



Document title: Arrowhead fPVN Deliverable D1.2

Version
1.1

Status
Final
Author
Elisa Londero

Date
2025-06-11

Contact
elisa.londero@eurotech.com

Deliverable D1.2

Requirements and base line update 1

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 D1.2 of the Arrowhead fPVN project, part of the **EUROPEAN PARTNERSHIP**.

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 will further collect and summarize common technology and use-case baselines with respect to the project objectives, to provide a consolidated way of validating and verifying the project advancement and objectives fulfillment.



ChipsJU project 101111977 - Arrowhead FPVN
Project Coordinator: Professor Jerker Delsing | Luleå University of Technology



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.....	4
2.1	The nature and role of requirements	4
2.2	The elicitation process.....	5
2.3	The requirements matrix and its future use and maintenance.....	6
2.4	Requirement elicitation and evaluation: second cycle.....	7
3.	The baselines	11
3.1	The concept of “baseline”	11
3.2	Building blocks baselines.....	12
3.2.1	Micro-service paradigm	12
3.2.2	Major digital languages.....	18
3.2.3	Autonomous translation	21
3.3	Use case baselines	29
3.3.1	WP6 - Use-case #1.6 – Automotive Battery Innovation fPVN.....	30
3.3.2	WP6 - Use-case #2.6 – Humans in the interoperable System.....	37
3.3.3	WP7 - Use-case #1.7 – Interoperable intelligent management of production lines: Towards Model-based Enterprise.....	39
3.3.4	WP7 - Use-case #2.7 – Aircraft Health Management System (AHMS) for Trend Monitoring, Predictive Maintenance and Fleet Operations & Maintenance Simulation .	50
3.3.5	WP9 - Use-case #1.9 – Pump Station Engineering.....	61
3.3.6	WP9 - Use-case #3.9 – Interoperability for technical information exchange in process industry.....	66
4.	Collaboration with other technical WPs.....	73
5.	Conclusions	75
6.	Annex	76
7.	Revision history.....	77
7.1	Contributing and reviewing partners.....	77
7.2	Amendments.....	77
7.3	Quality assurance	77

1. Introduction

This deliverable reports the results achieved during the second 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 M13-M24.

In the first part, the deliverable provides a recap about the concept of requirements, illustrating the process adopted for the requirements elicitation and for their evaluation. It further provides an updated list of the identified requirements and a complete analysis of the status at M24.

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

The final evaluation of requirements satisfaction and the final version of the baselines will be presented in Deliverable D1.3.

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 will be 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 analyzing 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 fulfillment of requirements has been evaluated. The new 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 will also serve 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

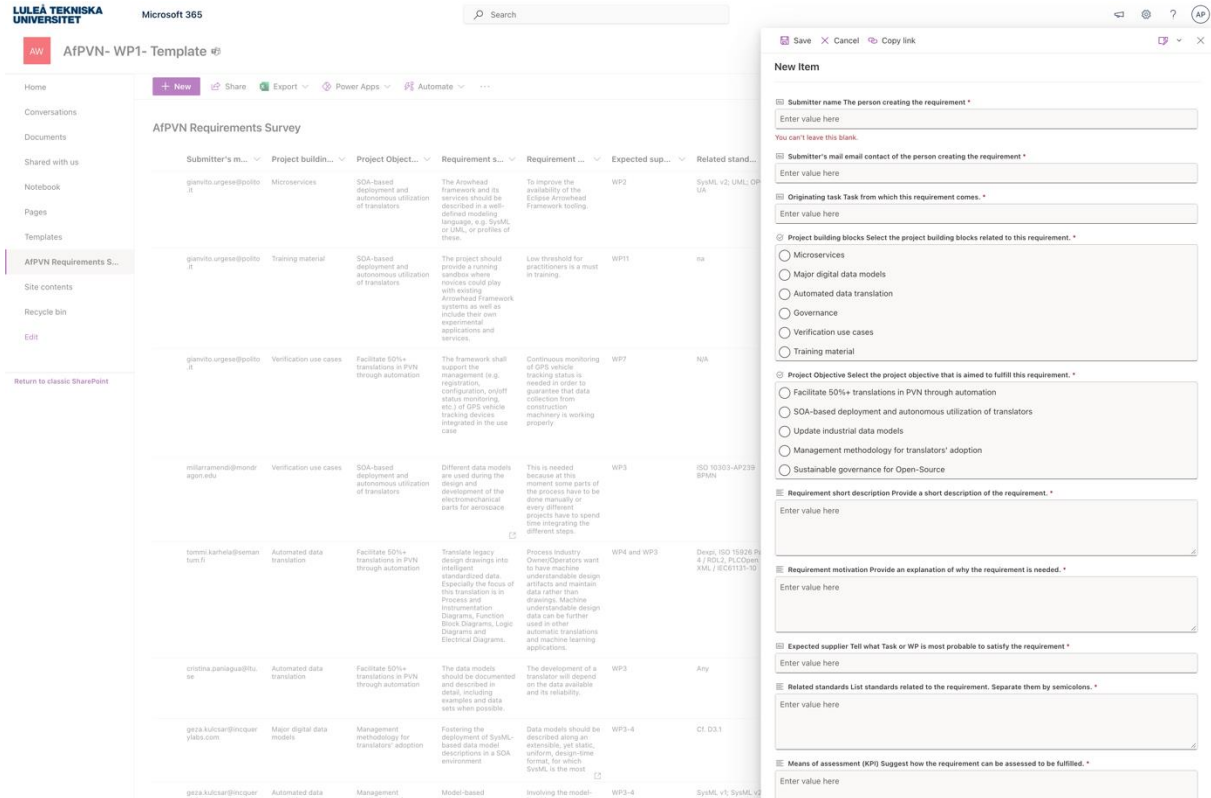
reasonable rule to ensure we don't miss the main targets of the projects, avoiding getting lost in an endless list of technical requirements with excessive granularity.

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 fulfill the requirement.
- **Related standards:** the list of relevant standards related to the requirement.
- **Means of assessment (KPI):** explain how the requirement fulfillment can be assessed.
- **Validating Use Case:** the use case in which the requirement will be validated.
- **Validation Means:** explanation of how the use case will validate the requirement.
- **Y1 Fulfillment (%):**1st year assessment of the fulfillment of the requirement in percentage.
- **Y1 Evaluation:** explanation of the fulfillment percentage achieved in the 1st year.
- **Y2 Fulfillment (%):**2nd year assessment of the fulfillment of the requirement in percentage.
- **Y2 Evaluation:** explanation of the fulfillment percentage achieved in the 2nd year.
- **Y3 Fulfillment (%):**3rd year assessment of the fulfillment of the requirement in percentage.
- **Y3 Evaluation:** explanation of the fulfillment 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 name	Project building blocks	Project Objectives	Requirement status	Requirement description	Expected supplier	Related standards
giarvito.urgesa@polito.it	Microservices	SOA-based deployment and autonomous utilization of translators	Not started	The Arrowhead framework and its services should be described in a well-defined modeling language, e.g. SysML, or UML, or a subset of these.	WP3	SysML v2, UML, CP
giarvito.urgesa@polito.it	Training material	SOA-based deployment and autonomous utilization of translators	Not started	The project should provide a training sandbox where services could interact with existing Arrowhead Framework systems as well as include their own experimental applications and services.	WP1	na
giarvito.urgesa@polito.it	Verification use cases	Facilitate 50%+ translations in fPVN through automation	Not started	The framework shall support the management (in g. registration, configuration, output status monitoring) etc. of GPS vehicle tracking devices integrated in the use case.	WP3	N/A
milarramendi@mendiagon.eu	Verification use cases	SOA-based deployment and autonomous utilization of translators	Not started	Different data models are used during the design and development of the electromechanical parts for aerospace.	WP3	ISO 10303-AP238 BPMN
tommi.kahkonen@vaimenturn.fi	Automated data translation	Facilitate 50%+ translations in fPVN through automation	Not started	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.	WP4 and WP5	Devic. ISO 15926 P4 / R02, PCCOpen, ANL / SC04131-30
crstina.pantagada@polito.it	Automated data translation	Facilitate 50%+ translations in fPVN through automation	Not started	The data models should be documented and described in detail, including variables and data sets when possible.	WP3	Any
geca.fulcarin@inroper.yaleo.com	Major digital data models	Management methodology for translator adoption	Not started	Fostering the deployment of SysML-based data model descriptions in a SOA environment.	WP3-4	CF D3.1
geca.fulcarin@inroper.yaleo.com	Automated data translation	Management methodology for translator adoption	Not started	Model-based annotations	WP3-4	SysML v1, SysML v2

Figure 1 - The Sharepoint web site for requirements elicitation.

The requirements matrix mentioned in the Document of Action is and will be 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 will be cyclical and will allow the project partners to review, refine, update, extend the existing requirements and add new one: this will be 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 will evaluate the consolidation of requirements and their level of fulfillment.

2.3 The requirements matrix and its future use and maintenance

Deliverable D1.1 and D1.2 contain the requirements matrix as of end of December 2023 and June 2025 respectively (see Annex 1). This defines a first baseline and starting point for the project, providing a preliminary indication of the expectations for the years to come. As

anticipated, the Matrix has been and will be continuously updated to guide and follow at the same time the evolution and progress of the project. In the first elicitation cycle (M1-M6), we have the preliminary set of requirements and the related evaluation in terms of fulfillment at the end of the first project year. During the following 18 months, we continuously updated the Matrix until the second step of fulfillment evaluation at M24. The final cycle of consolidation and fulfillment evaluation will be at the end of the project.

2.4 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.

Regarding the evaluation of requirements fulfilment at the end of year 2, the analysis of the 50 requirements reveals a mean fulfilment level of **58%** (more than half of the requirements), with values ranging from **20% to 100%**. This range reflects the natural progression and maturity differences between work packages and use cases, particularly in a project of this complexity, where technical developments are interdependent and iterative.

The analysis identifies three major categories in terms of requirement fulfillment:

- **Advanced Progress:** a significant portion of the requirements exceed **70% fulfilment**, often in cases where the architecture and design have been finalized, and partial implementations are underway. These requirements are typically linked to foundational infrastructure, semantic data handling, and architectural alignment with Eclipse Arrowhead framework. This confirms that the project has moved past preliminary phases in several strategic areas.
- **Moderate Progress (40–70%):** this range includes the majority of requirements and represents use cases and WPs where prototyping, technology alignment, or partial integration has started. These requirements often involve coordination among partners, adaptation to evolving specifications, or waiting on technological components like the WP4 translators or AI modules. The high percentage of requirements in this segment is aligned with the project timing.
- **Low Fulfilment (<40%):** a smaller group of requirements still show limited progress. This is generally due to factors such as:
 - Dependence on external components not yet delivered (e.g., translation engines, external data models).
 - Scheduled for later implementation in the work plan (e.g., integration and validation in year 3).
 - Complexity of aligning different standards, systems, or vertical domains.

This evaluation reflects a healthy and realistic development pace in line with a three-year project, with year three expected to focus heavily on integration, validation, and consolidation across the different use cases and technologies.

3. The baselines

This chapter introduces the concept of “baseline”, provides a recap of the methodology adopted to analyze 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 current 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, have been currently updated with this deliverable (D1.2) and will be finalized in D1.3. The continuous updating of the baselines allows us to keep them consistent with the evolution of the project.

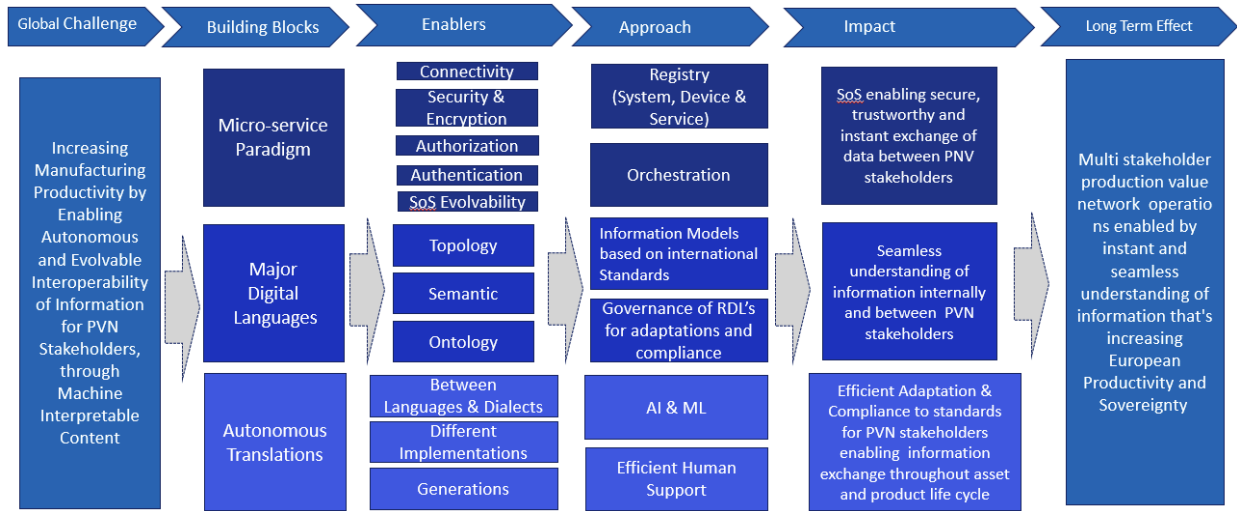


Figure 2 - Approach and Ambition outline in logical steps (from DOA).

3.2 Building blocks baselines

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

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

As anticipated in D1.1, these baselines are in a stable state and will be consolidated during the last year of the project (D1.3).

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.

Arrowhead core system	Capability	TRL M0
ServiceRegistry v5.0	Registry for available service within a local cloud	7
Orchestration v5.0	Responsible for distribution of orchestration rules to service consumers	7
Authorisation v5.0	Primary system for checking authentication and authorization rule, ensuring edge security	7
SystemRegistry v5.0	Registry for available systems within a local cloud	6
DeviceRegistry v5.0	Registry for available devices within a local cloud	6
CertificateAuthority v5.0	Responsible for distribution of Eclipse Arrowhead compliant X.509 certificates	6
On-boarding v5.0	Managing the on-boarding procedure to ensure only authorized devices, systems, and services will allow into a local cloud.	6
DataManager v5.0	Primary data storage from service producers in a local cloud	7
Gatekeeper and Gateway v5.0	Secure interoperability in-between Eclipse Arrowhead local clouds – control and data channel	7
PlantDescription v5.0	Multi-view descriptor for plant configuration mapping	6
EventHandler v5.0	Providing authorized publish-subscribe type, generic event handling capabilities	7
Translation v5.0	Translating between HTTP-REST, CoAP, MQTT and Web socket protocols	6
FiWare adaptor	Enabling interoperability from/to Arrowhead systems to/from the FiWare broker system.	5
OPC-UA adaptor v5.0	Autonomous registration of OPC-UA services with Eclipse Arrowhead	5
Z-wave adaptor v5.0	Autonomous registration of Z-wave services with Eclipse Arrowhead	5
Modbus TCP adaptor v5.0	Autonomous registration of Modbus-TCP services with Eclipse Arrowhead	5
ISO 10303 adaptor	Arrowhead adaptor enabling integration to ISO 10303 compliant tools like PLM, CAD	7 for specific tools
WEB of Things adaptor	Adaptor enabling Web of Things to be integrated to Eclipse Arrowhead	5

Table 4 – 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.

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.

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 5 summarizes the draft KPIs defined for the autonomous translation evaluation and validation. They will be finalized in the next project phase (D1.3).

Table 5: Micro-service paradigm KPIs

Key Performance Indicators (KPI)		
Name	KPI Description	Means of assessment
2.1.1	Cookbook presence	Review, approve and use the cookbook
2.1.2	Number of patterns present in the cookbook.	Counting of implemented /verified patterns
2.1.3	Governance process	Percentage of WP defined services under version control.
2.2.1	TRL level of Eclipse Arrowhead framework components.	Assess percentage of components that have reached M36 TRL level.
2.2.2	Number of reports on interconnection analyses.	<TBD>
2.2.3	Number of available translators in Eclipse Arrowhead.	Counting
2.2.4	Existence of a process for Contracting and Invoicing	Review, approval and verification of implementation.
2.2.5	Existence of access control to components in the Eclipse Arrowhead framework.	Verification of a purchase process.
2.3.1	Presence of detection of non-interoperability.	Demonstration of function.
2.3.2	Allocation of translators in the Eclipse Arrowhead framework.	Demonstration of function.
2.3.3	Translators use IA core systems	Demonstration of function.



Document title: Arrowhead fPVN Deliverable D1.2

Version
1.1

Status
Final

Date
2025-06-10

	in Eclipse framework	Arrowhead	
--	-------------------------	-----------	--

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 3.

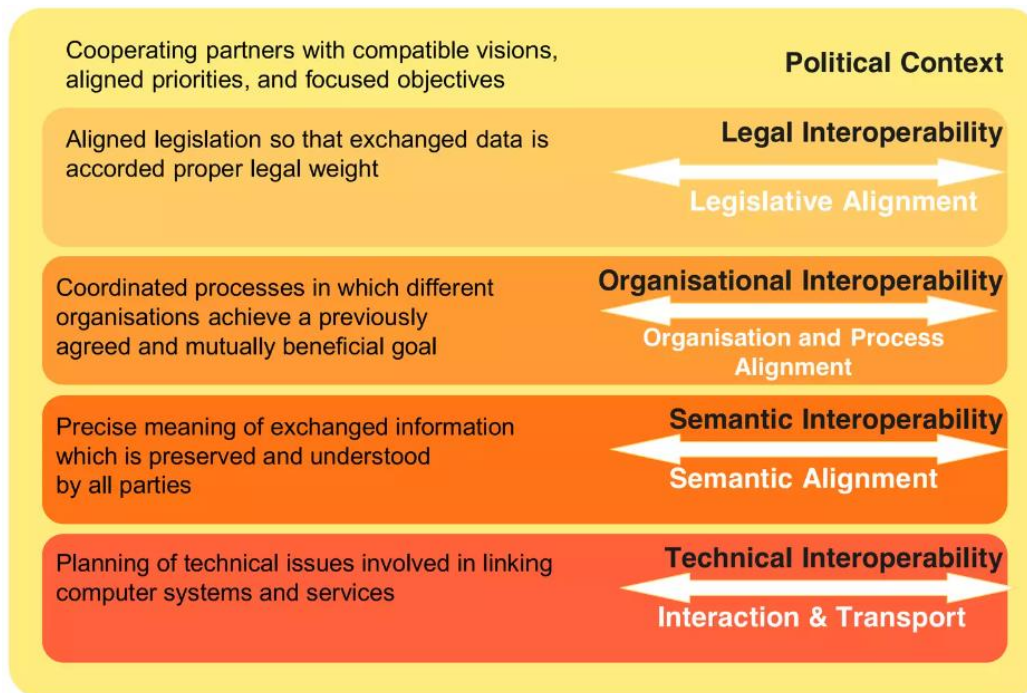


Figure 3: Levels of interoperability described in the European Interoperability Framework

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 19526 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 6 summarizes the draft KPIs defined for the autonomous translation evaluation and validation. They will be finalized in the next project phase (D1.3).

Table 6: Major digital languages KPIs

Key Performance Indicators (KPI)		
Name	KPI Description	Means of assessment
3.1	Number of collected and analyzed standards regarding Automotive fPVNs	Inventory (Reports and documents in the Owncloud)

3.2	Number of collected and analyzed standards regarding Aerospace fPVNs	Inventory (Reports and documents in the Owncloud)
3.3	Number of collected and analyzed standards regarding Process Industry fPVNs	Inventory (Reports and documents in the Owncloud)
3.4	Number of collected data model formats regarding use case partners	Inventory (Reports and documents in the Owncloud)
3.5	Number of analyzed data model formats	Inventory (Reports and documents in the Owncloud)
3.6	Number of analyzed communication protocols	Inventory (Reports and documents in the Owncloud)
3.7	Number of analyzed modelling and knowledge representation languages	Inventory (Reports and documents in the Owncloud)
3.8	Number of standard comparisons	Inventory (Reports and documents in the Owncloud)
3.9	Number of found synergies	Inventory (Reports and documents in the Owncloud)
3.10	Number of found interoperability problems	Inventory (Reports and documents in the Owncloud)

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-xmldata11-1-20120405/> (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 7 summarize the planned actions regarding each WP4 objective.

Table 7: Planned actions regarding WP4 objectives.

Objective	Actions Planned
<i>Provision of data set for translation development and early assessment of translation quality.</i>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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.

	<ul style="list-style-type: none"> - 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>	<ul style="list-style-type: none"> - 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.

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 4 summarizes the expected timeline.

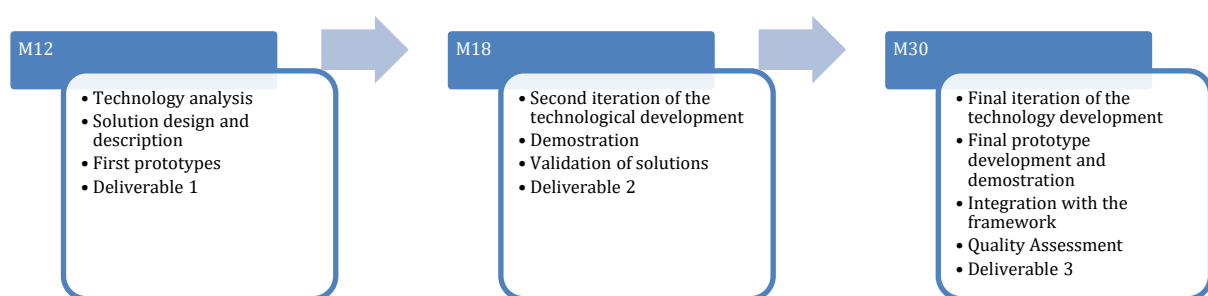


Figure 4: Autonomous translation timeline

3.2.3.3.2 KPIs, Evaluation, and validation process

Table 8 summarizes the draft KPIs defined for the autonomous translation evaluation and validation. The KPIs are explained in more detail after the Table 8. They will be finalized in the next project phase (D1.3).

Table 8: Autonomous translation KPIs

Key Performance Indicators (KPI)		
Name	KPI Description	Means of assessment
4.1	Number of translation solutions provided during the project	Inventory (File in the Owncloud)
4.2	Range of data-models, standards and protocols translated.	Inventory and demonstrators
4.3	Translation accuracy	Measure the correctness and precision of translations (Metrics)
4.4	Resource utilization on the use cases.	Monitor the utilization of translation resources (translators, tools, and software).
4.5	Number of scientific publications in the work package	Number of register publications.
4.6	Evolution of the prototypes available	Tools and prototype version (version id)
4.7	Integration of the solutions on the Eclipse Arrowhead Framework	Number of solutions integrated in the framework

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.
- 4.4 - Resource utilization in use cases. The solutions will be provided to the use cases, and this Key Performance Indicator (KPI) will quantify how useful the translation solutions are for these use cases. It will measure the percentage of solutions and resources used.

- 4.5 - 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.
- 4.6 - Evolution of the available prototypes. The prototypes will evolve during the project's iterations, with new versions being presented each year.
- 4.7 - 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.

3.3 Use case baselines

This section reports results of the baseline analysis at the end of the second 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.

With respect to D1.1, use case 1_8 “System-Driven Modularization and Digitalization for Offshore Renewables” was canceled, while use case 2.9 is merging with 3.9.

At the end of the second year, the definition and refinement of the use case baselines show tangible and consistent progress towards the finalization. In particular, several use cases have evolved from early conceptualization to delivering stable and detailed baseline versions. Use cases 1.6 and 1.7 have provided a new stable version, building on their already mature definitions from the first year. Use case 2.7 has maintained its solid baseline with minor enhancements.

We highlight also the advancements that have been made in use cases that were still under development at the end of the first year. Use cases 1.9 and 3.9 have moved from a stage of exploratory analysis to a more consolidated status that can be defined as “formative”—signaling that a foundational structure is now in place, even if some components are still under elaboration. Use case 2.6 remains still in a formative phase and, in the next months, will have to speed-up the analysis and provide a stable version of the baseline.

Overall, this positive development confirms that most use cases are converging towards a coherent baseline consolidation. The final year of the project will be dedicated to completing the refinement and validation of the remaining use case baselines, to reach a consolidated state that will allow the evaluation of the project results towards the baselines themselves. The final version of the baselines will include also the analysis of the engineering process and the related costs, two fundamental aspects to evaluate the impact of the technologies designed and developed in the project.

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 5:

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.

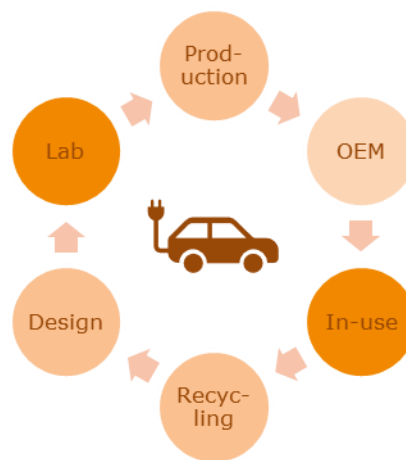


Figure 5: UC-1.6 Full life cycle of EV batteries.

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

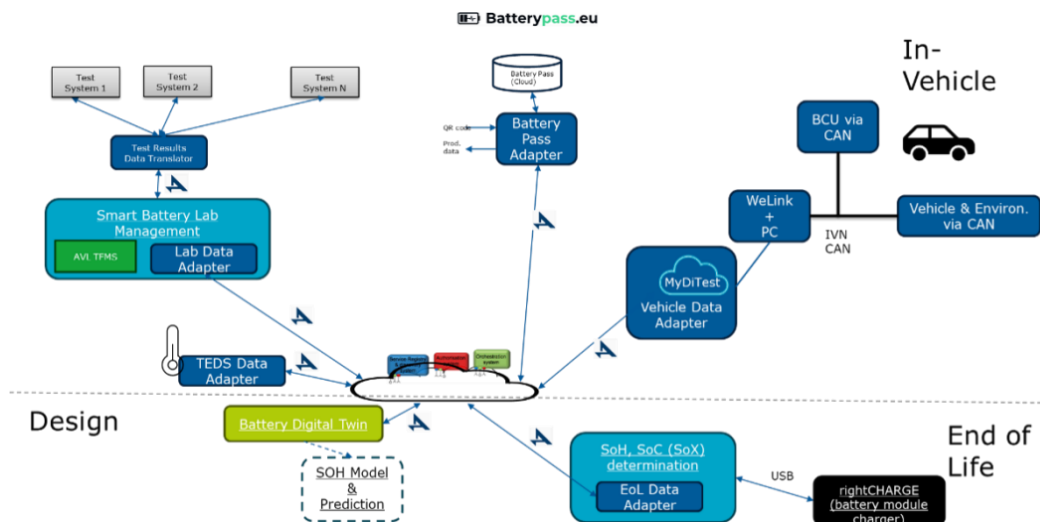


Figure 6: Updated UC-1.6 Architecture

Table 9 lists functional blocks planned for prototypical implementation in the project (see Figure 5)

Table 9: UC 1.6 Functional Blocks planned for prototypical implementation.

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 ¹)
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

¹¹ <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

Table 10: UC 1.6 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 9
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 9
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

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 6.

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

Table 11: UC 1.6 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

3.3.1.2.5 Key Performances Indicators

Table 12: UC 1.6 KPIs

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

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 6)
- 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

Table 13: UC 1.6 - 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
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%
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
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
Obj 5 - Established a	n/a	-	-	-	-	-

sustainable governance of open-source architecture and implementation platform						
--	--	--	--	--	--	--

3.3.1.3.3 KPIs, Evaluation, and validation process

See Table 12.

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

3.3.2.1 Baseline summary

Industry is not just machines! The future of production lies in the efficient and effective collaboration between humans and machines. However, to facilitate a smooth holistic interoperability, we must optimize the interaction between humans and machines. The challenges to be addressed and overcome are three-fold. First, we need to adjust the processes to the workers’ needs, thereby creating a more intuitive, less error-prone or stressful and overall quicker way of collaborative human-robot interaction. Second, automation pressure is high in the European car manufacturing industry. At the same time, there is diversity due to different car manufacturers focusing on collaborative industrial robots in different production areas, such as logistics, assembly, quality inspection, and dispensing. This requires a different, and a differently intensive human-robot interaction (HMI); it also requires different repertoires of commands and feedback signals within the interaction (both in terms of vocabulary size and complexity). Third, workers of European car manufacturers form international teams. Many languages are involved. Sound and voice interfaces have to cope with this degree of accents, languages, etc. – and, moreover, need to be prepared to noisy environments.

The analysis of this baseline is still ongoing.

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

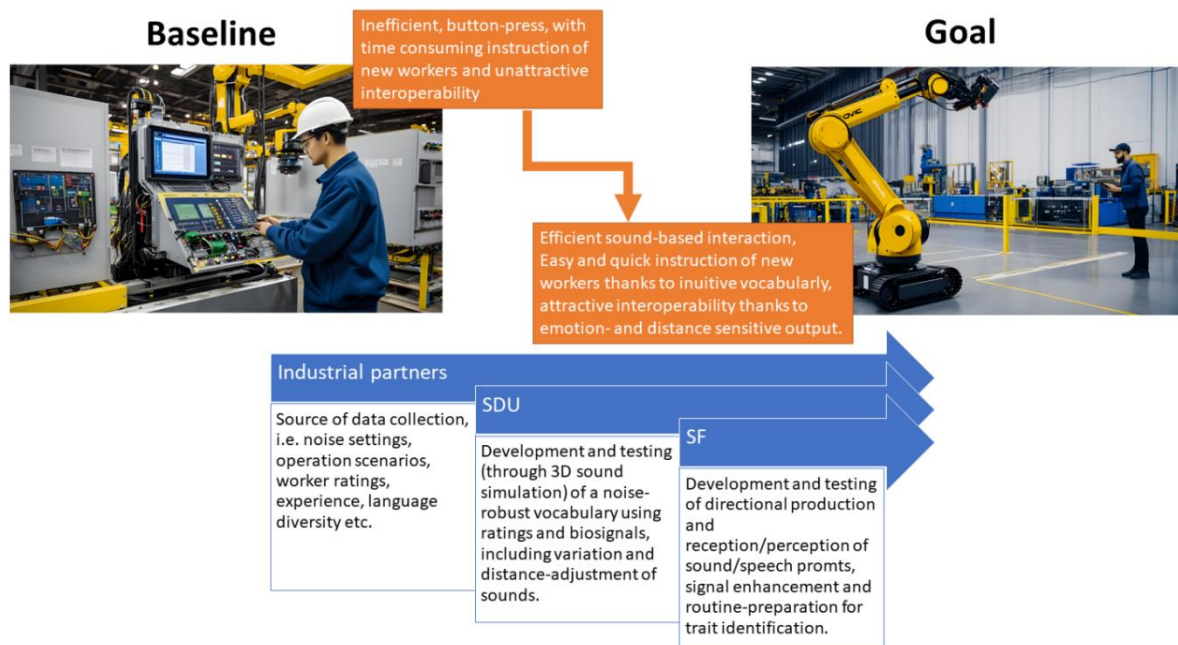


Figure 7 - Use case architecture.

3.3.2.1.2 Objectives that are linked to the Arrowhead fPVN objective

Table 14: UC-2.6 Arrowhead fPVN objective

Project Objective	UC Contribution
<p>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.</p>	<p>(1) A more intuitive interoperability “vocabulary” makes Human-Robot-Interaction more acceptable and attractive both for employers and employees (workers) and, moreover, accessible to age groups who have not been in touch with robots (i.e. elderly workers).</p> <p>(2) A more noise-robust interoperability “vocabulary” can help make robots applications run in tough environmental settings where places are noisy and a distance between human worker and robot is hard to avoid (i.e. no button press is possible).</p>
<p>Obj 2 - Microservices/SOA enabling of dynamic deployment and autonomous utilization of information translation in PVNs</p>	
<p>Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.</p>	
<p>Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN</p>	
<p>Obj 5 - Established a sustainable governance of open-source architecture and implementation platform</p>	

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

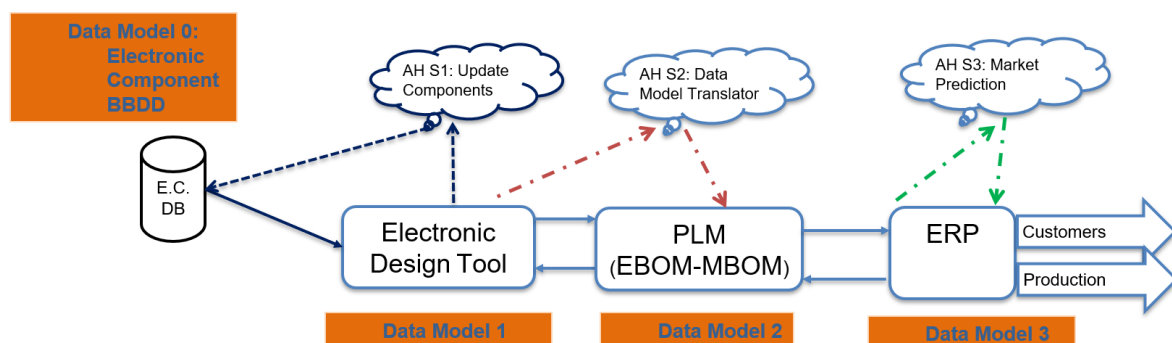


Figure 8: UC 1.7 Architecture

The use case represented in Figure 8 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

Table 15: UC 1.7 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	The methodology to be used during the product lifecycle will follow the MBSE methodology based on Arrowhead services

autonomous translation within a PVN	
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	The translation services and intelligent services applied for optimising the sales department decisions will be implemented in the Arrowhead platform

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

Current state: The requirements have been defined and at this moment, the research group is working in the functional design.

3.3.3.3.2 Enablers

The use case will follow the suggested Arrowhead – EP process. Once the Procurement and Engineering phase is completed, a Deployment phase will be launched but initially, this deployment will be installed in a parallel system that Sener is using as a Validation platform. Additionally, at this point we believe that the phase of Decommissioning and Recycling make no sense in this use case.

3.3.3.3.3 Arrowhead-EP of the Use Case

Not defined yet.

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

Not defined yet.

3.3.3.3.5 How the Arrowhead-EP allows matching the WP2 objectives

Not defined yet.

Arrowhead-EP Objective	Contribution of the Arrowhead-EP of the Use Case
Obj. 1 - The change from design time to run time engineering	Not identified
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.

<p>Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation</p>	<p>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.</p>
<p>Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle</p>	<p>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.</p>

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.

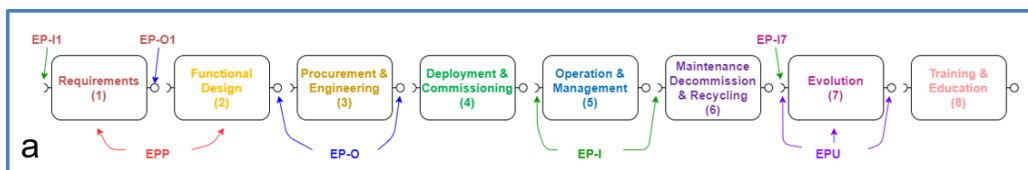
3.3.3.3.7 Toolchain automation

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.

3.3.3.4 Use case and engineering costs



Engineering Costs:

Requirements: 3 PMs

Functional Design: 7 PMs

Procurement & Engineering: 20 PMs

Deployment & Commissioning: 10 PMs

Operation & Management: 10 PMs

Maintenance, Decommissioning & Recycling: 5 PMs

Evolution: 5 PMs

Training & Education: 5 PMs

3.3.3.4.1 Standardization requirements

Table 16: UC 1.7 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

3.3.3.4.2 Key Performances Indicators

Table 17: UC 1.7 KPIs

Key Performance Indicators (KPI)			KPI Fulfilment & Evaluation (expected year results)		
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%
Market Prediction	Electronic Componentes market prediction service	Intelligent service able to predict market of electronic components deployed	10%	60%	100%
SOA	Service Oriented Architecture	The use case will be deployed in	10%	40%	100%

Model		a SOA architecture			
-------	--	--------------------	--	--	--

3.3.3.5 Beyond the baseline

3.3.3.5.1 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.

3.3.3.5.2 Contribution to project objectives

Table 18: UC 1.7 - Contribution to project objectives

Project Objective	Status at M0	Expected improvement	Planned actions	Status at M12	Status at M24	Status at 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.	Not automatic translation	Automatize process by using automatic translators	Develop or use services/tools able to translate data models such as DITAG	Translators Tools/Services defined	Translators Developed	Translators Integrated in the use case
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
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
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
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.

3.3.3.5.3 Use case and engineering costs

See previous section.

3.3.3.5.4 Micro-service paradigm

3.3.3.5.4.1 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.

3.3.3.5.4.2 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.

3.3.3.5.4.3 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.

3.3.3.5.5 Major digital languages

3.3.3.5.5.1 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.

3.3.3.5.5.2 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.

3.3.3.5.5.3 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.

3.3.3.5.6 Autonomous translation

3.3.3.5.6.1 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.

3.3.3.5.6.2 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.

3.3.3.5.6.3 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).

3.3.3.5.7 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.

3.3.3.6 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.

3.3.4 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:

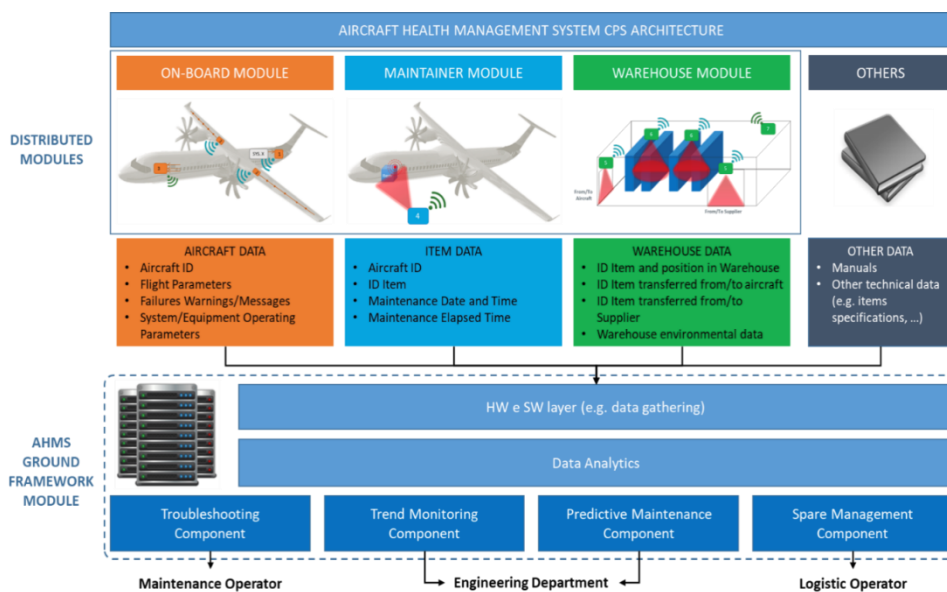


Figure 9 - The use case architecture.

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

Table 19: UC 2.7 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

<p>Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN</p>	<p>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.</p>
<p>Obj 5 - Established a sustainable governance of open-source architecture and implementation platform</p>	<p>Currently n/a</p>

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.

3.3.4.2.4 Standardization requirements

Table 20: UC 2.7 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	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	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

3.3.4.2.5 Key Performances Indicators

Table 21: UC 2.7 KPIs

Key Performance Indicators (KPI)		
Name	KPI Description	Means of assessment
Data exchange capability	<ol style="list-style-type: none"> 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: ?]	<ol style="list-style-type: none"> 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.
Data exchange model in accordance with S5000F	Compliance between data exchanged through S5000F message and defined business rules [Y/N - M0: no - M36: ?]	the compliance of the exchange data with a defined set business rules will be verified
Data integrity	Compliance between data exchanged through S5000F message and raw data [Y/N	the data exchanged through S5000F messages will be tested with respect to raw data

	- M0: no - M36: ?]	
Orchestration of the 3 service providers and 1 service consumer	Adoption of SOA oriented solution [Y/N - M0: no - M36: ?]	The architecture of the adopted solution must be based by design on the SOA paradigm and microservice architecture

3.3.4.3 Beyond the baseline

3.3.4.3.1 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.

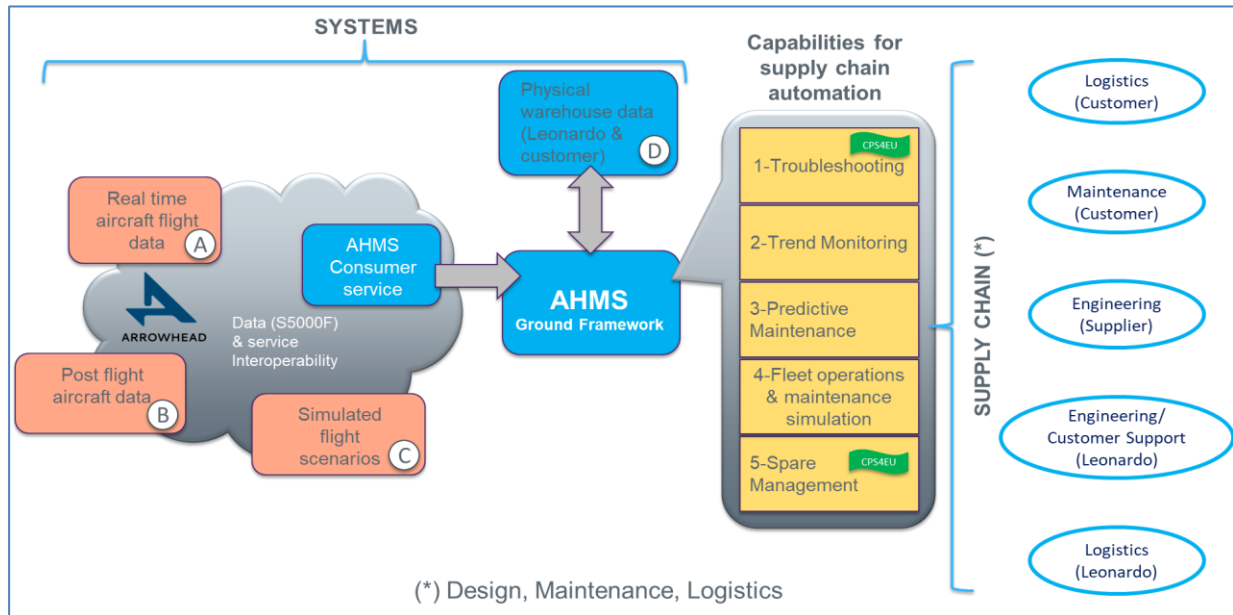


Figure 10 - The Arrowhead local cloud, the interconnected systems, and their capabilities.

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.

3.3.4.3.2 Contribution to project objectives

Table 22: UC 2.7 - 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.	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.
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 architecture	Adoption of Eclipse Arrowhead Framework

<p>Obj 3 - Update proposals for major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.</p>	<p>No standard adopted</p>	<p>Data shall be exchanged based on S5000F and S-SERIES data model. We don't plan to update the standard.</p>	<p>Adoption of S5000F</p>
<p>Obj 4 - Digital transformation management methodology for the introduction of seamless and autonomous translation within a PVN</p>	<p>Low level of automation, mainly based on a human operator.</p>	<p><u>The Orchestration module of AHMS</u> 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.</p>	<p>Development of an orchestration module for the automation of maintenance process operations.</p>
<p>Obj 5 - Establish a sustainable governance of open-source architecture and implementation platform</p>	<p>Currently N.A.</p>		

3.3.4.3.3 Micro-service paradigm

3.3.4.3.3.1 Approach

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

3.3.4.3.3.2 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

3.3.4.3.3.3 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.

3.3.4.3.4 Major digital languages

3.3.4.3.4.1 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.

3.3.4.3.4.2 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.

3.3.4.3.4.3 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.

3.3.4.3.5 Autonomous translation

3.3.4.3.5.1 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.

3.3.4.3.5.2 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.

3.3.4.3.5.3 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.

3.3.4.3.6 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.

3.3.5 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 11 - The initial engineering process

The initial overview of the engineering process (Figure 11) 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 12) 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.

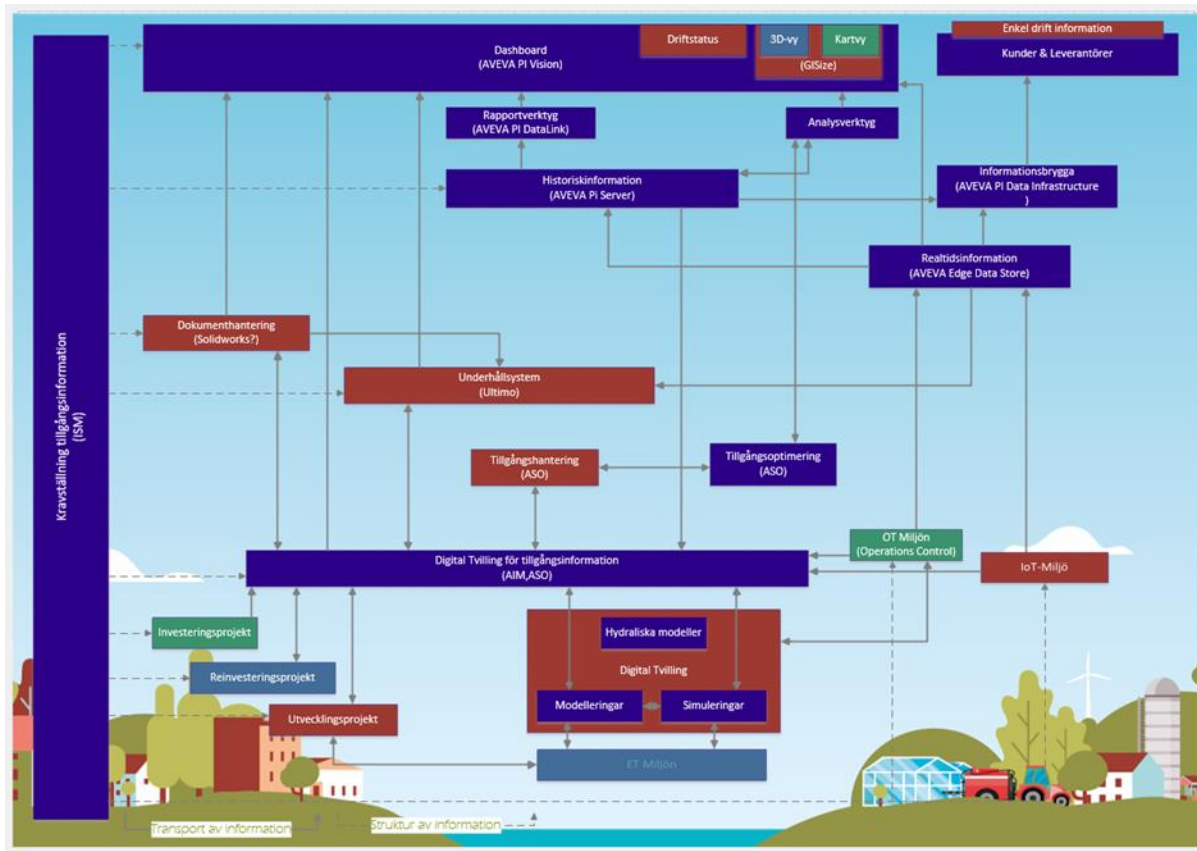


Figure 12 - Outline for the systems that are operating (or planned for the near future) at Roslagsvatten.

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.

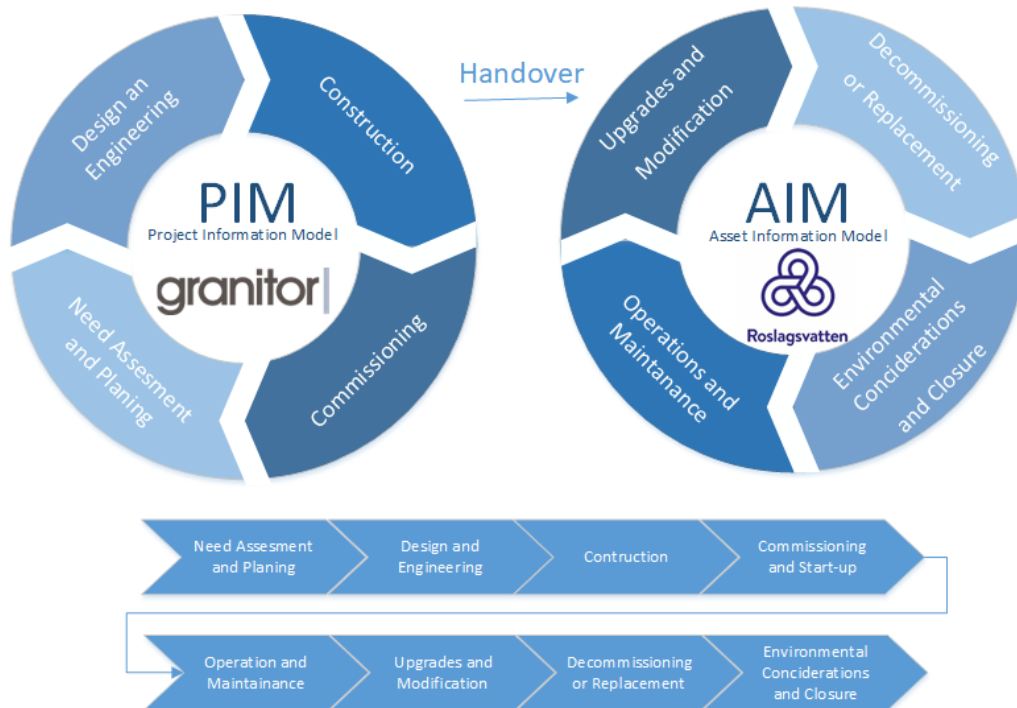


Figure 13- Use case architecture.

3.3.5.1.2 Objectives that are linked to the Arrowhead fPVN objective

Table 23: UC 1.9 - Contribution to project objectives

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	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.

within a PVN	
Obj 5 - Established a sustainable governance of open-source architecture and implementation platform	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.

3.3.5.2 Micro-service paradigm

3.3.5.2.1 State of the art

Currently, no micro-service architecture is implemented.

3.3.5.2.2 Enablers

RSV is actively developing a micro-service architecture based on established standards and proven technologies. The focus is on interoperability and scalability.

3.3.5.3 Major digital languages

3.3.5.3.1 State of the art

No major digital languages are currently in use.

3.3.5.3.2 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.

3.3.5.4 Autonomous translation

3.3.5.4.1 State of the art

RSV will only work with selected standards, but RSV and Granitor see the benefit of translation between RSV picked standards and Granitor standards and an automatic translation between them.

3.3.5.4.2 Enablers

Not applicable.

3.3.5.5 Use Case Engineering Process

3.3.5.5.1 State of the art

Currently, the engineering process is fragmented, relying on disconnected tools and manual integration. No formal adoption of the Arrowhead-EP has been documented.

3.3.5.5.2 Enablers

The use of IEC 81346 and RAMI models offers a framework for structured, standardized engineering processes. Future efforts will focus on aligning workflows with these standards.

3.3.5.5.3 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.
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.
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.
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.

3.3.5.6 Use case and engineering costs

An analysis of engineering costs at M0 will 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.

The overall system consists of different modules, located both on-board and on-ground, providing data and HW / SW framework, whose objective is to collect and correlate all data in order to support AHMS users.

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

3.3.6.1 Baseline summary

3.3.6.1.1 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.

3.3.6.1.2 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 passport² (DPP).

² 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

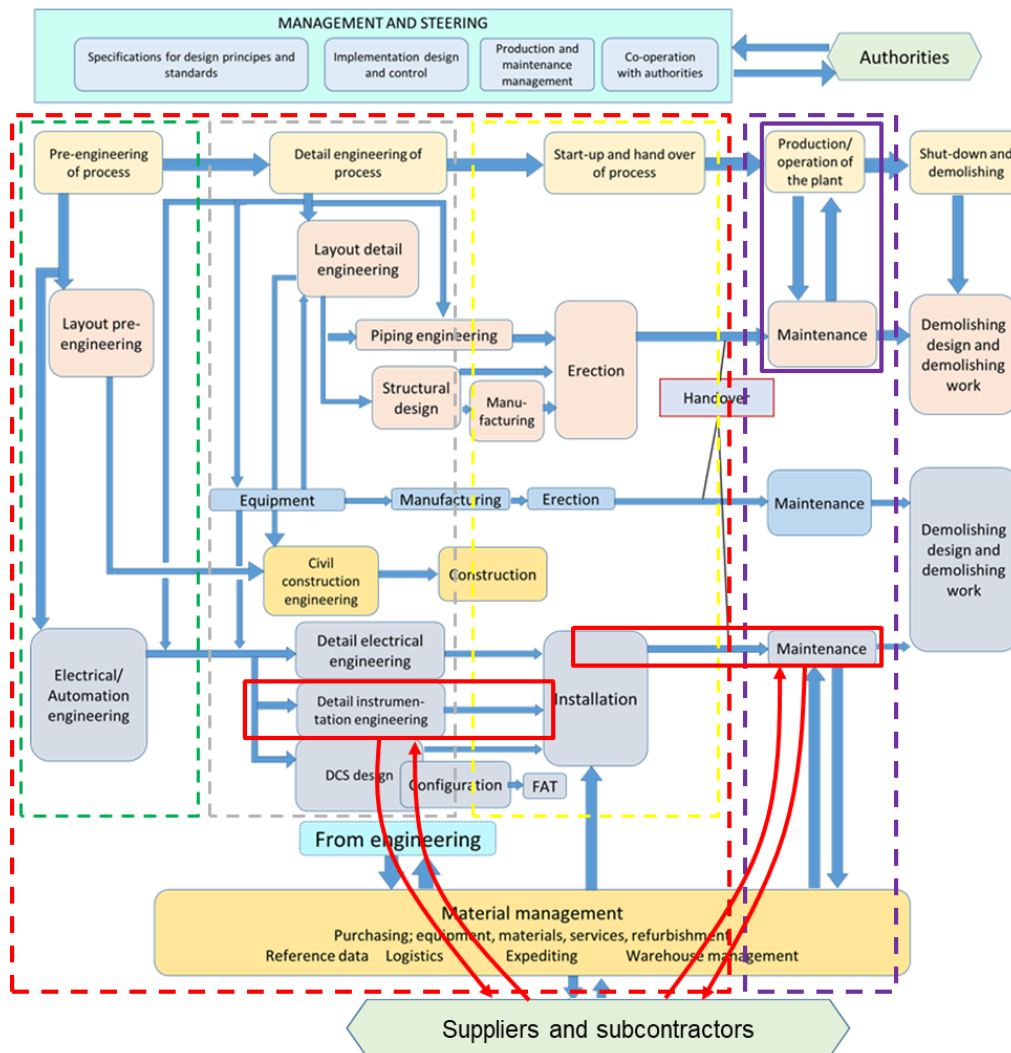


Figure 14 - General information flow diagram for plant lifecycle data from an investment project (red dashed line box) to plant demolition (source PSK 5990 standard). Thick blue lines describe information flows. The investment project and supplier service pilots for valves and instrumentation are described 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).

3.3.6.1.3 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.	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	Several important standards have been identified in use for some aspects on the baseline level

major digital data model for industrial production between which autonomous and seamless data model understanding is enabled.	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	-

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

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.

3.3.6.2.1.2 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.

3.3.6.2.2 Major digital languages

3.3.6.2.2.1 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.

3.3.6.2.2.2 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.

3.3.6.2.3 Autonomous translation

3.3.6.2.3.1 State of the art

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

3.3.6.2.3.2 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.

3.3.6.3 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”.

3.3.6.3.1 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.

3.3.6.3.2 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

3.3.6.3.3 Micro-service paradigm

3.3.6.3.3.1 Approach

Use of Eclipse Arrowhead to integrate systems at run-time.

3.3.6.3.3.2 Impact

Standardize how information systems are discovered and bound at run-time. Standardize how authentications and authorizations are performed.

3.3.6.3.3.3 Long-term effect

Improved interoperability, reduced engineering effort.

3.3.6.3.4 Major digital languages

3.3.6.3.4.1 Approach

Set of selected standard data models and digital languages, and clear guidelines how to use them.

3.3.6.3.4.2 Impact

Improved data interoperability, improved quality of data, reduced number of errors.

3.3.6.3.4.3 Long-term effect

Improved usability of data, lower data management and governance costs.

3.3.6.3.5 Autonomous translation

3.3.6.3.5.1 Approach

Automatically invoke translations to provide added interoperability

3.3.6.3.5.2 Impact

Better chances to integrate different systems and new processes.

3.3.6.3.5.3 Long-term effect

More competitive network, improved business collaboration prospects.

4. Collaboration with other technical WPs

WP1 is linked and influences all the horizontal WPs and the use cases, with an impact on the entire project, both in terms of the requirements and specifications of the technologies that the project will develop and for the evaluation and validation of the project results. The following Pert diagrams illustrate the relations with the other WPs.

WP1 forces us to brainstorm about the use case architecture, adopted technologies and solutions, about the reuse of the horizontal technologies in different use cases, about the advantages obtained by their adoption and how they contribute to project objectives, looking forward beyond the baseline with an action plan and a validation/verification process. The surveys play an important role in this regard, providing concrete support for all the aspects of brainstorming and mitigating the complexity and heterogeneity of the use cases, and the complexity of the horizontal technologies.

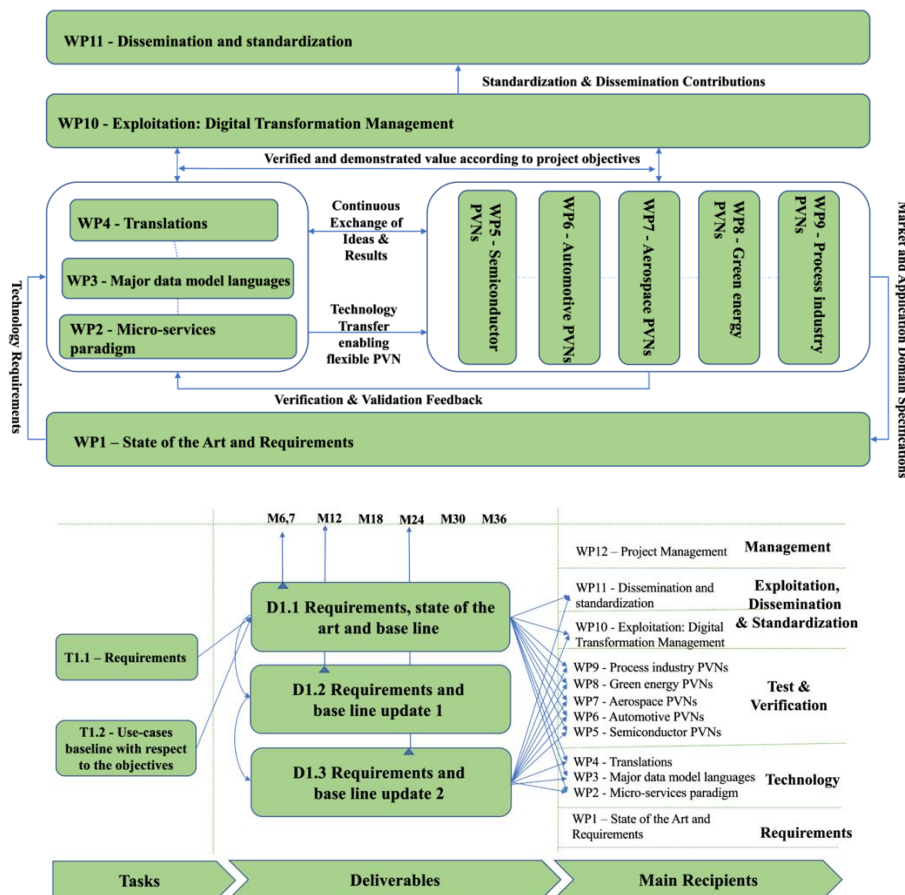


Figure 15 - Project and WP level Pert diagrams

WP1 plays a strategic role in the overall evaluation of the Arrowhead fPVN project. By defining the baselines for each use case, WP1 establishes the reference conditions representing the technological and operational status at project start (M0). These baselines serve as the primary term of comparison for assessing the progress and improvements brought about by the project's activities and innovations. In parallel, WP1 is responsible for the systematic monitoring of

requirements fulfilment, tracking how the project responds to the needs and expectations identified across the technical work packages and use cases. These two evaluation dimensions, baseline comparison and requirement fulfilment, are essential components in measuring the success and impact of the project. From this perspective, WP1 maintains a strong link with WP10, which is responsible for the overall impact assessment. The outcomes produced by WP1 directly support WP10's activities by providing structured, traceable, and quantifiable evidence of the value delivered by the project across its objectives.

5. Conclusions

This deliverable presented the results of the requirements elicitation and of the baselines definition. The methodologies adopted to carry out these two tasks have been described and we reported the preliminary results of both tasks.

During the second year of the project, significant advancements have been achieved, in the definition and consolidation of the baselines and requirements. The results presented in this deliverable reflect a step forward compared to the status at the end of year 1, confirming the effectiveness of the work methodology and the growing maturity of the analysis in both horizontal and vertical work packages.

Section 2 reported a systematic and cross-cutting effort in the elicitation and refinement of use case and horizontal work package requirements. Building on the initial list of 37 requirements identified in the first year, the second year focused ensuring the coverage of the three building blocks by all the use cases and, in the last part of the year, on validation and assessment, resulting in a consistent and traceable mapping of requirements against project objectives and technical building blocks. The updated list of 50 requirements includes a more detailed characterization of each requirement's purpose, scope, validation strategy, and progress. An evaluation of fulfilment levels at the end of year 2 shows that many requirements have moved forward from early architectural or conceptual maturity to advanced implementation phases, although full integration is largely scheduled for year 3. This more comprehensive requirements list now provides a robust basis for technical convergence across use cases and for ensuring alignment with the broader goals of the Arrowhead fPVN project.

In Section 3, we observed several progresses in the definition of use case baselines. Compared to the uneven and often preliminary characterizations seen at the end of the first year, several use cases now rely on fully structured and stable baselines. In particular, UC 1.6, UC 1.7 and UC 2.7 have consolidated their design and implementation scope, while UC 1.9 and UC 3.9 have transitioned from exploratory to well-outlined baseline definitions. UC 2.6 is still behind the schedule, but efforts are ongoing to ensure the baseline will be available. Overall, the development of baselines now shows a convergence trend, which will be capitalized upon in the final year of the project.

The second year marks an important point in the alignment between design/plan intentions and actual results. The results presented in this deliverable create the conditions for an effective evaluation of the project results planned for the second part of year 3. To this regard, during the last year of the project, we plan to continue monitoring the requirements fulfilment and finalize the baselines definition, including the study of the engineering process costs and their impact.

6. Annex

1. Requirements Matrix: external file “AfPVN WP1 - Requirements elicitation and evaluation Y2.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”

7. Revision history

7.1 Contributing and reviewing partners

Contributions	Reviews	Participants	Representing partner
TOC definition and first version		Elisa Londero, Gianvito Urgese, Massimo Vecchio	ETH, Polito, FBK
Deliverable contents		Elisa Londero, Gianvito Urgese, Massimo Vecchio	ETH, Polito, FBK
Deliverable contents		WP2, WP3, WP4 leaders	
Deliverable contents		Use case leaders	

7.2 Amendments

No.	Date	Version	Subject of Amendments	Author
1	04/04/2025	0.1	Document structure and intro	Elisa Londero
2	10/04/2025	0.2	Requirements section	Elisa Londero, Gianvito Urgese
3	12/05/2025	0.4	Use cases baselines	Elisa Londero, Gianvito Urgese
4	22/05/2025	0.9	Pre-final version	Elisa Londero, Gianvito Urgese
5	04/06/2025	1.0	Final revision	Elisa Londero, Gianvito Urgese, Massimo Vecchio
6	10/06/2025	1.1	Final version	Paolo Azzoni

7.3 Quality assurance

No	Date	Version	Approved by
1	12/06/2025	1.1	Jerker Delsign