform			Promote Eclipse Arrowhead

Table 65: contribution to project objectives.

3.3.7.6.3 Use case and engineering costs

Engineering effort and cost reductions are expected to be in the ranges from 30-80 %, depending on if it includes only interoperability of data eliminating manual mappings through automated translations, dynamic integration of information systems via compatible system interfaces, or also interoperability of business processes with complex information exchange sequences.

3.3.7.6.4 Micro-service paradigm

3.3.7.6.4.1 Approach

Use of Eclipse Arrowhead to integrate systems at run-time. The microservices are mainly related to the integration of technical information exchange between numerous different systems.

Microservices provide an interoperable infrastructure for service discovery, user authentication, and authorization as part of production value networks. Special needs regarding information security and integrity, as well as information access management and information sharing concept must be taken into account.

In the Investment project and Supplier service use-cases technical information can be passed to IEP and commercial information is only displayed in the Mockup user-interface.

Additional requirements for the integration architecture are imposed by the combination of Peppol interfaces for commercial processes and Arrowhead interfaces for technical data exchange. The defined solution options will be tested in pilot implementations.

In the dependability sub use-case, the relevance of microservice-based preprocessing was evaluated for different data sources (engineering, CMMS/SAP, and MES) to make data usable for dependability analyses. The conclusion was that there is limited direct utilization in current dependability deliveries, but there are potential future opportunities for modular preprocessing services.

3.3.7.6.4.2 Impact

Standardize how information systems are discovered and bound at run-time. Standardize how authentications and authorizations are performed.

3.3.7.6.4.3 Long-term effect

Improved interoperability, reduced engineering effort.

3.3.7.6.5 Major digital languages

3.3.7.6.5.1 Approach

A set of selected standard data models and digital languages, and clear guidelines how to use them.

3.3.7.6.5.2 Impact

Improved data interoperability, improved quality of data, reduced number of errors.

3.3.7.6.5.3 Long-term effect

Improved usability of data, lower data management and governance costs.

3.3.7.6.6 Autonomous translation

3.3.7.6.6.1 Approach

Automatically invoke translations to provide added interoperability

3.3.7.6.6.2 Impact

Better chances to integrate different systems and new processes.

3.3.7.6.6.3 Long-term effect

More competitive network, improved business collaboration prospects.

3.3.7.7 Cross-use-case Arrowhead-EP observations

Observation	Implication for Arrowhead-EP
All use cases start from fragmented, domain-specific tools and data models.	EPP2 and EPP3 need explicit mapping of data languages, translators and adapters.
Microservices are the common mechanism for moving from isolated toolchains to interoperable PVNs.	EPP4 and EPP5 should include ServiceRegistry, Orchestration, Authorization and data/service contracts.
Autonomous translation maturity differs strongly by use case.	UC1.7 and UC2.7 are the most mature in demonstrated translators; WP9 and UC2.6 still require broader pilot validation.
Cost data is not yet quantified consistently across use cases.	The VCEP-map cost row should be completed later with normalized partner cost data.
Training and education are often implicit rather than formally engineered.	EPP8 should become an explicit deliverable element for each use case.

Table 66: cross-use-case Arrowhead-EP observations.

3.3.7.8 KPIs, Evaluation, and validation process

KPIs and validation in demonstrators is covered in deliverables D9.1, D9.2, D9.3 and D9.4.

3.3.7.9 Final assessment

3.3.7.9.1 KPI Assessment

UC3.9 successfully demonstrated the feasibility of interoperable technical information exchange across process-industry value networks by validating both autonomous translation capabilities and run-time integration mechanisms.

The first KPI, Automated Data Translation, targeted the coverage of technical information through machine-supported translation mechanisms. Starting from a baseline where no automated translation capability was available, the demonstrator achieved 80% translation coverage of the technical information required by the selected industrial use case. The validation confirmed that semantic translation and harmonization mechanisms can effectively integrate engineering, maintenance, and operational datasets while supporting the intended

industrial workflows without significant information gaps. This result demonstrates a substantial reduction in the need for manual mappings and data transformation activities that traditionally characterize information exchange in process-industry ecosystems.

The second KPI, Run-Time Connection of Systems, measured the share of system components integrated dynamically through Eclipse Arrowhead-based interoperability mechanisms. The project achieved the planned target of 30% run-time integration, demonstrating operational connectivity between selected engineering, maintenance, and operational information systems through the semantic Digital Twin platform. The validation showed that distributed information sources can be discovered, connected, and utilized during operation, enabling cross-system information exchange and reducing the dependence on static point-to-point integrations.

3.3.7.9.2 Contribution to project objectives

UC3.9 provided a substantial contribution to the Arrowhead FPVN objectives by addressing interoperability challenges across engineering, operational, maintenance, and sustainability information domains.

For Objective 1, the use case demonstrated the practical applicability of autonomous translation technologies by achieving 80% coverage of the technical information required in the selected industrial scenario, significantly reducing the need for manual data transformations.

Regarding Objective 2, the implementation validated the use of Eclipse Arrowhead as an interoperability infrastructure supporting service discovery, authorization, and run-time integration of distributed systems. The achieved run-time connectivity demonstrated the feasibility of dynamic information exchange across heterogeneous environments.

The use case also contributed to Objective 3 by promoting the use and harmonization of major industrial standards, including ISO 15926, DEXPI, CFIHOS, IEC 61987, and ISO 10303, demonstrating how semantic technologies can support interoperability across different information models.

For Objective 4, UC3.9 supported digital transformation activities through the identification of organizational and technical measures needed to introduce interoperable digital processes, in close collaboration with WP10.

Finally, the use case contributed to Objective 5 by promoting Eclipse Arrowhead and open interoperability approaches as a foundation for sustainable and scalable industrial digital ecosystems.

3.3.7.9.3 Improvement in the engineering process

UC3.9 demonstrated a clear improvement over the baseline engineering approach, where information exchange was largely based on proprietary interfaces, manual mappings, and project-specific integrations. These practices resulted in high engineering effort, limited interoperability, and poor reuse of information across lifecycle phases.

The developed solution introduced a more structured approach based on semantic interoperability, autonomous translation services, and service-oriented integration. Technical information can be represented in machine-understandable formats and exchanged through standardized interfaces, reducing the need for custom integrations and manual data transformations.

The adoption of Eclipse Arrowhead supports the transition from static integrations to dynamic run-time connectivity, while semantic technologies improve information quality, reuse, and

accessibility throughout the lifecycle. As a result, the use case establishes a foundation for more scalable and maintainable production value networks and is expected to reduce engineering effort by approximately 30–80%, depending on the scope of interoperability and process automation achieved.

4. Conclusions

Deliverable D1.3 is dedicated to the presentation of the final outcomes regarding requirements elicitation and their fulfilment, alongside with the consolidation and the evaluation of the project baselines with respect to the horizontal building blocks and the industrial use cases.

Regarding the final requirements assessment, the technological goals of Arrowhead fPVN have been achieved successfully, and its results proved to be highly applicable in multiple industrial contexts. Overall, the evaluation of the requirements fulfilment shows a fairly high weighted rate at 88.4%, which means that the objectives established at M0 have been met to a large extent. Furthermore, the gaps that exist in some use cases are mainly caused by external factors rather than any limitation in the technologies being employed.

The evaluation of the three horizontal building blocks reveals a significant improvement with respect to the baseline and a state-of-the-art solution. WP2 successfully matured the Eclipse Arrowhead ecosystem, delivering a robust service-oriented interoperability framework, which includes features like autonomous service discovery, management, orchestration, authentication, mismatch detection and translator allocation capabilities. In this regard, WP2 has contributed to the development of a solution for future federation of digital ecosystems and production value networks.

WP3 established a comprehensive framework for semantic interoperability based on the concept of Interoperability Design Objects (IDOs), enabling the systematic analysis, alignment and harmonisation of heterogeneous industrial standards and information models. The results demonstrate that semantic interoperability can be addressed in a scalable and structured manner, creating the necessary foundation for autonomous translation and cross-domain interoperability.

The objective of WP4 was to implement and validate several autonomous translation solutions based on ontology-, AI-, and model-driven approaches. Through developing and validating seventeen prototype translators that have been tested both independently and embedded into the Eclipse Arrowhead framework within a number of realistic industrial use cases, the feasibility of an automated interoperability solution was proven, fulfilling the requirements and with translation accuracies and applicability exceeding initial expectations.

The industrial use cases provided a wide variety of validation settings that include several different application areas, such as battery lifecycle management, human-machine interactions, aerospace manufacturing processes, aircraft health management systems, municipal infrastructure, knowledge graph digital twin technologies, and process industry information exchange. The assessment of the requirements fulfilment and the KPIs evaluation allow to conclude that the concepts proposed by Arrowhead fPVN proved themselves to be transferrable across a number of industries and could be effectively used to implement interoperability, semantic integration, autonomous translation, and other aspects related to service-oriented digital ecosystems.

The KPI evaluation in the use case baselines show that there is significant improvement compared to the situation initially observed at the beginning of the project. The use cases

showed significant improvements in the areas of interoperability, automation, engineering productivity, information accuracy, and digital integration. In several cases, the expected reductions in engineering effort and integration complexity were achieved or exceeded through the adoption of SOA, semantic interoperability mechanisms and autonomous translation solutions.

In particular, the most successful results were attained within use cases 2.6, 1.7, 2.7, and 3.9, where almost all initial requirements were fulfilled, while the combination of the three building blocks led to considerable improvement in interoperability capabilities, engineering efficiency, and automation levels. All the other use cases also demonstrated substantial benefits and validated the technological concepts.

Moreover, the use cases provided a thorough validation for the project objectives. With regard to Objective 1, related to the automation of data translations within production value networks, this was done by realizing and deploying several autonomous translation mechanisms within different industry sectors. Objective 2, related to the service-oriented deployment of translators and their autonomous use, was successfully validated thanks to the widespread adoption of Eclipse Arrowhead and effective management of the distributed interoperable services. Objective 3, related to industrial data model harmonization and evolution, was addressed via semantic analysis, standards alignment activities and realization of heterogeneous information model integration. Regarding Objective 4, which was related to methodologies for adopting translators and interoperability technologies, the validation process took place thanks to the engineering, integration and digital transformation efforts carried out in the use cases. Objective 5, finally, was addressed through the continuous evolution, validation and promotion of the Eclipse Arrowhead ecosystem.

Overall, according to the results of the final baseline assessment, the vision of Arrowhead fPVN regarding autonomous interoperability of production value networks has successfully been implemented as a set of validated technological and methodological solutions. Consequently, the project managed to establish a solid technological base that includes service-oriented architecture, semantic interoperability, and autonomous translations thus creating the foundation for further interoperable ecosystems.

5. Annex

1. Requirements Matrix: external file “AfPVN WP1 - Requirements elicitation and evaluation Y3.xlsx”
2. Template of the Building block baseline survey: external file “AfPVN WP1 Survey.WP2.3.4 Survey Template.v2.0.docx”
3. Template of the use case baseline survey: external file “AfPVN WP1 Survey.Use Case Survey Template.v2.0.docx”

6. Revision history

6.1 Contributing and reviewing partners

Contributions	Reviews	Participants	Representing partner
TOC definition and first version		Elisa Londero, Gianvito Urgese	ETH, Polito
Deliverable contents		Elisa Londero, Gianvito Urgese	ETH, Polito
Deliverable contents		WP2, WP3, WP4 leaders	
Deliverable contents		Use case leaders	

6.2 Amendments

No.	Date	Version	Subject of Amendments	Author
1	06/04/2026	0.1	Document structure and intro	Elisa Londero
2	13/04/2026	0.2	Requirements section	Elisa Londero, Gianvito Urgese
3	20/05/2026	0.5	Use cases baselines	Elisa Londero, Gianvito Urgese
4	31/05/2026	0.6	Use cases baselines	Elisa Londero
5	06/06/2026	0.7	Use cases baselines	Elisa Londero
6	08/06/2026	0.8	Requirements section Use cases baselines	Elisa Londero
7	08/06/2026	1.0	Final update for internal review	Elisa Londero, Paolo Azzoni

6.3 Quality assurance

No	Date	Version	Approved by
1	09.06.2026	1.0	Jerker Delsing