Version	**Status**	**Date**
4.0	Final	2021-05-26
Author	**Contact**	
Gianvito Urgese	gianvito.urgese@polito.it	

D2.2 Deliverable Appendix – Updated Use Cases analysis for AHT-EP definition

POLITO: Gianvito Urgese
gianvito.urgese@polito.it

Abstract

In this document, we provide the information describing the Engineering Process Phases adopted in the 22 Use Cases that have been collected by using WP1-WP2-WP4-WP10 survey. This information is analysed for extracting a table for each Use Case (UC) that summarise what are the Arrowhead Tools Engineering Process Phases (AHT-EPP) used in the UC and what are the objectives that each EPP can potentially match during the project.

ECSEL EU project 826452 - Arrowhead Tools
Project Coordinator: Professor Jerker Delsing | Luleå University of Technology

Table of contents

Updated Use Cases summary and AHT-EP analysis

This document contains the updated information collected from Use Case leaders that have been analysed during the second iteration of analysis performed for provide a final definition, feature, structure, and components to be included in the AHT-EP model (Figure 1) that is used for defining the life cycle management of all the use cases supported in the Arrowhead Tools project. We provided also a short summary for each Use Case describing the main product/system under development. A more detailed explanation can be found in WP1 and WP7-8-9 deliverables.
All the information here reported have been extracted from the named *WP12410_survey* [1] documents filled by the Use Cases leaders.

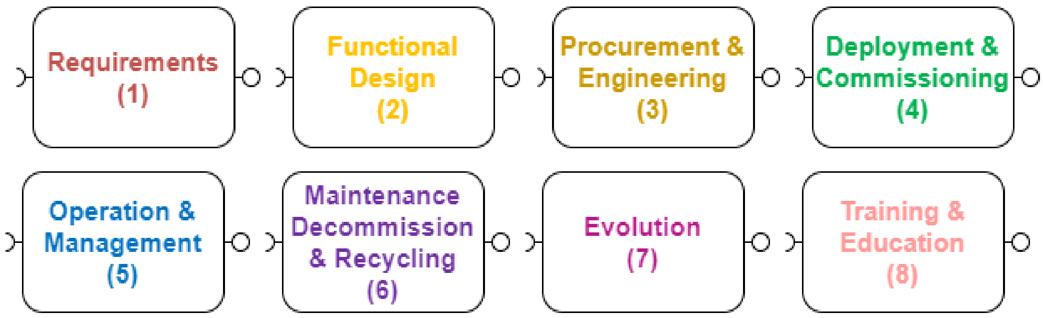


Figure 1 The Arrowhead-Tools Engineering Process (AHT-EP)

A table similar to Table 1 is produced for each use case where we report the AHT-EPPs used by the Use Case and the objectives that the UC will potentially match during the project.
In this Table 1 we have the AHT-EP and unknown phases in the first row. In the second row we have a letter that indicates if the UC have EPPs that match with the AHT-EP. In the following the meaning of each letter adopted in second row:

- **C** indicates that the EPP phase is currently used in the Use Case
- **P** indicates that the EPP should be planned to be used for a full management of the life cycle
- **N** indicates that the phase is not adopted
- **U** indicates that from the input material is not clear if the phase is used or not

The following six rows (OBJ-AHT) reports if the particular EPP can potentially match one of the Arrowhead-Tools objectives. Whereas, we reported on the last four rows EPPs of the UC that can potentially match the WP2 objectives.
The objective AHT #1 is overlapped to the objective of WP1.
With this notation, we can read the Table 1 as described in the following:
In the UC we use only the Requirements (AHT-EPP1) and Procurement & Engineering (AHT-EPP4) phases. Currently only the Requirement phase is adopted, the AHT-EPP4 will be adopted during the project. The Functional Design phase is not specified in the input material. For the Requirement phase, we can potentially match the objective #1 of the AHT project and objectives #1 and #3 of that are targeted by the WP2.
Whereas, for the Procurement & Engineering phase we can potentially match objectives #4 and #5 of the project and objective #1 of the WP2.
In the last column, we see that all the UC-EPPs have found a match with one of the AHT-EPP.

Table 1 Example of table that match the UC and the AHT Engineering Processes with focus on the potential objectives that can be matched in each phase.

AHT-EPP	**1 (R)**	**2 (FD)**	**3 (PE)**	**4 (DC)**	**5 (OM)**	**6 (M)**	**7 (E)**	**8 (TE)**	**Unknown**

UC-EP match	C	U	C	P	P	N	U	C	N
OBJ-AHT #1	X			X	X				
OBJ-AHT #2			X	X					
OBJ-AHT #3									
OBJ-AHT #4				X					
OBJ-AHT #5				X				X	
OBJ-AHT #6			X					X	
OBJ-WP2 #1	X		X	X					
OBJ-WP2 #2									
OBJ-WP2 #3	X		X	X					
OBJ-WP2 #4			X					X	

1. UC-01 Automated Formal Verification (CAMEA)

The Automated Formal Verification use case (UC-01) primarily aims at one of the phases critical in the development of embedded devices – namely, analysis and verification. In that sense, it is somewhat atypical and different from other use cases of the project as it focuses more on one group of methods used when developing embedded devices rather than on the development of a particular embedded device. Due to this, some inputs concerning UC-01 may differ or be limited with respect to those of other UCs.

However, the considered analysis and verification methods are studied not completely in general but with a stress on their applicability primarily in the area of intelligent traffic monitoring systems being developed by the CAMEA Company. Moreover, during the initial phase of the project, the focus was narrowed down to one part of a complex traffic monitoring system concerning, e.g., travel times or section speed control. In particular, a CAMEA's smart camera (or a system of several such cameras) connected to the cloud were selected as an ideal candidate for investigation within the UC. The first development iteration of the camera has been recently finished. The analysis and verification of the smart camera (or a system of such cameras) will be performed both on the level of models of the camera and its surroundings, where the models are to be developed within the UC, as well as on the level of the code.

UC-01 is concerned with a systematic application of automated analysis, verification, testing, and synthesis techniques, possibly aided by modelling techniques to obtain the needed models, to the development of intelligent traffic surveillance and control systems. As discussed in the previous section we consider primarily smart-camera systems developed by CAMEA plus possibly the underlying technologies they exploit. These techniques are primarily applied to the early phases of development processes to reduce their time and cost – we thus focus solely on the stages including (1) Requirements, (2) Functional Design, and (3) Procurement and Engineering.

1.1 Overall description of the UC-EP

Currently, CAMEA applies only a very limited class of techniques focusing on automation and improvement of the early phases of the development process. Requirement analysis is done mainly based on meetings among customers and developers and processed using Word and Excel spreadsheets. During the Functional Design, developers build simple block models describing the main functional components (using Python, C#, C++, sometimes Matlab) – no advanced modelling and simulation techniques are used. During the engineering phases manual testing is used only.

Since Use Case 1 focuses primarily on methods of analysis and verification, not on a specific product, the engineering process used is not of a primary importance here (indeed, one would have to speak about a process used to develop analysis and verification methods). However, the considered methods are to be applied with a specific focus on smart camera systems developed by CAMEA and their underlying technologies. We can hence refer to some degree to the engineering process of CAMEA limited to the first three phases relevant from the point of view of analysis and verification. From this point view, we can say that the considered smart camera systems are currently developed using a waterfall model. Within the development process, no advanced analysis and verification techniques are used. Providing such techniques, tools implementing them, and experimentally applying these tools in the given context is the target of the UC. Likewise, no framework is currently in use to manage the engineering process, neither – which is more relevant from the point of view of the UC – the analysis and verification process.

We aim at applying the OSLC standard for communication with at least some of the involved analysis, verification, and testing tools. As for the code being analyzed, we mostly assume that it adheres to the relevant programming language standards (e.g., the C and C++ standards).

1.2 Engineering Process Description

By its nature, the UC does not focus on the development of a new system but on the methods of analysis, verification, and testing in general (including development of new/improved analysis, verification, and testing tools), using smart cameras for validation of the developed methods and tools. Therefore, this item is not directly applicable for the UC. However, we can say the modelling, analysis, and verification whose applications and further development are the subject of the UC can be mapped to the AHT-EP stages (shown in Figure 2) as follows:

- *Functional design*: Gamma Statechart Composition Framework, Uppaal, PRISM, Modica, Testona.
- *Procurement and engineering*: various analysis plugins for platforms such as Facebook Infer or Frama-C, ANaConDA, Perun, Testos, 2LS, Predator/Symbiotic.

Further, the UC covers specifically the first three phases of the EP. However, it may be the case that the later stages may lead to a need to revisit the requirements, functional design, or engineering, forcing the phases covered by the UC to be repeated (at least to some degree).

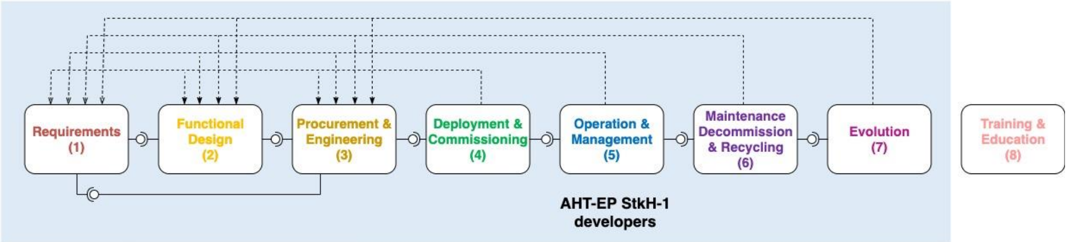


Figure 2 AHT-EP of use case UC-01

No technology is completely missing in the AHT-EP model. However, the existing analysis, verification, and testing technology needs to be suitably applied, optimized, and/or extended to support the needs of the considered systems.

Only the first three AHT-EP phases are considered as relevant, together with appropriate training.

Table 2 AHT-EP Phase focus of UC-01

Engineering process phase	Addressed/Focus
Requirements	**Activity** Requirements are obtained from past experience of CAMEA experts and from discussions between CAMEA representatives and customers. They are tracked using Word and Excel documents, sometimes even staying in the memory of CAMEA experts who then reflect them in the system design, code assertions, and the tests being manually prepared. **Tools** Input: none Output: At least some of the requirements on the smart camera systems, which are currently tracked completely informally (sometimes not even written down), should be formalised. For that, suitable specification means are to be chosen (e.g., structured English templates, formulae in a suitable logic, etc.).
Functional design	**Activity** Simple block models representing prototypes of the main functional components are developed using languages like Python, C#, or C++. Sometimes Matlab is used (in particular, when dealing with AI-based components). Repeated test or simulation runs are manually performed to assess properties of the developed prototypes. **Tools** Phase 1 (System Modelling) Inputs: Formally as well as informally described requirements on the smart camera system. Outputs: Models of critical components of the system or suitably abstract models of the entire system applicable for analysing critical properties of the components or the system as a whole before investing into their implementations. The models should be described using suitable high-level programming languages or modelling languages. For instance, UML usage models will be considered on the component as well as on the system level (obtained, e.g., using the tool MODICA), to which requirements can be directly linked and (executable) test cases can be automatically generated that ensure an appropriate coverage (e.g. requirement coverage). Furthermore,

Engineering process phase	Addressed/Focus
	classification trees (e.g., from the TESTONA tool), i.e. very basic segmentations of the relevant system parts, will be considered to analyse and compare certain system variations. Phase 2 (Model Analysis) Inputs: Executable models of the particular system components (e.g. time or probabilistic automata) obtained from the system modelling phase. Analysis models automatically synthesized from the high-level engineering model. Specification (functional and extra-functional) derived from the requirement analysis. Outputs: Analysis of the models with respect to the specification. Identification of requirements that are violated in the model. Counterexamples demonstrating the violations and identifying possible flaws in the models. Sensitive parts and components in the systems design or robustness issues.
Procurement & Engineering	**Activity** Based on experience gained from prototypes (provided they are built), final versions of the systems under development are written in C#, C++, or C (the latter is used in particular for drivers). Sometimes, prototypes developed in C# are just further optimised and used as the final product. In case of hardware-accelerated processing, VHDL code is produced, and synthesis for FPGA is subsequently used. For developing GUI, Visual Studio and C# are used. Unit tests and system tests are used in a manual way to check whether the system produced satisfies the collected requirements. Typing and basic static analysis checks built into Visual Studio and CLion are used. For the HW developed, manual comparisons of its behaviour with the behaviour of a C#/C++ prototype are used. **Tools** *Advanced automated static and dynamic analysis, formal verification* Input: code to be analysed (possibly coming in different, gradually created versions), properties to be checked (either generic, such as null pointer exceptions or absence of data races, or specified in an appropriate way, e.g., using assertions, code annotations, logical formulae, etc., depending on the tools used), analysis/verification/test harness if needed (sample inputs, models of the environment, etc.) Output: real or potential code defects, diagnostic information
Deployment & Commissioning	This phase is not in focus of the UC. However, it may trigger a need to change the requirements, update the functional design, and/or repeat activities done in the phase of procurement and engineering – then the activities described above are again performed.
Operations & Management	This phase is not in focus of the UC. However, it may trigger a need to change the requirements, update the functional design, and/or repeat activities done in the phase of procurement and engineering – then the activities described above are again performed.
Maintenance Decommissioning & Recycling	This phase is not in focus of the UC. However, it may trigger a need to change the requirements, update the functional design, and/or repeat activities done in the phase of procurement and engineering – then the activities described above are again performed.
Evolution	This phase is not in focus of the UC. However, it may trigger a need to change the requirements, update the functional design, and/or repeat activities done in the phase of procurement and engineering – then the activities described above are again performed.

Engineering process phase	Addressed/Focus
Training & Education	**Activity** No training in advanced analysis, verification, or testing approaches. **Tools** Input: analysis/verification/testing approaches identified as most promising, CAMEA developers selected for training Output: new/improved documentations of the tools considered, new/improved study materials concerning principles of the considered methods and tools, improved skills in testing, analysis, verification among CAMEA developers

1.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 3 UC-01 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	This objective is matched by the considered application of advanced analysis, verification, and testing methods applied by developers (AHT-EP Stkh1) in Phases 1, 2, and 3 of the AHT-EP, which should limit the number of errors discovered late in the development process where they are often the most difficult (timewise as well as moneywise) to correct. **Planned actions** • AHT-EPP 1: Attempts to formalize at least some requirements on the smart camera systems. Application of suitable modelling and model-based analysis, verification, and testing tools, possibly complemented by further improvements of these tools. • AHT-EPP 2: Application of suitable modelling and model-based analysis, verification, and testing tools, possibly complemented by further improvements of these tools. • AHT-EPP 3: Application of suitable code-level analysis, verification, and testing tools, possibly complemented by further improvements of these tools.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The planned new support of the OSLC standard in at least some of the considered tools should make it easier to integrate these tools with other similar tools from other providers as well as to integrate them into various IDEs. This will ease the work of developers (AHT-EP Stkh1) in Phases 1, 2, and 3 of the AHT-EP. **Planned actions** • AHT-EPP 3: Added/improved support of OSLC at least in some of the considered analysis, verification, testing tools.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Not relevant.

WP2 Objective	Focus & Planed actions
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	This objective can be supported by improved documentation of at least some of the considered analysis, verification, or testing tools, as well as by providing suitable examples of the how the tools can be used such that developers (AHT-EP Stkh1) can themselves play with them and learn what these tools can provide them with. This will improve the efficiency of developers (AHT-EP Stkh1) in Phases 1, 2, and 3 of the AHT-EP. **Planned actions** • AHT-EPP 2: Providing improved documentation and/or illustrating examples for the considered tools. • AHT-EPP 3: Providing improved documentation and/or illustrating examples for the considered tools.

Table 4 UC-01 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Reducing the number of errors discovered in smart camera systems (or, more generally, in smart traffic monitoring and control systems, or, even more generally, embedded systems) and their underlying technologies during late development phases or escaping into operation by application of suitable advanced analysis, verification, and/or testing techniques and tools. Improvements of scalability, generality, or automation of the considered techniques and tools.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Added/improved support of OSLC at least in some tools for advanced analysis, verification, and testing, allowing their better integration with other analysis tools, IDEs, etc.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Despite the use case primarily aims at verification of intelligent traffic control systems and similar classes of systems, a connector (or connectors) allowing to interconnect such systems developed in CAMEA to the Arrowhead framework will be developed (and suitably verified).
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Probably not applicable.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and	Probably not applicable.

Project Objective	Focus & Planed actions
automation solutions	
Obj. 6 - Training material (HW and SW) for professional engineers	New or improved documentation and/or illustrating examples for the considered analysis, verification, and testing tools. New or improved study materials on the background of the considered analysis, testing, and verification methods.

1.4 Engineering Process analysis

Table 5 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-01.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	P	N
OBJ-AHT #1	X	X	X					X	
OBJ-AHT #2		X	X						
OBJ-AHT #3			X						
OBJ-AHT #4									
OBJ-AHT #5									
OBJ-AHT #6		X	X					X	
OBJ-WP2 #1	X	X	X						
OBJ-WP2 #2			X				X		
OBJ-WP2 #3									
OBJ-WP2 #4		X	X					X	

2. UC-02 The MRI pTX Optimizer: Design tool framework for MRI RF transmit chains (PHC)

The speed and quality of MRI Scans is highly dependent on the B_1^+ component of the RF transmit field. Best results are obtained if this field is as high and homogeneous as possible in the Region Of Interest (ROI).
Especially on Ultra High Field (UHF) systems, like a 7 Tesla whole body scanner, this is a difficult challenge due to:

1) Intrinsic RF field inhomogeneity: On higher RF frequencies (300 MHz for 7T) the wavelength is much shorter than body parts, leading to very inhomogeneous RF fields if traditional RF body coils are applied
2) Heating: high frequent RF field radiation leads to heating of parts in the body. If the local power is too large, tissue may get too hot, leading to patient harm. This is translated in formal regulations to a limit of the SAR (the specific absorption rate) in the tissue, protecting both global heating of the body and local heating of small tissue parts

The RF inhomogeneity can be largely improved by pTX, parallel transmit coils. In that case the RF coils are not only one antenna, connected to a single amplifier, but an array of antennas, each connected to a separate RF amplifier, with independent amplitude (so power) and phase control.
The added flexibility of pTX coils requires sophisticated models, in order to determine the safe and optimal settings for the selected scan technique. This strongly depends on the patient anatomy and size, the clinical application and region of interest (ROI) and the applied pTX coil.
Problem Statement:

- Model based design is very important to allow optimal control of the pTX system
- Complex algorithms and models are required to analyze the characteristics of the coil in combination with the applications, to guarantee safe operation and to optimize performance
- Currently, the output data of models are evaluated manually and applied for the selected series of control settings of the scanner, captured in the coil parameter archives, which are indirectly selected by the operator
- The results in:
  - Time consuming development.
  - Limited solution space, so sub-optimal performance.

The goal of the MRI pTX Optimizer is to smoothen this workflow, leading to more productivity and higher quality for both the R&D organization and the users of the pTX coils.
The use case evolves across three different phases:

- Phase 1 (M0): this phase represents the past and collects the expertise of the stakeholders that will cooperate to design, model and apply the pTX coils. The current way of working is analysed and interfaces are captured.
- Phase 2: The optimized way of working is described, leading to faster and more interactive design cycles. Interfaces to the Arrowhead framework are defined (UC-EP).
- Phase 3 (M36): The interfaces are implemented and the optimized design process is thoroughly exercised.

2.1 Overall description of the UC-EP

The MRI pTX Optimizer improves the R&D productivity to create top quality parallel transmit coils for Ultra High Field MRI systems, better satisfying customer needs via better calibrated systems and optimized user productivity.

In this process the following stakeholders are distinguished as shown in Figure 3:

A. RF Coil engineers (StkH1) design application specific coils, using their knowledge and experience of analogue RF electronics and antenna design. RF coils are in most cases made by third party suppliers, specialized in creating dedicated RF coils.
B. RF Coil simulation engineers (StkH2) use models of coils and human bodies to determine the relevant coil properties (B_1^+ field, SAR), and share the results through a service architecture network (Arrow Head). The RF coil simulation engineers could be employed by the RF coil supplier or by Philips, the MR system manufacturer.
C. MR Methods engineers (StkH3) create MR scan technique models that optimize performance, based on the selected coil and application, in combination with the simulated RF performance

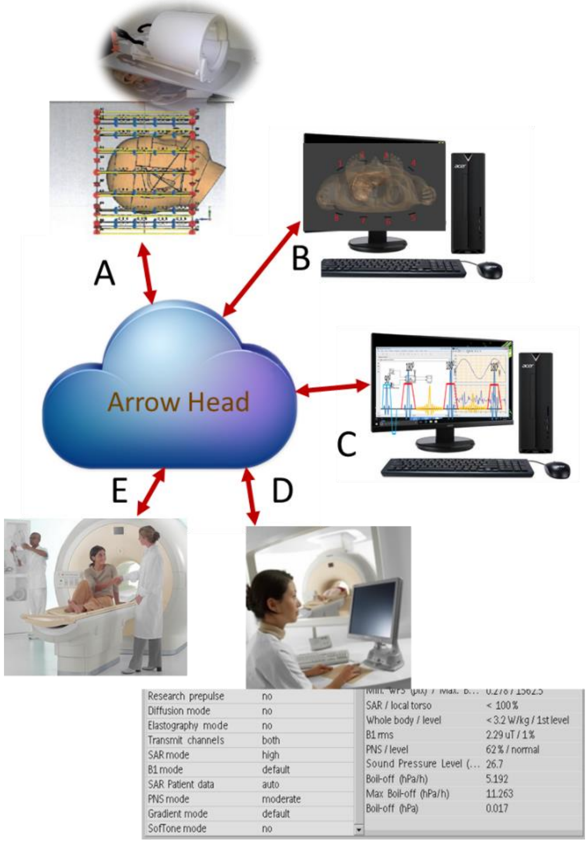


Figure 3 UC-02 stakeholders

D. MR Application engineers (StkH4) optimize scan protocols and ExamCards, based on the combined knowledge of RF coils and MR scan techniques, to reach optimal Image Quality and in the shortest possible scan time.
E. The Clinical MR users (StkH5) use the MRI system routinely for clinical examinations, together with the provided pTX coils.

The building blocks for the MRI pTX Optimizer are pictured in the schematics of Figure 4:

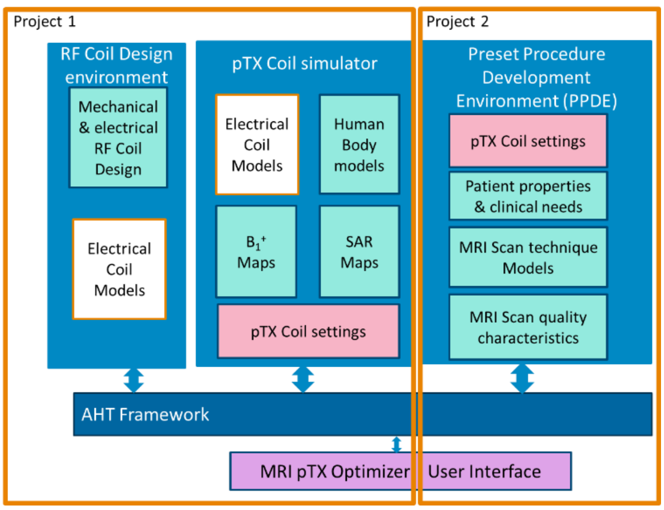


Figure 4 The Building blocks for the MRI pTX Optimizer

- RF Coil Design environment for StkH1, who design the coil, based on user needs provided by the clinical user, derived technical requirements and regulatory constraints.
- The pTX Coil Simulator: the tool for StkH1 and StkH2 to determine the important characteristics of the pTX Coil for different anatomies and applications. The simulator uses human body models and RF coil models to predict the RF transmit fields and to calculate the expected B1+ fields and SAR distributions.

- The MR Methods engineer (StkH3) designs and optimizes scan techniques and calibrations in order to use the coil most effectively.
- The Preset Procedure Development Environment (PPDE): The tool for StkH4 to determine optimal scan protocols for relevant clinical applications, using MRI pulses sequence algorithms, created by StkH3, in combination with the pTX coil characteristics, obtained from the pTX coil simulator. PPDE shows important parameters of the scan protocols to the user to guide the optimization, such as total scan time, image quality (Signal to Noise ratio, contrast), patient comfort parameters (SAR level, sound level).

The transfer of information between the various parts of the tool chains is, in most cases, manual. A framework is currently hardly defined.

2.2 Engineering Process Description

In the Figure 5 AHT-EP of the UC-02. The development of a new MRI pTX coil starts with the User Needs, so the requirements defined via direct communication with the end user. This is direct input for the requirements for the RF coil engineer (c34), together with input from evaluation of earlier versions (c11), field problem reports (c17) and regulatory requirements.

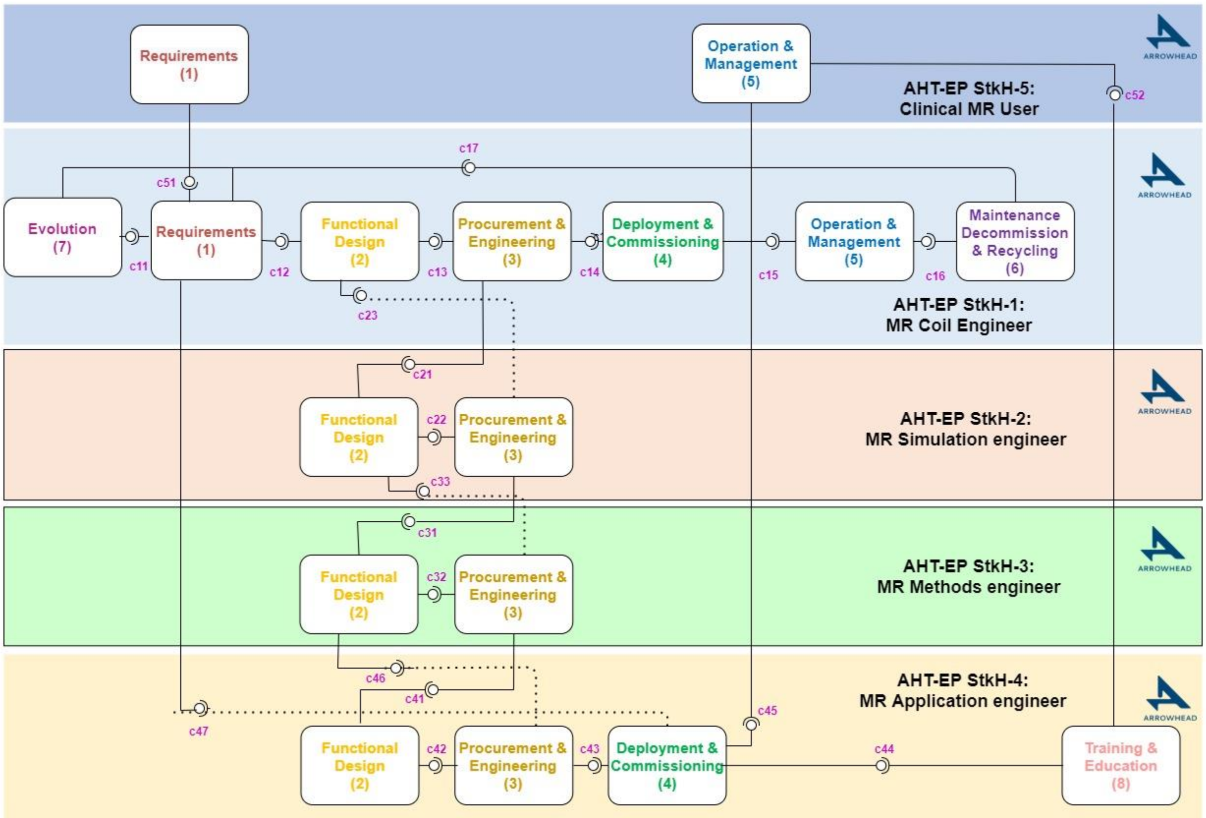


Figure 5 AHT-EP of use case UC-02

One of the design outputs, c21, the electrical model of the coil, is input for the MR Simulation engineer, who combines this with the other models to calculate the coil fields. The resulting

parameters are input for the MR Methods engineer (c31), who uses this to optimize the MR scan techniques using this coil. The resulting techniques are input for the MR Application engineer, c41, who uses this to optimize the protocols. The output of this engineering phase is shipped to the end user, the Clinical MR user (c45) and is used to create training material for the customers (c52).
During these steps the engineers might encounter issues or identify possible optimizations. This is used as feedback during the process to further improve the designed solution (c23, c33, c46 and c47).

All the AHT-EP phases are considered in the MRI pTX Coil optimizer use case.
At the current state we did not identify any lack of technology for our multi stakeholder use case.
The EPP, as presented in the Figure 5, is applicable for any MRI coil development and is completely scalable regarding the size and number of channels of the coils. Also all clinical applications can be addressed via this procedure.
The Philips Product Data and Life Cycle Management (PDLM) process is applied for all phases in the EPP for clinical products. Products for research customers, who use the coils for research studies, are created according to RDD process, a tailored version of the PDLM process.

Table 6 AHT-EP Phase focus of UC-02

Engineering process phase	Addressed/Focus
Requirements	**Activity** Requirements are obviously key to create solutions that both completely satisfy the customer needs, within the limits posed by MRI physics and target MRI system, and satisfy all safety regulations. Since most 7T coils are purchased from third party suppliers, the purchasing specification needs to be carefully formulated, together with financial and maintenance aspects. Due to the complexity of the TX chain at Ultra High Fields, iterations on requirements are inevitable to find the optimal balance between requirements for the RF coil, human safety, the calibration procedure and the clinical performance. Also experiences with earlier versions of the coils, including defect reports and customer complaints and appreciation, need to be taken into account. Since the products have a long life time (typically support for 10+ years is guaranteed) End of Life issues of applied components also influence the requirements and design choices. **Tools** ***StkH1: RF Coil engineer***: StkH1 obtains the user needs from the MR Clinical User (typically via the product marketing organization). These are input for the technical requirements, together with the regulatory requirements, evolution of older models, and maintenance insights. Requirements are maintained in ALM and communicated via Word documents. ***StkH5: MR Clinical user***: StkH5 defines the User Needs, which are important input for the technical requirements of the Coil.
Functional design	**Activity** The special conditions, posed by the Ultra High Field MRI, combined with the specialized clinical applications, make an iterative design process is inevitable.

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
Smooth links between the tools from different domains (RF coil design, RF field analysis, RF chain calibration, MRI scan techniques, and clinical scan procedures) will have significant impact on speed and quality of the design process.
The three main design phases all play an important role:
Conceptual modelling
Architectural modelling
Detailed design
Tools
StkH1: RF Coil engineer: StkH1 creates the complete design of the coil. The design is reported in Word files.
StkH2: RF Coil simulation engineer: StkH2 obtains the electrical design model of the coil, resulting the engineering phase of StkH1, and combines this with the human body models. He sets up the framework to simulate the RF properties of the coil
StkH3: MR Methods engineer: StkH3 obtains the report with optimal parameter settings for this coil from StkH2 and translates that to settings that can be used for the controlling software
StkH4: MR Application engineer: StkH4 uses the combination of coil and optimized MR methods software to define the optimal scan protocols for the clinical application, required by the MR Clinical user (StkH5).
</td></tr>
<tr><td>Procurement & Engineering</td><td>
Activity
Procurement:
Since most 7T MR RF coils are designed by third party suppliers the procurement process plays an important role. The supplier need join in the iterative design process, within the constraints of the business relation between supplier and Philips (e.g. regarding costs, responsibilities, warranty, lead time).
Engineering:
The process to create pTX coils needs to deal with different interfaces:

Data: Interface between various tools and between the coil parameter archives and the scanner
Machine: Interface between different physical systems: the mechanical and electrical parts of the RF coil and with the body parts of the patient.
Hardware: Interfaces between physical systems in electrical engineering and electronics.
Network: The RF coils are connected to the MRI system via an analogue (power, control signals) and digital network (receiver data, digital control)
Software: Simulation SW interfaces and interfaces on the MRI scanner
User: Interface between designers and the tools, and between the clinical operators, the RF coils, and the MRI system

Tools
StkH1: RF Coil engineer: StkH1 acquires all components (via procurement) to build the coil and does the detailed design and engineering of the coil. If the coil is created by a third party supplier, in this stage the procurement of the complete coil takes place, based on the requirements defined in the previous EP. The detailed design is captured in Word documents, electronica drawings (Mentor Graphics) and mechanical CAD drawings (Pro-Engineer).
</td></tr>
</table>

Engineering process phase	Addressed/Focus
	StkH2: RF Coil simulation engineer: StkH2 performs the simulations for all required anatomies on various human body models, and creates a report (Word and Excel) with the conclusions of optimal and safe setting of the RF transmit chains, using this coil. ***StkH3: MR Methods engineer***: StkH3 optimizes the MR Methods software, if needed, to optimally calibrate the RF transmit chain with this coil, and optimizes the scan techniques that apply this coil, if needed. This is documented in Word files, and the SW modifications are coded in C++. The coil parameters are stored in coil parameter files, interpreted texts files with a proprietary format. ***StkH4: MR Application engineer***: StkH4 optimizes the scan protocols and archives these in the Exam Card database.
Deployment & Commissioning	The MR application engineer first verifies the complete solution: pTX coil in combination with the scan techniques and available protocols. Secondly, the MR application engineer validates if all User Needs, defined based on customer input, are satisfied. **Tools** ***StkH1: RF Coil engineer***: StkH1 (or the linked verification engineer) verifies if the coil meets all defined requirements. ***StkH4: MR Application engineer***: StkH4 verifies the final result on MR systems via volunteer scanning and validates if all User Needs are met. StkH4 contacts customers (StkH5) for final evaluation.
Operations & Management	After finalizing and verifying the design of the coil, this design is transferred to Operations, who establish the manufacturing process and ship the manufactured coils to the finial customers. This process phase is not including in the study of this AHT use case. **Tools** ***StkH1: RF Coil engineer***: StkH1 transfers the design of the coil to Operations to enable manufacturing of the coil. ***StkH5: MR Clinical user***: StkH5 obtains the final product (coil, updated software and optimized Exam Card (protocols), as shipped from the factory and receives training material and education from StkH4.
Maintenance Decommissioning & Recycling	After delivery of the pTX coil the customer feedback is monitored and issues, like complaints and defects, are fed back to the RF coil engineer. Maintenance aspects need to be addressed by service organisation, and, in case of defects or malfunction related to design choices, by the RF coil engineers. This process phase is not including in the study of this AHT use case. **Tools** ***StkH1: RF Coil engineer***: StkH1 receives feedback from the field, which may include reported problems, customer complaints and information about availability of parts / End of Life issues.
Evolution	MRI systems and related technology is continuously evolving. Especially high end applications, like Ultra High Field systems, are driven by new insights and applications. The requirement ad design process therefor needs to be able to adopt these new insights and create further optimized coils, scan techniques and applications.

Engineering process phase	Addressed/Focus
	Often dedicated components are applied in the pTX coil, which have a risk of becoming End of Life during the life time of the coil. In that case, a resign might be required to adapt the coil to the newly available components. This process phase is not including in the study of this AHT use case. **Tools** ***StkH1: RF Coil engineer***: StkH1 evaluates the design of the coil, based on the feedback, and may create a new version to solve issues, like reliability issues, End of Life problems or customer complaints.
Training & Education	**Activity** The pTX chain optimizer on the Arrowhead framework should be, if this all works as expected, be a good starting point for further improvements of the design process of other parts and chains of the MRI system. Therefor training how to use and extend the framework for other applications is important to inspire and allow others to adopt this. The MR application engineers are responsible for the training and education of the end users, to use the coil and provided scan techniques most effectively to satisfy the customer needs. **Tools** ***StkH4: MR Application engineer***: StkH4 creates training material for the MR Clinical User, via IfU (Instructions for Use) documents and additional presentations (power point and, occasionally, videos to demonstrate the optimal use.

The EPP phases 5, 6 and 7 (see table above) are certainly relevant for the complete EPP, but are not target for optimization with the AHT framework.

2.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 7 UC-02 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The data transfer between the various stakeholders is currently manual. If this is automated by the AHT-EP the engineering process can be significantly improved. By also arrange fast feedback from stakeholders, the improvement cycles can also be made much fast and of higher quality. Feedback data is analysed by the stakeholders for identifying rooms of improvement of the component for which each of them is responsible. Usually, the analysis drives the update of the requirements that trigger the following phases, with the aim of revising the original project and deploying a new enhanced release of the product.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The stakeholders in the current process merely work independently and sequentially, typically with long time intervals between their activities. The AHT-EP aims to have much more integration of their activities, due to the automated and fast feedback from stakeholders further in the development chain back to stakeholders earlier in the chain (e.g. from MR methods engineer back to the MR simulation engineer and to the RF coil engineer.
Obj. 3 - Handling of substantially increased number of I/O's	Since the modelling, data analysis and deriving of resulting control settings s currently very time consuming, not all technical options for the possible clinical applications can be optimized. An automated framework should allow a more thorough analyse

WP2 Objective	Focus & Planed actions
due to much more fine grained automation	of the coil, used for various human models and anatomical areas, resulting in a more dedicated and optimized set of parameters, used for clinical studies.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Each AHT-EP considers the development of documentation and reference manuals. The StkH 4, which will develop the user application, is in charge of producing tutorials and interactive tutorials for final users (StkH 5). These are also very useful for the internal stakeholders to get a better understanding of the applicability of their output.

The use case aims at automating the process the design process of MR coils. This objective is linked to the reduction of the EP costs, but has also a significant impact on the quality and on the safety of the final product.
A detailed list of actions planned to reach the use case specific objectives and technical measures will be provided in a later phase of the project.

Table 8 UC-02 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	The use case aims at automating the process the design process of MR coils. This objective is linked to the reduction of the EP costs, but has also a significant impact on the quality and on the safety of the final product.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Not yet identified.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Not yet identified.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Not yet identified.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and	Not yet identified.

Project Objective	Focus & Planed actions
automation solutions	
Obj. 6 - Training material (HW and SW) for professional engineers	Each AHT-EP considers the development of documentation and reference manuals. The StkH 4, which will develop the user application, is in charge of producing tutorials and interactive tutorials for final users (StkH 5). These are also very useful for the internal stakeholders to get a better understanding of the applicability of their output.

2.4 Engineering Process analysis

Table 9 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-02.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X	X				X	
OBJ-AHT #2	X		X						
OBJ-AHT #3			X						
OBJ-AHT #4			X						
OBJ-AHT #5			X						
OBJ-AHT #6			X	X				X	
OBJ-WP2 #1	X	X	X	X					
OBJ-WP2 #2	X	X	X	X					
OBJ-WP2 #3	X	X	X	X					
OBJ-WP2 #4			X	X				X	

3. UC-03 Integration of electronic design automation tools with product lifecycle tools (ULMA)

This use case is applied to whole development lifecycle of critical systems for IoT and SoS. More specifically, it focuses on the possibility of making a better reuse of physical models covering the abstraction, selection, representation and customization of system artefacts from the whole development lifecycle. The reuse of any system artefact goes beyond the mere discovery of potential reuse and it must focus on evaluating what can be reused (requirements, analytical models, descriptive models, test cases, etc.) when a match is discovered. To do so, quality usually has some impact since it is assumed that high-quality system artefact may help to improve the reusability factor of a system artefact. Furthermore, in this use case, there is another major objective focusing on the improvement of traceability to be able to keep trace from the very early stage of development to the final release of a complex product. Traceability is a critical activity to ensure that every system artefact exists for a reason.

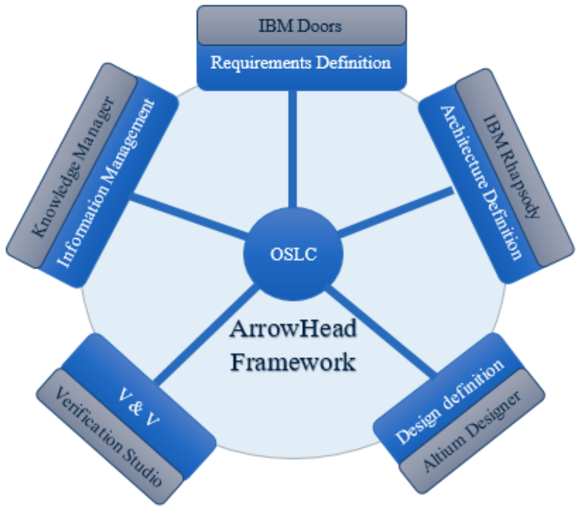


Figure 6 Tools developed in UC-03

The main motivation of the UC-03 relies then on being able to access and exchange different system artefacts (and tools) enabling a reusability layer based on interoperable standards.

The system artefacts will be accessed and exchanged by the following tools represented in Figure 6:

- IBM Doors for system requirement definition process
- Requirement Authoring tool for system requirement definition process
- IBM Rhapsody for Architecture Definition process
- Altium designer for Design Definition process.
- Verification Studio for Verification and Validation process.
- Knowledge Manager for Information management process.

All these tools will share the information via OSLC Standard (mainly through a specific resource shape for the System Representation Language-SRL, commonly named as OSLC KM, for those artefacts which are not covered by OSLC domains) to create and manage links between engineering processes artefacts.

3.1 Overall description of the UC-EP

The engineering process defined for this use case is based on the technical engineering processes of ISO 15288.

Then, different engineering methods and tools are used to implement the use case. More specifically, the mapping between the UC-EP engineering processes and those in the AHT-EP defined lifecycle is shown in Table 10.

Table 10 Mapping of UC-EP to AHT-EP processes within UC-03.

UC-EP Engineering process	AHT-EP	Tools

System Requirements Definition	Requirements	IBM Doors, Requirement Authoring tool
Architecture Definition	Functional Design	IBM Rhapsody
Design definition	Functional Design	Altium Designer
Implementation	Procurement & engineering, Training & education	Traceability Studio
Verification & Validation (Measurement process)	Procurement & engineering, Deployment & Commissioning	Verification Studio
Information Management	Not available	KnowledgeManager

In general, it is a matter of nomenclature. However, the main difference between UC-EP engineering processes and the AHP-EP is the notion of Information Management that in this use case, it is explicitly defined as a cornerstone to provide reuse capabilities and traceability management. More specifically, this case focuses on the concept of Knowledge-Centric Systems Engineering as a means to guide the engineering process exploiting all the data, information and knowledge that is generated during the development lifecycle and encoded in the system artefacts.

3.2 Engineering Process Description

In Figure 7 AHT-Ep of the UC-03 where:

- ***Stakeholder 1***: Is the Project manager. This stakeholder will define the project requirements in IBM DOORS tool. These requirements will be validated by the Stakeholder 4 (Q&A manager) using the Requirement Authoring Tool and will be stored in Knowledge Manager tool. In this process, some requirements could be reused using the Knowledge Manager tool.
- ***Stakeholder 2***: Is the Safety manager. This stakeholder will define safety related requirements in order to achieve the certifications. This requirements will be validated by the Stakeholder 4 (Q&A manager) by the Requirement Authoring Tool and will be stored in Knowledge Manager tool. In the process of requirement definition some requirements can be reused using the Knowledge Manager tool.
- ***Stakeholder 3***: Is the hardware designer. This stakeholder will collect all the requirements and will define the Functional design and Hardware design with IBM Rational Rhapsody tool and Altium Designer. The defined artifacts will be linked to requirements using Traceability Studio. All those artifacts will be stored in Knowledge Manager tool. In the process of the functional and hardware design definition some artifacts can be reused using the Knowledge Manager tool.
- ***Stakeholder 4***: Is the Q&A Manager. This stakeholder will verify the quality of system artefacts.

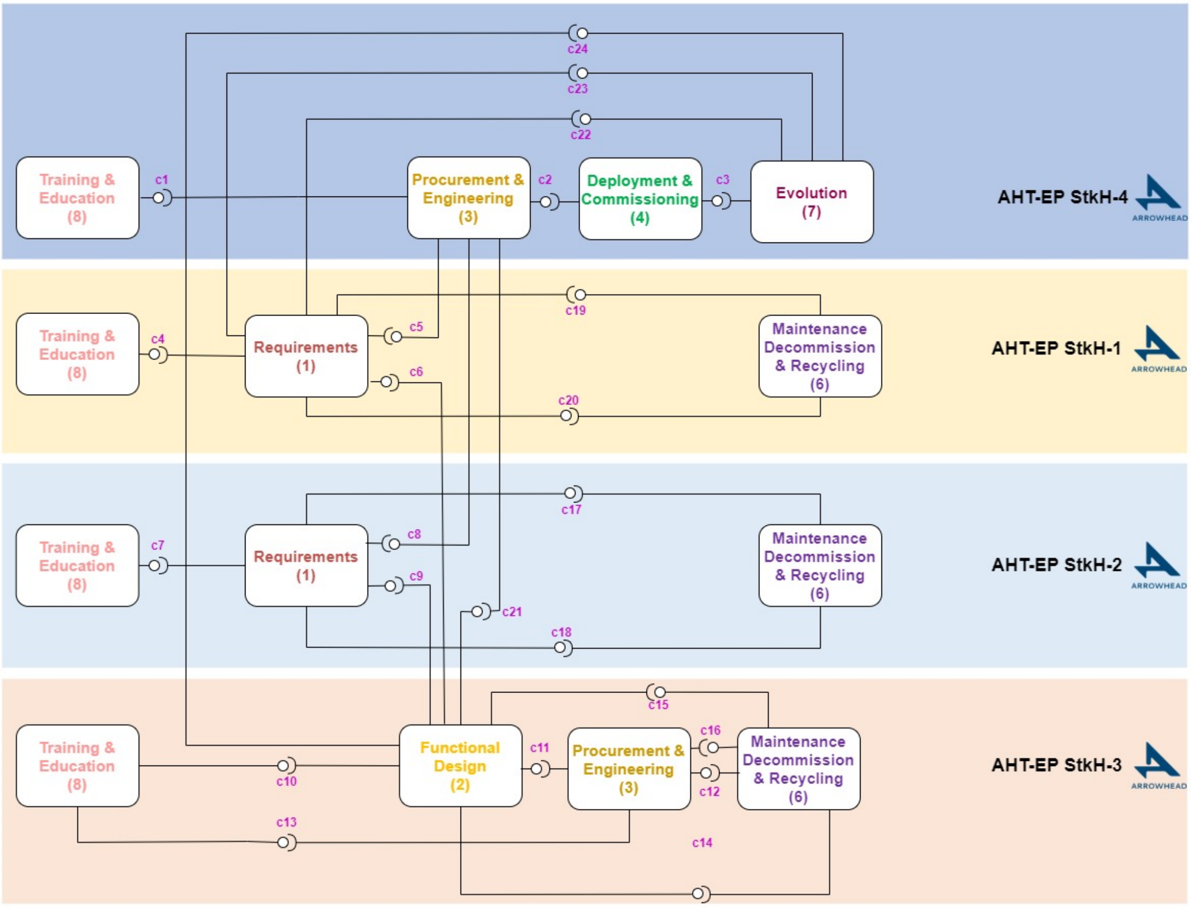


Figure 7 AHT-EP of use case UC-03

Those AHT-EP phases regarding *Maintenance* and *Evolution* are not explicitly defined in the UC-EP. This is mainly because the system under definition and construction will not be deployed in a production environment but in a relevant scenario (TRL 4-6).

Technology to implement the use case is generally available but not in the frame of Knowledge-Centric Systems Engineering. This means, common data, access and communication models for both data and operations in each of the tools that are part of the toolchain.

The use case is based on OSLC standard, which will use the arrowhead framework. Any tool that uses both of them will be able to connect to the toolchain described in this UC.

Without using AHT-EP, requirement management tool, hardware design tool and analysis of the design-requirement toolchain are not connected, so we cannot use the benefits of this solution.

The use of AHT-EP makes possible these connections, and makes the process of hardware design more agile.

Table 11 AHT-EP Phase focus of UC-03

Engineering process phase	Addressed/Focus
Requirements	**Activity**

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>System Requirements Definition. In this process, the objective is to specify the systems requirements that define the System of Interest. First, requirements are extracted from different sources. Then, they are analysed to build a first specification (document). Afterwards, a requirements verification process is performed to ensure the quality of requirements. Finally, the client validates the specification. Usually, the process is not linear and contains different feedback loops until the requirements specification is accepted. Additionally, this process requires some management related to traceability and configuration. This is a commonality in all processes.
Elicitation
Analysis
Specification
Verification and Validation (quality management)
Management (traceability and configuration management)
Tools
IBM Doors, Requirement Authoring tool</td></tr>
<tr><td>Functional design</td><td>Activity
Architecture Definition. Based on a verified and validated set of requirements, a first conceptual model is created to graphically understand the system. Afterwards and applying the 4+1 architectural view model, different views of the system are defined using a diagramming technique based on SysML/UML.
Conceptual modelling
4+1 architectural view model: use cases, process view, logical view, physical view and development view.
Verification and Validation (quality management)
Management (traceability and configuration management)
Tools
IBM Rhapsody

Activity
Design definition.
Detailed designed
Physical modelling
Verification and Validation (quality management)
Management (traceability and configuration management)
Tools
Altium Designer

Activity
Information Management.
Vocabulary definition
Taxonomy definition (semantic relationships)</td></tr>
</table>

Engineering process phase	Addressed/Focus
	• Pattern definition • Inference rules • Management (configuration management) **Tools** *KnowledgeManager*
Procurement & Engineering	**Activity** ***Implementation***. • Simulation and coding. • Verification and Validation (quality management) • Management (traceability and configuration management) **Tools** *Traceability Studio*
Deployment & Commissioning	**Activity** ***Verification & Validation (Measurement process).*** • Definition of quality metrics • Implementation of quality metrics • Management (configuration management) **Tools** *Verification Studio*
Operations & Management	**Activity** ***Information Management.*** • Vocabulary definition • Taxonomy definition (semantic relationships) • Pattern definition • Inference rules • Management (configuration management) **Tools** *KnowledgeManager*
Maintenance Decommissioning & Recycling	Those AHT-EP phases regarding Maintenance and Evolution are not explicitly defined in the UC-EP. This is mainly because the system under definition and construction will not be deployed in a production environment but in a relevant scenario (TRL 4-6)
Evolution	Those AHT-EP phases regarding Maintenance and Evolution are not explicitly defined in the UC-EP. This is mainly because the system under definition and construction will not be deployed in a production environment but in a relevant scenario (TRL 4-6)
Training & Education	**Activity**

Engineering process phase	Addressed/Focus
	Implementation. • Simulation and coding. • Verification and Validation (quality management) • Management (traceability and configuration management) **Tools** *Traceability Studio*

3.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 12 UC-03 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	This objective will be matched since in the UC we are adopting the evolution phase where we evaluate feedback data collected in the field (phase 4 of StkH4 AHT-EP). This data is analysed by the stakeholders for identifying rooms of improvement of the component for which each of them is responsible. Usually, the analysis done in the Evolution phase drives the update of the requirements that trigger the following phases with the aim of revising the original project and deploying a new enhanced release of the product.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	In this UC we represent each of the actors involved in the Digital Hardware design as a stakeholder. All the engineering phases are intended for a single stakeholder that can produce more than one component. All the EPs are integrated and connected between each other internally and externally (different StkH). Connections are not fully automated and several of them are manual. During the project we will automatize some of these connections by interfacing the modules with and without the AHT Framework.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Several phases of the EPs have multiple interactions with the other phases of the same EP and EPs from other stakeholders. Interactions, represented in the AHT-EP model with connection lines, are implemented by exchanging information between tools contained and operating in each of the AHT-EP phases. This exchange of information is made automatically using OSLC for some phases or manually for others.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Each AHT-EP and each stakeholder considers the development of documentation and reference manuals.

Table 13 UC-3 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	With the intermediate layer developed in the use case, the access to different artefacts will be unified. Thus, the number of connections that are needed to carry out the engineering process keeping the quality of the final product will be decreased significantly. This objective nowadays is not achieved but we have been working hard to achieve the following milestones: • Implementation of the OSLC layer on top of Altium Designer. • Validation of each service: The aim of this subtask has been to test functionalities of the OSLC layer on top of the different tools, in both cases, providers and consumers. We have tested the functionality of the OSLC layer, in IBM doors and Altium Designer tools. • Validation of the OSLC specifications: The aim of this subtask was to guarantee that the OSLC provider follows the specification of the target OSLC domain to make possible the links between tools. We have validated that the IBM doors and Altium Designer tool follow the specification of the target OSLC domain. In conclusion, it will be possible to link those tools using OSLC standard.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Make easier the interoperability between tools using OSLC and Arrowhead Framework. Reduce the effort to connect tools.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	In this use case we will use legacy data. We will integrate this into arrowhead framework using OSLC standard.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Same as obj 2: arrowhead framework and OSLC standard.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Not defined.
Obj. 6 - Training material (HW and SW) for	Creation of tutorials, manuals, videos, etc to explain how to use the toolchain.

Project Objective	Focus & Planed actions
professional engineers	

3.4 Engineering Process analysis

Table 14 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-03.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	P	C	C	C	N
OBJ-AHT #1	X	X	X					X	
OBJ-AHT #2		X	X						
OBJ-AHT #3			X					X	
OBJ-AHT #4		X	X						
OBJ-AHT #5									
OBJ-AHT #6	X	X	X					X	
OBJ-WP2 #1	X		X				X		
OBJ-WP2 #2	X	X					X		
OBJ-WP2 #3									
OBJ-WP2 #4	X	X	X					X	

4. UC-04 Interoperability between (modelling) tools for cost-effective lithography process integration (ASML)

ASML is the world's leading provider of complex lithography systems for the semiconductor industry. The design process of these Cyber-Physical Systems of Systems involves multi-disciplinary engineering teams focusing on functional specification and verification of scenarios and mono-disciplinary engineering teams focusing on the realization of these scenarios in a platform composed of mechanical, optical, electrical and software components.
The objective of this use case is to develop a tool chain, mainly focussing on the *functional design* phase that establishes seamless interoperability between modelling tools to facilitate multi-disciplinary engineering teams providing functional engineers with an efficient feedback loop to develop and qualify calibration, performance and diagnostic test scenarios.

The use case consists out of two toolchains shown in Figure 8:

1. CIDT: Calibration Performance and Diagnostics Integrated Development Toolkit; Connectors c01-c35 in Figure 9.
2. SSVP toolchain: Specification, Synthesis, Verification and Product Line Engineering; Connector c36 in Figure 9.

CIDT is considered TRL8 (commercial, main contributions ICT + ASML)

SSVIP is considered TRL4 (lab experiments, main contributions TU/e + ASML)

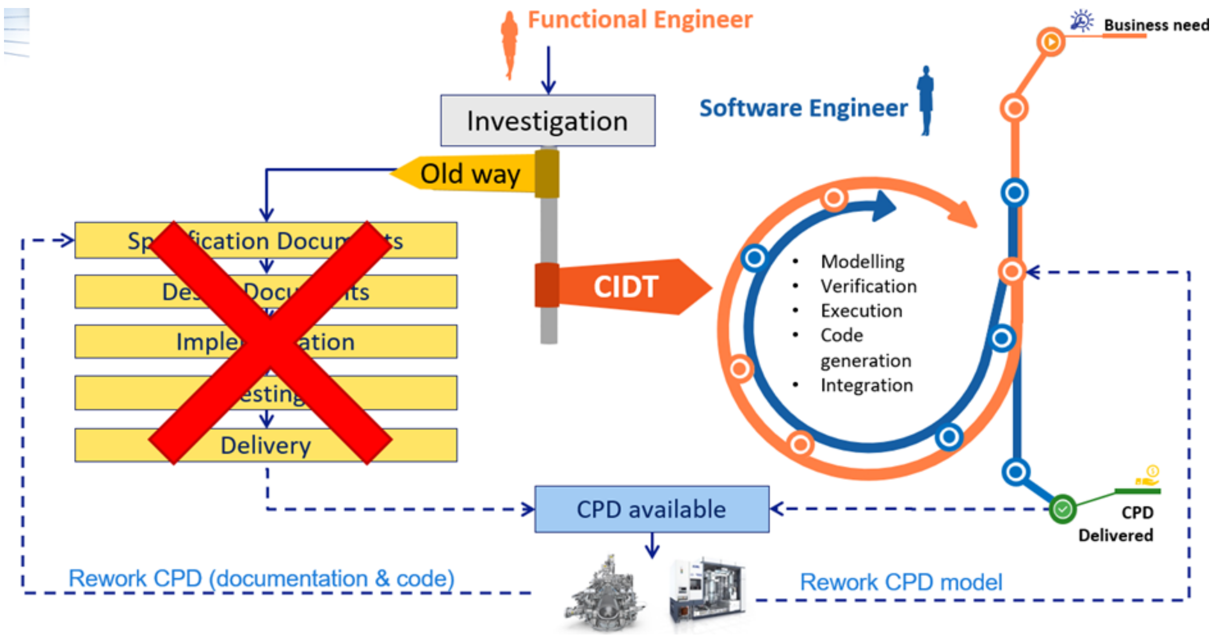


Figure 8 Toolchains of UC-04

4.1 Overall description of the UC-EP

All of the AHT-EP phases are applicable to the use case as it encompasses the total software life-cycle of calibration, performance and diagnostics test scenarios.
The AHF toolchain will strongly improve the quality and duration of the CPD development process as manual and error-prone handovers between the phases will be automated. This applies to the functional design and engineering phases, but also the commissioning and maintenance phases.
The executable requirements designed and built in the design phase have to be converted into usable code, and vice versa. This technology is not available and has to be developed.
Additionally the Mathworks tooling is not designed to communicate in real time with a target machine, hence a new component has to be developed to support this.
To reduce manual efforts in scenario modelling, we aim for modular and composable specifications, and use PLE techniques to enable re-use of specifications amongst different product families.
Supervisory controller synthesis suffers from scalability issues. To mitigate this, we use modular and distributed synthesis techniques i.s.o. monolithic synthesis, and attempt to use PLE-techniques to allow for re-use of (partial) synthesis results.
Formal analysis techniques such as model checking suffer from scalability issues due to the state-space explosion problem. To mitigate this, we use compositional model checking techniques next to abstraction techniques.
The creation of the model is checked by a Model Advisor. The off-the-shelves advisor of Mathworks will be extended with specific checks and advices. We will loosely use automotive standards as Modelling Standards for MISRA C:2012.

4.2 Engineering Process Description

In Figure 9Figure 7 AHT-EP of the UC-04 where. System engineering retrieves feedback from customer support (c30) and get performance requirements for the next machine (c04). System engineering decomposes these requirements toward several submodules using Word (c01, c06, c07, c08, c11). The different submodules are handled by the functional engineer that sharpens requirements by using MATLAB. The functional engineer together with the software engineer uses MATLAB/Simulink Stateflow and CIDT to make one model containing the whole model (c19, c20, c21). The CIDT Tool consists of interface generation, adapting generated code, generated build artifacts, documentation, model checks so that generate code will act as any other component and consequently can interact with other components. (c12, c13, c14, c15, c21, c22, c23)
The SSVP toolchain, consisting out of tools LSAT, CIF, mCRL2, PLE Tool, and model translation tool, is used by the functional engineer (c36). The work flow is such that a system product line is specified using LSAT. After configuring a product in PLE tool a system specification is derived. This specification is used by SDF3 to analyze the timing aspects of the system such as throughput making sure that the specification conforms to the timing requirements. The LSAT model is then translated to mCRL2 for formal verification of system requirements, and to CIF for supervisory controller synthesis.

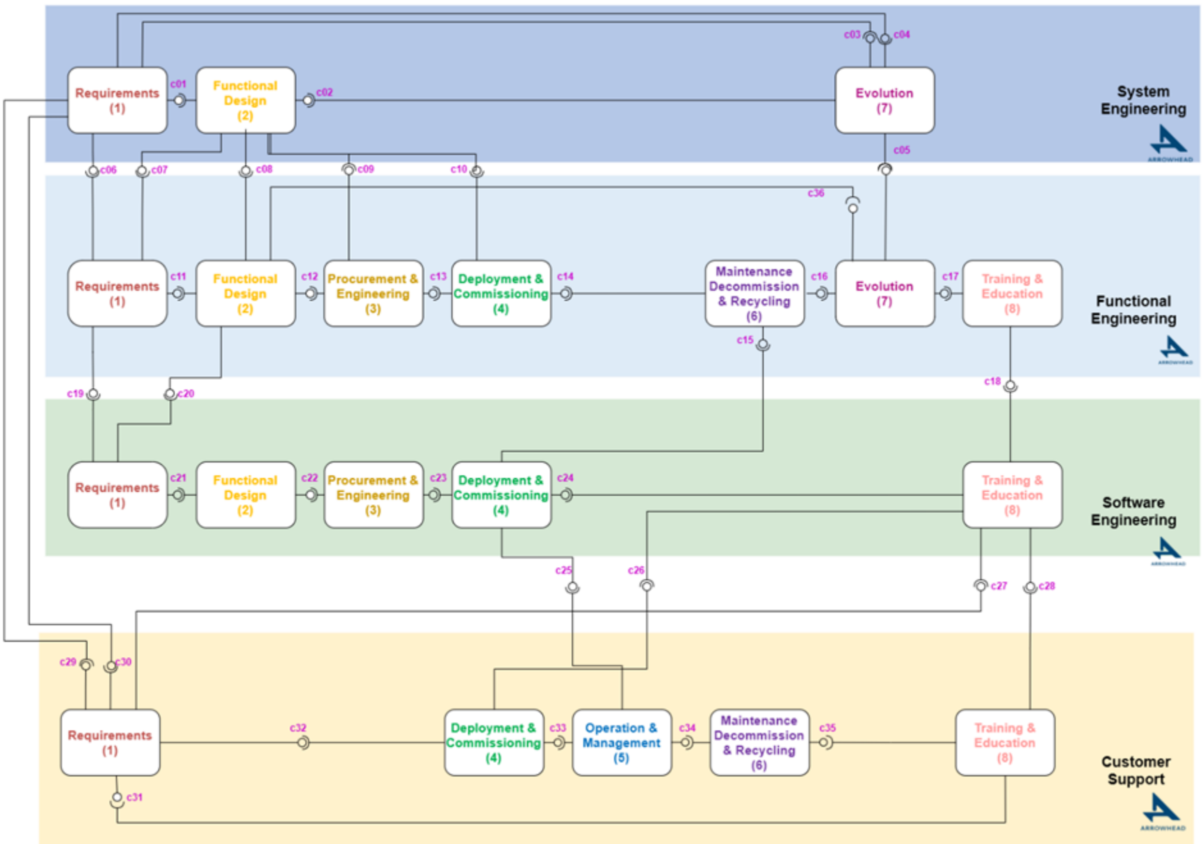


Figure 9 AHT-EP of use case UC-04

SSVP: Only the functional design phase is considered in the domain of the Use Case.
The use of integration of various tools that were previously stand-alone has increased the multi-disciplinary aspect of the functional design phase. The AHF provides a platform that automates the distribution of various intermediate design formats, as well as automation of various tasks that prior to integration included one or more manual steps.

Table 15 AHT-EP Phase focus of UC-04

Engineering process phase	Addressed/Focus
Requirements	**Activity** StkH2: MatLab is used as scratch book, to discover requirements **Tools** *CIDT MatLab* • Input: Design specifications • Output: Generated code and documentation
Functional design	**Activity** 1) StkH2 -> StkH3: These unprecise specifications are manually translated to a software specification during the procurement and engineering phase 2) Systems implemented by StkH1 are captured in a model by StkH3 and StkH2. 3) StkH3 and StkH1 can use the model in the LSAT component to analyze the system and provide timing analysis and guarantees. 4) Systems implemented by StkH1 are captured in a model by StkH2. System requirements specified in Word documents by StkH3 are formalized by StkH2. CIF is used to synthesize a safe supervisory controller for the system. 5) Systems implemented by StkH1 are captured in a model by StkH2. System requirements specified in Word documents by StkH3 are formalized by StkH2, and verified. 6) Variability information of the product line is captured in a model by StkH1, StkH2 and StkH3. 7) For baseline, models are manually translated by StkH1. An automated tool is being developed **Tools** 1) *CIDT Word/Visio*: • Input: Specifications and requirements • Output: Generated code and documentation 2) *LSAT Component*: System is specified by StkH3 in the tool • Input: Word documents and informal descriptions of the system • Output: Timed LSAT model 3) *SDF3 Component*: • Input: Timed LSAT model • Output: timing analysis and timing guarantees for the input model 4) *CIF Component*: • Input: Models from LSAT and requirements from StkH3, implemented by StkH1 • Output: Safe supervisory controller for the input system 5) *mCRL2 Component*: StkH1 can verify the requirements of StkH3 directly from their tooling.

Engineering process phase	Addressed/Focus
	• Input: requirements • Output: verified models and generated software 6) *PLE Component*: • Input: Models from LSAT • Output: Derived LSAT model according to variability configuration 7) Model Translation Component • Input: Models from LSAT, PLE models • Output: Models for LSAT, CIF3, SDF3, and mCRL2
Procurement & Engineering	**Activity** StkH3->4: use Linux patching to deploy to the field. **Tools** *CIDT Linux patching* • Input: Generated code and specifications. • Output: Integrated code
Deployment & Commissioning	This phase is not in focus of the UC. However, it may trigger a need to change the requirements, update the functional design, and/or repeat activities done in the phase of procurement and engineering – then the activities described above are again performed.
Operations & Management	**Activity** StkH2->4: the scenarios are captured in field procedures in Word **Tools** *CIDT Word/Visio*
Maintenance Decommissioning & Recycling	**Activity** StkH3: on the software the scenario is retrieved from ClearCase and updated **Tools** *CIDT ClearCase*
Evolution	Not Declared
Training & Education	**Activity** StkH4: field procedures **Tools** *CIDT Word/Visio*

4.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 16 UC-04 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	CIDT: AHT-EP triggered the usage of a Stateflow/Simulink/MATLAB model that is connected to a live machine. So feedback from a live machine is used during development. Also feedback from the field is incorporated in the next version of the components created. SSVP: For the Functional Design EP, a transformational synthesis algorithm and variable ordering heuristic have been developed to reduce computational effort of supervisory controller synthesis in CIF. This method can deal with product families with large amount of configurations and with creating supervisory controller solutions for various tasks. These are now being implemented. **Planned actions** • AHT-EPP 1: Even during requirements phase a live connection to the machine can be made to check responses. • AHT-EPP 2: A live connection to the machine or simulator can be made to check results. • AHT-EPP 3: A live connection to the machine of simulator can be made to check results. • AHT-EPP 4: Jenkins will make that you start with an empty but integrated piece of code. That stays integrated during previous phases.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	CIDT: ATH-EP brings multi stakeholder automation, within this UC we uses Stateflow/Simulink/MATLAB model is used a single communication language between multiple stakeholders. Consequently multiple views can be generated from one single source. Such as generated source code, documentation for custom support, documentation for maintenance. SSVP: For the Functional Design EP, the integration of the design, optimization, synthesis, verification and PLE techniques, allows multi-disciplinary tasks to be performed by single-disciplinary stakeholders without advanced intervention by domain experts. The actions involving integration, consist of planned multi-domain tasks as well as an automated translation between domain models. In CIF, the use of feature models was investigated to describe variability in system features and scenarios has been investigated. In LSAT, the integration of feature models as well as state charts are being investigated to facilitate specification and analysis of multiple scenarios on a system. LSAT will also incorporate event feedback data back to the supervisory controller to facilitate error-handling and more refined automated control of systems. **Planned actions** • AHT-EPP 1: From the requirements multiple stakeholders start to communicate from and within on model. • AHT-EPP 2: Functional design can be made from within the single model that is shared between phases 1..3 • AHT-EPP 3: Again the model is the single source to generate code and to generate integrated code. • AHT-EPP 4:
Obj. 3 - Handling of substantially increased number of I/O's due to much	CIDT: Out of scope for this use case. In general this is handled by (many) abstraction layers.

WP2 Objective	Focus & Planed actions
more fine grained automation	SSVP: For the Function Design EP, a transformational synthesis algorithm and variable ordering heuristic have been developed to reduce computational effort of supervisory controller synthesis in CIF. These are now being implemented. Investigation of state charts to drastically reduce the model size and facilitate scalability issues is planned for LSAT.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Training and learning are integrate part of ATH-EP with this case we invested from the beginning in good training for the engineering phase by providing wiki, video, as well as extended model and code checkers, next to the dedicated support team. Also the documentation for the field is always up to date as it is generated from the same model. SSVP: For the Functional Design EP, LSAT currently has learning material in the form of documents and tutorials. Future features will also include this level of training. SDF3 is being used in course material (TUE/Electronic Design Automation, TUE/Computational Modeling) for modelling and analysis of systems, presented as an online service for students and in a course where mCRL2 is used in verification of a system (TUE/System validation).

Table 17 UC-04 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	CIDT: The communication between stakeholders, the machine and the live machine will give us a reduction of about 20%. Generated code and integrated code will give us another 10%. Using the same tools during updates during maintenance phase will give another 5%
Obj. 2 - Interoperability for IoT and SoS engineering tools	CIDT: Tools within the engineering and outside of the will be made interoperable. SSVP: • Create interoperability between legacy automation engineering tools with more powerful verification and analysis tools, to provide diagnostics for detection and correction of potential undesired operation of systems, before it reaches to more costly testing phase of the design. • In CIF, the use of feature models was investigated to describe variability in system features and scenarios has been investigated. This is to be further researched and implemented, focus on researching dynamic feature configuration using this method. • Also, a transformational synthesis algorithm has been developed that uses previous synthesis result and a model delta to more efficiently come up with updated supervisors, which is now being implemented in CIF. The effectiveness of this method is further investigated. • A variable ordering heuristic has been investigated to reduce computational effort of symbolic supervisor synthesis. This method will be extended with a event ordering heuristic to further decrease the computational effort. • A method for efficient supervisor synthesis for models containing mutual state exclusion invariants was established. • The proposed method of integrating event feedback in LSAT will result in a reduction of engineering cost by formally defining the events and their interactions at the system level, resulting in less ambiguity in the specification and a more accurate analysis of the system.

Project Objective	Focus & Planed actions
	• State charts are also projected to reduce the time and costs to specify and analyse complex systems by creating a hierarchy of system models that result in more levels automation in synthesis and analysis steps, as well as facilitating the way to a design automation for flexible manufacturing systems in LSAT. • The methods to integrate PLE techniques with current tools are being investigated. These PLE techniques will provide the infrastructure for modelling variability in the product line which will support the system artefacts reuse. • A model translation component integrated in the AHF will be developed to manage model-to-model conversions and communicate between the tools in the use-case.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	CIDT: Tools and generated code should act as a legacy component to be part of the whole system. SSVP: • A method for translation from LSAT models to CIF models has been established. This is being put in a paper for dissemination and being implemented • A method for translation from LSAT models to mCRL2 models has been established. • A method for integrating variability models with LSAT models is being developed. • A model translation component integrated in the AHF will be developed to manage model-to-model conversions and communicate between the tools in the use-case.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	CIDT: Jenkins is used to integrate and produces different artefacts like generated code, generated documentation, and generated patches. SSVP: The integration of multi-disciplinary tasks in the design tools, as well as providing automatic translation between domain-specific models used for synthesis, optimization, verification and managing product line configurations.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Not applicable.
Obj. 6 - Training material (HW and SW) for professional engineers	CIDT: Training material is created for the engineering process. SSVP: • Assisting in teaching an academic course where CIF is used for modelling a production system. (TUE/Model based system engineering) • SDF3 is being used in course material (TUE/Electronic Design Automation) for modelling and analysis of systems, presented as an online service for students. • mCRL2 is being used in course material (TUE/System validation) for modelling and verification of systems by students. • For the PLE tool under development, proper training material will be

Project Objective	Focus & Planed actions
	developed.

4.4 Engineering Process analysis

Table 18 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-04.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X					X	
OBJ-AHT #2	X	X	X						
OBJ-AHT #3		X	X						
OBJ-AHT #4			X	X			X		
OBJ-AHT #5									
OBJ-AHT #6		X	X	X				X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2	X	X	X	X					
OBJ-WP2 #3		X	X						
OBJ-WP2 #4		X	X	X				X	

5. UC-05 Support quick and reliable decision making in the semiconductor industry (KAI)

To support quick and reliable decision making in the semiconductor industry, three self-developed existing tools will be improved and implemented in the course of the project. They are TePEx, WHF and DR.

- TePEx: (Test pattern extraction): An algorithm which is able to detect test patterns, which are related to malfunctioning testing equipment. With TePEx, malfunctioning wafer testing equipment is detected before the yield is affected.
- WHF (Wafer health factor): An algorithm which is able to detect process patterns, which are related to deviations during production. WHF is used to rate each wafer regarding its health. WHF is based on an ML-pipeline, which automatically detects and classifies each wafermap regarding pre-defined critical process patterns.
- With WHF, critical process patterns are detected at an early stage, before yield loss occurs.
- DR (Digital Reference): The Digital Reference is a Semantic Web Representation of the Supply Chain to guarantee interoperability as it creates an abstraction layer that defines concepts and relationships between heterogeneous data sources. The Digital Reference is a generic approach, allowing interconnectivity, to enable sharing and integration of information, data bases and tools.

5.1 Overall description of the UC-EP

UC-05 deals with different tools, supporting a quick and reliable decision making process within the semiconductor industry. The whole workflow is depicted in Figure 10.

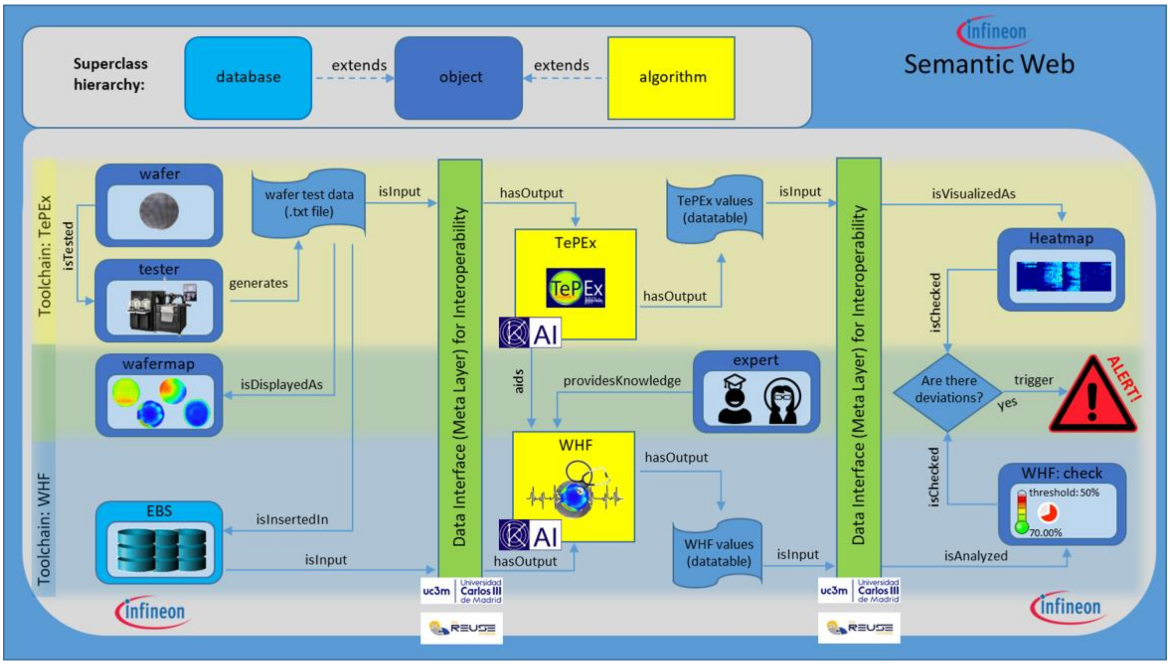


Figure 10 UC-05 overview.

Starting point of Figure 10 is the object called *wafer* in the left upper corner. A wafer undergoes hundreds of process steps until it is finally processed, i.e. single devices on the wafer are ready for usage. But before the wafer is sawed into single devices, they are electrically tested on a *tester*. The output, summarized as *wafer test data*, is written into .txt files and transferred into the *EBS* database. The visualization of each electrical measurement is called a wafermap. Some of these *wafermaps* show patterns, which want to be detected. Therefore, two algorithms can be applied, called *TePEx* (Test Pattern Extraction) and *WHF* (Wafer Health Factor) because they provide insight into testing issues or process deviations, respectively. The input data for both algorithms are *wafer test data* and additional information stored in the *EBS*. With a *data interface layer* it must be guaranteed, that the data points are adequately available for the algorithms. Afterwards, the algorithm's output is written into *data tables*, providing the input information for the visualization. Also here, a *data interface layer* must guarantee that the data are adequately available for the visualization. While a *heatmap* is generated as output for TePEx, a value between 0 and 100 is given for the WHF, both indicating the affectedness of a wafer by a test or process pattern, respectively, triggering an alarm in case of deviations.

DR is mainly located at the data end and with its incorporated semantic web features it is providing standardization (single entry point for heterogeneous data sources), enhanced inference mechanisms (enrichment of existing data), advanced data quality (inconsistency identification), as well as machine and human interpretability (extensive meta structure for interoperability). As a more focal excerpt with regards to the semantic web part, its relationship to the Eclipse Arrowhead framework and specifically for UC05, this may look like the interconnection structure represented in

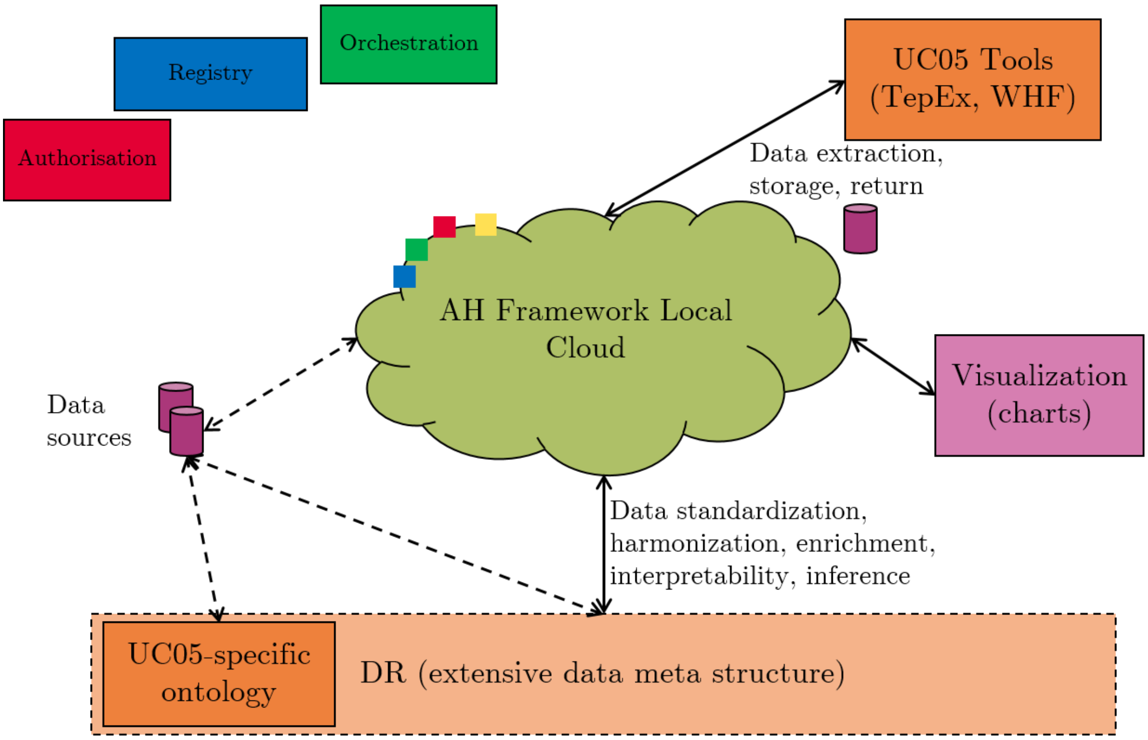


Figure 11.

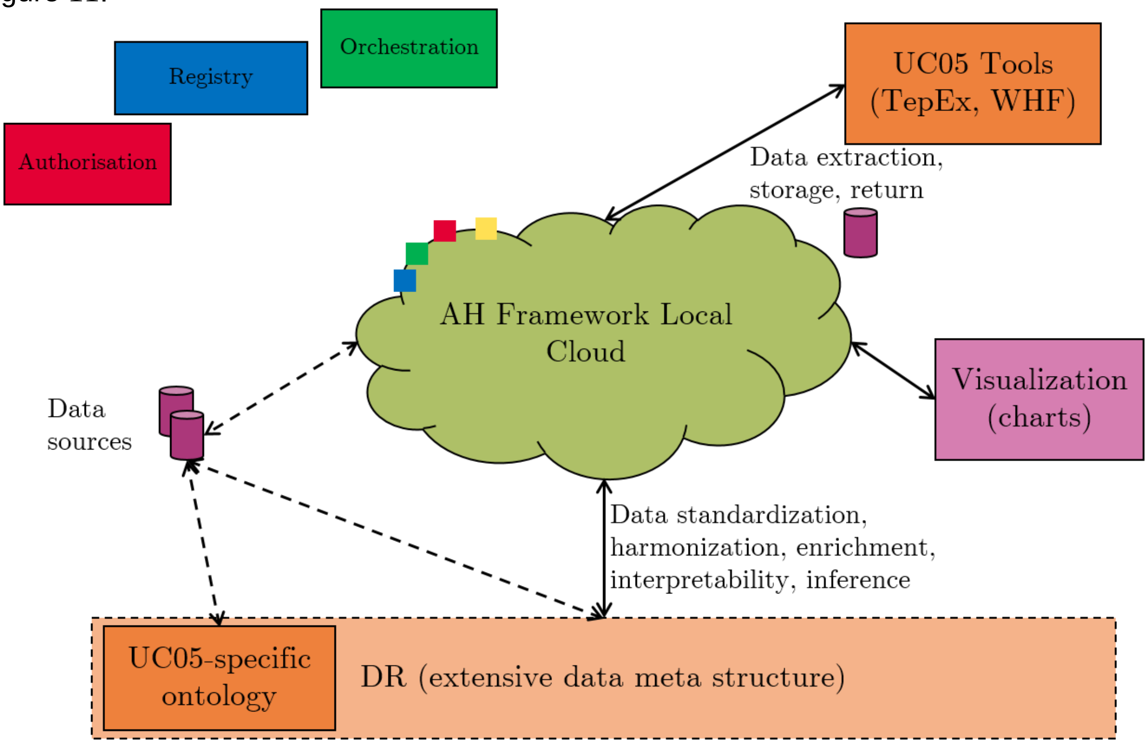


Figure 11 UC-05 tools in the Eclipse Arrowhead framework.

Current coverage of the UC-EP phases for TePEx and WHF:

The algorithms TePEx and WHF have been developed in SemI40, are available, but act as stand-alone tools, i.e. an extra software application (implemented algorithm, equipped with a GUI), which can be used by the user parallel to his daily work. The feasibility of the tools for a limited amount of product has already been proven but the scalability to other products and other company sites is still open. Hence, as a first step within the project, the tools have to be evaluated on a broad product portfolio, to prove their generalization and to detect further possible improvements, based on core user feedback. Then, as the second step, the tools improvements are implemented, but still in the "stand-alone version". Therefore, as a last step within this project, the tools are provided in a way that the final integration into the toolchain, or more specific, the tools output (results of the algorithm) can be implemented in the productive environment. This already touches all engineering phases, whereas the main focus within the Arrowhead Tools project is on *Engineering & Procurement and Deployment*.

Current coverage of the UC-EP phases for DR:
The Digital Reference is a tool developed in Productive4.0 that models semiconductor supply chains and supply chains containing semiconductors conceptually in a Semantic Web representation. With this extensive pre-work in place, the further development in the course of the AH Tools project hence can be described as another engineering iteration. The engineering phases covered in this use case hence contain requirements definition, functional design, model development, instantiation (with data) for this the specific ontology for UC-05.
In Figure 12 the old engineering process of the UC-05 before the AHT-EP model have been adopted.

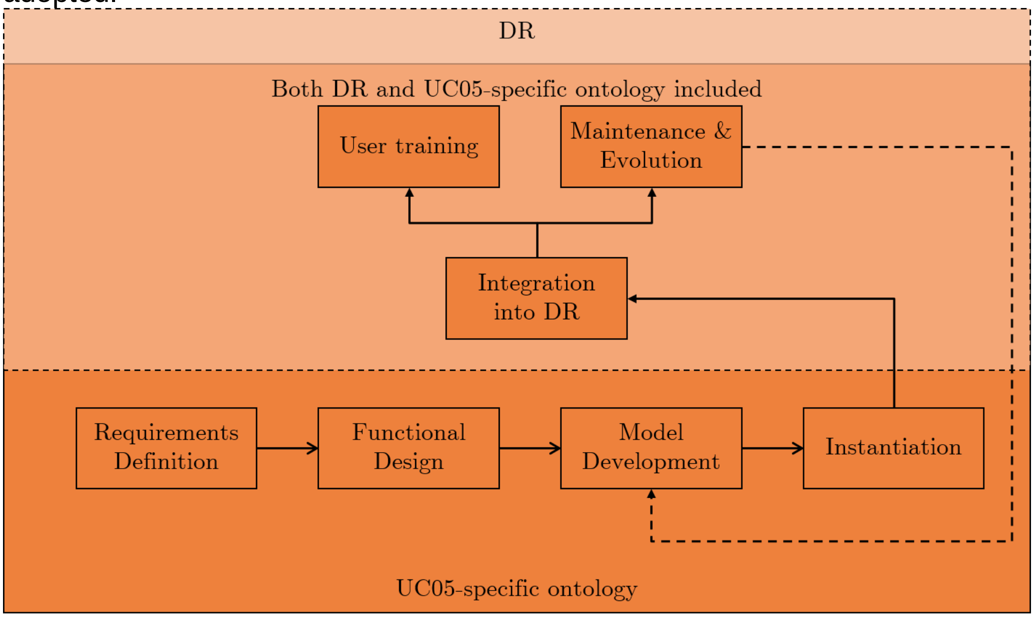


Figure 12 Old engineering process of UC-05.

Partners within UC-05 closely focus on special tool (TePEx, WHF, DR, UC-05 specific ontology) development. No special framework has been used so far. Within the AHT project, partners got in contact with formal engineering processes for the first time.

5.2 Engineering Process Description

The UC-05 engineering process, represented in Figure 13, has 3 stakeholders, following the AHT-EP and 2 further Stakeholders, where the EP is unknown to us. Main stakeholder is StkH-1, which is the algorithm developer. So, those who are developing the two algorithms TePEx, WHF and the ontology DR.
The second stakeholder is the semiconductor industry, which is the end-user of the developed algorithms from StkH-1.

As a third stakeholder we mention here the Arrowhead Framework (AHF), which is on the one hand the user of the product from StkH-1 and at the same time provider of this service for StkH-2. It can be seen as possible connection/service provider between StkH-1 and StkH-2. After consultation with AHF experts within the Arrowhead Tools (AHT) consortium it has been decided, that at that stage of development (i.e. within the timespan of the AHT project), an integration is not suitable. AHF concept for TePEx, WHF and DR at a more mature point in time might be suitable (beyond AHT project).
StkH-4 and StkH-5, for whom the EP is unknown to us, are for instance the supplier of semiconductor equipment and related IT infrastructure.
StkH-1 is the main stakeholder in UC-05.
StkH-1 develops artefacts such as algorithms or ontologies (in case of UC-05, these are TePEx, WHF and DR) and then provides it or delivers it to further StkHs.

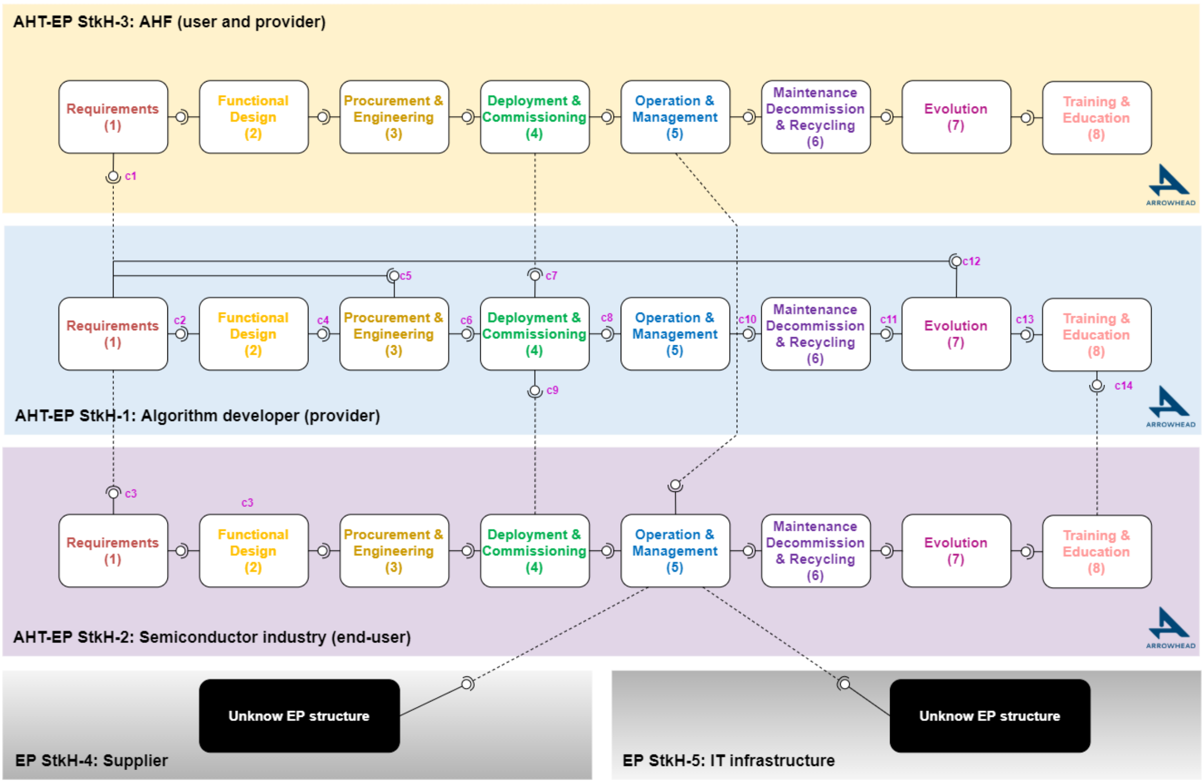


Figure 13 AHT-EP of use case UC-05

Generally speaking, all EPPs are to some extend considered in UC-05 and at related StkHs. Simply the planned activities within the AHT project do not cover each EPP in detail but focuses on specific ones as detailed in the following table.
For UC-05 specific ontology and DR the EP is very similar to AHT-EP, so no major changes/adoptions. Some phases do not apply in full extent, such as: procurement (rather already defined), commissioning (rather already defined), Decommissioning/recycling (not applicable).
In this UC there is no focus on standards within the AHT project.

Table 19 AHT-EP Phase focus of UC-05

Engineering process phase	Addressed/Focus
Requirements	**Activity** Requirements (c1, c2, c3): First of all, requirements are collected from all StkH. Thinking in the direction of providing a tool for data analytics this can mean for instance, to have a sufficient number of data available. Further, a requirement from StkH-2 is e.g. to have a timely efficient runtime of the algorithms. **Tools** 1. *UC05-specific ontology*: Define requirements based on documents and expert interviews. 2. *DR*: n.a. 3. *TePEx toolchain*: StkH1-3 define requirements. 4. *WHF toolchain*: StkH1-3 define requirements.
Functional design	**Activity** Functional design (c4): Here, the concept is created, which means that dependent on the requirements, different methods might be applicable. **Tools** 1. *UC05-specific ontology*: DrOWLings and basic modeling tools are used to gain a general understanding of the structure. 2. *DR*: only situative (for merging preparation). 3. *TePEx toolchain*: StkH1 builds single software blocks of TePEx. 4. *WHF toolchain*: StkH1 builds single software blocks of WHF.
Procurement & Engineering	**Activity** Procurement & Engineering (c5, c6): In this phase, the algorithm itself, the implementation so to say is build. Here, a first back-loop to the requirements is inserted (c5), which have to be constantly checked regarding their fulfilment or, if needed, can be adapted. **Tools** 1. *UC05-specific ontology*: Model development and instantiation. 2. *DR*: n.a. 3. *TePEx toolchain*: StkH1 builds the algorithm (TePEx) itself. 4. *WHF toolchain*: StkH1 builds the algorithm (WHF) itself.
Deployment & Commissioning	**Activity** Deployment & Commissioning (c7, c8, c9): To make the algorithm applicable to others, graphical user interfaces can be provided. When reaching a sufficiently good maturity level, the implementation of the algorithms are handed over to StkH-2 and StkH-3, afterwards following their own specific EPs. **Tools**

Engineering process phase	Addressed/Focus
	1. *UC05-specific ontology*: Integration into DR. 2. *DR*: Integration into DR. 3. *TePEx toolchain*: Making TePEx applicable @ StkH1-3. 4. *WHF toolchain*: Making WHF applicable @ StkH1-3.
Operations & Management	**Activity** Operation & Management (c10): In this phase, the application is in a kind of "productive" usage. For TePEx, WHF and DR this means that the developed model (the model is the outcome of the algorithm) is applied to datasets which want to be investigated. Mainly StkH-2 is active in that EP because StkH-2 is the end-user. Optional, parts of it can also be taken over by StkH-3. **Tools** 1. *UC05-specific ontology*: WebVOWL for visualization. 2. *DR*: WebVOWL for visualization. 3. *TePEx toolchain*: Productive usage of TePEx @ StkH1-3. 4. *WHF toolchain*: Productive usage of WHF @ StkH1-3.
Maintenance Decommissioning & Recycling	**Activity** Maintenance, Decommission & Recycling (c11): Regarding maintenance of such an application, in this case, an algorithm, one has to think about needed updating concepts in case of any influencing environmental changes. Further, also versioning concepts have to be considered here. **Tools** 1. *UC05-specific ontology*: WebVOWL for maintenance. 2. *DR*: WebVOWL for maintenance. 3. *TePEx toolchain*: Maintenance concepts @ StkH1 and StkH2. 4. *WHF toolchain*: Maintenance concepts @ StkH1 and StkH2.
Evolution	**Activity** Evolution (c12, c13): During the evolution phase, improvements of the existing procedure can be made. This includes, for instance, an extended functionality of the algorithm, debugging, but also to consider and integrate first user feedback. Usually, the analysis done in the Evolution phase drives an update of the requirements that triggers again all following phases (c12). **Tools** 1. *UC05-specific ontology*: WebVOWL for evolvement. 2. *DR*: WebVOWL for evolvement. 3. *TePEx toolchain*: Improvement of existing procedure @ StkH1. 4. *WHF toolchain*: Improvement of existing procedure @ StkH1.
Training & Education	**Activity** Training & Education (c14):

Engineering process phase	Addressed/Focus
	In the last phase, information on the developed algorithm is documented by StkH-1. This means, the purpose, the used data, meta-data and structure, tips&tricks, the usage, or simply, what it can/what it doesn't can is documented, for instance in form of a user manual or handbook, which provides important information to StkH-2, the end user of the developed algorithms. **Tools** 1. *UC05-specific ontology*: WebVOWL for training. 2. *DR*: WebVOWL for training. 3. *TePEx toolchain*: User training @StkH1 and StkH2. 4. *WHF toolchain*: User training @StkH1 and StkH2.

5.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 20 UC-05 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	While the artefacts (TePEx, WHF, DR) are designed kind of "off-line" at StkH-1, in the course of this project we want to go one step further into the direction of making them a run-time application. This means in concrete terms that: StkH-1 develops and improves the algorithms (this happens in Phase 3 and Phase 7) and delivers the results to StkH-2 and StkH-3 (this happens during Phase 4, which means to deploy the algorithms to the users).
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	There are multiple StkHs involved in UC5. StkH-1 are the providers or developers of the algorithms used in UC5 with the aim to support decision making in the semiconductor industry. Each algorithm developed here has its own requirements but additionally has to consider also the requirements from StkH-2 and StkH-3. StkH-2, the semiconductor industry is the end-user of the algorithms. As soon as the StkH-1 enters its deployment phase, a first handover to StkH-2 can be performed, who then has to integrate the algorithms into their own toolchain, again consisting of their own EP and further related stakeholders like suppliers of semiconductor equipment (shown as StkH-4) or available IT infrastructures (as StkH-5). StkH-3 is the AHF which is user and provider at the same time. It is the user from StkH-1 (this means that it gets information or input from StkH-1) and provides output to StkH-2. Globally speaking, this means that StkH-2 can get and integrate the information directly from StkH-1 or can use StkH-3 (the AHF) as a service in between.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Each EPP at least interacts with a second EPP. Some phases have multiple interactions within, but also across different EPs from different stakeholders. These interactions are marked as connection lines ("c") between the EPPs. For a flawless interaction between different EPPs, for each EPP, input and output has to be defined. While some of the I/O interfaces already exist (especially within the EPPs from one StkH), I/O interfaces across StkHs have to be created in the course of this project. For instance, using the AHF as a new tool within this pipeline, suitable I/O structures have to be defined (or even created), to communicate with StkH1 and StkH2.

WP2 Objective	Focus & Planed actions
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Each EP from every AHT Stakeholder includes phase 8, the Training & Education phase. For StkH-1 for instance, this means to document the functionalities of the developed algorithms and to provide a user manual, including recommendations on the usage and highlighting the purpose of the algorithms, for instance, the application area, the reason what it is made for, but also mentioning possible limitations. This is especially important to guarantee a satisfying and reliable outcome of the analysis. This user manual is handed over to StkH-2. There, it might has to be adapted or extended dependent on additional information which might be important for the end-user, or other software, which might be related to the one provided by StkH-1.

Table 21 UC-05 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	TePEx: Today, test patterns, occurring due to malfunctioning testing equipment, are detected, when already yield loss is affected. With TePEx (TePEx - Test pattern extraction), test patterns are detected in the analog measurements, provided by the tester, instead of using pass/fail information (triggered when specification limits are applied to the analog measurements) and hence, does not affect the yield. With TePEx, test patterns are detected earlier as of today and in an automated way (without manual/optical inspection). With this higher degree of automation, engineering costs are reduced. WHF: In contrast to TePEx, WHF offers the opportunity to detect critical process patterns – patterns originating from the process itself and not the tester – before yield loss arises in the production. Therefore, the production process can be modified at an earlier stage to lower the manufacturing of defect devices. This results in a potential reduction of engineering costs. DR: DR will support TePEx and WHF in their integration effort. If successful, a significant reduction of future solution engineering work is expected as classes with identical defintions can be re-used.
Obj. 2 - Interoperability for IoT and SoS engineering tools	n.a.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	n.a.
Obj. 4 - Integration platform interoperability with emerging digitalization and	After successful validation of the tools (TePEx, WHF) itself (which requires the application of the methodology to a broad product portfolio) and their improvement via integration of core user feedback, their integration into the existing toolchain is intended. Therefore, the interoperability of these tools with the existing toolchain has to be ensured (special focus on I/O interfaces). To improve interoperability and data accessibility, interaction with DR and data bases

Project Objective	Focus & Planed actions
automation framework	is intended.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	n.a.
Obj. 6 - Training material (HW and SW) for professional engineers	Already available: 1. TePEx: Master's thesis, manual, guideline 2. WHF: PhD thesis, several master theses, manual, guideline, papers 3. DR: Master's thesis, papers After Arrowhead Tools project: • For educational purposes and internal dissemination, TePEx and WHF shall provide a user documentation, including a tutorial and some examples on using the tool. • With DR the information exchange is possible among experts (and generally stakeholders) of different domains and with different backgrounds.

5.4 Engineering Process analysis

Table 22 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-05.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	C
OBJ-AHT #1	X	X	X	X				X	
OBJ-AHT #2									
OBJ-AHT #3									
OBJ-AHT #4			X	X			X		
OBJ-AHT #5									
OBJ-AHT #6		X	X				X	X	
OBJ-WP2 #1			X	X			X		
OBJ-WP2 #2	X			X			X		
OBJ-WP2 #3	X	X	X	X	X	X	X	X	
OBJ-WP2 #4		X	X				X	X	

6. UC-06 Production preparation tool chain integration (LIND)

In this use case, there are three main actors. Lindbäcks Bygg AB (Lindbäcks), Lundqvist Trävaru AB (Lundqvist) and PodComp AB (PodComp). The first two ones are building manufacturers and the third one is a bathroom module supplier to the first two stakeholders.
Lindbäcks has an engineering process that begins with an architect drawing to the assembly of wooden modules into apartment buildings. Our focus here is it automated wooden sub house modules factory. The technology used in the engineering process originates from the 1980s. The process consumes a lot of engineering time, since it has a lot of manual entry and verification while sometimes errors slip into production resulting downtime and delays in the production. To be able to keep the pace in the factory Lindbäcks buys completed bathroom pods from PodComp. These are lifted in place into the wooden modules that Lindbäcks produces in their factory.
Today Lindbäcks sends PDF drawings to PodComp. At PodComp, the drawings are manually transfer into a drawing system connected to their production system. Here machine files are generated and verified in a simulator before they are put into their production line's robots.
Lundqvist Trävaru AB has a system that enables their customers to configure desired building in a 3D configurator within a web browser. The 3D configurator has the engineering logic and transfers the information to drawings and bill of materials, which is the basis for purchasing and production. However, the system is not scalable, and it is getting harder to update and enhance it with new features. This in turn is an obstacle for growth and further digitalization in the production process.
PodComp has a good production and manufacturing line. However, a lot of their information flow from order to machine is handled manually. This limits PodComp to extend their footprint on the market towards other building manufacturers or even end users (i.e. more and smaller orders).
The three companies plan to collaborate within the Arrowhead Tools project to streamline the process from architectural drawing to the ordering and manufacturing of bathroom pods using a service-oriented architecture (SOA) within the cybersecure environment of the Eclipse Arrowhead framework.

6.1 Overall description of the UC-EP

The baseline process is very "manual":

1. Lindbäcks receives architectural drawings (often in the Revit format) from the architect. The quality of the ifc-files is not up to the standard needed to be used as a base for new drawings, and as DDS (baseline CAD application) lacks the capability to import ifc-files. There is no other option then to just utilize them as printed paper or pdf plans.
2. Engineering starts from blank paper in DDS:
    a. House drawing AutoCad 2D (Dwg)
    b. Production Drawing 2D (DDS arkitekt)
    c. DDS construction 2D initial version
    d. DDS Architect (electricity & plumbing separately)
    e. DDS Construction verification of all three disciplines

    f. Loop until all known issues are resolved
3. Lindbäcks sends order with pdf drawings to PodComp.
4. Lundqvist has a 3D configurator, which enables a customer to configure a building in three dimensions using a web browser. To complement this user interface, a back-end system streamlines the order handling, purchasing and engineering. The 3D configurator has all the logic around what is possible to build, and the material needed for the design that the end customer configures. The back-end system gathers information about material needed and creates purchase orders to cover the material needs for planned orders in production as well as creates drawings for the production. This is however not scalable and hard to develop further with any new requirements.
5. PodComp has a good and highly automated production line. However, a lot of their information flow from order to machine is handled manually. When an order is received, PodComp manually transfers drawings from pdf into their system to be able to create machine files for their production line.
    a. The machine files are simulated in a software that acts as a digital twin to verify that the robot cell can interpret the machine files in a correct way.
    b. Adjustments are made if needed.
    c. When the simulation has verified that the files are working, the machine file is deemed ready. The file is transferred to the robot cell according to schedule and initial production can start.
6. Verification in the PodComp production: even though the simulation is executed successfully there remains a risk that the entered information is incorrect, due to manual mistakes earlier in the process. Because of this, a lot of double checking is done for every individual version of a product.

6.2 Engineering Process Description

In the UC-06 we have described the end customer StKH5 Requirements goes via StkH 4 Architect and have two ways to transfer the information of how the bathroom will be designed. One way is true StkH1 Lindbäcks and the other is true StkH2 Lundqvist Trävaru. StkH3 Podcomp is the producer of the bathroom.
Within the Arrowhead Tools project, we adopted the Arrowhead Tools Engineering Process model as it clearly addressed our situation with multiple stakeholders. This revealed clearly some of our bottlenecks.
The tool chain before the adoption of the Arrowhead Tools Engineering Process included many manual tasks, which creates costs, delays, and potential errors.
After the adoption of the Arrowhead Tools Engineering Process along with the selection of new tools (e.g., Vertex BD) and development of new tools (translation from CAD to robot machine files and SOA file transfer) the tool chain is has a much faster response time with a long term lower cost.
In Figure 14 the AHT-EP of the use case. Main activities are the product design and production, in the following the list of the main activities:

- Product design:
    1. Building requirements and drawings [product]
    2. From engineering design and Procurement to Bathroom pod requirements and drawings [product]

   3. From engineering design procurement to production request [product]
   4. Produced bathrooms pods being shipped [product]
   5. Building modules being shipped to end customer [product]
- Production:
   1. Operation drives evolution [production]
   2. Evolution influences requirements [production]
   3. Evolution requires new training material [production]
   4. Training material to educate engineering staff [production]

Lindbäcks has made a drastic change and is in the process of changing their CAD software from DDS to Vertex BD; a move from 2D to 3D with improved import and export functions (Functional design, Engineering phases). This has implication it several of the engineering phases: it allows the absorption of Requirements form the architect by importing Revit files and checking functional design. In its Procurement phase, it is able to export its requirements in 3D for the bathroom pods in a way that PodComp can use directly.
PodComp has been developing (Engineering a software that takes in the design (Requirements phase) of the bathroom pods and generates machine files for the ABB Robots that the bathroom walls (including electrical, ventilation and plumbing holes). The machine files are still verified using ABB Robot Studio (Functional design, Operation phases).
Lundqvist is developing (Engineering) its own file format export from the 3D Configurator so that its output can be proceeds by PodComp's software (Procurement)
BNearIT, with LTU, is planning (Engineering) for the three main stakeholders a software tool that would take the output files from the CAD and 3D Configurator software and look for a services suppliers with the aim of building a service oriented architecture (SOA) tool chain.

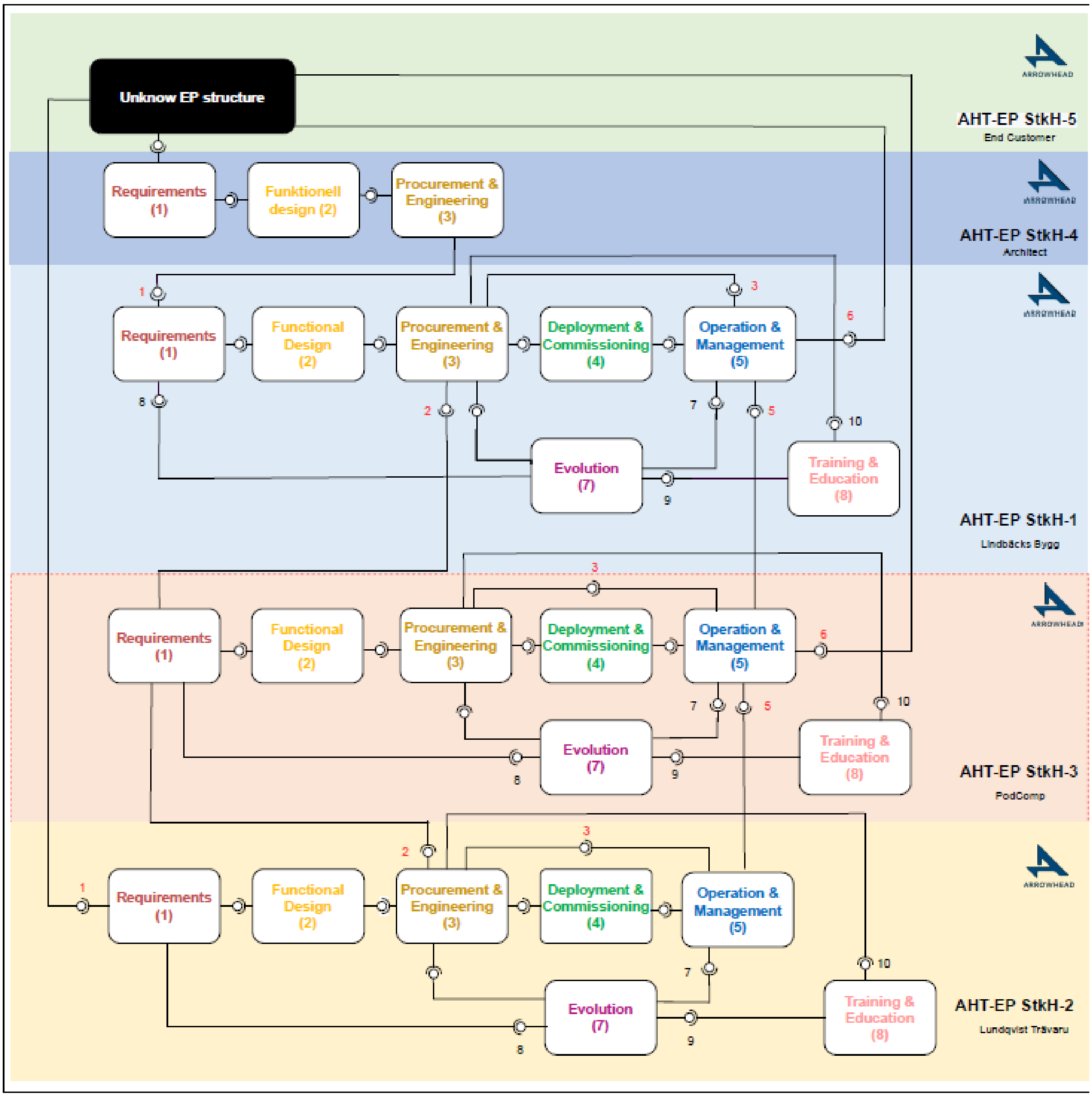


Figure 14 AHT-EP of use case UC-06

No technology is completely missing in the AHT-EP model. However, the existing analysis, verification, and testing technology needs to be suitably applied, optimized, and/or extended to support the needs of the considered systems.

Only the first three AHT-EP phases are considered as relevant, together with appropriate training.

Table 23 AHT-EP Phase focus of UC-06

Engineering process phase	Addressed/Focus
Requirements	**Activity**

Engineering process phase	Addressed/Focus
	PodComp has been developing (Engineering a software that takes in the design (Requirements phase) of the bathroom pods and generates machine files for the ABB Robots that the bathroom walls (including electrical, ventilation and plumbing holes). The machine files are still verified using ABB Robot Studio.
Functional design	**Activity** Lindbäcks has made a drastic change and is in the process of changing their CAD software from DDS to Vertex BD; a move from 2D to 3D with improved import and export functions. PodComp has been developing (Engineering a software that takes in the design (Requirements phase) of the bathroom pods and generates machine files for the ABB Robots that the bathroom walls (including electrical, ventilation and plumbing holes). The machine files are still verified using ABB Robot Studio. **Tools** 1. Tool Chain Lind-Podcomp: Vertex 2. Tool Chain Lundq-Podcomp: 3D configurator
Procurement & Engineering	**Activity** Lindbäcks has made a drastic change and is in the process of changing their CAD software from DDS to Vertex BD; a move from 2D to 3D with improved import and export functions. Lundqvist is developing (Engineering) its own file format export from the 3D Configurator so that its output can be proceeds by PodComp's software (Procurement) BNearIT, with LTU, is planning (Engineering) for the three main stakeholders a software tool that would take the output files from the CAD and 3D Configurator software and look for a services suppliers with the aim of building a service oriented architecture (SOA) tool chain. **Tools** 1. Tool Chain Lind-Podcomp: Vertex 2. Tool Chain Lundq-Podcomp: 3D configurator
Deployment & Commissioning	**Tools** 1. Tool Chain Lind-Podcomp: File Converter, ABB Robot studio 2. Tool Chain Lundq-Podcomp: File Converter, ABB Robot studio
Operations & Management	**Activity** PodComp has been developing (Engineering a software that takes in the design (Requirements phase) of the bathroom pods and generates machine files for the ABB Robots that the bathroom walls (including electrical, ventilation and plumbing holes). The machine files are still verified using ABB Robot Studio. **Tools** 1. Tool Chain Lind-Podcomp: Abb robot cell 2. Tool Chain Lundq-Podcomp: Abb robot cell
Maintenance Decommissioning & Recycling	Not specified.

Engineering process phase	Addressed/Focus
Evolution	Not specified.
Training & Education	**Activity** Lindbäcks has been learning to use Vertex BD, PodComp and Lundqvist have been developing their tools. The training material that is missing is how to develop Eclipse Arrowhead compliant tools. It feels more like black magic than an engineered solution.

6.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 24 UC-06 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The change from design time to run time engineering: the use of the SOA modules to select service providers at run time (e.g., when Vertex CAD are saved, a SOA module seek a bathroom service provider {potentially to negotiate a price} and then passes on the CAD information. At PodComp the CAD file is processed to generate ABB robot machining files as well as other ERP and MES functions). **Planned actions** • AHT-EPP 1: Stakeholders who need services from other stakeholders can use the idea of SOA. [Procurement from one to Requirements to the next one] • AHT-EPP 2: Integrated within the new tools, e.g., Vertex BD and ABB Robot Studio
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The move from single to integrated multi stakeholder automation and digitalization: analysis and mapping to link tools within the different engineering phases of all stakeholders associated with the use case have been performed (c.f. UC-06 engineering process map). This is followed by the development of tools to promote the necessary service oriented architecture.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Handling of substantially increased number of I/O's due to much more fine grained automation: with a clear mapping of all tools in the different stakeholders' engineering phases, one can use input from different tools to optimize production. The enterprise resource planners of different authenticated and authorized stakeholders can now interact together to select the best service providers. Lindbäcks's system will be able to consider different bathroom manufacturers while PodComp can provide bathroom modules to several customers (e.g., Lundqvist).
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Address digital learning and training activities as an integral part of the engineering cycle: to generate a paradigm shift of an interconnected set of service oriented tools across multiple stakeholders as well as the introduction of new tools has required learning and training activities. This ranged from Vertex BD to the Arrowhead Framework. They have not been all well documented to make them general activities across the project and beyond.

Table 25 UC-06 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	We plan to automate the information flow exchange between the architect and Vertex including 3d configurator at Lundqvist and the toolchain to the ABB robot.
Obj. 2 - Interoperability for IoT and SoS engineering tools	We have adapt the information format that Vertex and the 3D configurator at Lundqvist.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	We use file converter as a tool for legacy automation that will be integrated to the AHF.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	We will use the Test Tool to verify interoperability within the systems in the tool chain.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	We will use AHF to communicate between stakeholders.
Obj. 6 - Training material (HW and SW) for professional engineers	During the project we will continues provide Training material for the engineers using the systems.

6.4 Engineering Process analysis

Table 26 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-06.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	N	C	C	C
OBJ-AHT #1	X	X	X	X	X			X	
OBJ-AHT #2		X	X						

OBJ-AHT #3			X						
OBJ-AHT #4			X	X					
OBJ-AHT #5			X	X	X				
OBJ-AHT #6		X	X					X	
OBJ-WP2 #1	X	X	X		X				
OBJ-WP2 #2			X	X	X		X	X	
OBJ-WP2 #3	X		X	X	X		X		
OBJ-WP2 #4		X	X					X	

7. UC-07 CNC machine automation (FAUT)

Around a machine tool construction and use there are different stakeholders but, as a Computerized Numerical Control (CNC) manufacturer, we provide the interface to many or all of the functionalities of the machine. The CNC (with its integrated PLC (Programmable Logic Controller) and the digital drives) is incorporated as a component in the Machine Tool builder engineering process. We provide drawings in standard formats (dxf, stp) for the different components (being the electrical motors CAD drawings the most relevant) and many different manuals covering from selection of the right motor or digital drive to installation and commissioning.
We can guess that the Machine Tool (MT) builder uses an Engineering Process similar to the standard, but we don't know.
Supposing a "standardized" MTool Builder, the interaction between that and Fagor Automation's provided information, products and tools would be like the schematic shown in Figure 15:

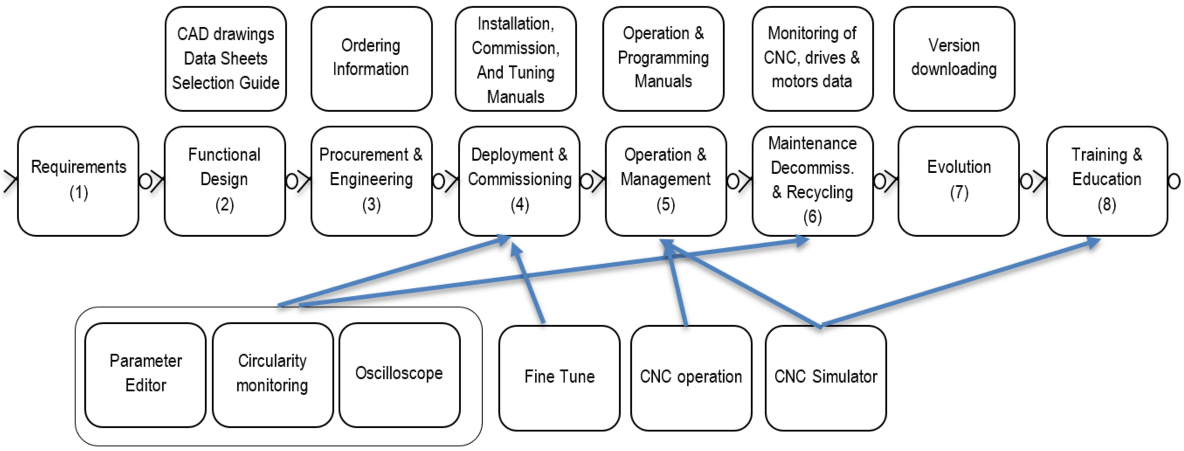


Figure 15 interaction between that and Fagor Automation's provided information, products and tools

It is important to note that Fagor Automation must provide information and tools for both the MT manufacturer and MT end user. Moreover, many of the machines are highly customizable and, regarding the control of their components, the CNC must be adapted to many manufacturers and to different machines at any customer.

A small explanation of the life cycle follows to help understand the use case(s) defined for the project:

1) A MT manufacturer decides the design of a new Machine Tool (or a revision of an existing model). Hopefully, they write some requirements as dimensions, loads…
2) For the mechanical design, a CAD system is used. Fagor provides CAD drawings in standard formats for the motors, digital drives, input/output hardware and CNC. Motor drawings are by far the most important.
3) Regarding selection, the MT manufacturer must use the datasheets (manuals) of the motors and drives to select the right ones according to loads, speeds…The machine tool prototype is built (many providers can be involved, both for components and for mounting of the different parts). Being a programmable system, the MT manufacturer usually includes specific routines for controlling and using
4) Regarding the CNC, the adaptation (number of axis, number of parallel part programs, PLC program, etc…) is usually done once when the first prototype is built, and Fagor Automation usually provides engineering services to help adapt the many aspects of the machine and periphery and control an synchronize its parts. This is done once for the prototype. But the deployment needs a differentiation .For the next machines of the same model, those parameters are copied but the deployment is usually done individually for every machine and, for big machines, bust be done at the installation place. As this is a very time consuming and critical phase for the quality of the produced parts, tools have been developed to help the engineer at the shop floor.
5) In operation, a different user, the final user, incorporates a new use case. They have their own engineering process for every part piece that must be produced, and the CNC must be programmed to achieve that goal. We will write another explanation, as it is also very relevant for the project. But, in short, Fagor Automation provides operating and programming manuals, as well as a simulator that greatly simplifies rapid detection of programming faults and realistically view the "simulated" part.
6) During the life of the machine tool, periodic revisions are made for wear detection of the mechanical parts and corresponding adjustment of the tuning parameters. Tools used at deployment are also relevant here. The CNC provides also logging of the variables (both internals and coming from digital drives and motors and PLC controlled periphery).
7) Fagor Automation periodically publishes new versions of the CNC code with error correction or new functionalities. Under contract, these versions can be downloaded to the CNC.
8) Finally, regarding training (we are in the MT builder engineering process) the CNC simulator can accept, after adaptation, CAD data with the 3D drawing of the machine, along information on movement relationship between parts. This allows complete simulation of the MT builder machine in a PC where the final user can program the part and see the results.

As it is relevant for the project, the process to produce a part with a machine tool is briefly described. For small companies, all the phases can be done by the same, skilled, operator.

1) The MT owner (part producer) receives as requirement a part drawing accompanied with material and tolerances. The drawing can be received in 3D(.stp) format or 2D5 (typically dxf). But it is not unusual to receive a blueprint of the part.

2) The “design” process implies determining the operations needed to make the part. This can be quite difficult to optimize and, for true 3D operations in 5-axis milling machines, must be done with a CAM system and through a number of iterations. It is a time consuming task that needs very skilled people and where any help is useful. The result is a program or set of programs and the tools needed. It is also common to programm directly at the machine tool, with the same, or near the same, tools and applications that are used at the office.
3) Procurement and engineering can be here mapped to the definition of the machining process, based on material, machine power and characteristics (speeds…), available tools and operation type and tolerances.
4) And to produce the part, the machine must be commissioned, putting the right offsets in the CNC, finding origins of the part, setting axes, and taking care of the lubricants, cooling, etc.
5) Once the program starts (that can produce many identical parts) the operator uses the CNC to monitor and modify in real time machining conditions to cope with variations of the material, tool wear, chatter...but the machine can sometimes work unattended for long periods.

It must be noted that, once the first phase (finding the right program and machining conditions) is done, the rest can be highly automated for large batches or be done by less skilled people, for small batches, depending on cost ratios.
From the above description, it is clear that there are at least two different companies (apart from FAUT involved in the full process. We envision improvements in both processes by focusing on the work where the CNC is concerned and the cost reduction (measured in highly skilled people time) is higher.

7.1 Overall description of the UC-EP

In Figure 16 the mapping of the two tools developed in this use case.

For the **MT builder** use case, the CNC main tools are mapped to the **deployment and commissioning** phase, with some applications spreading to the engineering, operation and maintenance phases.
For the **MT user** use case, the CNC integrated tools address **from Design to education**. With some limitations, a skilled operator can design, optimize, choice machining conditions and decide tools, prepare the machine (commission), find the origins and offsets, and make a part. He can eventually diagnose the machine components with the information provided by the CNC or download a version (with the appropriate rights). 3D Simulation of programs, even for another machine, can be done at the shop-floor also.

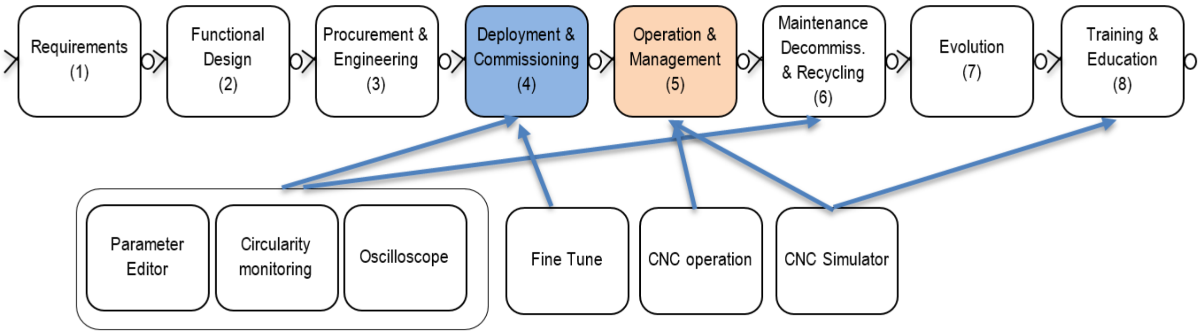


Figure 16 Mapping of MT builder tool and MT user tool on the AHT-EP phases

All the tools described in this paragraph are Fagor Automation proprietary toolchains. The current (baseline) architecture assumes tight integration of the different parts and DCOM use for HMI. Individual evolution of the components is aggregated in full product versions.
The Blocks defined come in two different products and a side application.
CNC8065: This is the real-time controller of channels and axes, and includes today all the applications, from 3D simulation (or graphical 3D representation of the machined part) to fieldbus mastering, operator HMi. The Fine Tune is a windows application that can run on the CNC hardware or on a separate PC and connects to the CNC via DCOM.
ALL the blocks run on the CNC and their interactions are integrated. Interfaces are usually through shared memory or relying on binary files with proprietary formats.
Simulator8065: The "same" software of the CNC without the need to connect to the buses or run in hard real time. With all the problems of its software base. PLC program can't access physical signals also.
From the analysis of the use cases, blocks and the baseline, a bunch of tools have been identified and three toolchains (shown in Figure 17), for three actors, seem to cover well many of the blocks. Let's start by describing these and their mapping to the EPPs.
Evolution will be treated probably in the Configuration Editor or via a specific application, not decided yet.

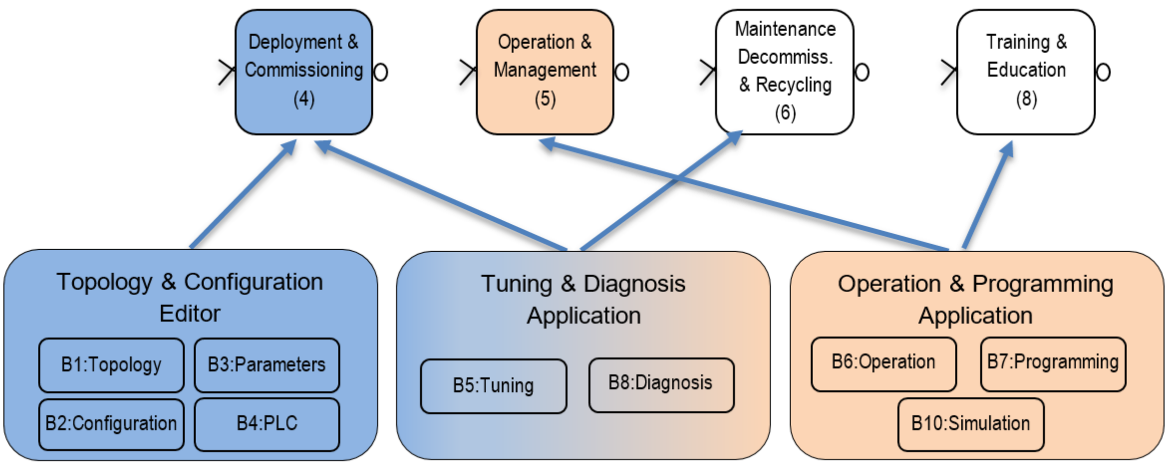


Figure 17 The three toolchains mapped on the AHT-EP phases

7.2 Engineering Process Description

In Figure 18, it is clear that the main relationships between the CNC manufacturer and the **MT-User** are carried out during Operation & Management phase (of the CNC), while the relationships with the **MT-Builder** are spread over most of their EP phases, with some concentration around Deployment & Commissioning and Maintenance, Decommissioning & Recycling.
Information produced during CNC development and production is used by the MT-Builder and those are represented by the lines c11 (Data sheets…), c12, c21 (CAD drawings, dimensions, manuals…), c31 (ordering handbook…). But there's a tighter relation in c41 (data produced by the MT-Builder like kinematics, dimensions, mechanical elements used…), c42 (CNC configuration and tuning software applications), c43 (CNC applications for machine health assessment), c51, c52 and c53 (CNC applications to gather, process and display operational data coming from the peripherals, drives and the own CNC).
The lines c54-c57 represent the different ways in which the operator uses the machine through the CNC or its simulator to design the program and see the result (c55), setting the machining conditions and tool offsets (c56), loading the tools and align the part (c57) and operating the machine in real time(c54).
c61 represents data gathering of the machine status that can be further be related to other operational data and lead to improvements in maintenance and diagnosis.
The c81, c82 and c83 (along some others not numbered) try to express the cooperation needed to configure a simulation environment (digital twin for specific processes) with information provided by the MT-builder (machine drawings, kinematics…) and the MT-user (tools, parts) where the application is provided by the CNC manufacturer. The resulting application can be used for training & education (c83) but also as the design tool of the MT-user (c84).

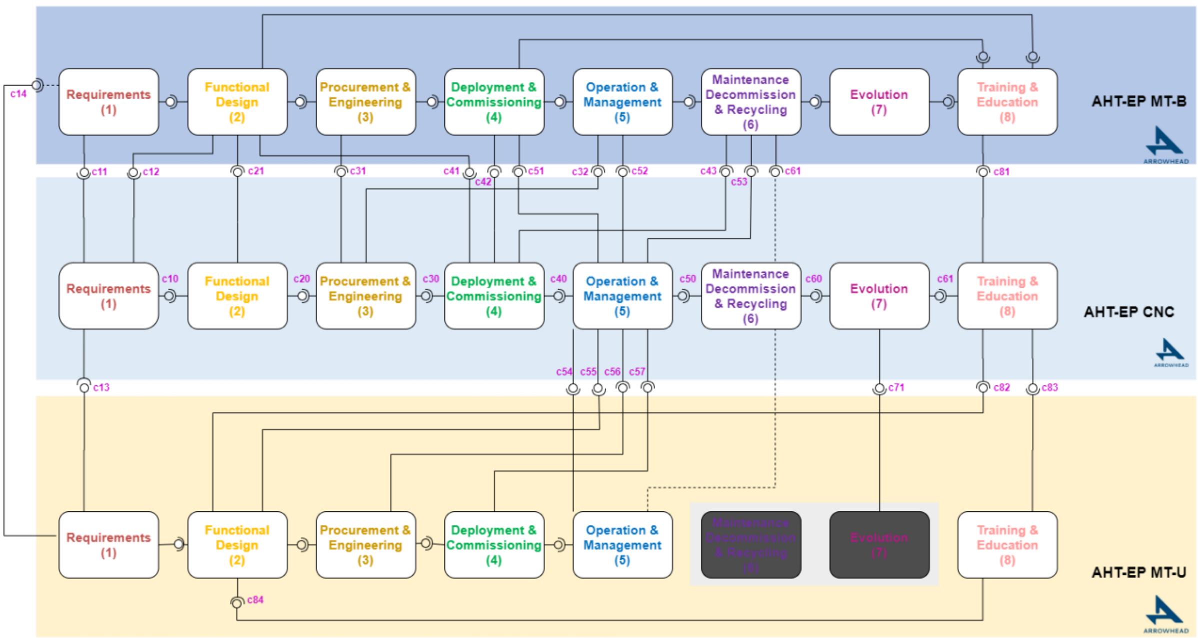


Figure 18 AHT-EP of use case UC-07

The following Engineering phases are not applicable in our use case: Requirements, Functional design in MT building, and Procurement & Engineering
It's important to remember that the use case identifies the integration and use of a CNC in a Machine tool. We don't look at the machine design and construction (in the machine tool builder case) nor the part piece definition (in the machine tool user case). Under such assumptions, no requirements, functional design or procurement phases are applicable.

Maintenance for the Machine Tool is out of scope, whilst the CNC gathers information and provides tools for diagnose and these tools have been already covered.
From the point of view of the use case, there's no knowledge of the internal Engineering phases of OEMs (MT builders) or End Users, but the CNC takes part in many of their Engineering phases and improvements on its use and functionalities will have a big impact on their EPs.

The HMI customization tool is used both by the OEM and by the final user. There is a brand customization possible (OEM) and individual Machine tool customization (End User). Tuning is done at machine level usually, but in some cases a fine tuning for specific parts is done for a part program. In the drawing and explanations, the scalability is applied to the machine tool building and the part piece production via Machine Tool use. The use case addresses both EPs and the projected tools try to reuse components from one EP into another. This can't be done without adhering to standards in persisted data

It seems that MT builders are adopting RAMI 4.0, being OPC_UA one motivation.
We have recently adopted OPC-UA for data acquisition related to maintenance and operation & management. Companion standards for machine tool are still in early phases or immature. We actively follow this track.
We address this EP standard by providing dictionaries and tools for it that are used en both EPs, for instance in monitoring process data or operational data.

While we don't exactly know every MTBuilder's EP, It seems that the AHT-EP can be mapped well to the "standard" way of building as we know it, as shown in the previous figures.

In the following table we describe activity and tools used in each phase by the two main tools developed within this use case.

- *Machine Tool Construction:* The Engineering process of every MTBld could be different, while we guess is not very different from the standard. What we address in the project are some toolchains that are used in the Engineering phases below. There are two toolchains identified in the project related with the MTBld:
    - TCh1: Configuration and mapping Toolchain
    - TCh2: Control Loop Tuning Toolchain identified Tools
- *Part Production:* The Engineering process to produce a part depends not only on the MTUser but also on the part itself, the market and the availability of data. What we address in the project mainly is a toolchain that will be used in the Engineering phases below.
    - TCh3: Smart Graphical 2D5 Editor and operations management

Table 27 AHT-EP Phase focus of UC-07

Engineering process phase	Addressed/Focus
Requirements	**Activity** *Machine Tool Construction:* The MTBld decides to construct a new model of MT or a customization of an existing model for a customer. These reqquirements usually determine dimensions, power, speeds and type of CNC. *Part Production:* The MtUser receives an order to produce a batch of parts. He can receive the requirements in many ways, from a blueprint and material to 3d geometry in a (hopefully) standard format. **Tools**

Engineering process phase	Addressed/Focus
	1. *Machine Tool Construction:* n.a. 2. *Part Production:* TCh3: Smart Graphical 2D5 Editor and operations management decide the machine and conditions. • Input: MTBld manual for the machine to check limits, dimensions, power, max. speeds. • Input: CNC operating and programming manual, and CNC provided info on what options are available • Input: requirements on material and tolerances • Output: Machining conditions and machine selection
Functional design	**Activity** *Machine Tool Construction:* MTBld designs mechanically the machine. CAD systems and FEM applications are used. To select components from the CNC supplier, Cad drawings and Data sheets are available. In some cases, structural analysis provides resonance frequencies from design phase. *Part Production:* The programmer decides the best way of producing the part in the machine. He can import dxf files (or any other standard format), select geometric elements are of interest, define the operations to be done with them (drilling, machining, threading…), modify the geometries or make new ones from a blueprint... **Tools** 1. *Machine Tool Construction:* n.a. 2. *Part Production:* The TCh3: Smart Graphical 2D5 Editor and operations management. The programmer decides the strategy of machining, perhaps aided of a CAM system. • Input: Requirements from previous phase (including dxf files...). • Input: MTBld data from manuals. • Input: CNC G-code set, and canned cycles availability • Input: MTBld specific routines • Output: CNC G-code program(s) • Output: List of necessary tools
Procurement & Engineering	**Activity** *Machine Tool Construction:* "MTBld places orders for all the components. Depending on the company, many of the components are bought or subcontracted to their parties. CNC, drives, motors and linear scale models are somewhat complex and choice is usually done assessed by commercial people with technical background. After selection is made, the ordered is placed and material served. For new models or new customers, technical advice from Fagor Automation is also offered for configuring and tuning. The engineering team will develop the documentation of the HW and SW developed for the ECU that will be provided as a reference manual to the StkH 1. A working prototype of the ECU will be tested in a real machine for quality test in the vendor laboratories." *Part Production:* "The programmer can iteratively refine the program using different features and machining conditions using the geometric aids and the technology tables. The phase produces a G-Code file (or several files) and the needed tools. These are retrieved and must be loaded in the tool magazine." **Tools**

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>1. Machine Tool Construction: n.a.
2. Part Production: TCh3: Smart Graphical 2D5 Editor and operations management adjustment of machining conditions, loading of tool magazine.
• Input: MTBld provided data on power
• Input: available set of tools and wear conditions
• Input: technology tables (material, cutting conditions …)
• Output: customized G-code for that machine and tool magazine</td></tr>
<tr><td>Deployment & Commissioning</td><td>Activity
Machine Tool Construction: "MTBld, after the MT is built, needs to diagnose for proper work of the system, align axes, program the PLC, connect the Ios to the fieldbuses, and configure them. After that, the tuning of the control loops is carried out. In this phase, for new models, is common that both the MTBld technician and the Fagor Automation service people work together. For the next machines, only the tuning optimization must be done, configuration is copied from the first machine."
Part Production: The operator downloads the program (frequently the program is written or modified at the machine), does the setup: putting the raw material, the tools, setting the offsets (zeroing the part geometry), all that is needed to start operation.
Tools
1. Machine Tool Construction: TCh1: Configuration and mapping Toolchain. TCh2: Control Loop Tuning Toolchain identified Tools. Currently, this phase doesn't receive inputs from the MTBlds' functional phase, while it would be sensible to have a first set of data. The phase tools are autonomous.
2. Part Production: TCh3: Smart Graphical 2D5 Editor and operations management setup of the machine for a part piece.
• Input: MTBld provided routines for zero calibration
• Input: MTBld provided routines for kinematic calibration
• Input: Canned cycles for part piece offsetting via touch probe
• Output: customized G-code for that part setup</td></tr>
<tr><td>Operations & Management</td><td>Activity
1. Machine Tool Construction: This phase includes the first 5 Eps of the part piece production.
2. Part Production: The operator can start production for the number of pieces of the batch. For the relevant use case, the operator must care for final adjustment of the feeds and speeds, adapting to material and tool wear. Usually, every new part requires a new setup, but zeroing and measuring can be automated for machines that integrate touch probes.</td></tr>
<tr><td>Maintenance Decommissioning & Recycling</td><td>Activity
Machine Tool Construction: "This task can be done by different actors, and one of them is the MTBld. There can be a maintenance contract. Data from the commissioning phase is used as a baseline to diagnose the machine. Some of the tuning tools are relevant here. Comparison with baseline gives a ""health"" assessment."
Part Production: "The operator usually ""monitors"" the health of the machine. This is currently substituted (or at least supplemented) by logging systems and automatic diagnosis systems (with learning algorithms, etc.). Commissioning tools are not</td></tr>
</table>

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>directly used by the operator but specific tests based on the same tools can be carried out automatically under operator supervision."

Tools

1. Machine Tool Construction: TCh2: Control Loop Tuning Toolchain identified Tools for condition based maintenance and periodical diagnosis:
• Input: data files produced by the oscilloscope and/or the Fine Tuning Tool (EP4)
• Output: Today, visual comparison with that baseline data
2. Part Production: n.a.</td></tr>
<tr><td>Evolution</td><td>Activity

Machine Tool Construction: "New CNC versions would incorporate new features of interest to the MTBld and also correct software errors. New machines should use the last versions. Moreover, on occasion of maintenance work, or when any error is important enough for a customer, the CNC software version must be upgraded. This is a delicate operation that is done manually (giving access to directories, taking into account of options)".

Part Production: The MTUser can report field errors and, when those are solved, can download new versions. Moreover, the MTUser can decide to download, under license, new applications, tools or canned cycles of interest for his work. This requires an application designed specifically for that matter.

Tools

3. Machine Tool Construction: Software and Firmware updates for CNC available for download.
• Input: Files downloaded or received
• Input: Compatibility table between different hardware of CNC and components and software-firmware
• Output: Decision on modules and versions to update
4. Part Production: Software and Firmware updates for CNC available for download
• Input: Files downloaded or received
• Input: Compatibility table between different hardware of CNC and components and software-firmware
• Output: Decision on modules and versions to update</td></tr>
<tr><td>Training & Education</td><td>Activity

Machine Tool Construction: "The MTBld can use the CNC Simulator incorporating the kinematics and drawings of its own machine. This produces, in fact, a digital twin useful for the programming of the machine and, much important, to detect collision between machine and part, etc."

Part Production: "The simulator is in this case very helpful. Many licenses are today sold for training and education in CNC programming. Regarding this, the basic version is freely downloadable and is used at professional schools. The look and feel is just as that of the CNC, what improves familiarity for students."

Tools

5. Machine Tool Construction: n.a.
6. Part Production: CNC Simulation customization.</td></tr>
</table>

Engineering process phase	Addressed/Focus
	• Input: CAD data from Functional Design of MT (MTBld) • Input: Link data between geometric elements of the MT (kinematics), from MTBld • Output: CNC configuration data needed by the simulator

7.3 How the AHT-EP allows to match the Project and WP2 objectives

The AHT Engineering Process is a very good architectural paradigm to analyse the CNC automation and specially the relationship between the CNC applications and the engineering phases of the MTBuilder and MTUser.
From this perspective, a separation of the tools inside the CNC has been started and the three more important toolchains identified will be developed as independent applications.
The service oriented architecture of Arrowhead has inspired a further decomposition and we have followed a modular approach where nearly all of the tools can be installed or updated and whose interfaces will be based on standard file formats as much as possible, in the spirit of the interoperability objective of WP8.
The AHT-EP draw shows the relevant relationships between the CNC and other Engineering Processes. These have been mapped to the AHT-EP from the knowledge of many customers and users, and is a conceptual EP representing how the CNC interacts with them. Internal lines are unknown for us.

Table 28 UC-07 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The tuning tool incorporates the acquired know-how from field tuning of machines. The modular conception planned allows future adoption of the strategies for both machine transfer function identification and optimizing strategies for the control loops. The machining operations editor includes also the concept of technology tables. As shown by the analysis derived of the AHT-EP, these can be provided to the MT User by the MT builder or be built by the end user from its previous experience. Being modular and in standard formats, these tables can be transferred from third parties or modified by the operator for its specific machine. **Planned actions** • AHT-EPP 4: Modular description of identification and tuning algorithms. • AHT-EPP 5: Modular definition of technology in tables. Deep customization done at user or operator needs, individual adaptation to the user possible. • AHT-EPP 6: Data from tuning used as baseline for diagnosis implies standard persistence. • AHT-EPP 8: Reuse of field data to tune tool for education. Use of technology tables in simulator when including operations editor.
Obj. 2 - The move from single to integrated multi stakeholder	Offering modular, non-integrated tools with well-defined, standard file formats input and output. Developing legacy data converters, very important to technology adoption.

WP2 Objective	Focus & Planed actions
automation and digitalization	The AHT-Engineering Process shows the relationships to other Eps and the new tool chains will allow import or export information to their EPs. We have decided: • Develop or adopt third party data readers (standards) as input to the toolchains. • Decoupling Tuning Tools (with its oscilloscope and monitoring function) allows third party managing of maintenance for instance. **Planned actions** • AHT-EPP 4: Persistence of data in standard formats. Adoption of OPC-UA MT companion standard where possible. • AHT-EPP 5: Publishing of input-output information. Standard files. • AHT-EPP 6: Tools available for third parties for maintenance. Both standard file format and, when possible, standard naming convention. • AHT-EPP 8: Input and output of standard formats.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	OPC_UA will take an important role for us in information exchange and logging. We actively work on companion standard for MTools. This will allow share of information between EP phases and even between different EPs, as is our case for tuning and maintenance.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The toolchains developed will be used also in education and training. The Tuning tool can be a great education tool. Real data from machines will be read from the tool and the junior engineer will try to guess the best control parameters. The CNC simulator is already a very good tool for education. During the project, the new editors will include an operations editor that includes technology help. **Planned actions** • AHT-EPP 8: New features included in CNC simulator.

Table 29 UC-07 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	**Configuring and Tuning:** *State:* Manual configuration, manual or semi-automatic tuning, upgrade of rules for tuning only by versioning of the application, COM interface lacks security and/or is difficult to run it reliably. Configuration is standalone application. No simulated training possible. *Improvement:* Independent, client-server oriented tools for every phase and use case. Central repository for tuning rules, ever-green approach for tools, seamless integration of data from the different tools. Fully automatic tuning, smart configuration tool, automatic topology detection where possible. Expected reduction of 50% for deployment (from 2 days to less than one) *Actions:* Two main engineering tools developed in the project, one regarding configuration and topology, the other for tuning. Change in the simulation to allow for training on tools.

Project Objective	Focus & Planed actions
	CNC Programming: *State:* G-Code and Fagor specific language textual programming, separated dxf converter, contextual programming helper for text input, aided editor with graphic support. All tools are included in the CNC code. Simulation is possible with a dedicated tool. *Improvement*: Independent, client-server oriented programming tools for textual input. New fully graphical 2D5 drawing tool with dxf import integrated with advanced sketching features (tangents, pattern, symmetries…). G-Code and Fagor code generation from drawing. Direct 3D visualization of the results for every operation during programming. Expected reduction for programming time between 20%-60%, depending on part complexity. *Actions:* Development of a new, independent, programming tool with plug-ins. Well defined, standard interfaces for file input-output. **CNC HMI customization:** *State*: Fagor tool for component programming and screen design. The solution is based on MFC, works on PC and subject to versioning. Results can be downloaded to CNC. No tool available for the new HMI based on web technologies. *Improvement:* new HMI customizing tool. Native and custom screens fusion and styling. Connection with RealTime CNC data and execution state machine. Expected reduction of programming time of 40% for simple screens, much higher for full customization. *Actions:* Development of a new tool based on web technologies. Well documented interfaces and standard input-output files for OEMs and Users. File management via standard integration tools like npm, etc.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Extensive use of standards in programming languages and file formats for information exchange. Use of standard communication protocols and data semantics where available.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The new tools are very specific to the use case, thus, even if the services are useful for different users the tool itself will be private. Matlab .m format reader will be necessary for compatibility with own legacy systems. Matlab functions needed for machine identification will be developed in ES6, translated from octave sources or using open source files.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	It is expected to develop translation commands from defined, standard format files like JSON and XML to popular proprietary tools like Matlab. Actively work on OPC-UA Machine Tool companion standard and use the defined semantics whenever possible.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and	Only authorized people will be able to use the services and the tools. License and access management will be dealt with in one of the described tools.

Project Objective	Focus & Planed actions
automation solutions	
Obj. 6 - Training material (HW and SW) for professional engineers	Compatibility paths for data import/export to proprietary standard engineering tools (eg: commands for Matlab import or export). Data visualization tool (oscilloscope, etc.) tied to free available engineering tools (eg: octave). CNC Simulation tool available for training in CNC programming or tuning.

7.4 Engineering Process analysis

Table 30 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-07.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1		X	X	X	X			X	
OBJ-AHT #2		X	X						
OBJ-AHT #3		X	X	X					
OBJ-AHT #4			X						
OBJ-AHT #5					X	X			
OBJ-AHT #6		X	X		X	X		X	
OBJ-WP2 #1	X	X	X	X	X	X		X	
OBJ-WP2 #2			X	X	X	X	X	X	
OBJ-WP2 #3		X	X	X	X	X		X	
OBJ-WP2 #4		X	X		X			X	

8. UC-08 SoS engineering of IoT edge devices (ST-I)

This Use Case is subdivided in five sub use cases which are discussed in the following five subsections.

8.1 UC-08.1 SoS engineering of IoT edge devices (Environmental Monitoring) (REPLY)

This use case will propose a highly pervasive sensing infrastructure must provide chemical, PM 2.5-10, noise, temperature, and weather data at sampling times of less than 1 minute and potentially with a spatial granularity of less than 100 m.

Such constellation of devices require an extreme attention on power consumptions, that can be accomplished only by using next generation silicon sensors, energy-aware software applications and low-power, medium-to-long range wireless communications.
An IoT architecture must take into account the environmental constraints where the sensors will be deployed.
The constellation of sensors and edge devices capable of collecting, processing and transmit data from the field will exploit wireless connections that preserve bandwidth, battery duration and extend the sensor's life spanning multiple years without maintenance.

Digital technologies are the building tools to tackle this problem head-on, but they have to face huge challenges on size, pervasively, cost and effectiveness of proposed solutions.
An end-to-end solution that collects air quality information across large areas must be deployed at a fraction of the cost (and size) of previous ICT systems and it must be deployed throughout a city or industrial zone, radically increasing the precision of air quality data.
High deployment costs are mostly due to "silos" from legacy systems and/or proprietary deployments with reduced or no interoperability. While cities might have once paid $150,000-$250,000 for a single unit, the deployment cost of AHT Tools-enabled, smart city environmental quality digital platforms must leverage on low-cost sensors and IoT infrastructure at all levels, affordable enough to be distributed throughout many neighborhoods or industrial areas. This is critical as disparities in traffic, population density, and industrial activity can mean drastically different levels of pollution across a city.
IoT systems will probably change the nature of city life. If we integrate an air quality monitoring system with a smart traffic network, we could detect traffic jams with high levels of air pollution as motors idle. It would be possible to redirect the flow of traffic or instruct drivers to turn off engines as they wait.
Given access to large new pools of city data, entrepreneurs can transform issues like congestion into opportunities with traffic management solutions, connected lighting, and smart parking. City planners and developers will integrate connected features into their design process. There is much to be excited about; the rise of such smart cities will bring many positive changes to residents in tomorrow's urban centers.
The vision goes beyond the monolithic monitoring platform because in the recent years it has been proven that top-down smart city deployment cases lag mainly to the difficulty of deriving actions from the huge amount of data streamed out from the sensors. Furthermore, the environmental data can be used in business value chains to increase the citizen awareness of public activity (and regulations) that target the urban safety and quality of air, water etc.
The cross-interoperability of sensor data is a major factor that drives the vision, together with the capability of deploying a large number of sensing points at a fraction of the current implementation costs, thanks to next-generation silicon and components.
That vision perfectly matches the Arrowhead Tools vision: "Enable collaborative automation by networked embedded devices". In the Smart City Use Case, automation is mainly pervasive automated sensing and localized actions to react to complex, multi-parameter functions that describe environmental situations and their changes.

8.1.1 Overall description of the UC-EP

Each partner has contributed accordingly to its own expertise and various solutions are implemented.
REPLY:
In this use case we will develop the Edge side processing and GUI Dashboard integration with Arrowhead Framework composed as below described.
In the lower level, we have sensing services useful for the monitoring of cities' pollution level and environmental health data.

Below are the integrated sensors:

- PM10: PM10 are very small particles found in dust and smoke. They have a diameter of 10 micrometres (0.01 mm) or smaller. PM10 particles are a common air pollutant. We measure PM10 at some of our air monitoring sites. PM10 are minute particles present in the air and exposure to it is very harmful to health. When the level of these particles increases and penetrate deeply into the lungs, you can experience several health impacts like breathing problems, burning or sensation in the eyes etc.
- PM2.5: fine particulate matter (PM2.5) is an air pollutant that is a concern for people's health when levels in the air are high. PM2.5 are tiny particles in the air that reduce visibility and cause the air to appear hazy when levels are elevated.
- NO2: Nitrogen Dioxide sensor designed to measure low ambient levels of NO2 associated with the irritation of the eyes, nose, throat, and lungs. NO2 and other NOx interact with water, oxygen and other chemicals in the atmosphere to form acid rain. Acid rain harms sensitive ecosystems such as lakes and forests.
- CO2: A carbon dioxide sensor or CO2 sensor is an instrument for the measurement of carbon dioxide gas. Measuring carbon dioxide is important in monitoring indoor air quality, the function of the lungs in the form of a capnography device, and many industrial processes.
- O3: Ground-level ozone is not emitted directly; it is created by chemical reactions between oxides of nitrogen (NOx) and volatile organic compounds (VOC) in the presence of sunlight. The ozone molecule absorbs ultraviolet radiation, and most ozone monitors utilized in regulatory applications use ultraviolet absorption to accurately quantify ozone levels.

In Figure 19 the functional blocks of the UC-08.1.

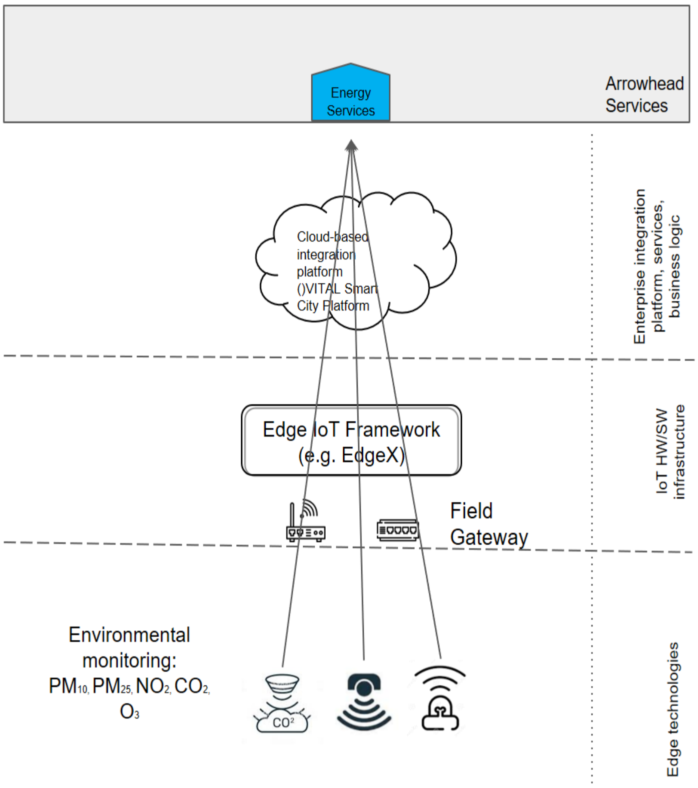


Figure 19 The functional blocks of the UC-08.1

BEIA:

Data collected from sensors connected to several RPi microcontrollers installed by BEIA team (e.g., uRAD, BEIA_GAS2 and temperature data) is parsed, the payload is built in the format requested by the partner (REPLY) and is sent further to BEIA telemetry topic meshlium3d4c / test / #

Data is automatically sent to the partners' broker when it is started.

To ensure the integration of BEIA IoT devices in the AHTOOLS architecture, a Node-RED application has been created that realizes several data acquisition flows specific to each device. Node-RED is a visual programming flow development tool originally developed by IBM to connect hardware devices, APIs, and online services as part of the Internet of Things.

Thus, for the acquisition of data from the uRADMonitor device, it was necessary to implement at predefined intervals, HTTP requests according to the REST interface.
Received messages are parsed using a specific function and then published to the MQTT broker in the AHTOOLS suite.
The diagram below (Figure 20) shows the data flows that ensure the acquisition of measurements from the uRADMonitor industrial device, as well as the Libelium Waspmote BEIA_GAS devices, respectively, RPI.
In the case of Libelium and RPI devices, the data acquisition was done with the help of MQTT clients that allow connections to certain predefined topics.
For easily duplicate these programming flows on any Node-RED server can be used the "import" file.

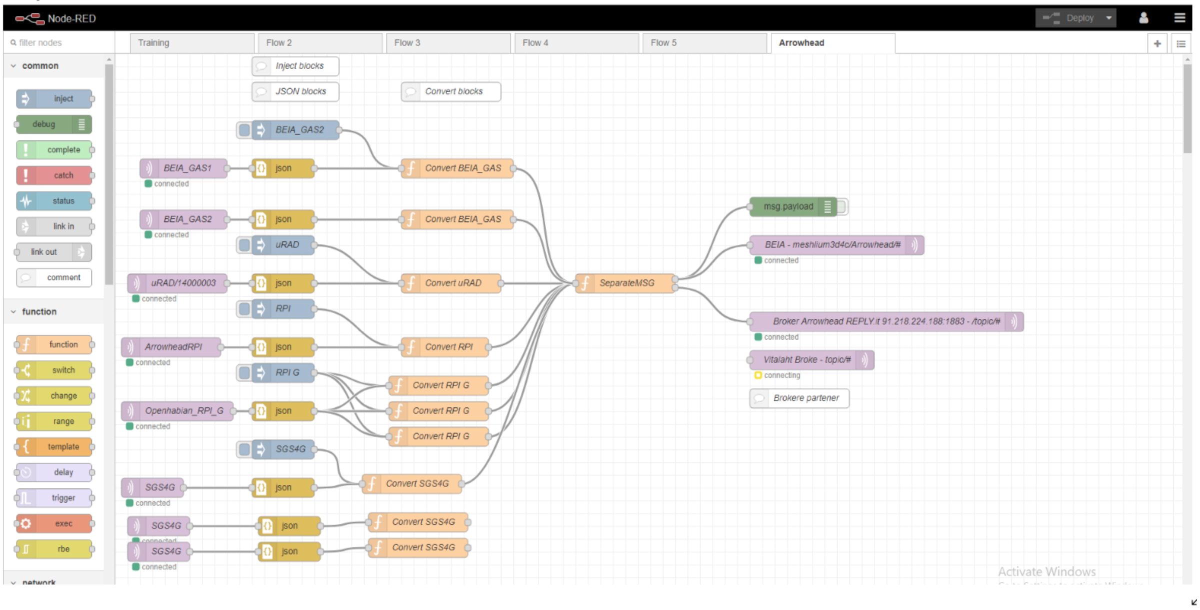


Figure 20 Measurement data flows from BEIA sensors to Arrowhead Platform

ROPARDO:

In the Figure 21 the functional blocks for ROPARDO activity.

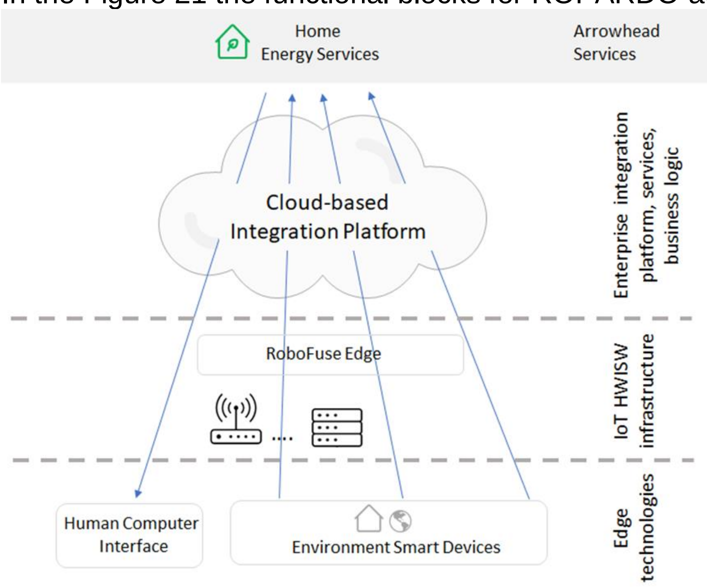


Figure 21 Functional blocks for ROPARDO activity

8.1.2 Engineering Process Description

In this Use Case we have two Stake Holders.
The developers of the Use Case (REPLY) and the End User of the solution.
The Engineering process during the activity of development is pretty equal to the Arrowhead Engineering process, meaning that the envisioned boxes are the same
The specific activities of the team of development during each phase are described in the following table.
In Figure 22 we add the layer of the End User for which we have the following envisioned activities per each phase:
Requirements: the end user participates into the description of the requirements (for example the requirements of the GUI of Vital-IoT) and we envision a process of exchange and tuning of the Requirements with the StkH-1, symbolized with the arrows in-out from the box "Requirements".
Functional Design: the End User participates, to some extent, into the definition of the Functional Design. Mostly he participates in terms of being aware of the outputs of the Functional Design process managed by the Developer of the Use Case, without having a technical impact on it.
Operation & Management: the end user takes part in the phase of Operation & Management, in the sense that, after a phase of Training & Education on the documents and material provided by the Development Team, the end user starts owning and managing the deployed platform.
The phase of Training & Education is a mostly manual phase, fed by inputs coming from the "Functional Design", "Procurement & Engineering" and "Evolution phase", whose primary output is to train people of the End Customer who have a role in the "Operation & Management" phase on his side, but whose additional output is our Activity of Education and Dissemination.

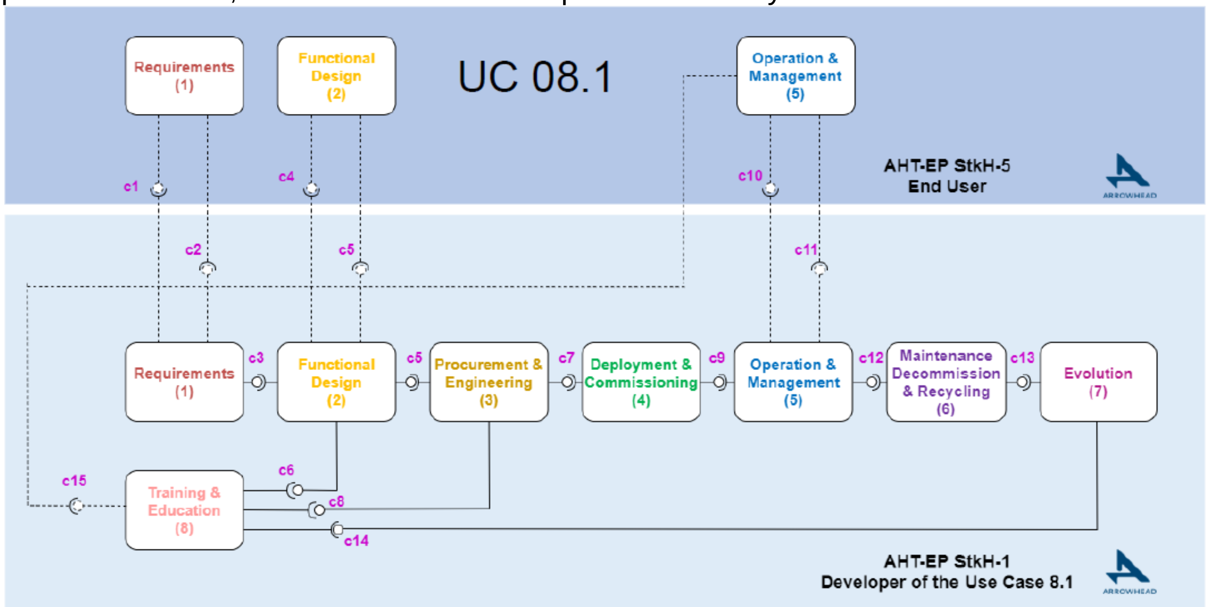


Figure 22 AHT-EP of use case UC-08.1

In the following we discuss the framework used to manage the engineering process that supports the life-cycle of the product/service developed in this use case.

We have been using Jira as our life-cycle management framework during the majority of the engineering phases.
Jira Software is a product of Atlassian, and it was initially released as a pure issue tracking software for software developers. Over the years, it has been adopted by non-IT teams for tracking any type of issue, task, and other work items, becoming a popular project tracking solution. With the launch of the Atlassian Marketplace, many developers have created Jira plugins that extended the features of the software. Today, it continues to be a popular project tracking system in software development for agile PM teams. As a customizable workflow engine, it allows users to track issues, bugs, tasks, and other work items through a predefined workflow that can be modified to fit users' requirements.
Jira is one of the preferred project tracking and management software by software developers and IT teams. It supports both traditional and agile project management. It is centralized and highly customizable, and its workflows allow users to control the status of the project and how it transits to other statuses, providing excellent tracking information.
Agile teams use it to develop software using the Scrum method. They are able to stay focused on their iterations at a fast pace. Customizable scrum boards enable them to deliver incremental value in an organized way.
As a product or project management software, it allows users to assign the needed tasks to complete the product/project. It also has tools such as views and reports that provide information about due dates and statuses. The team is able to collaborate with each other, provide feedback, as well as manage approvals for requests or changes.

For all the components the EP phases can be applied at different scale levels for describing the life-cycle of the components and subcomponents.
However, using the AHT, it is possible to add other providers as long as the data they provide can be demonstrated in the Vital-IoT platform.
Moreover, on the Edge Processing side, scalability can be achieved by using K3S.

In the following table we describe activity and tools used in each phase by the two main tools developed within this use case.

- *Measurement services:* Information is passed using databases and text documents. The type of information is sensor data transmitted through the MQTT protocol. The data is stored in MySQL databases.
  Management of surveys collected from users are manually curated and analysed, the process can be partially automated.
- *Edge Computing:* Information is passed using databases and text documents.
  On the Edge Processing part, K3S helps us considerably on the monitoring and health check of the services.
- Vital/IoT*:* Information is passed using databases and text documents.
  On the Vital-IoT part, K3S helps us considerably on the monitoring and health check of the services.
- Robofuse.

Table 31 AHT-EP Phase focus of UC-08.1

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Measurement services:* Collection of User Requirements on Excel Spreadsheet.

Engineering process phase	Addressed/Focus
	2. *Edge Computing:* Collection of user and edge processing requirements together with integration ones to dispatch data to the cloud (e.g. JSON interface format). 3. *Vital/IoT:* Collection of user and application requirements for GUI and interfacing Arrowhead Framework. 4. *Robofuse:* Collection of user and application requirements for GUI and interfacing Arrowhead Framework.
Functional design	**Activity** 1. *Measurement services:* Definition of the main points of software development protocol to use. Production of a specific documentation to drive the development. 2. *Edge Computing:* Definition of the main points of software development: protocol to use. Production of specific documentation to drive the development. 3. *Vital/IoT:* Definition of GUI and interfaces among involved systems. Production of specific documentation to drive the development. 4. *Robofuse:* Definition of GUI and interfaces among involved systems. Production of specific documentation to drive the development.
Procurement & Engineering	**Activity** 1. *Measurement services:* Purchase of unavailable sensors and development of software code for interfaces to send data. Development Test of implemented functions. 2. *Edge Computing:* Purchase of unavailable industrial PC to run Edgex framework and related pieces of code. Development of the edge processing software code. 3. *Vital/IoT:* Development of code to adapt GUI and internal process of discovery to involve Arrowhead Frameworks architecture services. 4. *Robofuse:* Development of code to adapt GUI and internal process of discovery to involve Arrowhead Frameworks architecture services.
Deployment & Commissioning	**Activity** 1. *Measurement services:* Deployment of software on the different boards and sensors. 2. *Edge Computing:* Deployment of the software code by means of Dockerization. 3. *Vital/IoT:* Deployment of the software code by means of Dockerization. 4. *Robofuse:* Deployment of the software.
Operations & Management	**Activity** 1. *Measurement services:* Use of Jira for tracking bugs reported from the operational field and the management phase. 2. *Edge Computing:* Use of Jira for tracking bugs reported from the operational field and the management phase. 3. *Vital/IoT:* Use of Jira for tracking bugs reported from the operational field and the management phase. 4. *Robofuse:* Use of TAIGA for tracking bugs reported from the operational field and the management phase.
Maintenance Decommissioning & Recycling	**Activity** 1. *Measurement services:* Manual tracking of bugs discovered and reported by users. Dedicated resolution of reported bugs. 2. *Edge Computing:* Manual tracking of bugs discovered and reported by users. Dedicated resolution of reported bugs. 3. *Vital/IoT:* End/User usage test to ensure the correct performance of GUI. Manual tracking of bugs discovered and reported by users. Dedicated resolution of reported bugs. 4. *Robofuse:* End/User usage test to ensure the correct performance of GUI. Manual tracking of bugs discovered and reported by users. Dedicated resolution

Engineering process phase	Addressed/Focus
	of reported bugs.
Evolution	**Activity** 1. *Measurement services:* Software evolution is driven by: 1) Aggregation of major bugs to solve via a complete software evolution rather than a discrete bug resolution. 2) Strategic enhancements of sensor code to introduce new features. 2. *Edge Computing:* Software evolution is driven by: 1) Aggregation of major bugs to solve via a complete software evolution rather than a discrete bug resolution. 2) Strategic enhancements of sensor code to introduce new features. 3. *Vital/IoT:* Software evolution is mainly driven by strategic enhancements of GUI and Back-End code to introduce new features. 4. *Robofuse:* Software evolution is mainly driven by strategic enhancements of GUI and Back-End code to introduce new features.
Training & Education	**Activity** Production of installation manuals and training document for usage and configuration.

8.1.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 32 UC-08.1 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	Using AHT-EP reduced the design time significantly since integrating with the AHT services led to discarding considerable parts of the previous system's services and architecture thus reducing the amount and burden of the workload to be carried out. Likewise, run time engineering is affected since there are fewer services to be managed and executed on our system. **Planned actions** • AHT-EPP 3: By only interfacing the exposed services of AHT into Vital-IoT and Robofuse we got the goal of being able to handle an increased number of I/Os. • AHT-EPP 4: Introducing K3S services into our USE CASE, we can manage to initialize much more multiple instances of sensors services remotely.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Applying the integration with AHT, it is possible to merge services from different stakeholders with minor adaptation operations thus to empower the framework remarkably. **Planned actions** • AHT-EPP 3: By only interfacing the exposed services of AHT into Vital-IoT and Robofuse we got the goal of being able to handle an increased number of I/Os. • AHT-EPP 4: Introducing K3S services into our USE CASE, we can manage to initialize much more multiple instances of sensors services remotely.
Obj. 3 - Handling of substantially increased number of I/O's	Applying the integration with AHT to Vital-IoT and Robofuse we succeeded in making the activities to add new service producers much more quick and automated, so substantially increasing the number of I/Os theoretically manageable.

WP2 Objective	Focus & Planed actions
due to much more fine grained automation	**Planned actions** • AHT-EPP 3: By only interfacing the exposed services of AHT into Vital-IoT and Robofuse we got the goal of being able to handle an increased number of I/Os. • AHT-EPP 4: Introducing K3S services into our USE CASE, we can manage to initialize much more multiple instances of sensors services remotely.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	This goal is achieved through the documentation provided during the project at the end of every phase, usable in eventual training activities. **Planned actions** • AHT-EPP 8: Production of specific documentation during the project at the end of every phase, usable in eventual training activities.

Table 33 UC-08.1 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Using the AHT platform, we have completely removed the phase of sensors and services integration which were done by the manual installation. Details about the method we used to evaluate such an improvement in section G.g of the WP12410_survey.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Using the AHT API services through registration, authentication and orchestration phases it is possible to link different IoT devices and SoS engineering tools and make the communication happen using MQTT.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The whole process of service discovery of Vital-IoT (our legacy system) was evolved to integrate with ARROWHEAD Framework using HTTP-REST.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Applying the integration with AHT, it is possible to easily link with services from different stakeholders with minor adaptation operations thus to empower the framework remarkably.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and	Probably not applicable.

Project Objective	Focus & Planed actions
automation solutions	
Obj. 6 - Training material (HW and SW) for professional engineers	*Measurement services:* A few training materials are available. The sensors are provided with a short instruction showing how to install and configure the device and some examples to get and send data. *Edge Computing:* Documentation of EdgeX Foundry and K3S are the main source of the training material for this part. Edgex Foundry training material is available on the Web. *Vital/IoT:* Documentation is available from previous projects about Vital_IoT architecture, configuration and integration patterns. Online documentation is available about the integration patterns of Arrowhead. *Robofuse:* Documentation is available from previous projects about Robofuse architecture, configuration and integration patterns.

8.1.4 Engineering Process analysis

Table 34 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-08.1

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1			X	X	X			X	
OBJ-AHT #2		X	X		X				
OBJ-AHT #3			X				X		
OBJ-AHT #4			X	X	X				
OBJ-AHT #5									
OBJ-AHT #6		X	X				X	X	
OBJ-WP2 #1	X	X	X	X	X		X		
OBJ-WP2 #2			X	X	X				
OBJ-WP2 #3			X	X	X		X		
OBJ-WP2 #4		X	X				X	X	

8.2 UC-08.2 SoS engineering of IoT edge devices (AI-Driven Environmental Monitoring) (IUNET)

This use case is applied to whole development lifecycle of an IoT and SoS for a smart energy smart home application.
The UC-08.2 will integrate in the **Edge side processing enabling kit** the component of the “Artificial Intelligence Driven Camera” **at minimum marginal cost** consisting in Ultralow power QVGA camera, a vector processor based on RISC-V GAP8 architecture and a multi-protocol

radio capable to switch and select the most appropriate IoT wireless standard. The ultralow-power machine learning optimized core, will be used as vector processor with a Low Power QVGA Image Sensor. An on-board energy harvester will satisfy the energy requirements, and make the system energy-neutral.

The UC_08.2 is intended to demonstrate the impact of an AHT-EP supporting the integrated and automated design, deployment and configuration of **AI-driven monitoring applications**. Data processing in done on the IoT device, and information/results collection from AI-driven Cameras is facilitated by a gateway which acts as a bridge between the field devices and the cloud. The data collected and processed on edge is stored on the cloud, where it can be further processed and can be made available to third parties (including the end users) through Arrowhead Services and applications.

The Ai-Camera makes possible many added-value services and enables cross-domain applications, for example by making access control more efficient through the prompt identification of authorized people.

The following stakeholders compose the value chain of this use case:

- StkH1: company that designs, develops deploy the AI-cameras
- StkH2: IoT integration platform and end user applications provider
- StkH3: company that designs and develops the AI-Software for the AI camera

The use case focuses on an AI-access control system to be used for professional indoor/outdoor environments, such as the Structural Health monitoring scenario of UC-08.2.

The main advantages for the different stakeholder are the following ones:

- For the consumers it will be translated in an access control service, with no need of continuous training of the model. The service will be easily configurable on the cloud.
- StkH1 will benefit of a major flexibility and a reduction of risks: the solution will be updated remotely, without the need of multiple maintenance.
- StkH3 will update and provide much more tailored Ai algorithms, energy efficient for the dedicated Camera. The company will provides the algorithms, and will require only software update with reduction of costs in the evolution of the product.

8.2.1 Overall description of the UC-EP

The system is an end-to-end solution for the domestic market, and composed as described in the following.
In the following the functional blocks of the AI-drive Camera for Environmental monitoring:

1. AI-ZeroEnergy-camera: consisting in a multiradio protocol, a QVGA camera and a RISC-V vector processor;
2. Cloud-Based Services: Data are collected and sent in burst mode via Wi-Fi or LoRaWAN on the cloud;

In Figure 23 the functional blocks of the UC.
The main objective of this use case is to provide two Arrowhead Services that is transparent with respect his technical details, complexity and the heterogeneity of the deployment scenarios.
The use case is composed by an innovative Ai-Camera is an innovative HW design based on RISC-V microarchitecture, capable to process on the device images from the QVGA camera using advanced ML algorithms. The baseline of the AI-Driven Camera is represented by the RISC-V microarchitecture and its camera. A Multi wireless protocol is capable of connecting

to the internet using the technology available in the deployment scenario (e.g. wi-fi, Lora, BLE), being robust to legacy infrastructure already available.
The image sensors operate at 16fps, and periodically send processed results to the cloud for the services to be implemented using the Arrowhead Framework. Results of the local processing is permanently stored in a database,
The Arrowhead Framework provides a “global service” for manage the AI-Cameras and for providing services tailored to the market. Each customers of the proposed solution can publish its services and use data from the Ai-camera, At this level, the framework also enables the creation of multi-stakeholder or/and cross-domain applications.
End user applications are typically developed by third parties that don’t know the low-level, technological and architectural details of the specific vertical domain in this case the control of accesses in an area. This is a good example of the key factor provided by the Arrowhead Framework, as provided also in WP7, WP8 and WP9 deliverables.

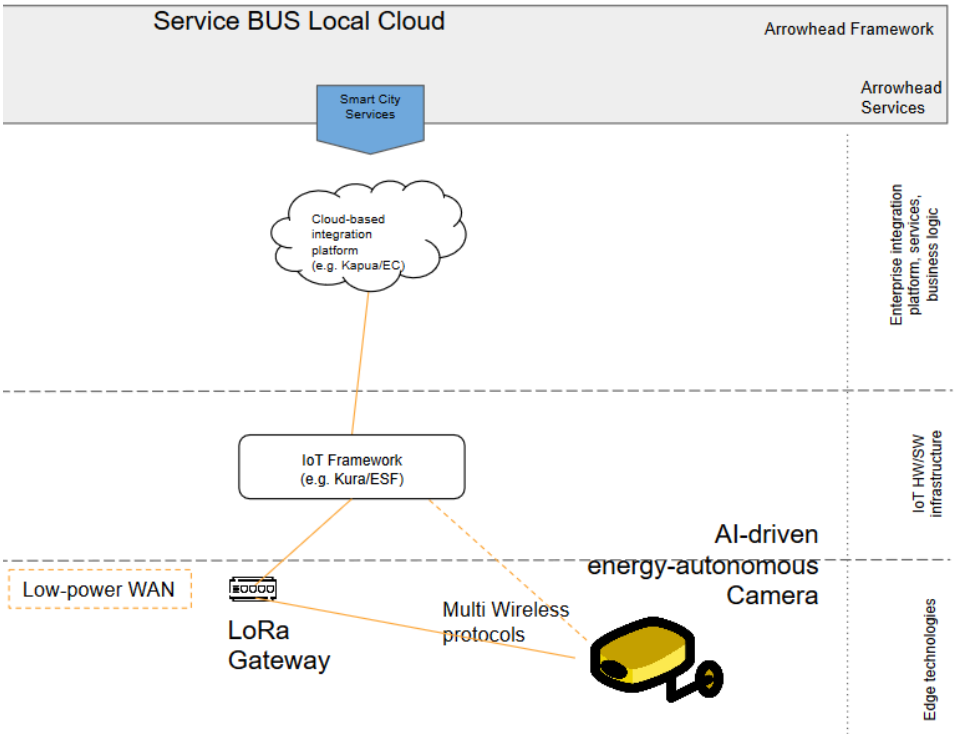


Figure 23 The functional blocks of the UC-08.2

8.2.2 Engineering Process Description

In the following the description of some of the activities implemented in the engineering process depicted in Figure 24.
Requirements: StkH1 and StkH3 acquire the specification describing the features of the camera and type of identification from the customer (StkH 2). Specification documents are passed in the form of Word files.
StkH3 together with experts of StkH 1 analyse the legacy wireless availability by the customer deployment environment. Documents are don in Word files.
StkH1 contributes to the requirements for the specific wireless and energy requirements. Documents are don in Word files.
Functional Design: StkH 2 design and simulate the AI-Camera Hardware. StkH 3 will develop a model of the Classification algorithm for the AI-Camera.
Procurement & Engineering: StkH 2 designs the PCB with Orcad and select the components and technologies for the PCB design. Develop the firmware for AI-Camera sensors using vector processing compiler and GCC toolchain.
Deployment & Commissioning: StkH1 produce the AI-Camera and supply them to the vendor product lines. StkH3 compile and install the program on the AI-Camera before to be supplied to the StkH1 product lines.
Operation & Management: StkH2 is in charge of monitoring the StkH1's instance of the IoT integration framework and ensure it is operating correctly.
Evolution: Data collected in the Operation & Management phase of StkH2 are analysed by StkH1 for identifying possible bottlenecks in the AI-Camera design. Moreover, firmware updates will be generated by StkH3 for keep a good standard of cybersecurity. Data collected

in the Operation & Management phase of StkH2 are analyzed by StkH3 for improving the algorithm and increase the energy performances.
Training & Education: StkH1 create the datasheet of Ai-Camera PCB and reference manual of low level API. StkH3 will document the code to be shared with the StkH1.
StkH3: source code is documented and specific tools are used to generate the related documentation. User manuals are edited using Microsoft Word.

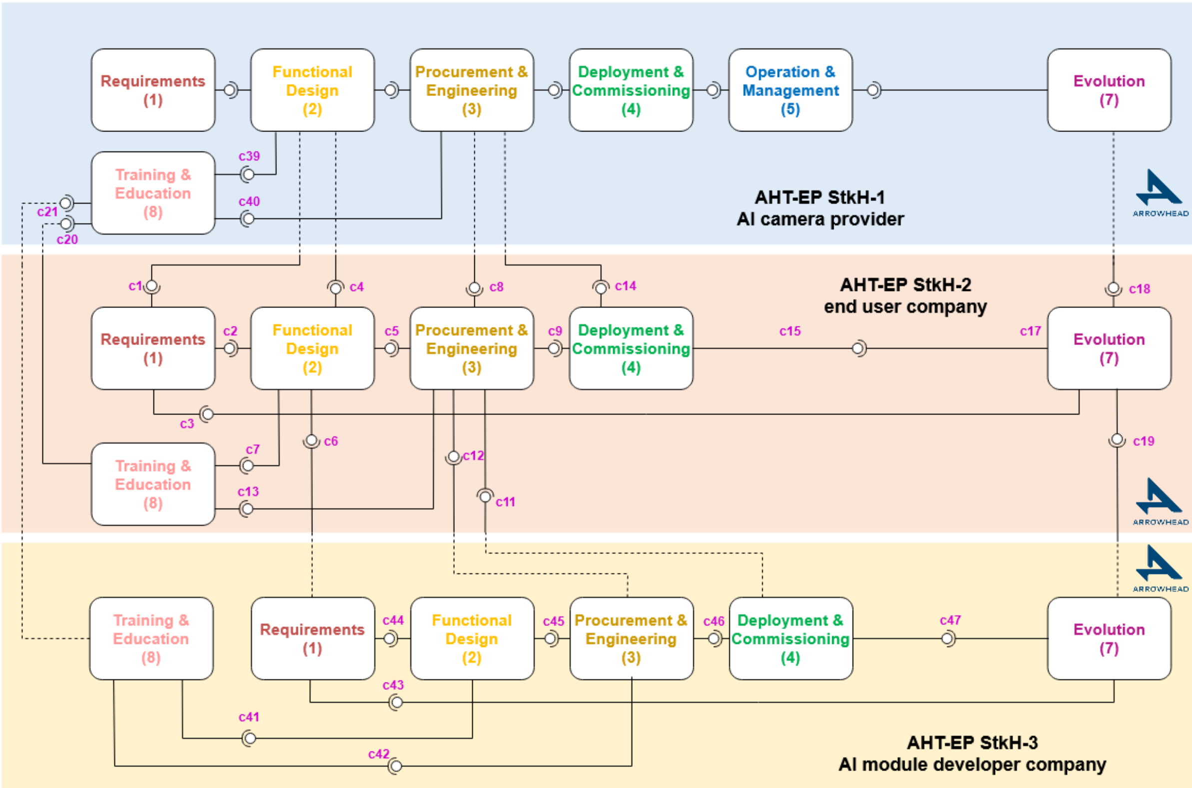


Figure 24 AHT-EP of use case UC-08.2

Each phase of the UC-EP can be perfectly mapped to the corresponding phase in the AHT-EP. The only difference is that the maintenance and training phases are missing from the UC-EP because they were not planned.
UC08_02 component will be directly implemented by using the AHT-EP. The input and output of each phase are described in table WP12410_survey.xlsx, sheet C.b.
UC08_02 components will be perfectly integrated with the Arrowhead Framework once implemented.

The maintenance and training phases are not considered in the engineering process of the AI-Driven Camera component, because they were not planned given the simplicity of the system. The IoT integration platform has a service oriented architecture that will simplify the integration with the AF. The tools adopted for the documentation management are not service oriented and will require the development of a specific tool (adapter) for their automation.
The deployment and maintenance will require the development of a specific tool for their automation.
Moreover, Management of the AI-drive cameras need to be fully automatized; and Evolution Phase is currently not supported.
The AHT-EP will help in promoting this phase by providing new technologies to automatize novel methodologies.

The information passed between the requirements and the functional design phases is in the form of documentation and is managed manually. The same is valid for the information passed between the functional design and the procurement/engineering phases.
The information passed between the procurement/engineering and the deployment phases is represented by a zip files file containing the entire application system to be deployed. No information is passed between the deployment and the operation phases. Finally, the engineering phase passes all source codes to the evolution phase.

The AHT-EP can be applied separately to each component, as well as to the whole system. Furthermore, the AHT-EP can cover multiple stakeholders working independently. In particular the described the EP of the end-user company (stakeholder 2) as a main AHT-EP connected to other AHT-EPs describing the life cycle of two sub-components: the AI camera, and the AI module algorithm for the camera. With this use case we demonstrated that we can adopt the AHT-EP models from different stakeholders, and forming a System of Systems.
The engineering process of the use case follows a standard waterfall model, and compliant with some of the dimensions represented in the RAMI 4.0 model.

Now, the UC-EP perfectly matches the AHT-EP. In the original version, the AI camera provider and the AI intelligent module software were developed by the same organization without any open-interface for external AI modules. Now, the AHT-EP permits to enhance the AI camera also with third party software modules.
The possibility to evolve the product is explicitly mentioned in the EP so we can support some aspects of the run-time engineering paradigm. Automatization of the process will be easier due to a set of training and learning resources produced for technicians, installers, associate shops, and final users.

Table 35 AHT-EP Phase focus of UC-08.2

Engineering process phase	Addressed/Focus
Requirements	StkH1 and StkH3 agrees with the customers about the Ai-camera features and the tailored access control requirements. Performance requirements collected by end-users, should be exhaustive.
Functional design	Performance requirements are analysed to generate a list of Technical requirements of the HW and SW design of the camera. For example, the type of algorithms to be executed with vector processing will be defined. The system modules are designed in order to satisfy the requirements defined in the previous phase.
Procurement & Engineering	*Procurement*: StkH1 receives from the customer the information about the wireless capability in the deployment environment. StkH1 will develop the system modules with specific Hardware configuration, StkH3 will provide the tailored implementation of the algorithm for the control of accesses. *Engineering*: • Image processing training and validation phases will be executed as offline procedure • Image Classification will eventually run on the AI-drive camera after an offline test. • New software releases will be assessed at server side and offline, before updating. • Release of the new software versions

Engineering process phase	Addressed/Focus
Deployment & Commissioning	The software is finally released. The provider module is deployed on a server, while the consumers are delivered from time to time to third parties.
Operations & Management	The system is up and running: the access control provide periodic data information about the confidence of the recognition and classification.
Maintenance Decommissioning & Recycling	Analysis of possible bugs raised from a large scale deployment.
Evolution	StkH3 may plan to extend the system with additional services derived from Python data analytics.
Training & Education	StkH3 produces the training material for both end users and developers.

8.2.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 36 UC-08.2 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The IoT integration platform introduces new remote monitoring functionalities that will allow to profile the access in an environment thanks to near-camera analysis. These functionalities could be exploited for identifying accesses, but in particular to automate the classification of people even in case of partially covered faces. This objective is matched since in the UC we are adopting the evolution phase where we evaluate feedback data collected in the field. This data is analysed by the stakeholders for identifying rooms of improvement of the component for which each of them is responsible. During run time engineering the system may be extended to provide additional services from Python data analytics. **Planned actions** • AHT-EPP 4: The IoT Integration Platform will provide feedback on the AI-Camera status. • AHT-EPP 5: The IoT integration platform will provide information on the AI-Camera that could be used for the engineering process (e.g. bug fix automation) and added value services (e.g. preventive maintenance). • AHT-EPP 6: The IoT integration platform will provide feedback on the AI-Camera in phase 5. • AHT-EPP 7: The IoT integration platform will provide feedback on AI-Camera status that could allow to identify new functionalities or new releases of the system.
Obj. 2 - The move from single to integrated multi stakeholder	The system will integrate different information from other sensors from multiple stakeholders, including third-party legacy infrastructures. All the engineering phases are intended for a single stakeholder that can produce more than one component. We decided to represent with the AHT-EP model all the

WP2 Objective	Focus & Planed actions
automation and digitalization	engineering processes from the end user (StkH2) to the AI camera provider (StkH1). All the EPs are integrated and connected between each other internally and externally (different StkH). Connections are not fully automated and several of them are manual. During the project we will automatize some of these connections by interfacing the modules with and without the AHT Framework. For example, thanks to the AHF, we will be able to connect multiple services (e.g. UC8_03) in a unique stream, connecting their toolchain in order to achieve automatic interactions. The baseline lacks severely of this automation as stakeholders have to iterate over manual interactions. **Planned actions** • AHT-EPP 5: The IoT integration platform, coupled with the AF, will allow the management of multiple streams of data coming from multi-stakeholder data sources. • AHT-EPP 6: The feedback provided by IoT integration platform can be useful for the maintenance operator.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Several phases of the EPs have multiple interactions with the other phases of the same EP and EPs from other stakeholders. Interactions, represented in the AHT-EP model with connection lines, are implemented by exchanging information between tools contained and operating in each of the AHT-EP phases. Each phase of the EP related to the StkH2 includes the toolchains for the two different components (AI camera, and AI algorithm). As a direct consequence of the improved interaction between the stakeholders, their engineering process during the steady state of the use case, increased in number of I/O. Functionalities, interactions and data streams are also increased in number.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	SthK1 and StHK3 will improve the automation level of the tools adopted for the editing of the documentation of the IoT integration platform and, partially, also of the use case (including: source code documentation, technical documentation, user manuals, application manuals, etc.). All the three StkHs will provide user guides and manuals. Moreover, interactive tutorials for installer technicians (StkH1) and final users (StkH2).

Table 37 UC-08.2 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	AI-Camera reduces the engineering costs by providing a general data-sharing platform that can be easily adapted to manage data for multiple solutions. The customer needs only small updates in order to introduce a new data source. Code documentation updates can save developers time by making the documentation editing process faster, coherent and more systematic. Indirect reduction of the costs is achieved by the improved quality of the final product which shows full automatization of the engineering process and therefore is less prone to human errors. Eventually, the save is also achieved in the amount of data transmitted to the cloud (Efficiency in data transmission (1000X reduction in data transfer))
Obj. 2 - Interoperability for IoT and SoS engineering tools	The AI-Camera guarantees interoperability between different sensing monitoring systems.

Project Objective	Focus & Planed actions
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The AI-Camera component allows the interoperability between different third-party legacy infrastructures, and permits to connect to multiple wireless technology, (Lora, WiFi, BLE, NB-IoT).
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	The IoT integration framework is based on popular solution for IoT and edge computing. The framework will be integrated with the Arrowhead Framework.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The Arrowhead Framework provides a security mechanism for ensuring that only authorized third-party consumers can access the data. Furthermore, the Arrowhead Framework provides the encryption of communications between consumers and providers.
Obj. 6 - Training material (HW and SW) for professional engineers	The full documentation of the source code will be available for supporting the developers who are in charge of integrating new legacy infrastructures. The end users will be provided with a short description for deploying the Consumer application in their working stations, in addition to the API documentation of the provider application.

8.2.4 Engineering Process analysis

Table 38 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-08.2

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	P	C	C	N
OBJ-AHT #1	X	X	X		X		X	X	
OBJ-AHT #2		X	X						
OBJ-AHT #3	X	X	X	X		X			
OBJ-AHT #4			X						
OBJ-AHT #5				X	X				
OBJ-AHT #6		X	X	X	X		X	X	
OBJ-WP2 #1	X	X	X	X	X	X	X		
OBJ-WP2 #2	X	X	X		X	X	X	X	

OBJ-WP2 #3	X	X	X		X			X	
OBJ-WP2 #4		X	X	X	X		X	X	

8.3 UC-08.3 SoS engineering of IoT edge devices (Condition Monitoring) (IUNET)

UC-8.3 is one of the demonstrators of UC-8, and it is intended to demonstrate the impact of an AHT-EP supporting the integrated and automated design, deployment, configuration, etc. of SoS for Condition Monitoring, based on IoT edge devices. Condition Monitoring is the process of measuring the status of an entity over time. Entity related condition data are collected to establish trends, recognize anomalies, degradation, failures, and failure risk increase and estimate remaining life. Through condition monitoring the maintenance process can be optimized, by improving the trade-off between accepted risk level and maintenance cost through efficient data analysis.

Condition Monitoring is usually based on Systems of Systems, consisting of sensor networks that monitor the integrity and the state-of-health of the addressed structure (e.g. P_SHM or Structural Health Monitoring Platforms). One of the limiting factors in the extensive use of Structural Health Monitoring systems are the initial costs related to the design of the sensing architecture, its installation and its maintenance costs. The current solutions are complex and require a big effort in terms of time and costs at design time. Moreover, there is no standardization yet in the data exchange between different P_SHMs, limiting the possibility to obtain a dataset usable for the implementation of new smarter, adaptive and fast ML and AI algorithms for an effective monitoring of the structure of interest.

The use case baseline consists of, e.g. dedicated sensors and sensor networks, organized in an IoT-inspired architecture, enabling, in principle, the creation of large data sets. Our previous engineering process of developing the sensor network employs dedicated algorithms for feature extraction and works with the Web of Things (WoT) paradigm, as a communication and entity representation standard. Although WoT is a newly accepted standard by the W3C, it still has little practical use in industrial scenarios and it demands for an integration with other interoperability frameworks. Furthermore, the deployed sensor network has no scope for runtime reconfiguration neither optimization nor the only visualization tool available cannot interact with entities outside the WoT ecosystem. From a hardware point of view, the sensor network is based on Peripheral Sensor Nodes connected in a daisy-chain fashion and orchestrated by a Cluster Head, a system currently affected by some constraints. A first limit is given by the CH, because it's not a purposely devoted central node, but it's a PC (i.e. Raspberry Pi) that is used for data storage and local data processing. The PSN circuitry and task organization is not optimized for power consumption minimization. The whole network is powered by the electric mains and this lack of autonomy hampers the deployment of such SHM solution. Furthermore, the scenario includes other stakeholders, one of them developing a Gas Sensor that is under constant improvement, which is not connected nor interoperable with the inertial sensor network mentioned above. Finally, another stakeholder needs to develop a unified dashboard and cloud-based controller, however, the isolation of the sensor networks presented makes this integration difficult at M0.

Therefore, novel toolchains and a revised engineering process are proposed to make SHM and Condition Monitoring more sustainable, cost effective and interoperable. The baseline might be drastically moved forward with appropriate tools to overcome the costs and technical difficulties during the specification, design, setup, operation and maintenance phases. An appropriate engineering process and the related value chain could also increase the flexibility and the "learning"/"evolution" capability of condition monitoring platforms as well as decrease the training costs of the personnel involved along the life-cycle.

More specifically, to make SHM and Condition Monitoring more sustainable, within the scope of the project we propose to:

a) Define a new service oriented architecture based on the AF.
b) Define an approach to collect data from multi-vendor heterogeneous sensors and sensor networks (i.e. edges and extreme edges) and make the collected data consistent with a shared data model. This includes the potential inclusion of off-the-shelf sensors external to our initial architecture.
c) Create a tool chain to accompany the engineering process of IoT-based monitoring platforms that optimize the data collection process, run-time configuration and deployment and resource optimization with all the associated costs.
d) Create an infrastructure to enable future tools to support prognostics based on condition monitoring.
e) Generate new services for the creation of added-value domain and cross-domain applications.

The AF should be the enabling technology supporting data collection and integration also at edge level (as can be seen in the proposed architecture of UC-08.3 in Figure 25), and to demonstrate this, we propose a general architecture suitable for condition monitoring both in industrial and smart city scenarios (a "helicopter view" is shown underneath). The main purpose of the mentioned toolchain is to reduce the engineering costs of monitoring systems along their life cycle as well as the integration of legacy and Arrowhead-compatible third-party systems. Moreover, an improvement in the benefits for the end users is expected.
Starting from the baseline, the underneath figure, illustrates the new solution we propose and shows that our approach is in principle open, so that other demonstrators might be integrated in the proposed architecture: a variety of heterogeneous platforms (P_x) may offer their data to local and external services through the Arrowhead Framework, should the need or the opportunity arise.

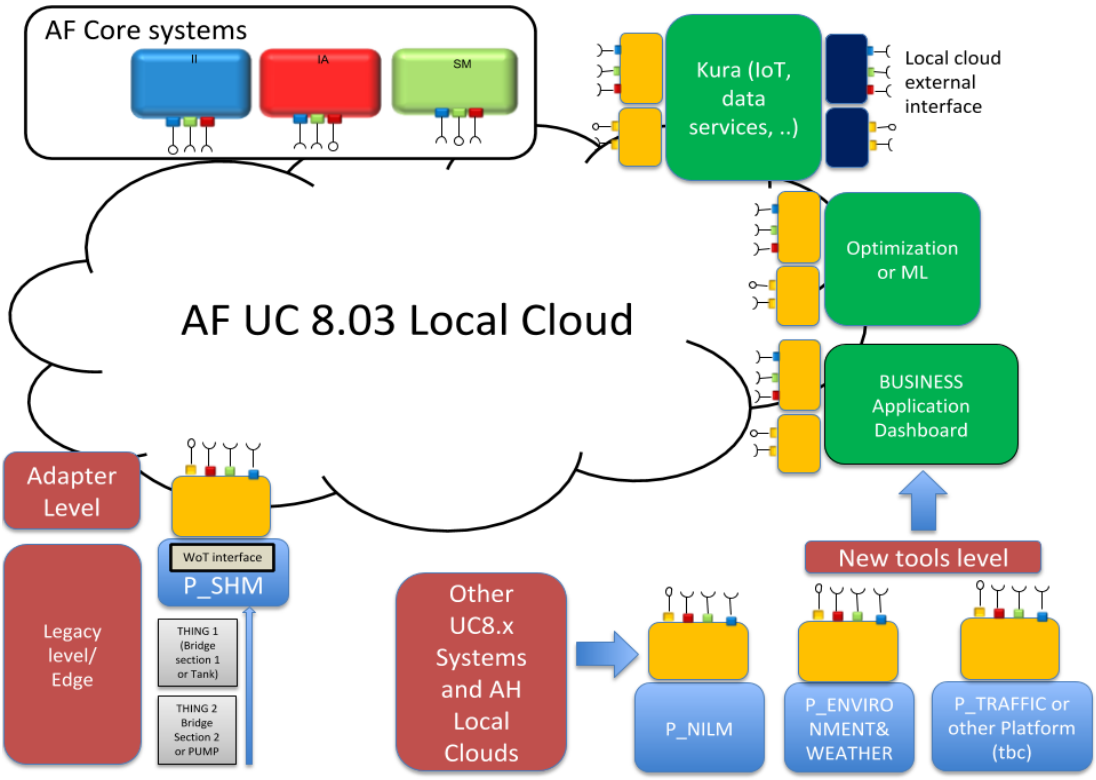


Figure 25 Proposed architecture of UC-08.3 integrated within UC-08 use case through the AF

8.3.1 Overall description of the UC-EP

A subset of the architecture shown in Figure 25 will be set up, as UC08.3, to demonstrate the proposed solution. The demonstrator itself is going to be a structural health monitoring system (including P_SHM) for a target structure (e.g. a section of a bridge or a construction that could be civil or industrial).

The SoS will consist of multiple IoT networks, including networks of sensors installed on the target structure, a monitoring station for environmental monitoring, that collects and analyses air quality parameters, electromagnetic fields, and ionizing radiation, and additional third parties data sources, such as for example meteo data services and other SHM network such as the PMUT-based one. The demonstrator has at hand a Gas Sensor developed within ST-I, which at the baseline stage, has no connection with the rest of the structure.

The baseline is shown in Figure 26, where we see the SHM multitude of sensors already hooked to a WoT ecosystem and the Gas Sensor as standalone resources.

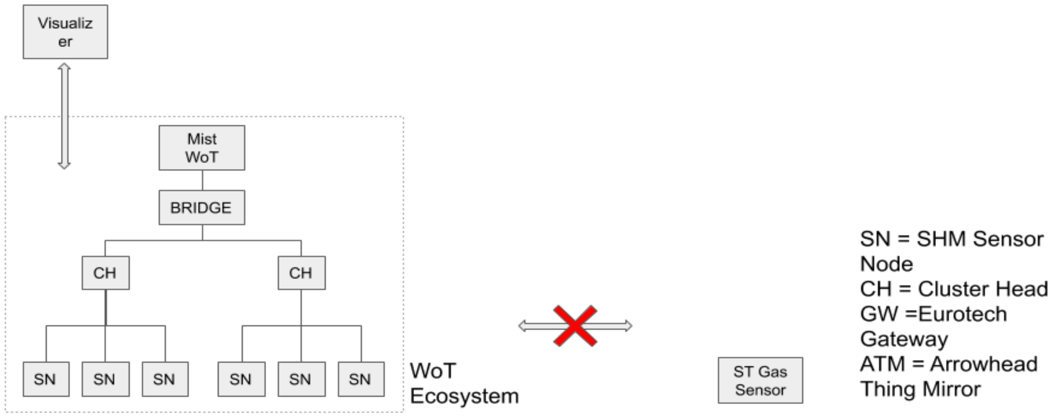


Figure 26 Baseline architecture of UC-08.3

In particular, the UC is realized and utilized by four distinct stakeholders (a fifth one is mentioned at the end of this section as a future extension):

- Stkh1 (IUNET): Develops the inertial SHM WoT-based sensor network
- Stkh2 (ST-I): Develops the Gas Sensor
- Stkh3 (ETH-IUNET): Develops the control dashboard and takes up the aggregation of all the edge devices in an IoT interoperability Framework
- Stkh4: Is the final user, making use of the aggregated data and performing, if necessary, the NDT (Non-Disruptive Test) check.

For Stkh1, in detail, sensor nodes are devoted to collect data, which is organized and collected by their Cluster Head (Figure 27). Several Cluster Heads are in communication with each other through a hardware bridge and on top of everything we have a WoT Mist device. This software device is able to get the data and instruct commands to the sensor nodes and associates each of them with a Web Thing, so that sensor nodes are virtually reachable from the external via WoT. A WoT-compatible visualizer is developed and communicates with the Mist WoT device to visualize the data produced by the sensors. The plan at M0 is to extend the Visualizer to gather data also from the Gas sensor with an ad-hoc extension. The Visualizer should also be extended to provide configuration to the nodes, but it has no automated policy, instead an operator can push the configuration manually.

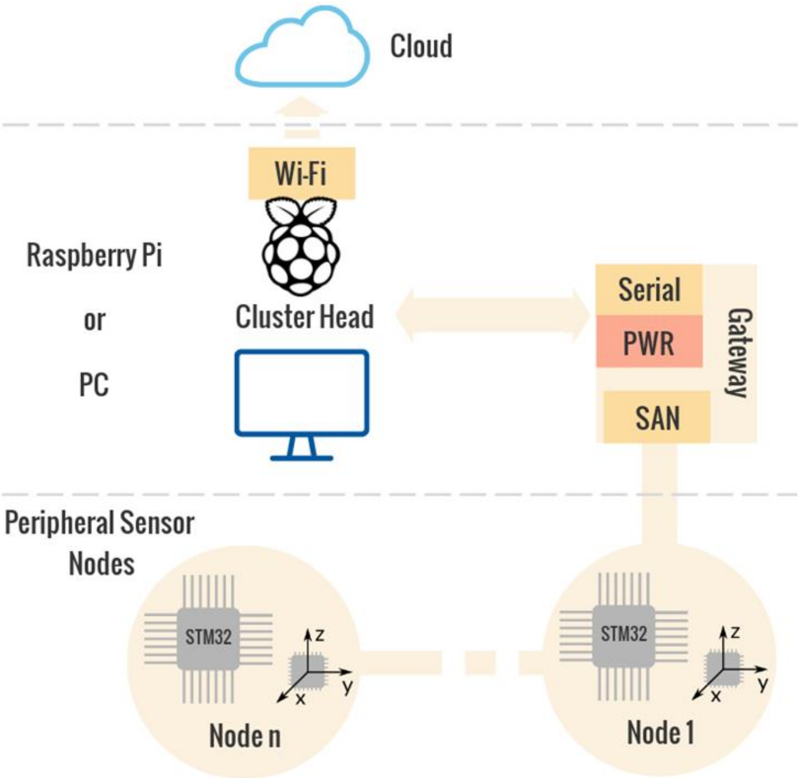


Figure 27 SHM baseline hardware architecture - M0

For Stkh2, at M0 the gas sensors are not yet in the market; they are stand alone sensors with no interconnection with any network. During the project execution the final testing and customization to different gases will be conducted. Furthermore the sensors will be interconnected to the STM32 board to constitute the node of the gas sensors network.
For Stkh3, we have a Kura instance that needs to be connected to all the edge devices manually, as all of them use a dedicated interface.

Stkh4 then can only aggregate data manually, as all the edge devices work on their own as isolated ecosystems.
Within the scope of the project, we expect to create an interconnected architecture as shown in Figure 28.

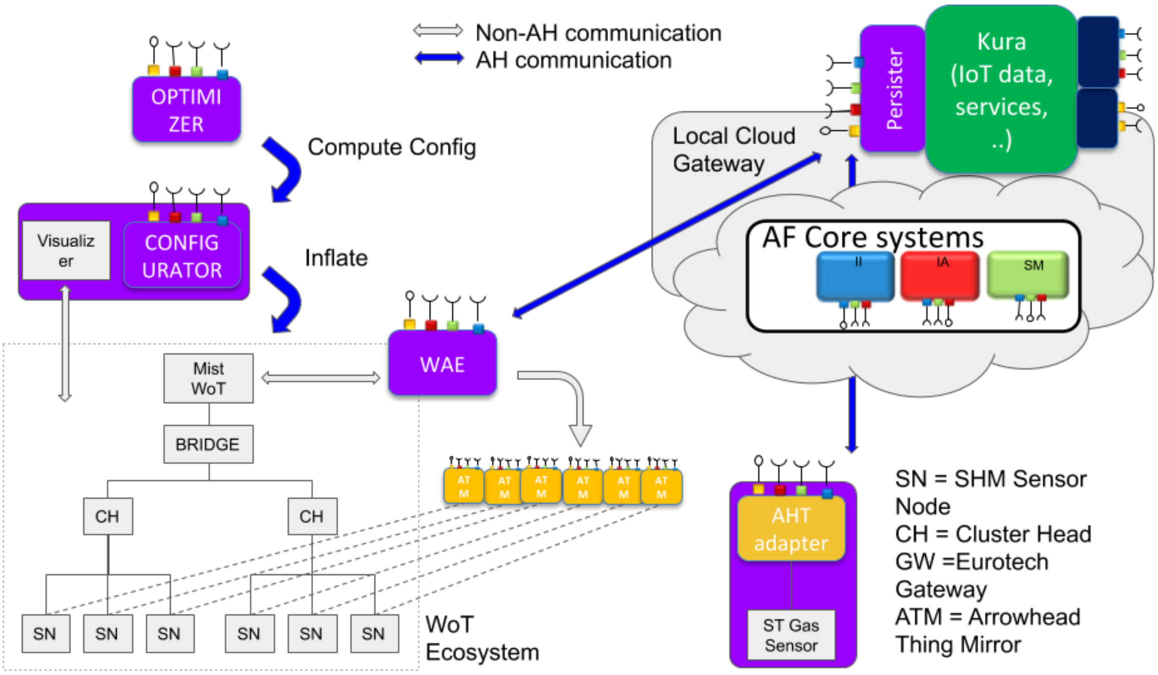


Figure 28 Interconnected vision of the outcome of the project (M36).

8.3.2 Engineering Process Description

Figure 29 shows the current state of the AHT-EP expected for M36, taking into account all the interactions between the various stakeholders. Stakeholders were introduced earlier and we can see that all of them use all the EPPs 1 to 5 (which is pretty much normal when building up an artifact that should be operative sometime. Also all of them use EPP8 because training is always present as an output (sometimes as an input too). StkH 1 and 3 also make use of EPP 6 and 7 because they are in charge of detection of any anomaly concerning the whole SoS and gathering data about the usage to trigger a new development cycle (i.e. they are the operative stakeholders). StkH 4 only uses EPP5 because it represents the final user who does not produce anything. Interesting interactions are connections 9..14 between sth1 and stkh3, as stkh1 is the main data producer and stkh3 is the main consumer/analyzer/reader, therefore it is obvious that constant data has to be streamed from stkh1 to stkh3 at runtime. c28-c29 tell that the user has access to the interface provided by stkh3 during runtime phases. c30 gives input to the PMUT developers (which is not present in the baseline) on whether, when and where it is necessary in the SoS so a simulation scenario will be designed. c15-c16 represent deployment information and streaming data from the gas sensor to the WoT sensor system.

Regarding Training, we can observe how in most cases training comes from EPP3 of the respective AHT-EP, which represents the actual instruction manual produced while implementing/assembling the artifact(s). Also, it is notable how all the training activities are interacting among each other (seminars and tutorials) between stakeholders and they are all sinking into stkh3 training which feeds into EPP3 of its AHT-EP. This happens because the Local Cloud Gateway is the final aggregator of the data and inputs from the other artifacts.

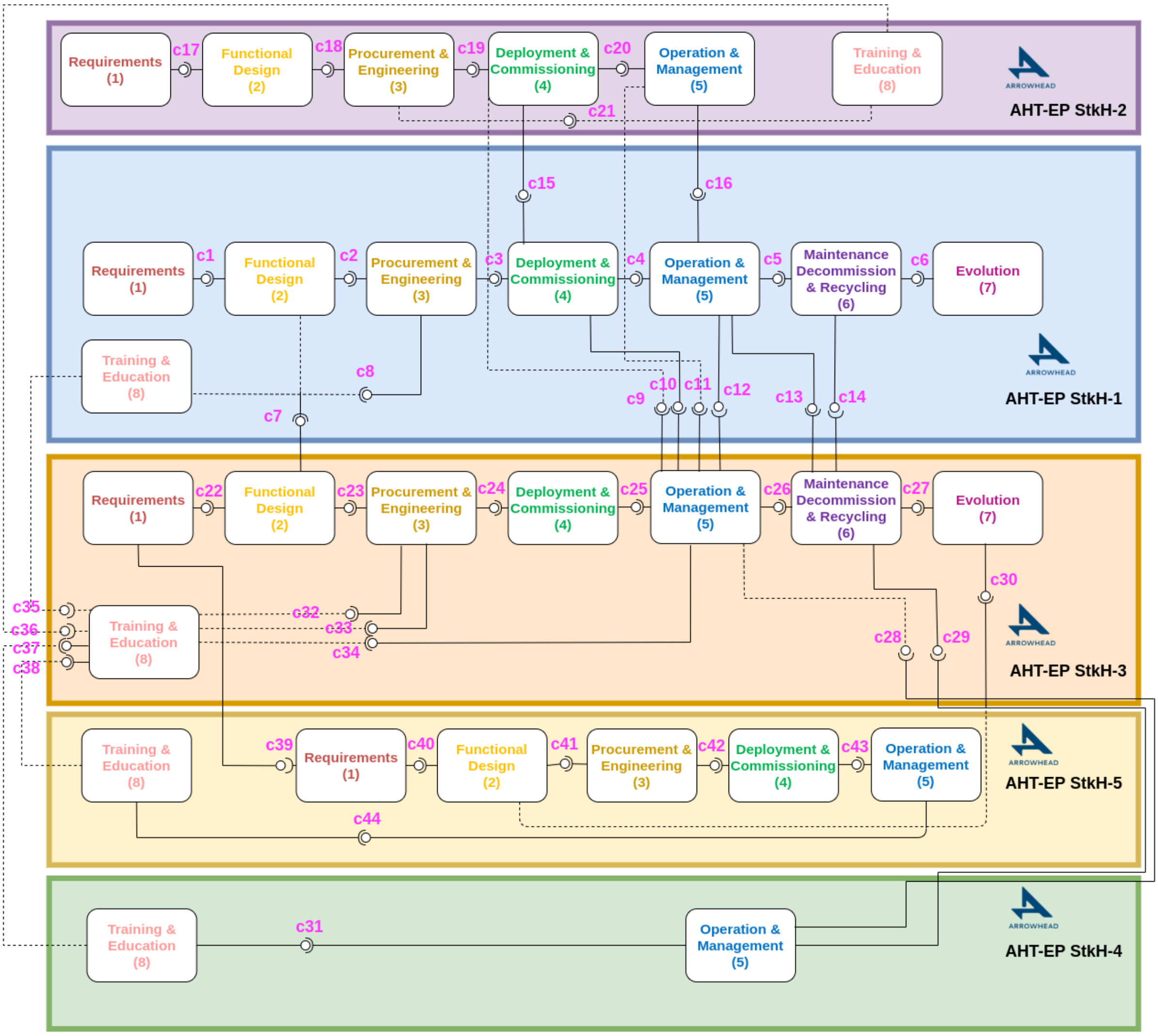


Figure 29 AHT-EP of use case UC-08.3

The engineering process of the whole use case is already aligned with IEC 81346 and its Arrowhead Tools extensions. Currently, with respect to the Local Cloud Gateway, only the Eclipse Integrated Development Environment is used for phases 3-4 and there isn't a framework that could be adopted to manage the EP. Many large enterprise already has frameworks for the management of the EP of IoT based applications (e.g. Tesla, BMW, Siemens, IBM, etc.), but these frameworks are proprietary and closed solutions, that have been conceived for specific vertical applications.

The WAE has been implemented in Y1 and it is interoperable with the AHF.
The Gas Sensor has been successfully integrated with the AHF as well and we have shown, as an outcome of Y1, that we are able to consume data from both the platforms by using a simple version of an AHF- compatible persister.
The second year of the project will see a full integration of the Configurer within the main stream of tools.
The extension/integration of the IoT framework based on Eclipse Kura is ongoing. The development of Dataflow IDE is ongoing. While the development of the use case specific business logic and the REST API for the high level services will start during the second year of the project.

Stkh1 engineering process covers all the phases of the AHT-EP.
Stkh2 engineering process covers phases 1 to 5. Further phases are not needed as the Gas Sensor helps such phases for the main stakeholder.
Stkh3 engineering process covers all the phases of the AHT-EP.
Stkh4 is operating only in phase 5, as the final user is considered to be an operative agent which should not participate in decisions concerning the design phases of the use case nor the evolution. Obviously the participation of the user is essential also in phase 8.
StkH5 is reported only for EPP2, because we envision this as a future development for our use case and only explorative studies are performed within the project.

The EP adopted for the Local Cloud Gateway is not currently service oriented and not integrated with the AF. If required, the open source tools could be easily adapted to export services on the AF, while the proprietary ones require a deep analysis of the available interfaces, API and shareable information.
Lack of multiple cloud access (possible thanks to the Local Cloud Gateway)
Complete lack of fleet management
Absence of a solution to manage interoperability between WSNs
The solution proposed in this use case will address both the enhancement of the EP and lack of technologies/solutions in the use case specific domain.

For the Sensor Network and the Visualizer, the EP span across all the conceptual layers, while for the Gas Sensor it currently span only the product layer (further layers will be included throughout the project).
For the Local Cloud Gateway, the EP covers all the abstraction levels from the hardware to the final product, across all the phases of the EP.
The multi-stakeholder objective is reached as multiple stakeholders are already involved in the Use Case since the baseline was defined.

For the Local Cloud Gateway, the AGILE project management methodologies are adopted for software development: StkH3 (ETH) adopts the SCRUM methodology. The security aspects of the EP are addressed following the OWASP Secure Coding Practices, with the future adoption of tools that automatically check the code and evaluate the security levels.

Table 39 AHT-EP Phase focus of UC-08.3

Engineering process phase	Addressed/Focus
Requirements	**Activity** • *SN & CH*: excel, project management, MatLab executable specs. • *Local Cloud Gateway*: Requirements elicitation. Specifications definition.
Functional design	**Activity** • *SN & CH*: Design of the sensors. • *Gas Sensor*: Task partitioning, programming, simulation, configuration. • *Sensor Network (WAE)*: Study of the placement of the sensors on the structure, study of how to connect each other and where to put Cluster Heads. • *Local Cloud Gateway*: Functional architecture definition. • *PMUT Network*: IoT-Ticket Application Templates, Versioning, Publishing, Access Management, Device Management, Arrowhead Framework.

Engineering process phase	Addressed/Focus
Procurement & Engineering	**Activity** • *SN & CH*: Procurement of PCB manufacturing. • *Gas Sensor*: Board prototyping, Virtualization of gas metering value in an API that IoT node /gateway can call. • *Sensor Network (WAE)*: Acquire the necessary gateways and hardware for the Mist WoT with Public IP address. Development of the Mist WoT. • *Visualizer (Configurer*): Acquire the necessary gateways and hardware for the Visualizer with Public IP address. Development of the Visualizer. • *Local Cloud Gateway*: During this phase, two processes develop in parallel: 1) design, development, test and debug of the Local Cloud Gateway hardware and software; design, development, test and debug of the use case specific business logic; 2) interaction with suppliers to acquire electronic and mechanical components, define external services to manufacture hardware prototypes; interaction with suppliers to acquire source code, software libraries and software licenses potentially required for the software part of the gateway. • The objective is the production of the Local Cloud Gateway golden sample. • *PMUT Network*: IoT-Ticket Application Templates, Versioning, Publishing, Access Management, Device Management, Arrowhead Framework.
Deployment & Commissioning	**Activity** • *SN & CH*: Virtual sensors fielding and estimation of positioning. • *Sensor Network (WAE)*: Linking of the Visualizer and the Mist WoT with the SN. • *Visualizer (Configurer*): Linking of the Visualizer and the Mist WoT with the SN. • *Local Cloud Gateway*: Deployment of the prototypes to test and debug them in a real environment to evaluate their maturity. Commissioning of the golden sample to start the production of the product.
Operations & Management	**Activity** • *SN & CH*: Sensors connected to CH and therefore to the cloud. • *Sensor Network (WAE)*: Usage of the Visualizer to infer potential Hazards. • *Visualizer (Configurer*): Infer Hazards. • *Local Cloud Gateway*: Integration of use case components on the field. Data collection, storage, local processing. Remote monitoring and control. Interfacing with cloud and enterprise level.
Maintenance Decommissioning & Recycling	**Activity** • *SN & CH*: Verify if something is broken. • *Sensor Network (WAE)*: Configure the sensor optimally in order not to waste energy resources. Done Manually. • *Local Cloud Gateway*: Remote monitoring and control for maintainance and decommissioning purposes.
Evolution	**Activity** • *Gas Sensor*: Task partitioning, programming, simulation, configuration. • *Sensor Network (WAE)*: Linking of the Visualizer to the Gas Sensor and PMUT as well. Manual redesign. • *Local Cloud Gateway*: Faults and bugs analysis to identify solutions, improvements and new product releases.
Training & Education	**Activity** • *Local Cloud Gateway*: Generation of design and source code

Engineering process phase	Addressed/Focus
	documentation. Editing of technical and user manuals.

8.3.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 40 UC-08.3 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The Optimizer and the Configurer are tools that, thanks to the improved interoperability offered by the AHF, can communicate to the edge devices of the SHM Inertial sensor network and change configurations at run-time, according to external conditions that are automatically detected. This was not previously possible. The IoT integration platform, partially running on the Local Cloud and on the remote cloud, will introduce new remote monitoring functionalities that will allow to profile the behavior of the infrastructure health monitoring system during phase 4, 5 and 6. These functionalities could be exploited for predictive and preventive maintenance, but in particular to automate the detection of faults, bugs, unplanned behaviors and trigger the design and development process to correct them. The objective is in particular met by means of the evolution phase used by stkh1 and stkh3. The data produced by both the WoT sensor system and the Local Cloud Gateway is analyzed to identify room for improvement, actually every year this might result into a new tool. c30 for instance shows how the outcome produced by StkH will provide design guidelines for the PMUT simulator. **Planned actions** • AHT-EPP 2: Here Optimization will run constantly to eventually come up with a new version of the configuration for the sensor network. • AHT-EPP 4: Configuration of the SHM inertial network will be automatic thanks to the Configurer. The IoT Integration Platform will provide feedback on the health monitoring infrastructure status already starting from phase 4. • AHT-EPP 5: The IoT Integration Platform will provide information on the health monitoring infrastructure status that could be used for the engineering process (e.g. bug fix automation) and added value services (e.g. preventive maintenance). • AHT-EPP 6: The persister thanks to Kura can perform basin NDT in order to automate the maintenance of the structure to be monitored (e.g. the bridge). The IoT Integration Platform will provide feedback on the health monitoring infrastructure status in phase 5. • AHT-EPP 7: The IoT Integration Platform will provide feedback on the health monitoring infrastructure status that could allow to identify new functionalities or new releases of the system.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Thanks to the AHF, we will be able to connect multiple stakeholders in a unique stream, connecting their toolchain in order to achieve automatic interactions. The baseline lacks severely of this automation as stakeholders have to iterate over manual interactions. The image shows each of the actors as a stakeholder of the system, each of them responsible for the development of one or more components of the system of systems. Each of the stakeholders follow the AHT-EP, some of them in full, some of them are active only during come of the phases. All the processes are interconnected

WP2 Objective	Focus & Planed actions
	with each other and most of these connections are supposed to be automatized during the project. **Planned actions** • AHT-EPP 2: All software tools mentioned so far will be designed to be compatible with the arrowhead framework in order for the interaction among stakeholders to be automatic. • AHT-EPP 4: Here the Gas Sensor and the PMUT will be easily deployed and advertise their presence thanks to the Arrowhead framework. The WAE will make this possible also for the inertial sensor network. The feedback provided by IoT Integration Platform can be useful for the operator in charge of installing the system. • AHT-EPP 5: The persister and the local cloud will be finally aware of all the edge devices coming from different stakeholders. The IoT integration platform, coupled with the AF, will allow the management of multiple streams of data coming from multi-stakeholder data sources. • AHT-EPP 6: The persister and the local cloud will be finally aware of all the edge devices coming from different stakeholders. The feedback provided by IoT Integration Platform can be useful for the maintenance operator.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	This is a direct consequence of the improved interaction between the stakeholders both in their engineering process and during the steady state of the use case. Functionalities, interactions and data streams are therefore increased in number. In particular, we can observe how many of the EPPs have multiple connection within and outside their own AHT-EP. Many of them are soon to be automatic, such as c 9-16, and many others are manual, such as the ones originated or directed to EPPs 7 and 8. **Planned actions** • AHT-EPP 4: The runtime configuration of the inertial sensor network was not an option in the baseline and constitutes a novel I/O interaction for the use case. The Integration platform will be able to manage an entire fleet of monitoring systems, dealing with a large amount of data streams. • AHT-EPP 5-6: The persister can access a great number of services, provided by the edge devices, which were not able to provide data as outputs in the baseline. This significantly improves the number of interactions and, consequently, of I/O ports. The Integration platform will be able to manage an entire fleet of monitoring systems, dealing with a large amount of data streams.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The software part of the Local CLoud Gateway is partially already automated: the source code documentation is generated automatically. SDK and Demo Kit of the IoT Integration Platform are not automatically generated, but the release of the platform in the kit follows the main development stream, therefore is always up to date. In general, many of the EEP3 phases output a user manual for their artifacts. Moreover, the Local Cloud Gateway is expected to produce training material in the form of tutorials to the final user who should be able to navigate the interface. **Planned actions** • AHT-EPP 1-2-3-8: the interaction with the AHF will be taught in the design phase of all the artefacts, making the development of them more agile and error-prone.

Table 41 UC-08.3 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Condition monitoring is the focus of a PhD program at UNIBO and a topic of several research contracts between UNIBO and primary companies and research institutions. As such, the distribution of engineering costs along the 8 stages of AHT-EP have their peaks on functional design and engineering, and could be thought to be partitioned as follows: requirements 10%, functional design 20%, engineering 30%, deployment 10%, operation 5%, maintenance 5%, evolution 10%, and training 10%. Operation and maintenance have low impact because the focus of the activity addresses pre-deployment innovation in condition monitoring. The proposed tools are not isolated, on the contrary they are conceived as a tool-chain that, thanks to their integration, should: • facilitate and smooth the first four stages, as well as training, with an expected reduction of their cost by 50%. • support phases 5 and 6, making them sustainable in real applications provide new benefits to the end user in block 7. In practice, the Energy Harvesting tool and the Optimizer tool will be the main responsible for the reduction of costs (because of the reduction in energy consumption) in the operational phases. Moreover, energy harvesting allows significant changes in pre-deployment phases because the network is going to be an autonomous system in which costs to have the mains electricity on site will be eliminated. For StkH3, the major cost reduction will come from the automatic integration of several tools that will result in service providers and consumers for the AHF. Automation improves the quality of the final product which becomes less prone to human errors. The business logic will be created by using a data flow management IDE that with its high level of abstraction and modularity, leads to additional savings in terms of man-months by speeding-up the prototype development and by supporting an easy scale-up of the fleet size. Automated code documentation updates can save developers time by making the documentation editing process faster, coherent and more systematic. Continuous monitoring of the remote health monitoring system allows a precise plan of the maintenance activities, also reducing the final indirect costs. Eventually, a significant cost reduction will be achieved by the fleet management capabilities introduced by Eclipse Kapua: the integration of Kapua with the AF will allow to monitor the remote local clouds from a single console with a single operator.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Interoperability is the main enabler of the cost reduction in EP and of innovative contributions to maintenance and evolution. This objective will be reached through the possibility by the sensor network of accepting sensor configuration designed by a third-party entity, in our case given by the Optimizer, which is a standalone tool that will be modified for accomplishing this task and needs parameters from the SN itself as well as other entities in the UC. The integration of Eclipse Kura and Kapua with AF will significantly improve the interoperability level of the IoT integration platform and of the services it offers: the IoT integration platform will enable data collection, local data processing and remote monitoring of the sensing infrastructure, and these functionalities will be exported on the AF in an inherently interoperable way.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework	Significant added values from the Target SoS are expected to be enabled by the integration of data originating from heterogeneous platforms through the AF (e.g. support ML from multidimensional data originating from: sensor networks for structural monitoring, environmental monitoring platforms, meteo stations, and possibly other platforms). Specifically, in the scope of this objective, we integrate with the Arrowhead Framework the Gas Sensor and all the SHM sensors through the WAE. All these are being used by the Optimizer, which will be integrated with the

Project Objective	Focus & Planed actions
integration platform	AHF as well, and fed as aggregate information to the ultimate consumer, the Persister. The AF will allow to improve the interoperability of the tools that SthK3 is using for the automatic generation of the documentation.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Preparing efficient modelling and simulations tools for the application of PMUT arrays in Arrowhead Tools applications, applied to SHM of structural components through bolted joints stress level monitoring. Preparing tools to support hardware and software configuration of structural health monitoring sensor networks, in our case made possible by the Configurator. The IoT integration capabilities of the Local Cloud Gateway are based on Eclipse Kura and Kapua, a popular solution for IoT and edge computing. This popular solution will be integrated with the Arrowhead Framework. The advantages of this integration will emerge both at Local Cloud and Global Cloud levels, in terms of data collection, processing and communication capabilities on the edge, and in terms of fleet management and cross-domain services availability at the Global Cloud level.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Probably not applicable.
Obj. 6 - Training material (HW and SW) for professional engineers	At the use case level, the contribution will be mainly derived from the proposed tool chain. For StkH3, the following training material will be provided: • demo showing the functionalities of the infrastructure health monitoring • code documentation as part of the Eclipse Kura and Kapua projects • user manuals for the developer • documentation of the adapters required to integrate legacy tools with the AF

8.3.4 Engineering Process analysis

Table 42 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-08.3.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1		X	X		X	X	X	X	
OBJ-AHT #2			X	X	X				
OBJ-AHT #3			X	X	X	X		X	
OBJ-AHT #4		X	X		X				
OBJ-AHT #5									
OBJ-AHT #6		X	X		X			X	

OBJ-WP2 #1	X	X	X		X	X	X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3	X			X	X	X		X	
OBJ-WP2 #4	X	X	X					X	

8.4 UC-08.4 SoS engineering of IoT edge devices (Smart Home) (ST-I)

UC-8.4 is one of the demonstrators of UC_8, and it is intended to demonstrate the impact of an AHT-EP supporting the integrated and automated design, deployment and configuration of smart home devices oriented to energy monitoring. Data collection from these smart meters is facilitated by a gateway which acts as a bridge between the field devices and the cloud. The data collected and processed on edge is stored on the cloud, where it can be further processed and can be made available to third parties (including the end users) through Arrowhead Services and applications. This monitoring solution allows to improve many added-value services and enables cross-domain applications, for example by making maintenance more efficient through the prompt delivery of malfunction warnings, by making the customer aware of consumption data, by improving the quality of utility management, etc.

The use case focuses on a Smart Energy System for domestic application. The system is an end-to-end solution for the domestic market, intended to extend a standard energy meters for domestic usage with connectivity and smart functionalities, integrate it into an IoT infrastructure and provide added value services for the manufacturer, installer, user, etc.

The main objective of this use case is to provide two Arrowhead Services that completely hide the technical details, the complexity and the heterogeneity of the smart metering infrastructure, offering a unified interface for metering functionalities and load monitoring management, respectively.

The following stakeholders compose the value chain of this use case:

- StkH1: utility managing the energy distribution
- StkH2: company that designs and develops the smart meters HW (ST-I)
- StkH3: IoT integration platform and end user applications provider (ETH + POLITO + UNIBO + UNIPI)
- StkH4: company that designs and develops the software applications (POLITO + ST-I + UNIBO + UNIPI)
- StkH5: smart energy end user

The use case is composed of several sub-systems:

- An electric and gas smart meter system, which allows to securely collect the electricity and gas consumed by the home appliances, process the consumption data on the edge (in the house), securely dispatch it to the enterprise level (e.g. energy utility cloud platforms or datacenters) for further analysis and extraction of insightful high level information, and publish them as a service in a controlled and secure way. These services are adopted by third parties to provide end user application, e.g. for the maintenance operator, for the householder, for different departments of the utility itself, for the authority managing the energy market, etc. Eventually, this subsystem allows the remote management of the smart meters monitoring infrastructure. The smart meters are designed and developed by StkH2, the IoT integration platform and the end user application are designed, developed and operated by StkH3, while the end user and StkH1 use these applications.

- The LF-NILM (NILM1) component implements an integration platform that consumes energy disaggregation data from an existing legacy infrastructure and supplies them to potential third-party consumers by using the Arrowhead Framework. The system is intended to simplify the dissemination of energy data among the different partners involved in the energy sector (end-users, grid operators, governments, etc.) by agreeing on a well-defined data format and by presenting information in a clear manner through the use of configurable web dashboards. The baseline of the LF-NILM is represented by the implementation of the legacy infrastructure that collects and stores the energy disaggregation data. The legacy infrastructure is provided by a third-party stakeholder operating in the energy sector, which already owns a fleet of sensors deployed across several households. The sensors operate in the low frequency range with a sampling frequency of 1 Hz and periodically send data to the cloud to be analysed. The collected data are pre-processed on the cloud and permanently stored in a database in both aggregated and disaggregated forms. The database represents the only interface between the legacy infrastructure and the LF-NILM integration platform, exposing all available data to the Arrowhead Framework through the database API.
- The HF-NILM (NILM2) component provides energy disaggregation data, with the information of the appliances detected, directly from the device. The stakeholder (potential third-party consumers) will use the Arrowhead Framework to collect information and configure triggers and alarms on events of interest. The baseline of the HF-NILM is represented by the implementation of a simple hardware meter capable of executing on-board classification algorithms that are developed and made available by the arrowhead framework. The sensors operate in the high-frequency range with a sampling frequency of 200KHz; on-board processing will send only detected appliances when they are switched on or -off to the cloud to be analyzed. The collected data are organized on the cloud and permanently stored in a database in disaggregated forms, and accessible through Arrowhead Framework.
- An Electricity Meter that communicate with the Arrowhead Framework using the Wireless Application Environment (WAE) technology for communication between wireless devices and Web servers.
- Cloud-Based Services where data are collected and sent in burst mode via Wi-Fi on the cloud (Microsoft Azure, Kura, WAE)
- Human-Computer Interface (HCI): A mobile application installed on users smartphones or a web services can visualise the results of the analysis.

The main advantages for the different stakeholder are the following ones:

- For the consumers it will be translated in a potential energy (and consequential costs) saving by leveraging a finer control in the consume thanks to a more frequent and reliable information that will be translated immediately into the meter reading / billing; the possibility to interface the meter with home automation / appliances can further improve the control of the energy consumption.

- The Energy Distributor will take advantage of a better efficiency by lower meter reading costs, and simplified processes for deactivation of the energy supply or customer/supplier switching (by remote procedures).
- The Energy provider will benefit of a major flexibility and a reduction of risks: the solution will provide multiple payment modes (prepaid and postpaid) and a prompt detection of tampering.
- The smart meter supplier will benefit from a software solution, where customization in different markets will require only software update with reduction of costs in the evolution of the product thanks to the automation provided by the tools of task partitioning and allocation.

8.4.1 Overall description of the UC-EP

A solution to provide metering and load monitoring services, starting from a multi-technology and multi-stakeholder use case, is currently not available. The solution we propose is based on the following functional components (see Figure 30):

- smart gas and electricity meters of different technologies;
- a multiservice gateway hosting an IoT framework;
- HF and LF NILM software;
- cloud platforms for system integration, data collection and big-data analytics;
- the Arrowhead Framework;
- End-user applications.

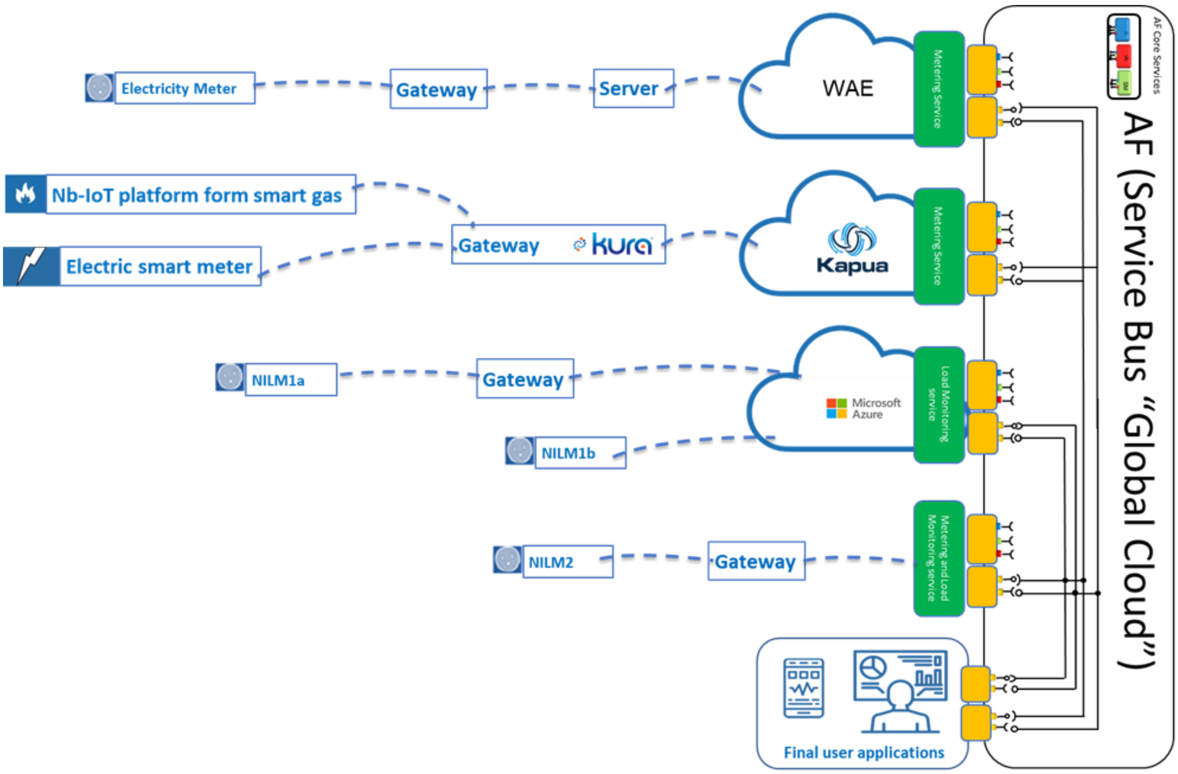


Figure 30 UC-08.4 functional architecture

The NB-IoT Gas/PLC Electricity smart meter solutions enable energy distributor to acquire automatically consumption and other information about the operating status of the meters. This can help to solve supply irregularities resulting from theft and leakage. In addition, interoperability and unified standards make maintenance and management easier. Reading the meters remotely instead of manually is more efficient, so it increases workforce productivity.
Customer management, field administration, engineering management, and scheduling analysis are improved, and customer service centre and call centre O&M is made easier.
The multiservice gateway is intended to integrate the metering components on the edge (in the house), providing an IoT framework (based on Eclipse Kura) that simplifies data collection, data processing on the edge, cloud connection and enables remote control of the metering infrastructure. The framework also provides a secure container where a set of service-based APIs simplify the business logic development. Eventually, the framework will include a data flow management IDE, intended to minimise the development effort of simple business logics and fully automate their execution.

In the baseline version of the LF-NILM (NILM1), there is not an integration platform for sharing the energy disaggregation data with potential additional consumers. Indeed, the customers of the start-up are the exclusive consumers of the legacy infrastructure, which can only access their own information by using a mobile application. In the baseline version there are no dashboards that can be configured based on the needs of the specific consumer. Furthermore, the integration of the available information with external sources of energy data would be costly, since they would probably adopt a very different data format for serving information.
On these premises, the LF-NILM integration aims at improving the legacy infrastructure by achieving the following objectives:

- establish a well-defined data format for transmitting energy disaggregation data from different sources directly to third-party consumers;
- develop an application system to access different data sources which can be easily extendable to include additional information;
- develop multiple services to supply energy information through the use of parameterized queries;
- implement configurable dashboards to access different data sources in a way totally transparent to the end user;
- Implement a security mechanism to grant access only to authorized consumers.

POLITO is in charge of designing, developing and operating the LF-NILM integration platform. The implementation of the legacy infrastructure provided by the start-up is a black box for the integration platform, except for the database which represents the only interface to interact with the legacy system.

UNIBO is in charge of designing and developing the HF-NILM as a processing algorithm in the IoT device and its integration in the Eclipse Arrowhead framework.

Different cloud platforms will be adopted for the management of the metering infrastructures adopted in the use case: Eclipse Kapua, Microsoft Azure, and WAE.

The cloud platform adopted for the full integration of first metering infrastructure, a fleet composed of all of the meters and gateways installed in houses, is based on Eclipse Kapua. Kapua is responsible for the creation of the system of systems and provides features for data collection, data processing, fleet management, user management, security, etc. Kapua is also responsible for the publication of the Metering Service on the Eclipse Arrowhead framework.

The Eclipse Arrowhead framework provides a “global cloud” where each sub-system of the proposed solution can publish its services and consume other services depending on its needs: the framework acts as a service bus improving the interoperability for information sharing, simplifying service usage, information exchange, service and users control and security. At this level, the framework also enables the creation of multi-stakeholder or/and cross-domain applications.

End user applications represent a good example of this key factor provided by the Eclipse Arrowhead framework: they are typically developed by third parties that don’t know the low-level, technological and architectural details of the specific vertical domain (in this case the energy domain). Examples of end user applications could be a console for the utility operator (StkH1), a mobile phone app for the maintenance operator (StkH1) or an external service company), a mobile phone app for the householder (StkH5), etc.

In Figure 31 we provide an overview of use case physical architecture. More details will be provided in WP7, WP8 and WP9 deliverables.

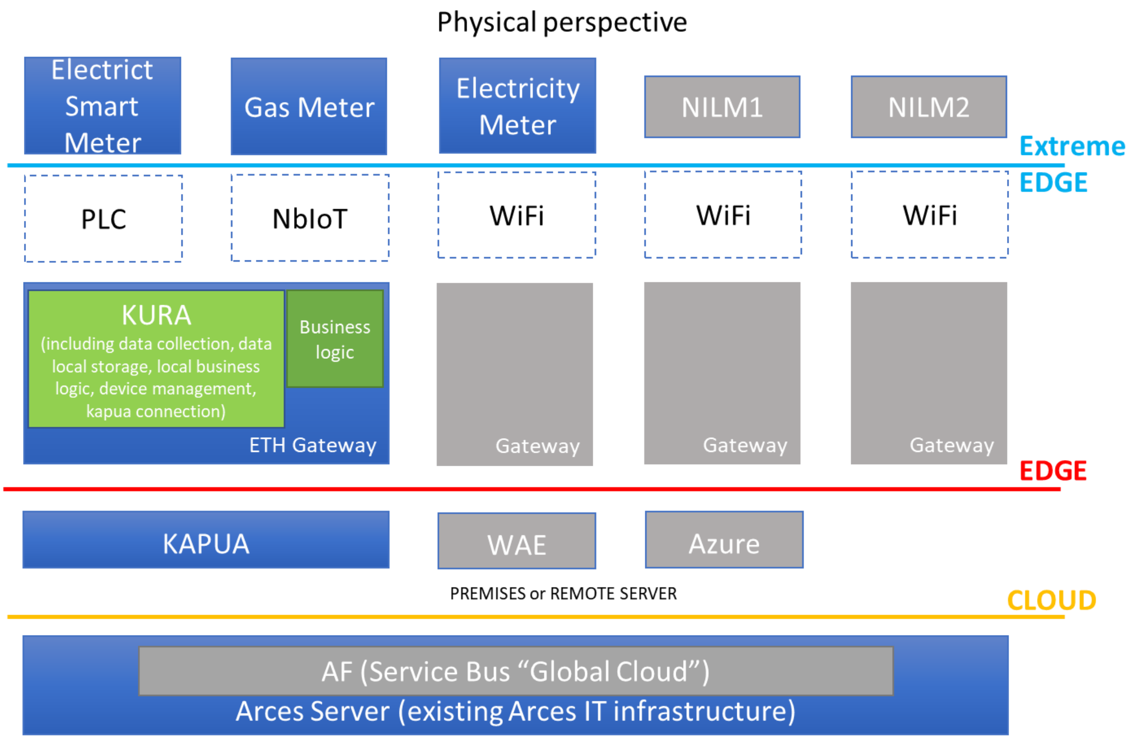


Figure 31 UC-08.4 physical architecture

8.4.2 Engineering Process Description

The following stakeholders compose the value chain of this use case:

- StkH1: utility managing the energy distribution
- StkH2: company that designs and develops the smart meters HW (ST-I)
- StkH3: IoT integration platform and end user applications provider (ETH + POLITO + UNIBO + UNIPI)
- StkH4: company that designs and develops the software applications (POLITO + ST-I + UNIBO + UNIPI)
- StkH5: smart energy end user

Partners involved in the use case are mapped in several tasks of the virtual stakeholders defined in a way to simulate a real interaction between four different companies placed in three different sectors which involve several internal departments in the development of the HW/SW sub-systems that will be used to compose the SoS product of the use case.

In Figure 32 the multi-stakeholder engineering process of the UC-08.4.

Actual solution for gas smart meter currently on the market can be divided as follows:

- Common mechanical meters. These meters are not intelligent. They require manual meter reading.
- IC card–based prepayment meters. These meters are controlled using IC cards on a per-user basis.
- Wired automatic meter reading system (AMR) meters. These meters require cable connections before being activated.
- Short-range wireless meters. Data is collected in a point-to-point or centralized manner. They can be controlled remotely.
- GPRS-based IoT meters. Scheduled data collection and remote control are made possible on GPRS cellular networks.

Gas meter reading across Europe is usually manual and carried out infrequently, from as seldom as once every two years (e.g. in Belgium) to once every two months in Spain. Some gas meters are equipped with a low frequency pulse output to enable remote reading. In the UK, prepayment meters are also commonly used by those on low incomes.
This use case takes as a baseline the actual situation in Italy, where electric smart meters are already in place but gas meters are based on a manual procedure. The baseline version of the gas meter was entirely designed and developed by the gas distributor. This version does not allow the product to be connected and controlled remotely. Once installed, the gas distributors receive manual feedback during the operation from the Installation technician.

The Engineering Process of the Gas Meter will define the interactions of StkH1 and StkH2. However, in the project the StkH2 (STMicroelectronics) is present: therefore the process will focus on the contribution of the Smart meter vendor (StkH2) that will design and deploy the smart meter. The EP of the process is aligned with the AHT-EP and uses 7 phases of the EP.
In the Requirements phase, technical requirements from StkH 1 are collected during the meetings of the focus group and saved in an excel spreadsheet.
During the Functional Design phase, the system model in MatLab will be designed for executable reference and then system design will take place by an EDA RTL cad suite for the hardware and a gcc toolchain based for the firmware and software development. This model will be tested and validated using RTL simulators and FPGA emulation.
In the Procurement & Engineering, the Engineering team of StkH 2 will work on the design and develop the smart meter prototype, which will be tested on a benchmark with synthetic signals

and then validated in operative conditions. In this phase, the StkH 2 integrates the support for the communication service provided by StkH 3 (the IoT Framework provider).
In the Deployment & Commissioning phase, the Embedded Control Unit (ECU) is deployed to the Engineering team of the StkH 1, together with the firmware level APIs. A final prototype of the smart meter is now available.
The prototype and all the production data is passed to the production department (represented with a black box in the StkH2 AHT-EP) through c41 and feedback for optimize the product with data acquired during the production are passed back through c38 to the engineering team. Produced product is ready for installation in customers' houses. The StkH 1 is responsible for installing the smart meter in customers' houses (StkH5).
In the Operation & Management phase StkH1 manages the operational phases, collecting periodically the gas consumption and monitors whether the gas grid is functioning properly.
In the Maintenance Decommissioning & Recycling phase, the operator StkH1 is responsible for repairing the smart meter and the gas grid, their decommissioning and for conferring the meter to the recycling center.
In the Evolution phase, the information collected during the lifecycle will be analyzed and StkH1 will identify possible modifications, updates, improvements, etc.
Eventually, the Training & Education phase, will be fundamental to provide technical manual, documentation of high level APIs for the App Developer and firmware level API to the smart meter vendor (to all the stakeholders involved).

Electric smart meters are already present in Italy (in 2018 Italy launched the second generation country wide rollout). The improvement respect to the baseline will be the provision of consumption data to consumers through a dedicated app. When consumers receive energy feedback information, they will gain insight in their energy consumption pattern and will be able to take measures to lower their consumption or to choose the better tariff according to their need.
Another aspect that will be improved for the electric smart meter component is the automation of the task partitioning and allocation of the software in the procurement and engineering phase that can allow StkH2 to save costs during the specialization of the electric meters in different countries (this typically implies different Power Line Communication protocols)
In the use case the electric smart meter is already present as a legacy component, therefore the Engineering phase that will be considered is the evolution where StkH1 or StkH2 will identify possible updates/improvements. Outcome of this phase is a change of the requirements or the addition of a new set of requirements (through c34) that will be reflected in the Procurement & Engineering (StkH2 AHT-EPP3) phases in a new software development, which will be tested on a benchmark with synthetic signals and then validated in operative conditions.

The engineering process of the StkH3 iot integration platform is in line with AHT-EP and uses several of its phases.
Requirements: the requirements elicitation and specifications definition is currently performed, adopting a traditional approach, by the engineering team without using any tool and related automation. Information about the system are flowing from StkH2 through c1 connector.
Functional design: the functional design of the IoT integration platform is currently performed by the engineering team without using any tool and related automation. The design of use case specific business logic, currently developed with standard coding methods, and its automatic execution will be simplified through a data flow management IDE, hosted on the multiservice gateway.
Procurement & Engineering: the IoT integration platform is currently designed and developed using a “standard” Java toolchain. The development tool-chain comprises Eclipse IDE for code development, Maven to simplify dependencies management and the building process, GitHub

for code management and versioning, Java as main programming language, Java Virtual Machine, etc.
Deployment and commissioning: the IoT integration platform is the main tool to manage deployment and commissioning. The multiservice gateway incorporates an initialization procedure which allows the auto-configuration of the infrastructure at the beginning of its life cycle. This procedure comprises security certificates exchange which allow for device authentication at iot integration platform level.
Operation and management: the IoT integration platform is the main tool to manage operations. The multiservice gateway is the enabling infrastructure in the operation and management phase because it allows remote administration and full control over the metering infrastructure. The remote management can be performed through the cloud platform Kapua while the addition of ad-hoc components is simplified by the development of osgi bundles in Kura.
Maintenance: the IoT integration platform is the main tool to manage maintenance. The remote identification of malfunctions is facilitated by remote control as described in “Operation and management”. Criticalities prediction is enabled by constant monitoring of the infrastructure status through the console offered by Kapua.
Evolution: the evolution phase is based on a ticketing tool to collect bugs, faults, and the evolution activities (bug correction, new releases, etc…) are managed manually by the R&D team. Information collect during the operation phase and elaborated in the evolution phase are triggering the requirement changes through the connector c3.
Training and education: StkH3 currently produces automatically the documentation for the developers by using JavaDoc and plans to improve upon, by implementing a feature that keeps track of new functionalities, added to the code, and provides a list of updates for the product documentation. The aim is to offer the developers a tool providing reminders and guidelines that ensure the inclusion of each new feature in future documentation (e.g. user manual). The tool is intended to make the whole process faster and more effective.

The engineering process of the LF-NILM team working in the StkH4 is made up by the following phases:
Requirements: The engineering team discusses with the owner of the legacy infrastructure developed by IoT integration platform and end user applications provider (StkH3) what kind of energy data can be exposed and how those data can be consumed by the system. Therefore, the requirements of the system are defined in accordance with the decisions of both parties (c6).
Functional Design: The system modules are designed in order to satisfy the requirements defined in the previous phase.
Procurement & Engineering: StkH4 receives from the owner of the legacy infrastructure the credentials to access the energy data stored in the database. StkH4 will develop the system modules (provider and consumer) and the web dashboards for data visualization to be integrated by the StkH3. StkH4 will implement a security mechanism to ensure that only authorized third-party consumers can access energy data.
Deployment: The provider module is deployed on a server, while the consumers are delivered from time to time to third parties.
Operation: The system is up and running. The provider (StkH1) disseminates the energy data to consumers and all the other stakeholders involved in the Operation & Management phase of the system.
Evolution: StkH4 may plan to extend the system with additional services derived from Python data analytics.
Training: StkH4 produces the training material for both end users and developers.

All the other services developed in the engineering process of StkH4 follows the same schema of activities and are intended to be developed by parallel engineering teams that work independently.

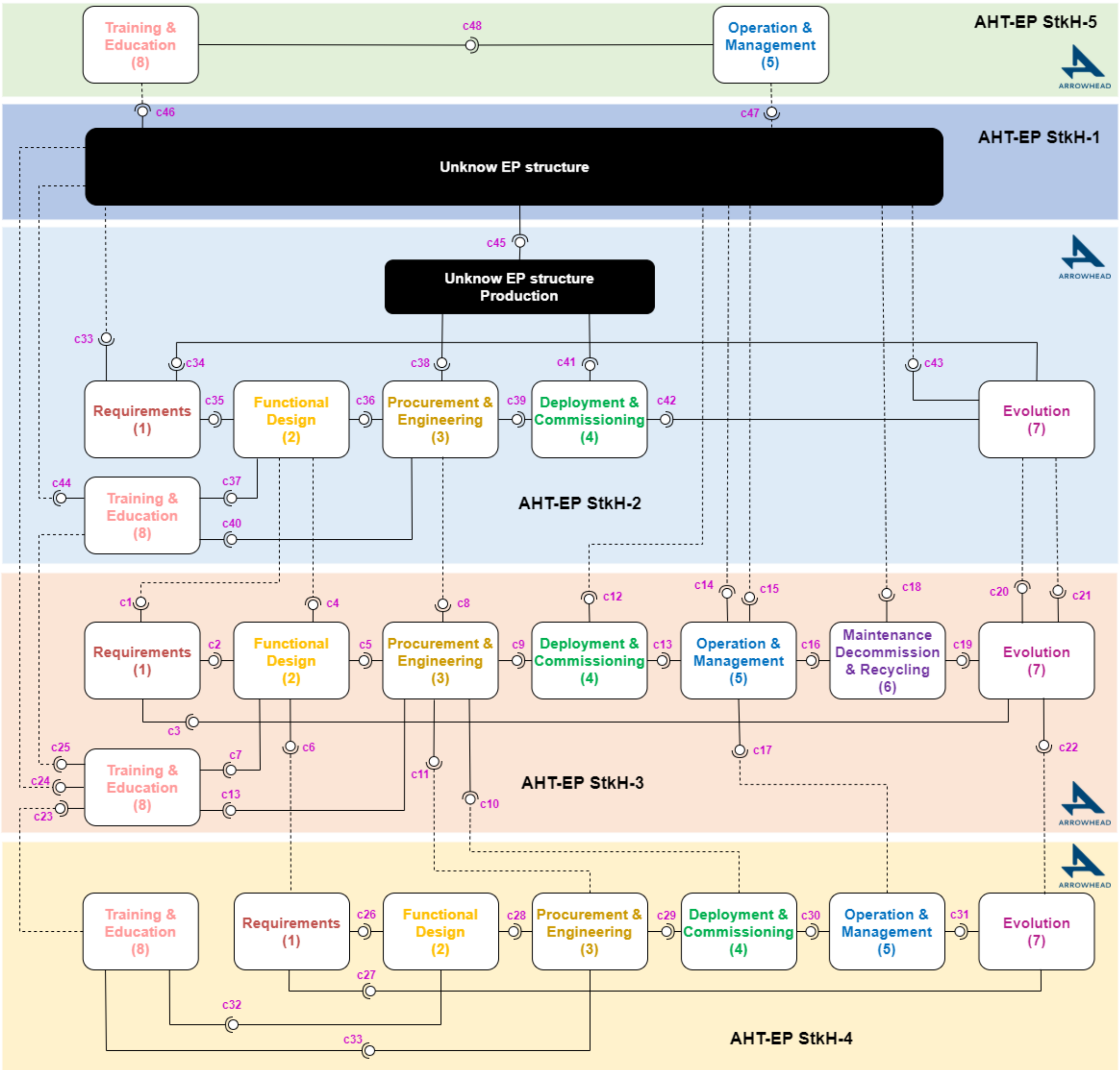


Figure 32 AHT-EP of use case UC-08.4

The LF-NILM component will be perfectly integrated with the Arrowhead Framework once implemented.
At M0, only two components of the ioT integration platform are integrated: Eclipse Kura and Kapua. The multiservice gateway, when available, will natively support the IoT integration platform. The tool adopted to manage the technical documentation tool is currently not integrated with the tool adopted for source code documentation generation.

The maintenance and training phases are not considered in the engineering process of the LF-NILM component, because they were not planned given the simplicity of the system.

Electric Smart meter: Requirements and functional design are not considered since it is a legacy component. Procurement and Engineering are considered only when evolution outcomes are present.
The EP of the IoT integration platform fully matches the AHT-EP.

The IoT integration platform based on Eclipse Kura and Kapua has a service oriented architecture that will simplify the integration with the AF. The tools adopted for the documentation management are not service oriented and will require the development of a specific tool (adapter) for their automation.

The AHT-EP can be applied separately to each component, as well as to the whole system. Furthermore, the AHT-EP can cover multiple stakeholders working independently.

The engineering process of the LF-NILM component follows a standard waterfall model.
For the IoT integration platform, the AGILE project management methodologies are adopted for software development: StkH3 (ETH) adopts the SCRUM methodology. The security aspects of the EP are addressed following the OWASP Secure Coding Practices, with the future adoption of tools that automatically check the code and evaluate the security levels.

Table 43 AHT-EP Phase focus of UC-08.4

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Smart Gas Meter*: StkH 2 acquire the specification from the vendor (StkH1). Mechanical parts, sensors and actuators are described with reference to the datasheets. Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the design of the smart meter. Requirements will be listed in a Word file. 2. *Smart Electric meter*: StkH 2 update the specification according to the evolution outcome. 3. *IoT integration platform*: Elicitation of the requirements; definition of the specifications. 4. *LF-NILM*: StkH4 discusses with the owner of the legacy infrastructure what kind of energy data can be exposed and how those data can be consumed by the system. Therefore, the requirements of the system are defined in accordance with the decisions of both parties. All system requirements will be clearly recorder by using MS Word + Excel + PowerPoint. Most of documentation will be saved in a PDF format and eventually consulted by using a PDF reader such as Adobe Acrobat Reader. **Tools** 1. *Smart Gas Meter*: Elicitate the requirements for the gas smart meter. • Input: <StkH1 - EP> specifications of the electromechanical part defined by the functional design team of the gas vendor (StkH 1). • Input: <StkH2 - EPP7> requirements are integrated with analysis performed in the evolution phase using the remote data collected from operating smart meters. • Output: <StkH2 - EPP2> requirements for the functional design of the smart meter. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Requirements elicitation for the LF-NILM integration. • Input: <StkH3 – EPP2> complete description of the legacy infrastructure with the list of available energy data • Input: <StkH4 - EPP7> requirements of planned evolution and functional modules that require an update on requirements.

Engineering process phase	Addressed/Focus
	• Output: <StkH4 - EPP2> requirements for the functional design of the LF-NILM integration system
Functional design	**Activity** 1. *Smart Gas Meter*: StkH2 design and simulate the system model of the meter/transmission algorithm of the Smart meter in MatLab. Design of the chip and the board are following traditional EDA design kit. 2. *Smart Electric meter*: StkH2 update the design and simulate the system model of the meter/trasmission algorithm of the Smart meter in MatLab. 3. *IoT integration platform*: The functional design is performed according to the guidelines given in the "Requirements" phase. 4. *LF-NILM*: StkH4 designs the system modules in accordance with the requirements defined in the previous phase. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Design the modules of the LF-NILM integration system. • Input: <StkH4 - EEP1> requirements • Output: <StkH4 - EEP3> design of the different functional modules composing the system with their corresponding responsibilities.
Procurement & Engineering	**Activity** 1. *Smart Gas Meter*: StkH2 finalise the design and produce the Smart meter supplied to the vendor product lines. 2. *Smart Electric meter*: The update firmware from evolution is developed. 3. *IoT integration platform*: • Supply chain planning based on functional design aiming at an efficient and cost effective procurement (acquire the software and hardware goods needed from suppliers to build the prototype) • Development of the IoT integration platform by using a well-defined toolchain; • Design and development of the use case specific business logic; • test of the IoT integration platform 4. *LF-NILM*: StkH4 receives from the owner of the legacy infrastructure the credentials to access the energy data stored in the database. StkH4 develops the system modules (provider and consumer) and the web dashboards for data visualization. StkH4 develops a security mechanism to ensure that only authorized third-party consumers can access energy data. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Implementation and development of the LF-NILM integration system. • Input: <StkH4 - EPP2> functional design of modules composing the system • Input: <StkH4 - EPP3> Java application to be tested before deployment • Output: <StkH4 - EPP4> Java application to be deployed • Output: <StkH4 - EPP3> Java application to be further tested before deployment

Engineering process phase	Addressed/Focus
Deployment & Commissioning	**Activity** 1. *Smart Gas Meter*: deploy the system to the internal production line. 2. *Smart Electric meter*: The update firmware from evolution is deployed to the different installations of the smart meters (if required) or deployed in a new generation of smart meter. 3. *IoT integration platform*: Deployment of the prototype and test in a real environment. At the end of the test, setup and deployment of the components of the use case, StkH3 is involved in the commissioning of the complete solution, focalising on the IoT integration platform. 4. *LF-NILM*: StkH4 mounts the provider module on a Docker container running on a server. The consumer modules are delivered from time to time to third parties interested in using the energy data supplied by the legacy infrastructure. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Deployment of the LF-NILM integration system. • Input: <StkH4 - EPP3> Java application to be deployed. • Output: <StkH4 - EPP5> provider system up and running + multiple deployed consumer systems.
Operations & Management	**Activity** 1. *Smart Gas Meter*: n.a. 2. *Smart Electric meter*: n.a. 3. *IoT integration platform*: The multiservice gateway allows data collection, edge processing and remote management of the metering infrastructure. 4. *LF-NILM*: The system is up and running: the provider disseminates the energy data, while the consumers use them. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Operation of the LF-NILM integration system. • Input: <StkH4 - EPP4> deployed Java application • Output: <StkH4 - EPP7> deployed Java application
Maintenance Decommissioning & Recycling	**Activity** 1. *Smart Gas Meter*: n.a. 2. *Smart Electric meter*: n.a. 3. *IoT integration platform*: Remote identification of malfunctions is facilitated by remote management. Criticalities prediction is enabled by constant monitoring of the infrastructure status. 4. *LF-NILM*: n.a.
Evolution	**Activity** 1. *Smart Gas Meter*: Data collected in the Operation & Management phase of StkH1 are analysed by StkH2 for identifying possible bottlenecks in the smart meter design. 2. *Smart Electric meter*: Data collected in the Operation & Management phase of StkH1 are analysed by StkH2 for identifying possible bottlenecks in the smart meter design. Moreover, firmware updates will be generated by StkH2 for update/extend communication standards

Engineering process phase	Addressed/Focus
	3. *IoT integration platform*: Faults and bug analysis to identify solutions, improvements and new product releases. 4. *LF-NILM*: StkH4 may plan to extend the system with additional services from Python data analytics. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: Firmware updates will be generated by StkH2 for functional improvements or update the communication standard collected information will be used for the next release of the software (hardware require a complete re design). • Input: <StkH2 - EPP6> Technical materials and serial number lists and updated status of the maintenance events. • Input: <StkH1 - EPP7> Information about the evolution of the system are evaluated for identifyng possible bottlenecks in the design. Log files collected in Operation & Management phase of StkH 1 are propagated. • Output: <StkH2 - EPP1> Analysis performed in this phase will be used to update the requirements and trigger the continuous engineering loop." 3. *IoT integration platform*: n.s. 4. *LF-NILM*: Evolution of the LF-NILM integration system with additional services. • Input: <StkH4 - EPP5> deployed Java application. • Output: <StkH4 - EPP1> requirements of planned evolution.
Training & Education	**Activity** 1. *Smart Gas Meter*: Production of user manuals. 2. *Smart Electric meter*: n.a. 3. *IoT integration platform*: Generation of source code documentation. Editing of technical and user manuals. 4. *LF-NILM*: StkH4 produces the training material for both end users and developers. **Tools** 1. *Smart Gas Meter*: n.s. 2. *Smart Electric meter*: n.s. 3. *IoT integration platform*: Integration of training material. 4. *LF-NILM*: Production of training material for end users and developers • Input: <StkH4 - EPP2> functional design of system • Input: <StkH4 - EPP3> source code documentation • Output: <StkH3 – EPP8> API documentation + source code documentation + deployment instructions.

8.4.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 44 UC-08.4 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to	IoT integration platform: The IoT integration platform, partially running on the Local Cloud and on the remote cloud, will introduce new run-time remote monitoring functionalities that will allow to profile the behaviour of the smart metering system during phase 4, 5 and 6. These functionalities could be exploited for predictive and

WP2 Objective	Focus & Planed actions
run time engineering	preventive maintenance, but in particular to automate the detection of faults, bugs, unplanned behaviours and trigger the design and development process to correct them. LF-NILM: during run time engineering the system may be extended to provide additional services from Python data analytics. **Planned actions** • AHT-EPP 4: The IoT Integration Platform will provide feedback on the smart metering infrastructure status already starting from phase 4. • AHT-EPP 5: The IoT integration platform will provide information on the smart metering infrastructure that could be used for the engineering process (e.g. bug fix automation) and added value services (e.g. preventive maintenance). • AHT-EPP 6: The IoT integration platform will provide feedback on the smart metering infrastructure status in phase 5. • AHT-EPP 7: The IoT integration platform will provide feedback on the smart meters status that could allow to identify new functionalities or new releases of the system.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The IoT integration platform, for its inherent nature, is intended to integrate multiple subsystems, designed, developed and operated by different stakeholders, in a system of systems. LF-NILM: The system will integrate different NILM technologies from multiple stakeholders, including third-party legacy infrastructures. **Planned actions** • AHT-EPP 4: The feedback provided by IoT Integration Platform can be useful for the operator in charge of installing the system. • AHT-EPP 5: The IoT integration platform, coupled with the AF, will allow the management of multiple streams of data coming from multi-stakeholder data sources. • AHT-EPP 6: The feedback provided by IoT integration platform can be useful for the maintenance operator.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	This is a direct consequence of the improved interaction between the stakeholders both in their engineering process and during the steady state of the use case. Functionalities, interactions and data streams are therefore increased in number. **Planned actions** • AHT-EPP 4: The Integration platform will be able to manage an entire fleet of monitoring systems, dealing with a large amount of data streams. • AHT-EPP 5: The Integration platform will be able to manage an entire fleet of monitoring systems, dealing with a large amount of data streams. • AHT-EPP 6: The Integration platform will be able to manage an entire fleet of smart metering devices, dealing with a large amount of data streams.
Obj. 4 - Address digital learning and training activities as an integral part of	SthK3 will improve the automation level of the tools adopted for the editing of the documentation of the IoT integration platform and, partially, also of the use case (including: source code documentation, technical documentation, user manuals, application manuals, etc.). LF-NILM: The system will be provided with the training material, in order to support both end users and developers in the usage of the system.

WP2 Objective	Focus & Planed actions
the engineering cycle	

Table 45 UC-08.4 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Functional design: the inclusion of task partitioning and allocation tools inside the STMicroelectronics toolchains can lead to a reduction of 20% in functional design and Procurement and engineering phases of metering application by PMs and licenses usage reduction. A typical example of evolution is the support of a new PLC (Power Line Communication) smart grid standards for new region market. Operation and management: Traditional manual meter reading leads to high labor and management costs. Given that an employee can read 80 to 100 in-house gas meters per day, a small or mid-sized natural gas utility serving 3 million residential houses must hire 150 to 200 employees to cope only with meter reading. With the introduction of gas smart metering the costs of manual reading (charged both on gas utilities and users) will be eliminated. Other added value relies on the more information about the operating status of all meters. This can solve supply irregularities resulting from theft and leakage, with a second added benefits. Taking in account that energy infrastructure have a life cycle in operation EP that can last for some decades, any reduction in this EP will have a big impact on the overall cost of the system. The LF-NILM component reduces the engineering costs by providing a general data-sharing platform that can be easily adapted to manage energy data from multiple legacy infrastructures. Indeed, the consumer application does not require any variation for switching between different data sources, thanks to the well-defined data format adopted by the system. The provider application needs only small updates in order to introduce a new data source, which consist of implementing the additional module for accessing the database of the new legacy infrastructure. However, the implementation of the new module counts for less than 20% of the whole application system, while the remaining parts of the system (provider, consumer and dashboards) constitute the majority of the engineering costs and are already provided by the LF-NILM component.
Obj. 2 - Interoperability for IoT and SoS engineering tools	The final system will provide interoperability between different smart sensors technologies.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The LF-NILM component allows the interoperability between different third-party legacy infrastructures in a way totally transparent to the end user. At operation phase the interoperability to legacy energy smart grid based on platforms that support e-smart meter PLC communication standards/gas smart meter standard Nb IoT.
Obj. 4 - Integration platform interoperability with emerging digitalization and	The IoT integration framework is based on Eclipse Kura and Kapua, a popular solution for IoT and edge computing. The framework will be integrated with the Arrowhead Framework.

Project Objective	Focus & Planed actions
automation framework	
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The gas smart meter solution will provide an E2E (End to End) security solution. For the LF-NILM, The Arrowhead Framework provides a security mechanism for ensuring that only authorized third-party consumers can access the energy data. Furthermore, the Arrowhead Framework provides the encryption of communications between consumers and providers.
Obj. 6 - Training material (HW and SW) for professional engineers	The full documentation of the source code will be available for supporting the developers who are in charge of integrating new legacy infrastructures. The end users will be provided with a short description for deploying the consumer application in their working stations, in addition to the API documentation of the provider application.

8.4.4 Engineering Process analysis

Table 46 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-08.4

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	C
OBJ-AHT #1	X	X	X	X	X		X	X	
OBJ-AHT #2		X	X	X	X				
OBJ-AHT #3			X	X	X				
OBJ-AHT #4	X		X		X				
OBJ-AHT #5					X	X			
OBJ-AHT #6		X	X	X	X		X	X	
OBJ-WP2 #1	X	X	X	X	X	X	X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3	X	X	X	X	X	X		X	
OBJ-WP2 #4		X	X	X	X		X	X	

8.5 UC-08.5 SoS engineering of IoT edge devices (Industrial Energy Monitoring) (ST-I)

In this project, Condition Monitoring applications in an end to end framework will be exploited by implementing a use cases strictly correlated with smart sensor vibration and ultrasound nodes.

A new cloud-based tool will be developed and adopted for Industry 4.0 combining the information coming from the physical entities object of monitor and surveillance on their operative status.

Condition monitoring and anomaly detection is the process to monitor relevant physical characterizes such as vibration, noise, temperature of an equipment and part of this to recognize the trend versus an anomalous behaviour. Advanced monitoring is based on the appropriate set of sensors in which sensor fusion capability reinforce the status detection and providing the required data collection enable their analysis through the different vertical layer providing advanced services on base of this information and elaboration chain.
Industrial scenario consists of:

- Smart sensor nodes deployed at the critical machines
- Collection of sensors data and pre-processing with / without machine learning at the edge (either the node or a gateway / industrial PC)
- High level analytics on company premises or on cloud for provisioning the nodes with updated algorithms

Today the applications enabled by this approach based on smart sensor vibration and ultrasound nodes are growing providing an advanced monitoring on production line and to ensure a condition maintenance for critical machineries avoiding downtime; the exploitation of new machine learning techniques are under development to facilitate and accelerate introduction in the field of advanced features as the predictive maintenance that allows the optimization of maintenance plan for the production line.
Predictive maintenance and workload optimization depend on the sourcing information capability that is implemented in the equipment and in the environments. The quality of this information is related to the capability to extract from each sensor and from the convergence and fusion among measurements from different sensors the adequate signal characteristics to synthesize the operative status of the monitored object.
Therefore, below the targets:

1. Define the appropriate set of sensors and measurements that allow the predictive maintenance and the optimization of the production paths on base of the workload recognition and anomaly detection.
2. Define an adequate chain of communication able to ensure the correct data exchange from the physical entity up to the FA, of preference, at physical level, providing several kind of communications to provide solution with large flexibility and compatibility on adoption for different scenarios and situation. Wired and the main wireless communication should be supported to show data and information to the upper levels.
3. Allow their easy and no invasive large application with constrains on size and on connection, making the updating to predictive analysis and workload recognition an additional operative value to exploit for the production.

8.5.1 Overall description of the UC-EP

The scope of the Use Case 08.5 is to develop a series of HW/SW tools that, organized in toolchains, are aimed at developing of full demonstrator for a condition based monitoring and predictive maintenance system. Starting from the Silicon manufacturing capability in provide advanced devices for communications, elaboration and MEMS sensor, the aims is to provide solution able to increase the value of local data processing, provide a modular connectivity covering the most common use so to allow and include different (sensor) node in the architecture, developing libraries that can re-use to accelerate the integration of our HW on sensor node with industrial gateway and at same time, achieve the required interoperability to connect and enable AF services and interactions. The focus will be in provide reference designs and design tool kits able to address our devices in the large adoption in the application field, enabling application (use case) tools and toolchains by means of these ”building blocks”. As illustrated in Figure 33, the Device to Cloud approach is adopted to publish and consume a set of Arrowhead services and to allow, for generality of the use case, the interfacing to third parties' producer of sensors platforms that could allow additional, but not in the scope of this use case, services and smart functionalities as possible tools of Arrowhead framework, such as localization, environmental awareness, optimization and ML tools.

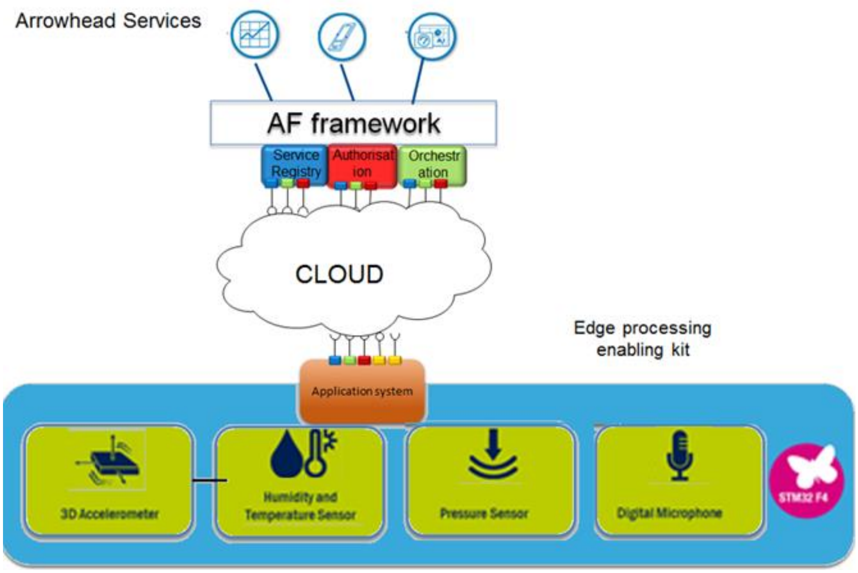


Figure 33 The Arrowhead Service in UC-08.5

In brief, the Use Case 08.5 describes a general architecture of IoT sensor networks devoted to multiple applications customizable through an SAAS platform as for Arrowhead 2.0 ambition. For this aim different industrial reference design kits for condition monitoring (CM) and predictive maintenance (PdM) have been developed to finalize the investigation of distinctive sensors parameter for sensors for predictive maintenance services, and to be complemented with inertial and environmental sensing to match better the operating condition of monitored equipment.
The introduction of sensor tool of ultrasound and vibrational sensors has been investigated to improve the early detection of faults, as described in the follow figure in which possible monitors for an electric motor as industrial actuator are related to the related time of detection before the fault.

8.5.2 Engineering Process Description

The main addressed phase of this use case is the Functional Design, whose results must ensure and enable functionalities and improved performance across each Eps. This is obtained through continuous refinement interactions on achieved and allowable HW/SW functionalities.
The use case is focused on HW manufacturer's EP (StkH1 in Figure 34) in providing HW, SW and FW tools to be organized in/to enable toolchains on condition based monitoring and predictive maintenance system referring industrial scenario.
The main scope is to define and enable this kind of solutions, in line with STMicroelectronics business core, as references for multiple implementations and innovative approaches. HW manufacturer is the only actor of the use case, although, System Integrator's activity it is

required to move, from prototypal shape into final and customized solution to be proposed to final user, the complete and organized set of "Building Blocks" and the enabling solutions provided, in the UC, by the HW manufacturer.
In the below description the EPs related to:

- Software Integrator (StkH2), in providing the required software tools;
- System Integrator (StkH3), to achieve the final implementation;
- Final User (STkH4) to drive the application scenario.

Were also considered with the limited purpose of providing a comprehensive overview of the value chain through stakeholders and highlighting interactions with the hardware manufacturer's EP, focus of this use case.
Two unknown EP structures, “IC Manufacturer” and “Application Context Scenario”, were considered respectively on top, as IC and components provider, and on bottom, as Industrial application scenario, for referred use case example.
Possible innovation requests, output of any “Evolution” phases, is promoted up to “IC Manufacturer” by means of the “Requirements phase” of the first stakeholders.
Requirements: Definition of requirements at sensor nodes: vibration bandwidth, processing capabilities, data latency collected by end-users and application scenarios. Input from software integrators, system integrator and final user contribute to define the requirements and address new product design following application needs and improvements (HW manufacturer). General requirements analysis referred HW manufacturer layer (functional design and procurement & engineering) and the addressed application scenario (Software Integrator).
Requirements analysis and elicitation. Specifications definition from the addressed application scenario (System Integrator). Use case definition and objectives driving overall specifications: vibration monitoring on wireless sensor network with multiple connectivity (End User).
Functional Design: It includes the selection of appropriate components (communication, sensing, computational) and the verification of feasible features. The interaction with software integrator/provider allows to introduce new requirement on software development and to receive the constraints to be complied. Design of the smart sensor node, this phase pass through a feasibility analysis -- FW/ SW Drivers (HW Manufacturer). Preliminary study and identification of FW/SW tools for the use of the sensors, their interconnection and next integration in addressed application scenario (SW Integrator). Functional architecture definition on base of deployed HW instruments and SW tools (System integrator).
Procurement & engineering: Procurement of PCB manufacturing and FW Driver and deployed software tools. Hardware components for the sensor node boards and their validation and functionality test on the HW & SW and FW drivers (HW Manufacturer). Functional architecture definition on base of deployed HW instruments and SW tools (SW Integrator). During this phase, the System Integrator is involved in two processes developed in parallel:

- design, development, test and debug of the Local Cloud Gateway hardware and software; design, development, test and debug of the use case specific business logic;
- interaction with suppliers to acquire electronic and mechanical components, define external services to manufacture hardware prototypes; interaction with suppliers to acquire source code, software libraries and software licenses potentially required for the software part of the gateway.

Deployment and Commissioning: Prototypal smart node is released to be connected by a specific protocol and specific connectors. Proof validation smart sensor nodes, system enabling tools made available in the chain of stockholders. Reference design and building blocks to be organized in a reference application tool and toolchain (HW Manufacturer). SW deployment (Software Integrator). Deployment of the prototypes to test and debug them in a real environment to evaluate their maturity. Commissioning of the golden sample to start the production of the product. (System Integrator). Prediction over Condition monitoring organization and Validation (End User).

Operation & Management: System Integrator provides measurements on specific physical aspect that characterize the machine operatively enabling the detection of anomalous state on operative machineries:

- Integration of use case components on the field;
- Data collection, storage, local processing;
- implement remote monitoring and control;
- Interfacing with cloud and enterprise level.

Prediction over Condition monitoring execution (End User).
The HW manufacturer/technology enabler provide, in this phase, proof of Value on certain numbers of node / system applications, test and application reference for the deployed solutions.
Maintenance Decommissioning & Recycling: The System Integrator provides remote monitoring and control for maintenance and de-commissioning purposes. Operational and Proof of Value on certain numbers of node / facilities (End User). The HW manufacturer evaluates the family product promotion based on market and on value chain feedback and elicitation.
Evolution: Cost evaluation and feedback from field back in requirements as input for new user's objectives/requirements to be proposed to the system integrator (as requirements), and to the HW manufacturer for innovation of products (End User). Faults and bugs analysis to identify solutions, improvements, and new product releases. (System Integrator). The HW manufacturer evaluates possible improvements on base of the operational verification done in the previous steps of EP.
Training & Education: Training people for operative conduction and maintenance (Final User). Generation of design and source code documentation. Editing of technical and user manuals (System Integrator). User manual, training tool, how to do, and library to support advanced functionality referring also application of deployed HW tools (Software integrator).
Application note, training tool on devices and how to do and how exploit advanced functionalities (HW manufacturer).

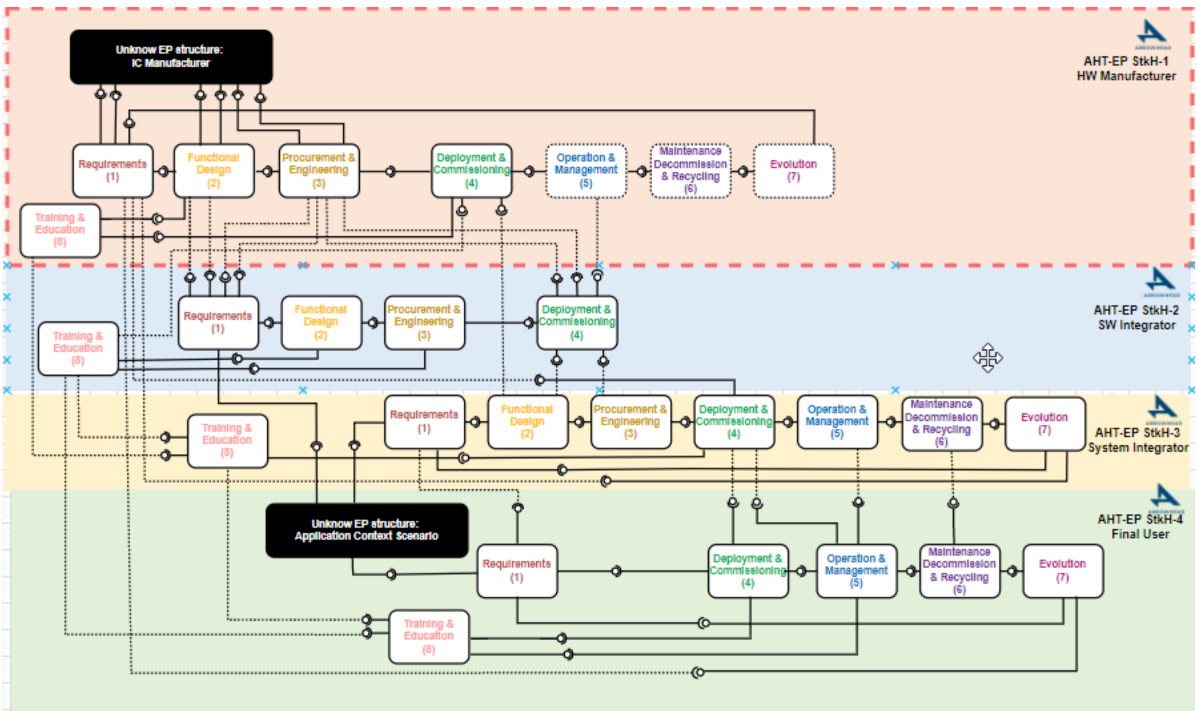


Figure 34 AHT-EP of use case UC-08.5

Many large enterprise already has frameworks for the management of the EP of IoT based applications (e.g. Tesla, BMW, Siemens, IBM, etc.), but these frameworks are proprietary and closed solutions, that have been conceived for specific vertical applications.
In the use case 8.5 we foresee to keep life cycle by implementing FOTA Services for the smart sensor nodes and the gateway in the end to end framework based on AWS/AZURE services. In relation to the evolution of AWS and AZURE, this is out of scope as the companies handle their life cycle; by the way to update new services or new evolution, a manual effort is needed for this use case 08.5.

The expected compatibility with the AHF happens in the integration of services offered by the AWS Cloud Application and the Business Cases offered by AH. This is based on data model and API to transfer data to AHF.

Use Case 8.5 is developed to help partners and medium / small enterprise in the implementations of products and services for condition-based monitoring. So far it is based on purpose on the main components that as STMicroelectronics we may offer. In this regard we can cover Requirements – Functional Designs – Training.

Table 47 AHT-EP Phase focus of UC-08.5

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Hardware Manufacturer*: Definition of requirements at sensor nodes: vibration bandwidth, processing capabilities, data latency collected by end-users and application scenario. 2. *SW integrator*: General requirements analysis referred HW manufacturer layer (functional design and procurement & engineering) and the addressed application scenario. 3. *System Integrator*: Requirements analysis and elicitation. Specifications definition from the addressed application scenario. 4. *Final User*: Use case definition and objectives driving overall specifications: vibration monitoring on wireless sensor network with multiple connectivity.
Functional design	**Activity** 1. *Hardware Manufacturer*: Design of the smart sensor node, this phase pass through a feasibility analysis -- FW/ SW Drivers. The interaction with software integrator/provider allows to put new requirement on software development and to receive constrains for the correct execution. 2. *SW integrator*: Preliminary study and identification of FW/SW tools for the use of the sensors, their interconnection and next integration in addressed application scenario. 3. *System Integrator*: Functional architecture definition on base of deployed HW instruments and SW tools. 4. *Final User*: n.a.
Procurement & Engineering	**Activity** 1. *Hardware Manufacturer*: Procurement of PCB manufacturing and FW Driver and deployed software tools. Hardware components for the sensor node boards and their validation and functionality test on the HW & SW and FW drivers. 2. *SW integrator*: Functional architecture definition on base of deployed HW instruments and SW tools. 3. *System Integrator*: During this phase, two processes develop in parallel: 1) design, development, test and debug of the Local Cloud Gateway hardware and software; design, development, test and debug of the use case specific business

Engineering process phase	Addressed/Focus
	logic; 2) interaction with suppliers to acquire electronic and mechanical components, define external services to manufacture hardware prototypes; interaction with suppliers to acquire source code, software libraries and software licenses potentially required for the software part of the gateway. 4. *Final User*: n.a.
Deployment & Commissioning	**Activity** 1. *Hardware Manufacturer*: Prototypal smart node is released to be connected by a specific protocol and specific connectors. Proof validation smart sensor nodes, system enabling tools made available in the chain of stockholders. Reference design/ building blocks to be organized in a reference application tool and toolchain. 2. *SW integrator*: SW deployment. 3. *System Integrator*: Deployment of the prototypes to test and debug them in a real environment to evaluate their maturity. Commissioning of the golden sample to start the production of the product. 4. *Final User*: PoC Organization and Validation.
Operations & Management	**Activity** 1. *Hardware Manufacturer*: Proof of Value on certain numbers of node/system applications, test and application reference for the deployed solutions. 2. *SW integrator*: n.a. 3. *System Integrator*: Integration of use case components on the field. Data collection, storage, local processing. Remote monitoring and control. Interfacing with cloud and enterprise level. 4. *Final User*: PoC execution.
Maintenance Decommissioning & Recycling	**Activity** 1. *Hardware Manufacturer*: Family product promotion based on market and on value chain feedback and elicitation. 2. *SW integrator*: n.a. 3. *System Integrator*: Remote monitoring and control for maintenance and de-commissioning purposes. 4. *Final User*: Operational and Proof of Value on certain numbers of node/facilities.
Evolution	**Activity** 1. *Hardware Manufacturer*: Evaluation of possible improvements on base of the operational verification done in the previous steps of EP 2. *SW integrator*: n.a. 3. *System Integrator*: Faults and bugs analysis to identify solutions, improvements and new product releases. 4. *Final User*: Cost evaluation and feedback from field back in requirements as input for new user's objectives/requirements to be proposed to the system integrator (as requirements), and to the HW manufacturer for innovation of products.
Training & Education	**Activity** 1. *Hardware Manufacturer*: Application note, training tool on devices and how to do and how exploit advanced functionalities 2. *SW integrator*: User manual, training tool, how to do, and library to support advanced functionality referring also application of deployed HW tools *System Integrator*: Generation of design and source code documentation. Editing of technical and user manuals.

Engineering process phase	Addressed/Focus
	3. *System Integrator*: n.s. 4. *Final User*: Training people for operative conduction and maintenance.

8.5.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 48 UC-08.5 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	Probably not applicable.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Thanks to the AHF, we will be able to connect multiple stakeholders in a unique stream, connecting their toolchain in order to achieve automatic interactions. The baseline lacks severely of this automation as stakeholders have to iterate over manual interactions. **Planned actions** • AHT-EPP 1: Inclusion of inputs from different stakeholders. • AHT-EPP 2: Edge side processing tools increasing the interoperability of the UC. • AHT-EPP 3: Identify and procurement of multiple heterogeneous sources of data. • AHT-EPP 4: Deployment of multiple heterogeneous sources of data for edge side processing tools. • AHT-EPP 5: Execution on base of multiple heterogeneous sources of data. • AHT-EPP 7: Evaluation possible improvements. • AHT-EPP 8: The interaction with the AHF will be taught in the design phase of all the artifacts, making the development of them more agile and error-prone.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Several edge side processing tools increasing the interoperability of the UC.
Obj. 4 - Address digital learning and training activities as an integral part of	**Planned actions** • AHT-EPP 2-4-8: The interaction with the AHF will be taught in the design phase of all the artifacts, making the development of them more agile and error-prone.

WP2 Objective	Focus & Planed actions
the engineering cycle	

Table 49 UC-08.5 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Condition monitoring is one of the focus applications for STMicroelectronics in contact with primary companies and research institutions. As such, the distribution of engineering costs along the 8 stages of AHT-EP can be estimated and their peaks are on functional design and engineering, and could be thought to be partitioned as follows: requirements 10%, functional design 20%, engineering 30%, deployment 10%, operation 5%, maintenance 5%, evolution 10%, and training 10%. Operation and maintenance have low impact because the focus of the activity addresses pre-deployment innovation in condition monitoring. The proposed end to end framework have been conceived after a survey on the most frequent use cases and developed smart sensor nodes, gateway and cloud components are not isolated, on the contrary they are conceived as end to end, thanks to their integration, should: • facilitate and smooth the first four stages, as well as training, with an expected reduction of their cost by 50%. • support stages 5 and 6, making them sustainable in real applications provide new benefits to the end user in block 7.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Interoperability is the main enabler of the cost reduction in EP and of innovative contributions to maintenance and evolution. This objective will be reached through the possibility by the sensor network of accepting sensor configuration through design by definition of protocol and usage of services like FOTA.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Significant added values from the Target SoS are expected to be enabled by the integration of data originating from heterogeneous platforms through the AF (e.g. support ML from multidimensional data originating from: sensor networks for structural monitoring, environmental monitoring platforms, meteo stations, and possibly other platforms). Specifically, in the scope of this objective, we integrate with the Arrowhead Framework the vibration sensors that provides also environmental and ultrasound analysis. Moreover we offer a unique integrated toolchain for machine learning life cycle management.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Probably not applicable.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and	The smart sensor nodes, especially the wireless, are equipped with Secure Element that combined with Security SW features and secure provisioning embed security since the beginning.

Project Objective	Focus & Planed actions
automation solutions	
Obj. 6 - Training material (HW and SW) for professional engineers	Important contribution with User Manuals, Getting Started and Application notes.

8.5.4 Engineering Process analysis

Table 50 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-08.5

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	C
OBJ-AHT #1	X	X	X	X	X	X	X	X	
OBJ-AHT #2		X	X	X					
OBJ-AHT #3	X	X	X	X					
OBJ-AHT #4									
OBJ-AHT #5			X	X	X				
OBJ-AHT #6		X	X					X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2	X	X	X	X	X		X	X	
OBJ-WP2 #3	X		X	X				X	
OBJ-WP2 #4		X	X	X				X	

9. UC-09 Machine operation optimisation (ACCIONA)

Use Case 9 focuses on the development of a digital platform for remote monitoring and optimization of earthworks in construction projects (e.g. construction of highways, railways, dams, etc.).
Earthworks are carried out by heavy machinery of different types (e.g. haul trucks, excavators, bulldozers, etc.), which perform sequences of tasks across wide areas. For instance, a typical sequence for a haul truck can consist of the following tasks:

- Loading: An excavator excavates soil and loads it onto the haul truck.
- Transport: The haul truck transports the soil to a dumping area.
- Dumping: The haul truck dumps the soil in the destination area.
- Return: The haul truck returns to a loading area and repeats a similar sequence.

The aim of the digital platform under development is to provide connectivity to the earthworks machinery in order to enable monitoring and quantification of the tasks carried out on site, and to provide the platform end users with suitable visualization and analysis tools to assess

the performance of earthworks operations, detect deviations from the original plan, and support the optimization of the process.
The use case evolves across the following phases:
Phase 1 (before M0): this phase represents a situation where no digital platform is used for earthworks monitoring, and therefore construction machinery do not have any kind of connectivity. This is the starting point for the creation of the digital platform.
Phase 2 (M0): this phase represents the start of the Arrowhead Tools project, when an initial prototype of the platform was made available by ACCIONA for local collection of data from earthworks machinery, processing of the collected data in the cloud, and visualization through a web interface. Project stakeholders have to collaborate in order to develop an evolved version of the platform that will include improvements and new functionalities at all levels, from the connectivity hardware to data visualization and analysis.
Phase 3 (M12): this phase represents the 1st iteration of the Arrowhead Tools project, when stakeholders have already developed and incorporated into the preliminary prototype of the platform a set of improvements and new functionalities. Furthermore, they have started to identify potential optimizations of the engineering process and how these can be supported by the Arrowhead Framework and other technologies of the Arrowhead Tools project.
Phase 4 (M36): this represents the final stage of the Arrowhead Tools project, when the final version of the improved digital platform will be made available, and the adopted engineering process will be optimized incorporating the use of the Arrowhead Framework and other technologies of the Arrowhead Tools project.

9.1 Overall description of the UC-EP

The digital platform for earthworks monitoring and optimization is a product aimed for its operation by a contractor (Construction Company). Its design, development and maintenance is carried out as well by the contractor in collaboration with different stakeholders.
The architecture (depicted in Figure 35) of the platform consists of 3 different layers, which can be further subdivided into 6 six different components or subsystems:
Connectivity layer: this layer consists basically of the subsystems in charge of the collection of data from the construction site and project design data. These subsystems are:

- Machinery tracking: this is the core connectivity subsystem, as it provides data about the operation of earthworks heavy machinery. The main requirement is to track the position, speed and status (on/off) of each machine. Optionally, it would be useful to collect additional data from the machine, e.g. weight transported, fuel consumption, etc., either through its internal CAN Bus (if available and accessible), or through external sensors attached to the machine. In some cases, it can be possible to collect additional data from the own IoT platforms of the machine manufacturers (Caterpillar, Volvo, etc.). Furthermore, the use case will explore the possibility of integrating measurements of the fuel supplied to the machines, either from fixed or from mobile supply stations.
- User manual inputs: this subsystem allows on-site staff (e.g. machinery drivers) to introduce manually additional data about the process, including for instance quality and safety checks.
- Planning: this subsystem collects the project design data, which will be used for interpreting the data collected from the other two connectivity subsystems (Machinery tracking and User manual inputs). The main categories of project design data needed are: subdivision of the project site into different areas according to the type of operation planned there (e.g. cutting, filling, etc.), quantification of the mass/volume of soil to be processed in each area, and planned dates for completion of tasks in each area.

Cloud layer: this layer carried out the processing of the data collected through the connectivity layer. The main output is the identification and measurement of the operation cycles of the earthworks machinery. The layer integrates just one subsystem:

- Analysis: this is the single subsystem of the cloud layer, in charge of merging and analysing the different data sources from the connectivity layer in order to identify and measure the operational cycles of the earthworks machinery.

User interface layer: this layer provides to the platform users a set of graphical user interfaces in order to support the visualization and analysis of the data collected and processed by the platform (i.e. both "raw" data collected through the connectivity layer, and the processed data provided by the cloud layer). The layer consists of two subsystems:

- Visualization: this is the main user interface for projects under execution, and it shall provide suitable visualization and analysis functionalities for accessing both "live" and historical data of a project. Among these functionalities, a map-based (both 2D and 3D) visualization/analysis tool shall be included.
- Business Intelligence: this interface shall be more oriented to the exploitation of data from projects already completed. The main objective is that this interface can support estimations of tasks costs and deadlines during the preparation of tenders for new projects, and during the detailed planning of already granted projects that are about to start.

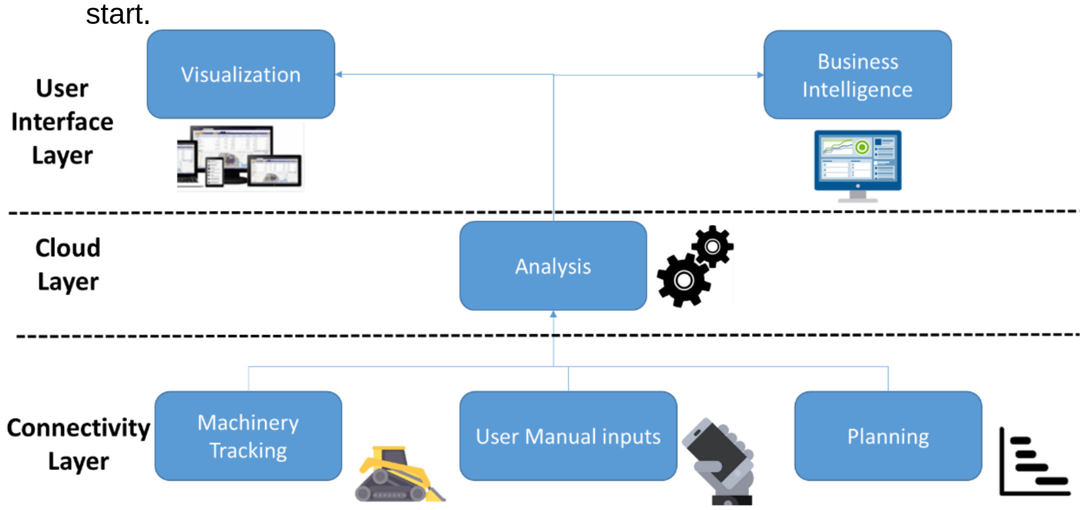


Figure 35 Architecture of the digital platform for earthworks monitoring

9.2 Engineering Process Description

As it can be seen in the engineering process represented in Figure 36, the AHT-EP of the use case encompasses the involvement of five different stakeholders. ACCIONA would play the role of both StkH-1 (as integrator of the earthworks management platform) and of StkH-5 (as end user of the platform, every construction project of ACCIONA adopting the platform would be an end user). dotGIS would play the role of StkH-2, as developer of the 3D visualization and analysis functionalities within the visualization tool. StkH-3 and StkH-5 are external subcontractors that respectively support ACCIONA in the development of the machinery tracking tool, and of the analysis, user manual inputs, and visualization tools.

StkH-1, as integrator of the earthworks management plan, is the only one that covers all the phases of the AHT-EP, and is the ultimate responsible for the development of all the tools integrated within the use case, and for each of them it is necessary to:

- Specify the requirements in the Requirements phase.

- To carry out a functional design in the Functional Design phase. For some tools, StkH-1 only produces only a preliminary Functional Design, which is then further refined by another stakeholder.
- The hardware/software development takes place in the Procurement & Engineering phase. As in the previous phase, there is an interaction with other stakeholders that provide certain hardware and/or software modules for each of the tools.
- StkH-1, according to its integrator role, is the main responsible for the Deployment & Commissioning phase, which targets the implementation of the earthworks management platform in real construction projects.
- During the Operation & Management phase, StkH-5 is responsible for overseeing the global operation of the earthworks management platform, and for carrying out advanced configurations that cannot be done by the end users.
- During the Maintenance, Decommissioning & Recycling phase, StkH-5 coordinates the routine maintenance actions of the rest of stakeholders, and identifies the need of carrying out special maintenance actions.
- StkH-1 is responsible as well for defining the evolution needed for each of the tools of the platform.
- Lastly, during the Training & Education phase, StkH1 compiles all the training material both for developers and for end users. The material for developers comes generally from the documentation generating during the Procurement & Engineering phase.

StkH-2, as developer of the 3D visualization and analysis functionalities within the visualization tool, covers all the phases except for Deployment & Commissioning, as this is carried out entirely by StkH-1. However, the 3D visualization and analysis functionalities could be adapted to be an independent tool commercialized separately from the rest of the earthworks management platform. In this scenario, StkH-1 would probably cover all the phases of AHT-EP.

StkH-3 and ***StkH-4*** cover a similar set of phases of the AHT-EP. Each of them have closed loops with StkH-1 for each of the phases where they participate. For instance, for the Functional Design phase, they use as inputs the results of the Requirements and/or of the Functional Design phases from StkH-1, and their respective outputs are then fed into the Functional Design phase of StkH-1 for final validation. One significant difference between StkH-3 and StkH-4 is that StkH-3 participates in the Deployment & Commissioning phase, as they provide a backend for the tracking devices deployed in the earthworks machinery.

StkH-5 participates firstly in the Requirements phase for providing high level requirements and business targets for the different tools of the use case, and then is involved in the Operation & Management phase as end user of the tools. They also participate in the Maintenance, Decommissioning & Recycling phase, as they have to contribute to certain routine maintenance actions of some components (e.g. tracking units attached to the machinery). Lastly, their participation in the Training & Education phase is as recipient of the training material generated by ShtkH-1.

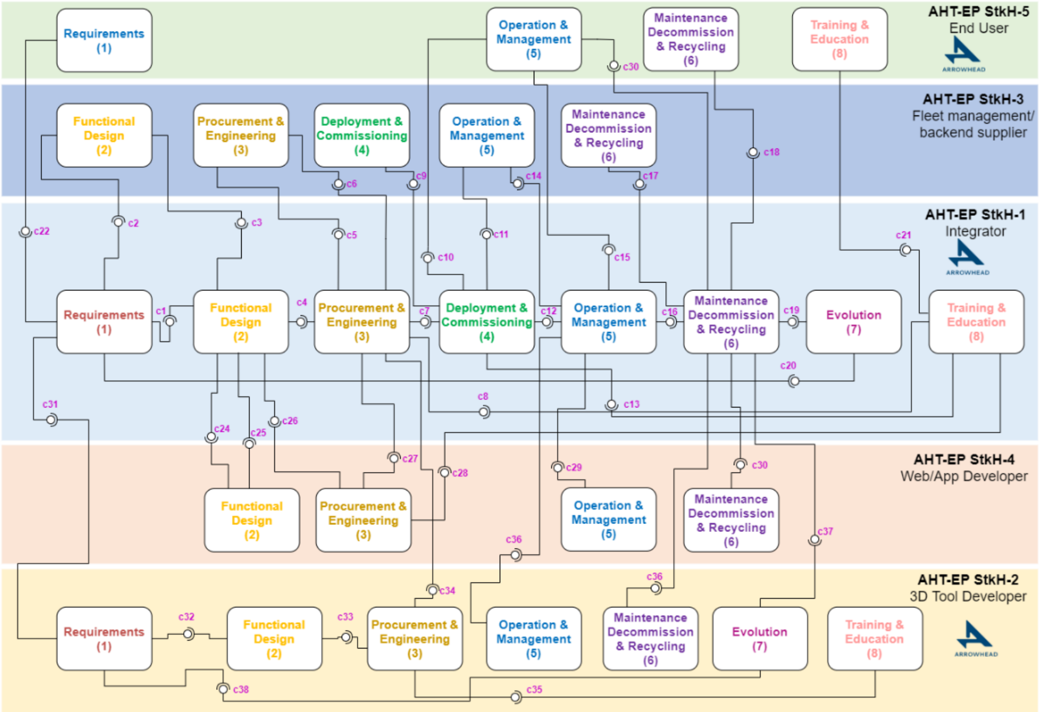


Figure 36 AHT-EP of use case UC-09

The main framework currently adopted for the development of the use case is SCRUM. This framework allows structuring the development of the platform into iterations with specific timing and objectives, called sprints. Each sprint aims to deliver an increment in the product that provides added value for the customers/stakeholders. Each sprint usually may contain tasks belonging to different phases of the UC-EP.
Although not in the focus of this use case, Building Information Modelling (BIM) is a standard framework within the construction domain for supporting the engineering process across the life-cycle of a project. BIM provides a set of guidelines, processes and technological tools for managing key design and project data in digital format throughout the life-cycle of a building or civil infrastructure.

All AHT-EP phases are considered within the use case, even if some of them have more relevance than others.
No specific lack of technology has been identified yet.
No other Engineering Process standards are adopted in the use case except for standards that have been previously considered for the development of the AHT_EP.

The EP described encompasses the development of both the system of systems level and of the different systems/tools that are integrated within the system of systems. Furthermore, the EP connects the different stakeholders involved in the development and operation of the systems of systems. The AP could be escalated to the development of a more generalized digital platform for construction projects, which would help to manage/optimize additional construction processes.

Table 51 AHT-EP Phase focus of UC-09

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Machinery Tracking*: StkH1 develops the specification of the tracking system needed for the construction machinery, including the requirements in terms of data to be gathered, electrical power, installation constraints, etc. This specification is shared with StkH3, who will be in charge of providing the GPS modules (and optionally, additional modules such as CAN Bus connector) and backend for the tracking system. These requirements are gathered in Word or PDF documents. 2. *User Manual Inputs (mobile app)*: StkH1 gathers requirements from StkH5 in order to understand which data shall be introduced through the app by the machinery operators or other stakeholders on site. StkH5 may provide this information through form templates that are usually filled in by the staff on paper or in Excel files. StkH1 will identify the main features to be included in the app. 3. *Planning*: StkH1 gathers from SthH5 the requirements in terms of the project planning and design information with which the real progress of earthworks shall be compared, including: tools currently used for creating/managing project planning and design, options for exchange of information with the tools, structure and logic followed for the planning and design, procedures for update of the planning and for comparison with the real progress of earthworks. 4. *Analysis*: StkH1 gathers requirements from StkH5 in order to identify how earthwork operations shall be identified and measured, and what KPIs are needed in order to assess project performance. According to these requirements, StkH1 defines the formulas and algorithms needed to analyse the data provided by the machinery tracking system, in combination with the user manual inputs and the planning system, and to calculate KPIs based on these data. 5. *Visualization*: StkH1 gathers requirements from StkH5 in order to identify the best ways to query and visualize the data from the analysis system as well as from other systems of the digital platform. StkH1 and StkH4 will agree on the main features to be included in the web application for query and analysis of data through tables, charts and 2D maps. StkH1 and StkH2 will agree on the features for the specific 3D Visualization. 6. *Business Intelligence*: StkH1 gathers requirements from StkH5 in order to identify project general data/KPIs that could be useful for the construction company in order to support estimation of costs and duration of tasks for future tenders. StkH1 will then analyse how the data obtained from the rest of the systems of the digital platform deployed in several construction sites can be further processed and correlated in order to generate the identified general data/KPIs. **Tools** 1. *Machinery Tracking*: Elicitate the requirements for the Machinery Tracking System. • Output: <StkH1 - EPP2> requirements for the functional design of the Machinery Tracking System • Output: <StkH3 - EPP2> requirements for the selection of the optimal hardware modules for the tracking system. 2. *User Manual Inputs (mobile app)*: Elicitate the requirements for the app for user manual inputs • Input: <StkH5 - EPP1> procedures definitions. Forms/templates typically used for manual collection of data on the field. • Output: <StkH1 - EPP2> requirements for the functional design of the User Manual Inputs app 3. *Planning*: Elicitate the requirements for the planning tool • Input: <StkH5 - EPP1> Project design and planning data. Planning management procedures. • Output: <StkH1 - EPP2>Requirements for the functional design of the

Engineering process phase	Addressed/Focus
	Planning tool 4. *Analysis*: Elicitate the requirements for the analysis tool • Input: <StkH5 - EPP1> Definition of project KPIs. Information about earthwork operations carried out on site. • Output: <StkH1 - EPP2>Definition of algorithms for earthworks operation identification and KPI calculation 5. *Visualization*: Elicitate the requirements for the 2D and 3D visualization tool • Input: <StkH5 - EPP1> Identification of needs of visualization of data from the rest of platform tools. • Output: <StkH1 - EPP2>List of requirements for 2D Visualization. • Output: <StkH2-EPP1>List of requirements for 3D Visualization. • Output: <StkH2-EPP2>Specification of requirements for the 3D tool development 6. *Business Intelligence*: Elicitate the requirements for the business intelligence tool • Input: <StkH5 - EPP1>Identification of company-wide KPIs/information of interest. • Output: <StkH1 - EPP2> Identified main functionalities of the tool and of the data sources to be used.
Functional design	**Activity** 7. *Machinery Tracking*: StkH3 proposes to StkH1 the most suitable GPS module and any other optional connectivity modules. StkH1 defines all the blocks of the tracking system, including the modules proposed by StkH3 and any other modules needed for power/energy harvesting, mounting accessories, enclosures, etc. Furthermore, StkH3 and StkH1 agrees on the format of the data to be stored in the backend provided by StkH3. 8. *User Manual Inputs (mobile app)*: StkH1 will develop a more detailed specification of the app producing mock-ups/wireframes for documenting the layout of the app, functionalities, etc. StkH4 may work on this specification in order to improve usability or functional aspects, and help to list data inputs to be collected from the users, outputs calculated automatically from the inputs, and potential connections with other systems of the platform. 9. *Planning*: Based on the requirements gathered, StkH1 and StkH4 will agree on the structure of the planning system, including: 1) which inputs will be received from specific external planning/design tools? 2) Which inputs will be provided through a dedicated user interface, and the layout and functionalities of such interface? 3) How the comparison between the planning and the real progress of earthworks will be displayed to the user? 4) Which outputs will be produced to be used by other systems of the digital platform? 10. *Analysis*: Based on the requirements gathered, StkH4 will define the different functional blocks of the analysis system and the data formats for storage in the backend. 11. *Visualization*: Based on the requirements gathered, StkH1 will agree with StkH4 on the structure of the 2D visualization system, and with StkH2 on the structure of the specific 3D Visualization. 12. *Business Intelligence*: Based on the requirements gathered, StkH1 will identify already available business intelligence tools that could provide the desired functionalities, how the candidate tools could be connected with the rest of systems of the digital platform, and whether specific additional modules shall be developed for complementing the tool functionalities. StkH1 will finally agree with StkH4 on the final structure of the Business Intelligence system to be developed. **Tools** 1. *Machinery Tracking*: Design of the machinery tracking system specifying all its hardware components and main data flows. • Input: <StkH1 - EPP1> requirements for the functional design of the Machinery Tracking system. • Input: <StkH3 - EPP2> selected hardware components for integration in the

Engineering process phase	Addressed/Focus
	functional design of the system. • Output: <StkH1 - EPP3> functional design of the complete tracking system. 2. *User Manual Inputs (mobile app)*: Detailed design of the app for user manual inputs • Output: <StkH4 - EPP2> initial mock-ups/wireframes of the app. • Input: <StkH4 - EEP2> refined mock-ups/wireframes. • Output: <StkH4 - EPP3> final functional design of the app. 3. *Planning*: Detailed functional design of the planning tool • Output: <StkH4 - EPP2>initial functional design of the planning tool (input data + mock-up of user interface). • Input: <StkH4 - EPP2>refined functional design, including connections with other systems of the platform. • Output: <StkH4 - EPP3>final functional design of the planning app. 4. *Analysis*: Detailed functional design of the analysis tool • Output: <StkH4 - EPP2>initial functional design of the analysis tool (detailed algorithms and KPIs calculation procedures) • Input: <StkH4 - EPP2>refined functional design, including functional blocks, data formats, backend procedures, etc. • Output: <StkH4 - EPP3>final validated functional design of the analysis tool 5. *Visualization*: Detailed functional design for the 2D and 3D visualization tool • Output: <StkH4 - EPP2>initial mock-ups/wireframes for 2D visualization tool • Input: <StkH4 - EPP2>refined functional design of the 2D visualization tool, including connection with other platform tools • Output: <StkH4 - EPP3> final validated functional design of the 2D visualization tool • Output: <StkH2 - EPP3> functional design of the 3D visualization tool 6. *Business Intelligence*: Detailed functional design of the business intelligence tool • Output: <StkH4 - EPP2>initial functional design of the business intelligence tool (data processing algorithms, KPIs definitions, wireframes, etc.). • Input: <StkH4 - EPP2>refined functional design of the business intelligence tool, identifying potential existing tools to be integrated, additional modules to be developed, etc. • Output: <StkH4 - EPP3>final validated functional design of the business intelligence tool.
Procurement & Engineering	**Activity** 7. *Machinery Tracking*: StkH1 purchases the GPS and any other optional modules from StkH3. Additional modules for electrical power/energy harvesting, mounting accessories, enclosures, etc. are purchased from other external suppliers. StkH1 carries out the detailed design of the tracking system, including mechanical/electrical designs. A working prototype (or a small set of prototypes) of the tracking system is produced and tested in real machinery in a controlled environment (machinery park owned by StkH1). StkH3 configures/adapts the backend for the tracking system, according to the previous functional design. Once the prototype is validated, StkH1 repeats the procurement process for all the components needed for producing the number of tracking system units needed by the customer (StkH5). The tracking system units are produced in a workshop owned by StkH1. 8. *User Manual Inputs (mobile app)*: The development team of StkH4 will develop the app (both the mobile user interfaces and the backend) according to the specifications of the Functional Design, including the connections that may be needed with the rest of systems of the platform. StkH1 will carry out the validation of the app prototype. 9. *Planning*: The development team of StH4 will develop the planning system (both web graphical user interface and backend) according to the specifications of the Functional Design, including the connections that may be needed with the rest of systems of the platform. StkH1 will carry out the validation of the system prototype. 10. *Analysis*: The development team of StkH4 will develop the analysis system

Engineering process phase	Addressed/Focus
	according to the functional design, including the connections needed with the rest of systems of the platform. StkH1 will carry out the validation of the system prototype using testing data generated by the Machinery Tracking system and checking that the analysis system produces the correct outputs. 11. *Visualization*: The development team of StkH4 will develop the 2D visualization module according to the functional design, including the connections needed with the rest of systems of the platform. StkH2 will follow the same approach for the 3D visualization module. StkH1 will validate the prototype of both visualizations. 12. *Business Intelligence*: The development team of StkH4 will develop the Business Intelligence system according to the functional design, including the connections needed with the rest of systems of the platform. StkH1 will validate the prototype of the Business Intelligence System. **Tools** 1. *Machinery Tracking*: Detailed design, production and testing of the tracking system prototype. • Input: <StkH1 - EPP2> functional design of the system. • Output: <StkH3 - EPP3> purchase order of components needed for manufacturing system units needed by the customer. • Input: <StkH3 - EPP3> components for integration in the machinery tracking system units to be produced. • Output: <StkH1 - EPP4> set of machinery tracking system units to be deployed on site. • Output: <StkH1 - EPP8> detailed documentation of the system for guiding the production of additional units. 2. *User Manual Inputs (mobile app)*: Development of the app for user manual inputs • Input <StkH4 - EPP3> developed app prototype. • Output: <StkH1 - EPP4> validated app prototype ready for deployment. • Output: <StkH1 - EPP8> technical documentation of the app prototype. 3. *Planning*: Development of the planning tool • Input: <StkH4 - EPP3> developed prototype of the planning tool. • Output: <StkH1 - EPP4> validated prototype of the planning tool ready for deployment. • Output: <StkH1 - EPP8> technical documentation of the app prototype. 4. *Analysis*: Development of the analysis tool • Input: <StkH4 - EPP3>developed prototype of the analysis tool. • Output: <StkH1 - EPP4> prototype of the planning tool validated with data from the machinery tracking tool. • Output: <StkH1 - EPP8>technical documentation of the analysis tool prototype. 5. *Visualization*: Development of the 2D and 3D visualization tool. • Input: <StkH4 - EPP3> developed prototype of the 2D visualization tool. • Input: <StkH2 - EPP3> developed prototype of the 3D visualization tool. • Output: <StkH1 - EPP4> integrated prototype of the 2D&3D visualization tool ready for deployment. • Output: <StkH1 - EPP8> technical documentation of the 2D visualization tool. • Output: <StkH2 - EPP8> technical documentation of the 3D visualization tool. 6. *Business Intelligence*: "Development of the business intelligence tool • Input: <StkH4 - EPP3>developed prototype of the business intelligence tool. • Output: <StkH4 - EPP3>validated prototype of the business intelligence tool. • Output: <StkH1 - EPP8>technical documentation of the business intelligence tool.

Engineering process phase	Addressed/Focus
Deployment & Commissioning	**Activity** 1. *Machinery Tracking*: StkH1 carries out the deployment of the tracking system units on the machinery of the construction site managed by StkH5. StkH1 coordinates with StkH3 for the commissioning of each tracking system unit, in order to verify that data from the system are correctly sent and stored in the backend from StkH3. 2. *User Manual Inputs (mobile app)*: StkH1 will deploy the app in a set of mobile devices (typically a rugged tablet) to be used by StkH5, and configures all necessary user accounts. 3. *Planning*: For the deployment, StkH1 will carry out the customization of the system according to the needs of StkH5, i.e. inserting the specific project planning/design information. 4. *Analysis*: The deployment of the analysis system in the construction site of StkH5 requires the previous deployment of the other systems providing data: the machinery tracking, the mobile app, and the planning system. StkH1 will carry out checks in order to assess the precision of the outputs produced, i.e. the correspondence of these outputs with real operations of the machinery. This can be done either through inspections on site, or by cross-checking with manual inputs introduced through the app, or combining both approaches. 5. *Visualization*: 2D and 3D visualization will be integrated for StkH5 as a single interface with 2D and 3D capabilities. The visualization will need to be connected to the rest of systems of the digital platform. 6. *Business Intelligence*: The Business Intelligence system will be connected to the rest of systems of the digital platform to allow collection of data from different projects. **Tools** 1. *Machinery Tracking*: Deployment and commissioning of the tracking system in a construction site • Input <StkH3 - EPP4> support for commissioning of tracking system units in StkH4 backend. • Output <StkH5 - EPP5> tracking system units deployed on machinery of construction site. • Output <StkH3 - EPP5> tracking system units commissioned in StkH4 backend. • Output <StkH1 - EPP5> commissioned tracking system. • Output <StkH1 - EPP8> documentation of tracking system deployment. 2. *User Manual Inputs (mobile app)*: Installation and configuration of the app in mobile devices for the construction site • Output <StkH5 - EPP5> app installed and configured in mobile devices in construction site. 3. *Planning*: Configuration and commissioning of the planning tool for the construction site • Output <StkH5 - EPP5> planning tool deployed on the cloud and configured/customized for its use by the customer at the construction site. 4. *Analysis*: Configuration and commissioning of the analysis tool for the construction site • Output <StkH5 - EPP5> analysis tool deployed in the cloud and tested on site in order to ensure precision of automated identification and measurement of earthworks. 5. *Visualization*: Configuration and commissioning of the visualization tool for the construction site • Output <StkH5 - EPP5>2 D&3D visualization tool deployed in the cloud and tested for the construction site. 6. *Business Intelligence*: Configuration and commissioning of the visualization tool for the construction site • Output <StkH5 - EPP5> 2D&3D visualization tool deployed in the cloud and tested for the construction site.

Engineering process phase	Addressed/Focus
Operations & Management	**Activity** 1. *Machinery Tracking*: During the Operation & Management phase, StkH1 and StkH5 may need to coordinate in order to dismount the tracking system unit from certain machines or mounting them in new ones, and/or introduce additional tracking system units. StkH3 will oversee the correct operation of the tracking system backend server, and StkH1 will check the communications with the rest of systems of the platform. 2. *User Manual Inputs (mobile app)*: StkH5 uses the app for the introduction of data by the machinery operators or other stakeholders on site. StkH4 will oversee the correct operation of the app backend. StkH1 will create and configure additional user accounts on demand, and check the communications with the rest of systems of the platform. 3. *Planning*: Project planning/design information may need to be updated by StkH5 during the Operation & Management phase. StkH1 will provide support if needed, and check the correct integration with the rest of systems of the platform. StkH4 will oversee the correct operation of the of the planning system backend. 4. *Analysis*: StkH4 will oversee the correct operation of the analysis system backend. StkH1 will check the correct integration with the rest of systems of the platform and collect any technical issues detected. 5. *Visualization*: StkH4 and StkH2 will oversee the correct operation of their respective system back-ends. StkH1 will check the correct integration with the rest of systems of the platform. 6. *Business Intelligence*: StkH1 will cooperate with StkH5 in order to fed into the Business Intelligence system data from different projects. StkH4 will oversee the correct operation of the business intelligence backend. **Tools** 1. *Machinery Tracking*: Operation of the tracking system in the construction site • Input <StkH3 - EPP5> ensure correct operation of tracking system backend server. • Output <StkH5 - EPP5> support for deploying new tracking system units or change tracking system units to other machines. • Output <StkH1 - EPP6> tracking system in operation. 2. *User Manual Inputs (mobile app)*: Operation of the app for user manual inputs in the construction site • Output <StkH5 - EPP5> User accounts administration. • Input <StkH4 - EPP5> ensure correct operation of app backend. 3. *Planning*: Operation of the planning tool • Output <StkH5 - EPP5>support for updates/modifications of the original project design/planning information and procedures • Input <StkH4 - EPP5> ensure correct operation of planning tool backend. 4. *Analysis*: Operation of the analysis tool • Input <StkH4 - EPP5> ensure correct operation of analysis tool backend. 5. *Visualization*: Operation of the 2D&3D visualization tool • Input <StkH4 - EPP5>ensure correct operation of the 2D visualization tool backend. • Input <StkH2 - EPP5>ensure correct operation of the 3D visualization tool backend. 6. *Business Intelligence*: Operation of the business intelligence tool • Input <StkH4 - EPP5>ensure correct operation of the business intelligence tool backend.
Maintenance Decommissioning & Recycling	**Activity** 7. *Machinery Tracking*: Usually the only maintenance action required is the periodical cleaning of the energy harvesting modules (e.g. solar PV panels) of the tracking system unit that shall be carried out by StkH5, as well as routine maintenance actions in the tracking system backend server, which is carried out

Engineering process phase	Addressed/Focus
	by StkH3. 8. *User Manual Inputs (mobile app)*: StkH1 will provide support to StkH5 in case any technical issue arises related to the use of the app. StkH4 will carry out routine maintenance actions in the app backend, and may use additional tools to monitor the app operation. 9. *Planning*: StkH1 will provide support to StkH5 in case any technical issue arises related to the use of the planning tool. StkH4 will carry out routine maintenance actions in the planning system backend. 10. *Analysis*: StkH4 will carry out routine maintenance actions in the analysis system backend. 11. *Visualization*: StkH4 and StkH2 will carry out routine maintenance actions in their respective system back-ends. 12. *Business Intelligence*: StkH4 will carry out routine maintenance actions in the business intelligence back-end. **Tools** 1. *Machinery Tracking*: Implementation of routine maintenance actions for the different components of the tracking system • Input <StkH3 - EPP6> backend server routine maintenance actions • Input <StkH5 - EPP6> routine maintenance actions of tracking system units deployed in construction site • Output <StkH1 - EPP7> list of bugs/technical issues detected. 2. *User Manual Inputs (mobile app)*: Maintenance of the app in order to solve any technical issues • Input <StkH4 - EPP6> routine maintenance actions of app backend. App logs monitoring. • Input <StkH5 - EPP5> detection of technical issues during app operation. • Output <StkH1 - EPP7> list of bugs/technical issues detected. 3. *Planning*: Maintenance of the planning in order to solve any technical issues • Input <StkH4 - EPP6> routine maintenance actions of planning tool backend. • Input <StkH5 - EPP5> detection of technical issues during operation of planning tool. • Output <StkH1 - EPP7> list of bugs/technical issues detected. 4. *Analysis*: Maintenance of the analysis tool in order to solve any technical issues • Input <StkH4 - EPP6> routine maintenance actions of analysis tool backend. • Input <StkH5 - EPP5>detection of technical issues during operation of analysis tool. • Output <StkH1 - EPP7> list of bugs/technical issues detected. 5. *Visualization*: Maintenance of the 2D/3D visualization tool in order to solve any technical issues • Input <StkH4 - EPP6> routine maintenance actions of 2D visualization tool backend. • Input <StkH2 - EPP6> routine maintenance actions of 3D visualization tool backend. • Input <StkH5 - EPP5>detection of technical issues during operation of 2D/3D visualization tool. • Output <StkH1 - EPP7> list of bugs/technical issues detected in the 2D visualization tool. • Output <StkH2 - EPP7> list of bugs/technical issues detected in the 3D visualization tool. 6. *Business Intelligence*: Maintenance of the business intelligence tool in order to solve any technical issues • Input <StkH4 - EPP6> routine maintenance actions of business intelligence tool backend. • Input <StkH5 - EPP5>detection of technical issues during operation of business intelligence tool. • Output <StkH1 - EPP7>list of bugs/technical issues detected in the business

Engineering process phase	Addressed/Focus
	intelligence tool.
Evolution	**Activity** 1. *Machinery Tracking*: Data collected by StkH1 during the Operation & Management phase is used for identifying potential improvements in the design of the whole tracking system. The improvements may apply to the GPS and other optional connectivity modules provided, and/or to the tracking system backend. 2. *User Manual Inputs (mobile app)*: Based on feedback from StkH5 or from potential malfunctions observed during the Operation & Management phase, StkH1 will agree with StkH4 on new releases of the app. 3. *Planning*: Based on feedback form StkH5, or due to the potential adoption of different planning tools/methodologies, StkH1 may agree with StkH4 on new versions of the planning system. 4. *Analysis*: Based on the validation of the system outputs, StkH1 may identify refinements needed in the formulas/algorithms that produce them. Accordingly, StkH1 and StkH4 would agree on new versions of the analysis system. 5. *Visualization*: Based on feedback from StkH5, StkH1 may identify potential improvements or new functionalities to be added to the system. Accordingly, new versions of the visualization could be agreed with StkH2 and/or StkH4. 6. *Business Intelligence*: Based on feedback from StkH5, StkH1 may identify potential improvements or new functionalities to be added to the system. Accordingly, new versions of the business intelligence system could be agreed with StkH4. **Tools** 1. *Machinery Tracking*: Identification of potential improvements of the tracking system • Output <StkH1 - EPP1> Identified new functionalities/improvements for the tracking system. 2. *User Manual Inputs (mobile app)*: Identification of potential improvements of the app • Output <StkH1 - EPP1> Identified new functionalities/improvements for the tracking system. 3. *Planning*: Identification of potential improvements of the planning tool • Output <StkH1 - EPP1> Identified new functionalities/improvements for the planning tool. 4. *Analysis*: Identification of potential improvements of the analysis tool • Output <StkH1 - EPP1>Identified new functionalities/improvements for the analysis tool. 5. *Visualization*: Identification of potential improvements of the 2D & 3D visualization tool • Output <StkH1 - EPP1>Identified new functionalities/improvements for the 2D visualization tool. • Output <StkH12 - EPP1>Identified new functionalities/improvements for the 3D visualization tool. 6. *Business Intelligence*: Identification of potential improvements of the business intelligence tool • Output <StkH1 - EPP1> identified new functionalities/improvements of the business intelligence tool.
Training & Education	**Activity** 1. *Machinery Tracking*: StkH1 generates the documentation of the tracking system for guiding the production of additional units. A user manual is also generated for StkH5 for guiding mounting/dismounting of a tracking system unit in a machine, and for maintenance. 2. *User Manual Inputs (mobile app)*: StkH4 will generate detailed documentation of the application for StkH1. StkH1 will elaborate a user manual of the app for StkH5.

Engineering process phase	Addressed/Focus
	3. *Planning*: StkH4 will generate detailed documentation of the planning system for StkH1. StkH1 will elaborate a user manual of the planning system for StkH5. 4. *Analysis*: StkH4 will generate detailed documentation of the planning system for StkH1. 5. *Visualization*: StkH4 will generate detailed documentation of the 2D visualization system for StkH1. StkH2 will internally develop detailed documentation of the specific 3D visualization. 6. *Business Intelligence*: StkH4 will generate detailed documentation of the business intelligence system for StkH1. StkH1 will elaborate a user manual of the business intelligence system for StkH5. **Tools** 1. *Machinery Tracking*: Collection and elaboration of documentation of the tracking system for both developers and end users • Output <StkH5 - EPP8> Manual for end user in construction site. 2. *User Manual Inputs (mobile app)*: Collection and elaboration of documentation of the app for both developers and end users • Output <StkH5 - EPP8> Manual for end users in construction site. 3. *Planning*: Collection and elaboration of documentation of the app for both developers and end users • Output <StkH5 - EPP8> Manual for end users in construction site. 4. *Analysis*: Collection and elaboration of documentation of the app for both developers and end users • Output <StkH5 - EPP8> Manual for end users in construction site. 5. *Visualization*: Collection and elaboration of documentation of the app for both developers and end users • Output <StkH5 - EPP8> Manual for end users in construction site. 6. *Business Intelligence*: "Collection and elaboration of documentation of the business intelligence tool for both developers and end users • Output <StkH5 - EPP8> Manual for company-wide end users.

9.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 52 UC-09 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The EP of the Use Case includes the collection of data during Phase 5 (Operation & Management) in order to assess the performance of the different tools integrated in the digital platform, and to implement the corresponding corrective actions either in Phase 6 (Maintenance, Decommissioning & Recycling) or in Phase 7 (Evolution). **Planned actions** • AHT-EPP 5: Collection of operational data to assess performance of the deployed tools/systems. • AHT-EPP 6: Use of collected operational data for implementing routine maintenance actions and tuning the performance of the tools • AHT-EPP 7: Use of collected operational and maintenance data for planning evolution of the tools.
Obj. 2 - The move from single to integrated multi stakeholder	The EP of the Use Case integrates the contributions from different stakeholders throughout all the phases, identifying the connections and exchange of data/information between stakeholders at each phase.

WP2 Objective	Focus & Planed actions
automation and digitalization	**Planned actions** • AHT-EPP 1: Integration of 3 stakeholders in the Requirements Phase. • AHT-EPP 2: Integration 3 stakeholders in the Functional Design Phase. • AHT-EPP 3: Integration of 3 stakeholders in the Procurement & Engineering Phase. • AHT-EPP 4: Integration of 2 stakeholders in the Deployment & Commissioning Phase. • AHT-EPP 5: Integration of 5 stakeholders in the Operation & Management Phase. • AHT-EPP 6: Integration of 5 stakeholders in the Maintenance, Decommissioning & Recycling Phase. • AHT-EPP 7: Integration of 2 stakeholders in the Evolution Phase. • AHT-EPP 8: Integration of 3 stakeholders in the Training & Education Phase.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Thanks to the graphical representation of the UC-EP, it is easier to visualize the different connections between the different phases of the EP and between the different stakeholders involved in each of these phases. Once these connections and the associated data/information exchanges are identified, it is easier to target the optimization/automation of part of these exchanges. **Planned actions** • AHT-EPP 1 to 8: I/O's mapped in the UC-EP
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Generation of documentation/training material (both for developers and for end users) is foreseen for each of the tools/systems that integrate the digital platform. **Planned actions** • AHT-EPP 8: Generation of documentation for end users and developers.

Table 53 UC-09 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	The reduction of solution engineering costs shall be achieved through the automation of the interactions/exchange of data/information between the different tools integrated in the platform for earthworks management. The main interactions to be optimized for cost reduction are the following: • User manual inputs: these inputs have been collected traditionally on paper forms that need later be consolidated in Excel files, which is a time consuming process. The use case introduces in first place a clear improvement through the digitization of the process, as the data will be created directly in digital format through the app. • Earthworks operations measurement and tracking: earthworks operations traditionally can only be measured accurately through topography (or lately, through measurements taken with drone flights), which involves a delay between the completion of an operation and the availability of the measurement. Although the platform does not remove the need of making these high-precision measurements, the machinery tracking together with the analysis tool allow close to real time tracking and measurement of

Project Objective	Focus & Planed actions
	operations carried out on site, thus helping to early detect any deviation or delay that can negatively impact on the project costs. • Planning: project design and planning data need to be constantly updated during the execution of a construction project. Each time an update is done, a significant workload is involved for crossing the new design/planning with the measurement of earthworks operations in order to track progress. The use case will help to automate this process so that new planning/design data can be easily imported and the earthworks operations will be automatically measured and tracked against the new planning data. • Business Intelligence: collection and consolidation of data and KPIs of earthworks operations in already finished construction projects is necessary for better estimation of time and costs for new tenders, and also for the elaboration of detailed planning at the beginning of new projects. The connection of the business intelligence tool to the platform will help to reduce the time needed to retrieve and compare data and indicators from different projects.
Obj. 2 - Interoperability for IoT and SoS engineering tools	This objective is not among the main priorities of the use case. However, the Arrowhead Framework will be assessed in order to identify opportunities for using it to improve the interoperability between the different tools of the project.

Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The main contributions of the use case to this project objective are the following:
	• The integration of data from legacy earthworks machinery from different vendors that may or may not be equipped with their own monitoring systems. The indicators to be assessed for the completion of these objective are:
	  - Cost effectiveness of the monitoring/tracking unit to be attached to the machinery.
	  - Low degree of intrusiveness and ease of deployment of the monitoring/tracking unit.
	  - Capability to integrate data regardless of the machinery vendor.
	  - Flexibility for carrying out either basic or advanced monitoring/tracking (basic monitoring □ tracking position, speed and on/off status of machinery; advanced monitoring □ collecting additional operational data either through internal CAN Bus or through additional sensors attached to the machine)
	  - Capability of combining data collected through monitoring/tracking units for some machines with data collected directly from vendors' own telemetry solutions for other machines.
	  - Capability of integrating data from other legacy systems supporting operation of earthworks machinery (e.g. mobile and fixed fuel stations on site).
	• Integration of data from legacy planning tools: capability of the platform to automate integration and further updates of project design/planning data from legacy planning tools.

Obj. 4 - Integration platform interoperability with emerging digitalization and	This objective is not among the main priorities of the use case.

Project Objective	Focus & Planed actions
automation framework	
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	This objective is not among the main priorities of the use case, although the tools developed will implement the necessary standard security mechanisms.
Obj. 6 - Training material (HW and SW) for professional engineers	This objective is not among the main priorities of the use case. Anyway, the use case will produce different training materials targeting on one hand the end users of the different tools, and on the other hand the developers in charge of the operation, maintenance and evolution of the different tools.

9.4 Engineering Process analysis

Table 54 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-09

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X		X	X		X	
OBJ-AHT #2		X	X				X		
OBJ-AHT #3			X		X				
OBJ-AHT #4			X						
OBJ-AHT #5			X		X				
OBJ-AHT #6		X	X		X			X	
OBJ-WP2 #1	X	X	X		X	X	X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3	X	X	X	X	X	X	X	X	
OBJ-WP2 #4		X	X		X			X	

10. UC-10 Rapid HW development, prototyping, testing and evaluation (ARCELIK)

In the last decade, industry 4.0 alongside with IoT, have changed the understanding of manufacturing by introducing the digitalization of manufacturing processes. Labour work on the manufacturing is limiting the digitalization of manufacturing process due to the lack of connected machines in the process. Turning manpower into machine power in manufacturing creates an opportunity to utilize big data and to digitalize the manufacturing processes. One other aspect of manufacturing is test & verification which requires lots of manpower within the operations. In consumer electronics, there are many standards and compliances to satisfy before putting the products into market. Today, in many manufacturing houses, most of the test and verification processes are handled by manpower and it leads to the same problems that manufacturing processes have: lack of utilizing the real time data to build a big data cluster and digitalization of the processes.

For example, in Arcelik, PSUs (Power Supply Boards) are tested for engineering validation. The general term for these tests are EVT (Engineering validation Test). These tests are done by manpower and it takes 10 to 30 days for a test to finish. And the outcome of the test is only an excel file that indicates whether the device under test passes the test of it fails. This simple procedure provides no surplus value to utilize in bigger frame of reference. To reduce the manpower in this operation and to utilize the test measurement data by converting the operation into automated test procedure that allows the real time measurement tracking and storing each measurement data in the cloud for further calculations, this test should be handled by a machine and should be able to communicate with other machines in the network.

Arcelik aims to design and produce an EVT tool that achieves connection to the machines in higher hierarchy to receive commands through Eclipse Arrowhead framework, reports the operation results and measurements in suitable formats to be stored, and handles the entire operation without manpower.

10.1 Overall description of the UC-EP

The architecture of the use case (Figure 37) is composed by the following functional building blocks:

- Enter setup configuration and measurement set by operator via UI.
- Receive test configuration data via AH local cloud.
- Configure AC power supply and variable load.
- Take measurement via FMC board.
- Save measurement data and calculate necessary values.
- Prepare test results table and send operator via AH Local Cloud.

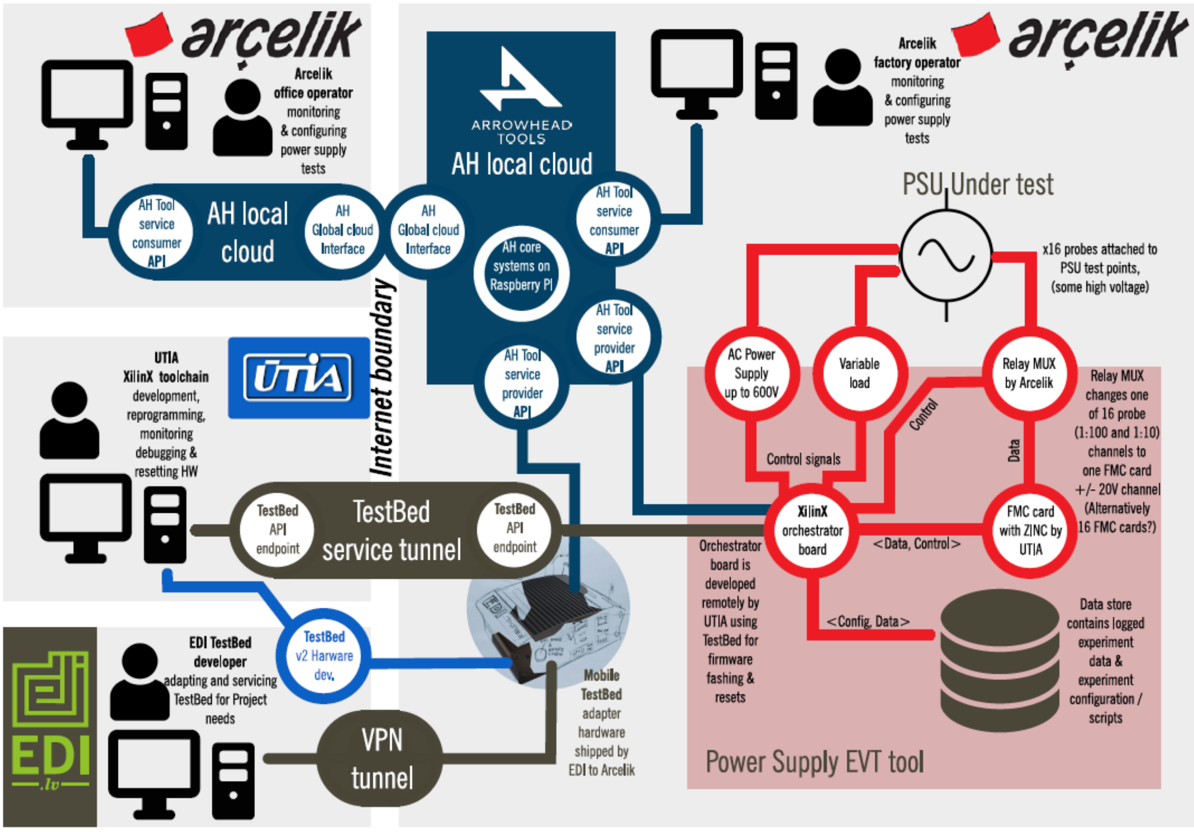


Figure 37 Architecture of the UC-10

For more information about the use case description see Deliverable D7.1 (Task 7.3).

10.2 Engineering Process Description

UC-10 involves development and integration of of these HW/SW components (tools):

- Embedded Zynq Ultrascale+ Unit (ZynqU+)
- FMC A/D (A/D)
- Relays(Relay)
- FPGA Control Unit for Power/Load (FPGA)
- Remote reconfiguration of FPGA (RemCtrl)
- Accelerated digital design on multiple PCs (ParDesign)

StkH1 = End User company (ARCELIK)
StkH2 = HW developer (UTIA)
StkH3 = Embedded SW developer (UTIA)
StkH4 = IoT communication provider (EDI)
StkH5 = Embedded SW provider (EDI)

Engineering activities performed in each phase and interaction of the AHT-ETP are presented in Figure 37 and can be described as follows:

Embedded Zynq Ultrascale+ Unit (ZynqU+)
[c1] StkH 2 acquires the specification describing the measurement system from the vendor (StkH 1). Specification documents are passed in the form of PDF files that are analyzed by the engineering team that generate a list of requirements for the design of the ZynqU+.

Requirements will be listed in a Word file. StkH 2 will design and simulate the HW part of the ZynqU+ interfaces in Xilinx Vivado simulation tool xsim.
[c26] StkH 2 will specify the of-the-shelf for the ZynqU+ module and carrier PCBs. Develop the firmware for Arm and related to PL logic interfaces to A/D using Xilinx GCC and Vivado toolchains.
[c8] Request quotes from the supplier for the electronic and sensor components selected in the PCB design process. The engineering team will develop the documentation of the HW and SW developed for the ZynqU+ that will be provided as a reference manual to StkH 1.
A working prototype of the ZynqU+ will be tested by StkH1 in the vendor laboratories.
[c20], [c13] StkH2 and Stkh3 produce the ZynqU+ SW firmware and HW design and supply them to the vendor StkH1.
[c11] StkH 3 will utilize embedded AH framework SW C++ clients running on the embedded Debian Linux for communication of the ZynqU+ with remote control/visualization PC clients.
[c18] Data collected in the Operation & Management phase of StkH 1 are analyzed by StkH2 for identifying possible bottlenecks in the ZynqU+ design. Moreover, firmware updates and HW design updates will be generated by StkH2 as needed by experiments.
[c27] StkH2 and StkH3 will create the datasheet of ZynqU+ system and reference manual of low level SW API for A53 CPU running Debian embedded Linux OS.

FMC A/D (A/D)
[c26] StkH2 will analyze existing design templates provided by company Analog devices for Xilinx University boards. StkH2 will model design templates ported to selected ZynqU+ module and carrier and model it in Xilinx Vivado simulation tool xsim.
[c14], [c21] StkH1 and Stk2 will buy identical A/D data acquisition card for the Zynq U+ system.
[c23] StkH2 produce the ZynqU+ SW firmware and HW design to control A/D and supply them to the vendor StkH1.
[c24] StkH2 produce the Linux version of ZynqU+ SW firmware and HW design to control A/D and supply them to the vendor StkH1.
[c28] StkH2 create the datasheet of A/D integration in ZynqU+ and reference manual of low level SW API for A53 CPU running Debian embedded Linux OS.

Relays(Relay)
[c1] StkH1 will specify relay system needed for switching of analog measurement points
[c2] StkH1 will model in Matlab isolation properties of the Relay in respect to isolation of high voltage environment.
[c10] StkH 1 will design the Relay PCB with Altium Designer and select the components and technologies for the PCB design. Develop the firmware for drive sensors and actuators using STM32 GCC toolchain for STM32 Nucleo controller syb-system. StkH1 will produce the Relay SW firmware and HW design and supply those to the vendor StkH2 and StkH5. StkH1 will produce the Relay SW firmware and HW design updates and supply them to the vendor StkH2 and StkH5. StkH1 create the datasheet of Relay and reference manual of low level SW API for STM32 MPU.

FPGA Control Unit for Power/Load (FPGA)
[c41] StkH 4 will design and simulate the HW part of the FPGA interfaces in Modelsim. StkH 4 will specify the of-the-shelf for the FPGA system module and carrier PCB.
[c52] StkH5 will develop the firmware for MicroBlaze softcore and HW interfaces to RS232 serial interfaces using Xilinx GCC and Vivado toolchains.
[c42] Request quotes from the supplier for the electronic and sensor components selected in the PCB design process. The Ethernet is embedded on the platform for supporting the services offered by the telecommunication provider (StkH 4). The engineering team will develop the documentation of the HW and SW developed for the FPGA that will be provided

as a reference manual to the StkH 1. A working prototype of the FPGA will be tested in a real machine for quality test in the vendor laboratories."StkH4 and StkH5 produce the FPGA SW firmware for MicroBlaze and HW design and supply them to the vendor StkH1
[c43] Data collected in the Operation & Management phase of StkH 1 are analyzed by StkH5 for identifying possible bottlenecks in the FPGA design. Moreover, firmware updates and HW design updates will be generated by StkH5 as needed by experiments. StkH4 and StkH5 will create the datasheet of FPGA system and reference manual of low level SW API for MicroBlazs soft core running in FPGA..

Remote reconfiguration of FPGA (RemCtrl)
[c41] StkH 4 will design and simulate the remote communication part of the RemCtrl interfaces in Matlab. "StkH 4 will specify the of-the-shelf for the RemCtrl for FPGA. StkH5 will develop the firmware for MicroBlaze softcore and HW interfaces to RS232 serial interfaces using Xilinx GCC and Vivado toolchains. Request quotes from the supplier for the electronic and sensor components selected in the PCB design process.
[c42] The Ethernet is embedded on the platform for supporting the services offered by the telecommunication provider (StkH 4). The engineering team will develop the documentation of the HW and SW developed for the FPGA that will be provided as a reference manual to the StkH 1. A working prototype of the FPGA will be tested in a real machine for quality test in the vendor laboratories." StkH4 and StkH5 produce the RemCtrl SW and HW and supply them to the vendor StkH1. Data collected in the Operation & Management phase of StkH 1 are analyzed by StkH4 for identifying possible bottlenecks in the RemCtrl SW/HW. [c43] Moreover, firmware updates and HW design updates for RemCtrl will be generated by StkH4 and StkH5 as needed by experiments.
[c27] StkH4 and StkH5 will create the datasheet of RemCtrl system and reference manual of low level communication protocol description.

Accelerated digital design on multiple PCs (ParDesign)
[c1] StkH2 will acquire specification for maximal HW design compilation times related to modifications and upgrades ZynqU+ programmable logic designs for single PC installation and requirements for their acceleration on several PCs from StkH1. Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the ParDesign of the ZynqU+.
[c24] StkH 2 will design and simulate the parallel compilation HW part of the ZynqU+ interfaces in Matlab parallel toolbox. StkH2 will specify required design tool sequence and define ParDesign compilation schedules acceleration on several PCs.
[c23] StkH 2 will utilize AH framework for control of execution of tools in case of use of ParDesign. Data related to recompilation of HW collected in the Operation & Management phase of StkH 1 will be analysed by StkH2 for identifying possible bottlenecks in the ParDesign.
[c3] Arrowhead firmware updates related to accelerated design scheduling will be generated by StkH2 as needed by experiments.
[c27] StkH3 will create the datasheet of Design acceleration flow and reference manual descrbing configuration of AH framework for ParDesign acceleration on multiple PCs in a local cloud for StkH 1.

In UC10, AH Framework will support automated connection of the chain of our design tools on multiple computers in the frame of a single AH local cloud. This toolchain will support definition of the sequence of tool executions to create store and document all what is needed for the embedded SoC system design needed for the Arcelik use case. This will be the use of AH framework in the design time.

In UC10, the embedded SoC system will be compatible with the AH framework. The AH framework will help in the deployment phase with data exchange to remote clients etc. This will be the use of AH framework in the runtime.

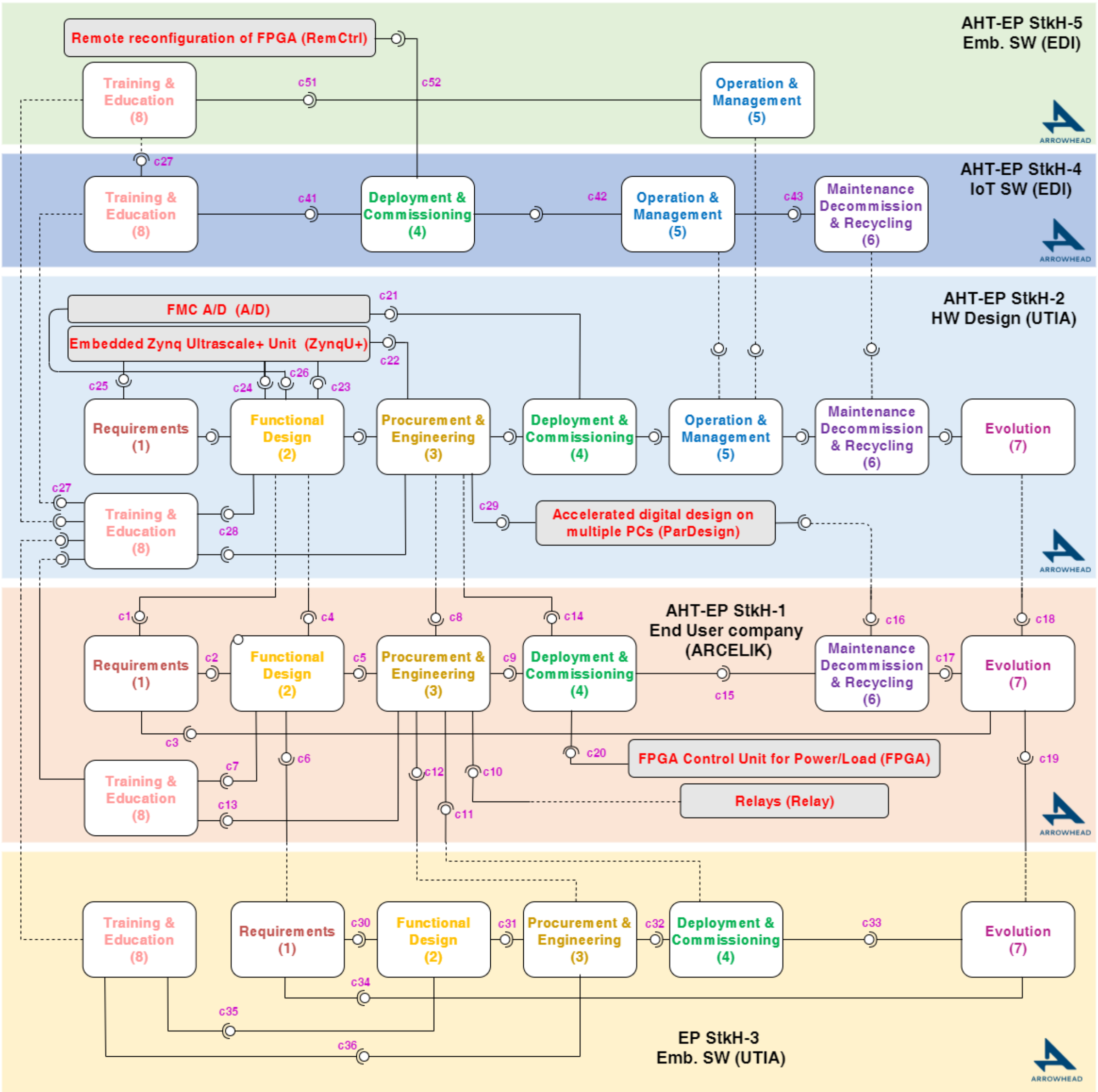


Figure 38 AHT-EP of use case UC-10

UTIA will develop for the use case the unified configuration and project-bringup-scripts to address industrial modules fitted with variable sizes of Zynq Ultrascale devices. These scripts will support maintenance and support upgrade of commercial tool chain (Xilinx Vivado HLS) and will support HW upgrades at level of Zynq Ultrascale+ module and upgrades of FMC data acquisition adapter cards in the design phase and on site testing phases.
UTIA will mainly support the design and early testing phases. The long term maintenance of products is not supported by UTIA in the frame of the project.

The configuration and project-bring-up scripts developed in UTIA to address industrial modules fitted with variable sizes of Zynq Ultrascale devices will simplify the process of selecting of an optimal industrial module and device. Design process will start on larger device. The finally

released system can be simply re-targeted to industrial module with minimal required size of device and memory.
We will use AH framework in two contexts:

- It will support automated connection of the chain of our design your tools on multiple computers in the frame of a single AH local cloud. This toolchain will support definition of the sequence of tool executions to create store and document all what is needed for the embedded SoC system design needed for the Arcelik use case. This will be the use of AH framework in the design time.
- The embedded SoC system will be compatible with the AH framework. The AH framework will helps in the deployment phase with data exchange to remote clients etc. This will be the use of AH framework in the runtime.

At this stage of development we move from single PC manually operated design flow to the AH framework supported design time flow utilizing several computers.
We are working on transformation of the scheduling of compilation stages (execution of .tcl or make scripts calling the tools) defined up to now for a single computer to AH framework based on several PCs with AH framework synchronization.
This would result in faster (shorter) execution of the entire design flow due to (for example) multiple Vivado HW compilation runs executed on several possibly remote servers.
This would also enable to get in one batch the actually used solution not only for the currently deployed specific Zynq Ultrascale+ industrial module TE0808-06EG-1E but also in parallel solution for another industrial modules like TE0808-09EG-1E and TE0808-09EG-1I and TE0808-15EG-1E with practically same compilation time, if the other 3 would be compiled in parallel on another 3 PC servers.
The AH framework automation will take care for the collection of tool execution logs with possible issues and performance parameters etc. needed for "certification" or "reproducing" or for "bug identification".
The embedded SoC system is already compatible with the AH framework ver 4.1.3. It can use the framework for secure run-time data of measured data for SW clients running on PCs connected in the local cloud.
In the present stage of development we generate sinus wave with variable amplitude. We sample it with 1.0 Gsample/second speed and perform measurement of the actually measured peak voltage. This information is provided by the run-time C++ client to the AH Framework 4.1.3. The edge device is the Xilinx Zynq Ultrascale+ device, running Debian Linux OS with user application capable to measure the 1.0 Gsample/second via the Analog Device FMC data acquisition card supported by the HW DMA system present in the programmable part of the Zynq device.
EDI TestBed v2 (iteration upgrade) is in the development process. The software MVP with the basic functionality in the form of CLI is already made available. The hardware prototype for the v2 is in the prototyping stage nearing the final testing.

We plan to focus on use of the AH-Framework in the Deployment & Commissioning phase, and in the Operation & Management phase where the benefits are largest. In Y3 we will also extend to use it in support for Training & Education.

The existing Xilinx Vivado HLS is targeting one concrete Zynq Ultrascale + device and the design flow can run only on a single PC.
The configuration and project-bring-up scripts can target complete families of industrial modules with Zynq Ultrascale+ modules with different size, temperature grades, memory size etc. Automation of the execution of the sequence of Xilinx tools by AH framework will add the opportunity to perform multiple potentially very long HW compilation steps on several PCs in the local cloud for several HW configurations.

EDI TestBed is scaled up to 100 workstations at the moment and the possible scaling capability is not tested or known. The limit comes purely from the network bandwidth capabilities of the central server.

The existing Xilinx Vivado can target only single configuration of a Zynq Ultrascale + device and the design flow can run only on a single PC.
Combination of AH framework with database of configuration and project-bring-up scripts will be able to autonomously execute sequence of compilation runs for complete families of industrial modules with Zynq Ultrascale+ modules with different size, temperature grades, memory size etc. on several PCs in the local cloud for several HW configurations.

Table 55 AHT-EP Phase focus of UC-10

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: StkH 2 acquire the specification describing the measurement system from the vendor (StkH 1). Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the design of the ZynqU+. Requirements will be listed in a Word file. 2. *FMC A/D (A/D)*: StkH2 will analyse existing design templates provided by company Analog devices for Xilinx University boards. 3. *Relays (Relay)*: StkH1 will specify relay system needed for switching of analog measurement points. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH 4 acquire the specification describing the FPGA from the vendor (StkH 1). Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the design of the FPGA. Requirements will be listed in a Word file. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH 4 acquire the specification describing the RemCtrl requirements from the vendor (StkH 1). Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the design of the FPGA RemCtrl. Requirements will be listed in a Word file. 6. *Accelerated digital design on multiple PCs (ParDesign)*: StkH2 will aquire specification for maximal HW design compilation times related to modifications and upgrades ZynqU+ programmable logic designs for single PC installation and requirements for their acceleration on several PCs from StkH1. Specification documents are passed in the form of PDF files that are analysed by the engineering team that generate a list of requirements for the ParDesign of the ZynqU+. Requirements will be listed in a Word file. **Tools** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: Elicitate the requirements for the ZynqU+. • Input: <StkH1 - EPP2> specifications of the electromechanical part defined by the functional design team of the vendor measurement site (StkH 1). • Input: <StkH2 - EPP7> requirements are integrated with analysis performed in the evolution phase using the remote data collected from operating measurement site of StkH 1. • Output: <StkH2 - EPP2> requirements for the functional design of the ZunqU+. 2. *FMC A/D (A/D)*: Elicitate the requirements for the A/D. • Input: <StkH1 - EPP2> specifications of the sensors installed on the vendor measurement site. Legislation of measurement safety for high voltages will be evaluated with StkH1.

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>

Input: <StkH2 - EPP7> requirements are integrated with analysis performed in the evolution phase using the remote data collected from operating measurement site.
Output: <StkH2 - EPP2> requirements for the functional design of the A/D.

3. Relays (Relay): StkH1: is responsible for the technical requirements related to Relay switching of measured analogue signals.

Input: <StkH1 - EPP2> technical aspects of the measurement site functionalities required to manage/switch multiple analogue high voltage measurement signals.
Input: <StkH1 - EPP7> requirements are integrated with analysis performed in the evolution phase using the remote data collected from operating measurement site.
Output: <StkH2 - EPP3> requirements for the engineering team.

4. FPGA Control Unit for Power/Load (FPGA): STKH 1, StkH 4 and StkH 5 engineers will discuss FPGA control unit for control of Power/Load features to be offered on the user and installer application.

Input: <StkH2 - EPP2> functional models of Power/Load units and API interfaces.
Input: <StkH3 - EPP7> requirements are integrated with analysis performed in the evolution phase analysing the reports of requested updates.
Output: <StkH4 - EPP2> requirements for the development/update the user application.

5. Remote reconfiguration of FPGA (RemCtrl): StkH1 collect ElectroMechanical requirements and analyse the solution available on the market for include new intelligent features.

Input: <StkH1-UEP> report collecting: 1) requirements of the measurement site, 2) legislation law related to safety in case of high voltage measurements, 3) features identifyed as necessary from the team working in the evolution phase.
Output: <StkH1 - EPP2> requirements for the development/update the electromechanical framework at the measurement site.

6. Accelerated digital design on multiple PCs (ParDesign): StkH2 collect requirements on time needed to recompile HW/SW design of the complete ZynqU+ system.

Input: <StkH2-UEP> report collecting: 1) requirements of the design compilation local cloud with multiple connected PCs, 2) features identified as necessary from the team working in the evolution phase.
Output: <StkH2 - EPP2> requirements for the development/updates of the ZynqU+ HW redesign and I/Os framework at the measurement site.

</td></tr>
<tr><td>Functional design</td><td>
Activity
1. Embedded Zynq Ultrascale+ Unit (ZynqU+): StkH 2 will design and simulate the HW part of the ZynqU+ interfaces in Xilinx Vivado simulation tool xsim.
2. FMC A/D (A/D): StkH2 will model design templates ported to select ZynqU+ module and carrier and modell it in Xilinx Vivado simulation tool xsim.
3. Relays (Relay): StkH1 will model in Matlab isolation properties of the Relay in respect to isolation of high voltage environment.
4. FPGA Control Unit for Power/Load (FPGA): StkH 4 wil design and simulate the HW part of the FPGA interfaces in Modelsim.
5. Remote reconfiguration of FPGA (RemCtrl): StkH 4 will design and simulate the remote communication part of the RemCtrl interfaces in Matlab.
6. Accelerated digital design on multiple PCs (ParDesign): StkH 2 will design and simulate the parallel compilation HW part of the ZynqU+ interfaces in Matlab parallel toolbox.
Tools
1. Embedded Zynq Ultrascale+ Unit (ZynqU+): Design and simulation of the cyber physical system model of the measurement site (measured power-supply units) in matlab.
</td></tr>
</table>

Engineering process phase	Addressed/Focus
	• Input: <StkH1 - EPP2> specifications of the working parameters of the measured power suply units defined by the functional design team of the vendor (StkH 1). • Input: <StkH2 - EPP1> requirements. • Output: <StkH2 - EPP3> model of the measurement system to be engineered. • Output: <StkH2 - EPP8> documentation about the model of the power-supply units to be elaborated and provided to the vendor. • Output: <StkH2 - EPP1> details of the model for extracting the specifications and requirements for the user application. 2. *FMC A/D (A/D)*: StkH 2 will develop a model of the A/D data acquisition FMC unit. The model will be validated on surrogate data generated by D/A converters on prototype working in a controlled environment for evaluate the A/D precison at the 1G samples/s speed. • Input: <StkH2 - EPP2> dataset extracted from the D/A output of the prototype working in a controlled environment (StkH 2). • Input: <StkH2 - EPP1> requirements for the A/D board. The A/D has to support 1 G sample/s. • Output: <StkH2 - EPP3> model of the HW DMA interface of the A/D to be engineered. • Output: <StkH2 - EPP8> documentation about the model to be elaborated and provided to the vendor. 3. *Relays (Relay)*: n.s. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH 4 will develop/update a mock based on the platform developed for the remote control and management of the FPGA. • Input: <StkH4 - EPP1> requirements of defined for the FPGA. • Output: <StkH5 - EPP3> mock of the user interface and model of the application to be engineered. • Output: <StkH4 - EPP8> documentation about the model to be elaborated and provided to the vendor. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH 1 design and simulation of the system. • Input: <StkH1 - EPP1> requirements of the measurement site remote reconfiguration. • Input: <StkH1 - UEP> report of good practices from teams involved in the design of previous versions of the measurement site. • Output: <StkH1 - EPP1> list of functional requirements to be provided to the StkH 4. • Output: <StkH4 - EPP2> draws of the model for the development of a virtualised version of the measurement site to be used in the model developed by StkH 4. • Output: <StkH1 - EPP3> draws and technical document of the model to be engineered. • Output: <StkH1 - UEP> draws and technical document of the model to be validated by experts in the development of electromechanical parts of boiler. • Output: <StkH1 - EPP8> documentation about the model. 6. *Accelerated digital design on multiple PCs (ParDesign)*: n.s.
Procurement & Engineering	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: StkH 2 will specify the of-the-shelf for the ZynqU+ module and carrier PCBs Develop the firmware for Arm and related to PL logic interfaces to A/D using Xilinx GCC and Vivado toolchains. Request quotes from the supplier for the electronic and sensor components selected in the PCB design process. The engineering team will develop the documentation of the HW and SW developed for the ZynqU+ that will be provided as a reference manual to the StkH 1. A working prototype of the ZynqU+ will be tested by StkH1 in the vendor laboratories. 2. *FMC A/D (A/D)*: StkH1 and Stk2 will buy identical A/D data acquisition card for

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
the Zynq U+ system.
3. Relays (Relay): StkH 1 will design the Relay PCB with Altium Designer and select the components and technologies for the PCB design. Develop the firmware for drive sensors and actuators using STM32 GCC toolchain for STM32 Nucleo controller sub-system.
4. FPGA Control Unit for Power/Load (FPGA): StkH 4 will specify the of-the-shelf for the FPGA system module and carrier PCB. StkH5 will develop the firmware for MicroBlaze softcore and HW interfaces to RS232 serial interfaces using Xilinx GCC and Vivado toolchains. Request quotes from the supplier for the electronic and sensor components selected in the PCB design process. The Ethernet is embedded on the platform for supporting the services offered by the telecommunication provider (StkH 4). The engineering team will develop the documentation of the HW and SW developed for the FPGA that will be provided as a reference manual to the StkH 1. A working prototype of the FPGA will be tested in a real machine for quality test in the vendor laboratories.
5. Remote reconfiguration of FPGA (RemCtrl): StkH 4 will specify the of-the-shelf for the RemCtrl for FPGA. StkH5 will develop the firmware for MicroBlaze softcore and HW interfaces to RS232 serial interfaces using Xilinx GCC and Vivado toolchains. Request quotes from the supplier for the electronic and sensor components selected in the PCB design process. The Ethernet is embedded on the platform for supporting the services offered by the telecommunication provider (StkH 4). The engineering team will develop the documentation of the HW and SW developed for the FPGA that will be provided as a reference manual to the StkH 1. A working prototype of the FPGA will be tested in a real machine for quality test in the vendor laboratories."
6. Accelerated digital design on multiple PCs (ParDesign): StkH2 will specify used design tool sequence and define Pardesign compilation schedule acceleration on several PCs.

Tools

1. Embedded Zynq Ultrascale+ Unit (ZynqU+): StkH 2 design the PCB of the ECU and the firmware.
• Input: <StkH1 - EPP3> prototype of the electromechanical system for creating a full prototype.
• Input: <StkH2 - EPP2> model simulated and validated of the cyber physical system.
• Input: <StkH3 - UEP> The telecommunication provider provides the specification of the telecommunication module to be installed on the ECU system. The modules are acquired and installed on the PCB.
• Output: <StkH2 - EPP8> The engineering team will develop the documentation of the HW and SW developed for the ECU that will be provided as a reference manual to the StkH1.
• Output: <StkH2 - EPP4> The ECU project is passed to the deployment team that will trigger the production line.
• Output: <StkH1 - EPP3> A working prototype of the ECU will be tested in a real machine for quality test in the vendor laboratories.
2. FMC A/D (A/D): StkH2 will implement the HOA model in C++ for being executed on the ECU. A set of interfaces will make available the customisation of selected parameters from external tools that will use the available API. A first release of the SW will be tested on the ECU prototype working on the real condition. A Continuous Integration approach is adopted for the development of the core and following updates.
• Input: <StkH1 - EPP3> prototype of the electromechanical system integrated with the ECU is used for creating a full prototype for testing the algorithm in a real scenario
• Input: <StkH2 - EPP2> model simulated and validated
• Output: <StkH2 - EPP8> The engineering team will develop the documentation of the HOA algorithm that will be provided as a reference manual to the StkH1.
• Output: <StkH2 - EPP4> The algorithm is passed to the deployment team
</td></tr>
</table>

Engineering process phase	Addressed/Focus
	that will compile and upload in the ECU on the production line. 3. *Relays (Relay)*: StkH2: Design and development of the IoT integration framework • Input: <StkH1 - EPP3> architecture of the server system which will host the IoT framework. • Input: <StkH2 - EPP1> requirements. • Output: <StkH2 - EPP8> The engineering team will develop the documentation of the IoT framework that will be provided as a reference manual to the StkH1. • Output: <StkH2 - EPP4> The IoT framework is passed to the deployment team that will install and set-up on the vendor servers. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH4 will customise the proprietary platform that allows an easy generation of applications for web and mobile devices. • Input: <StkH2 - EPP3> architecture and APIs of the IoT framework. • Input: <StkH4- EPP2> mock and model. • Output: <StkH4 - EPP8> The engineering team will develop the documentation of the User application that will be provided as a reference manual to the StkH1. • Output: <StkH4 - EPP4> The user application is passed to the deployment team that will deploy to the StkH2. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH1 will engineer the simulated design and create a list of components to be quoted by the providers. • Input: <StkH1 - EPP2> draws and technical document of the electromechanical model. • Input: <StkH1 - UEP> engineering reports of the teams involved in the design of previous versions of boiler. • Input: <StkH2 - EPP4> ECU for the development of a full prototype with standard sensors and actuators for test in working scenarios. • Output: <StkH2 - EPP3> preliminary prototype (with high precision sensing and testing system) for checking fluid dynamics and mechanical satisfaction of the requirements. • Output: <StkH1 - EPP4> Quotations of the components and engineered draw of the system integrated with ECU which hold an interface of the IoT framework. 6. *Accelerated digital design on multiple PCs (ParDesign)*: STkH2 collect requirements on time needed to recompile HW/SW design of the complete ZynqU+ system. • Input: <StkH2-UEP> report collecting: 1) requirements of the design compilation local cloud with multiple connected PCs, 2) features identified as necessary.
Deployment & Commissioning	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: StkH2 and Stkh3 produce the ZynqU+ SW firmware and HW design and supply them to the vendor StkH1. StkH 3 will utilize embedded AH framework SW C++ clients running on the embedded Debian Linux for communication of the ZynqU+ with remote control/visualisation PC clients. 2. *FMC A/D (A/D)*: StkH2 produce the ZynqU+ SW firmware and HW design to control A/D and supply them to the vendor StkH1. 3. *Relays (Relay)*: StkH1 will produce the Relay SW firmware and HW design and supply them to the vendor StkH2 and StkH5. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH4 and StkH5 produce the FPGA SW firmware for MicroBlaze and HW design and supply them to the vendor StkH1. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH4 and StkH5 produce the RemCtrl SW and HW and supply them to the vendor StkH1. 6. *Accelerated digital design on multiple PCs (ParDesign)*: StkH 2 will utilize AH

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
framework for control of execution of tools in case of use of ParDesign.

Tools

1. Embedded Zynq Ultrascale+ Unit (ZynqU+): StkH2 produce the ECU and supply them to the vendor product lines.
• Input: <StkH2 - EPP3> Technical sheets and cad documents of the PCB to be produced in series.
• Output: <StkH1 - EPP3> Deployed and produce pieces with relative serial numbers.
• Output: <StkH2 - EPP6> Technical materials and serial number lists of the ECU for maintenance.
2. FMC A/D (A/D): StkH2 compile and install the program on the ECU before to be supplied to the StkH1 product lines.
• Input: <StkH2 - EPP3> C++ code to be compiled and installed on the produced ECU.
• Output: <StkH1 - EPP3> Deployed the code on the produced pieces.
3. Relays (Relay):
• Input: <StkH2 - EPP3> IoT framework prototype to be Deployment and pre-commission test.
• Output: <StkH1 - EPP3> instance of the IoT integration framework installed on the StkH1 servers.
4. FPGA Control Unit for Power/Load (FPGA): StkH 4 will deploy a web interface to be embedded on the vendor website and two version of mobile applications for google play store and apple store.
• Input: <StkH4 - EPP3> implementation of the user application.
• Output: <StkH2 - EPP3> deployed user application.
• Output: <StkH4 - EPP7> documentation of the application developed.
5. Remote reconfiguration of FPGA (RemCtrl): StkH1 will set-up production lines for the measurement of power supply units with powe supply and load controlled by FPGA.
• Input: <StkH1 - EPP3> Quotations of the components and engineered draw of the system integrated with measurement site which hold an interface of the IoT framework.
• Output: <StkH5 - EPP4> Product to be installed in the house of the customer
• Output: <StkH1 - EPP5> Serial number of the sell product associated to the technician and final user for starting the collection of data during the lifecycle of the product. Moreover, the IoT framework configured by the engineering teams of StkH2 and STkH1 is installed on the StkH1 servers and cloud services.
6. Accelerated digital design on multiple PCs (ParDesign): StkH 2 will set-up set of PCs with installed tools needed for design and configuration of the ZynqU+.
• Input: <StkH2 - EPP3> Quotations of the components and engineered draw of the local cloud of PCs IoT framework. Output: <StkH5 - EPP4> Product to be installed in the house of the customer.
• Output: <StkH2 - EPP5> The IoT framework configured by the engineering teams of StkH2 and is installed on the StkH1 PCs.
</td></tr>
<tr><td>Operations & Management</td><td>
Activity

1. Embedded Zynq Ultrascale+ Unit (ZynqU+): n.s.
2. FMC A/D (A/D): n.s.
3. Relays (Relay): n.s.
4. FPGA Control Unit for Power/Load (FPGA): n.s.
5. Remote reconfiguration of FPGA (RemCtrl): n.s.
6. Accelerated digital design on multiple PCs (ParDesign): n.s.

Tools

1. Embedded Zynq Ultrascale+ Unit (ZynqU+): n.s.
2. FMC A/D (A/D): n.s.
3. Relays (Relay): Input: <StkH2 - EPP4> Technical documents and reports of the
</td></tr>
</table>

Engineering process phase	Addressed/Focus
	IoT framework instantiated on the private servers of the vendor that will implement the active monitoring of the fleets. • Input: <StkH1 - EPP5> Report of the status of the IoT framework. • Output: <StkH1 - EPP5> Consulting for keep the monitoring system working in the correct conditions. • Output: <StkH2 - EPP6> Pass the maintenance information to the Maintenance phase and aggregated data to the evolution phase. 4. *FPGA Control Unit for Power/Load (FPGA)*: n.s. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH4 produce the IoT framework as a tool used to monitor a FPGA by RemCtrl technology instantiated on the StkH1 measurement site. • Input: <StkH1 - EPP4> The acquired IoT framework form StkH 4 are instantiated on the measurement site of the vendor that will implement the active remote reconfiguration of FPGA by RemCtrl technology from StkH 4. • Input: <StkH4 - EPP5> Report of the settings and installation details provided by the technician during the control on the operation site. • Input: <StkH4 - EPP5> Consulting on the utilisation of the IoT framework for collecting operative data. • Input: <StkH5 - EPP5> Parameters and feedbacks are collected from the reconfigurable FPGA and the measurement site. • Output: <StkH5 - EPP5> Trigger for the technician to open a maintenance session. • Output: <StkH5 - EPP5> Report of the status of the IoT framework. • Output: <StkH1 - EPP6> Pass the maintenance information to the Maintenance phase and aggregated data to the evolution phase. • Output: <StkH5 - EPP5> Set-up remotely the FPGA parameters. 6. *Accelerated digital design on multiple PCs (ParDesign)*: n.s
Maintenance Decommissioning & Recycling	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: n.s. 2. *FMC A/D (A/D)*: n.s. 3. *Relays (Relay)*: n.s. 4. *FPGA Control Unit for Power/Load (FPGA)*: n.s. 5. *Remote reconfiguration of FPGA (RemCtrl)*: n.s. 6. *Accelerated digital design on multiple PCs (ParDesign)*: n.s. **Tools** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: In case of faulty ZynaU+ the StkH2 make the diagnosis by analysing the log files collected remotely in the operative measurement site. Eventually request to the vendor to receive back the faulty board for a deeper investigation. • Input: <StkH2 - EPP4> Technical materials and serial number lists of the ZynqU+ for maintenance. • Input: <StkH1 - EPP6> Log file acquired remotely and eventually the board substituted in the faulty ZynqU+. • Output: <StkH2 - EPP7> Technical materials and serial number lists and updated status of the maintenance events. Log files collected in Operation & Management phase of StkH 1 are propagated. 2. *FMC A/D (A/D)*: n.s. 3. *Relays (Relay)*: • Input: <StkH2 - EPP4> draws and documents of the ECU and HOA are accessible to be shared with Maintenance and Evolution teams. • Output: <StkH2 - EPP7> information collected during the maintenance intervention and documents of ECU and HOA. • Output: <StkH1 - EPP6> maintenance and update of the IoT framework instantiated on the StkH1 servers. 4. *FPGA Control Unit for Power/Load (FPGA)*: n.s. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH1 identify in the operation and

Engineering process phase	Addressed/Focus
	management phase anomalous behaviour of the measurement site and trigger the related maintenance activities reconfiguring the FPGA. It uses the tool to manage also the decommissioning and recycling activities of the measurement system when the final user uninstall it. • Input: <StkH1 - EPP5> report on the anomaly detected during the monitoring of the operative conditions. • Input: <StkH5 - EPP6> report on the interventions on the field. • Output: <StkH4 - EPP6> StkH1 send request of maintenance of the IoT framework when dysfunctions are detected. • Output: <StkH2 - EPP7> Aggregated data to be analysed for improving the project of the measurement site. 6. *Accelerated digital design on multiple PCs (ParDesign)*: n.s.
Evolution	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: Data collected in the Operation & Management phase of StkH 1 are analysed by StkH2 for identifying possible bottlenecks in the ZynqU+ design. Moreover, firmware updates and HW design updates will be generated by StkH2 as needed by experiments. 2. *FMC A/D (A/D)*: StkH2 produce the Linux version of ZynqU+ SW firmware and HW design to control A/D and supply them to the vendor StkH1. 3. *Relays (Relay)*: StkH1 will produce the Relay SW firmware and HW design updates and suply them to the vendor StkH2 and StkH5. 4. *FPGA Control Unit for Power/Load (FPGA)*: Data collected in the Operation & Management phase of StkH 1 are analysed by StkH5 for identifying possible bottlenecks in the FPGA design. Moreover, firmware updates and HW design updates will be generated by StkH5 as needed by experiments. 5. *Remote reconfiguration of FPGA (RemCtrl)*: Data collected in the Operation & Management phase of StkH 1 are analysed by StkH4 for identifying possible bottlenecks in the RemCtrl SW/HW. Moreover, firmware updates and HW design updates for RemCtrl will be generated by StkH4 and StkH5 as needed by experiments. 6. *Accelerated digital design on multiple PCs (ParDesign)*: Data related to recompilation of HW collected in the Operation & Management phase of StkH 1 will be analysed by StkH2 for identifying possible bottlenecks in the ParDesign. Moreover, Arrowhead firmware updates related to accelerated design scheduling will be generated by StkH2 as needed by experiments. **Tools** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: Firmware and HW design updates will be generated by StkH2 for keep a good standard of functionality and also cybersecurity for ZynqU+. Collected information will be used for the next HW/SW re-designs. • Input: <StkH2 - EPP6> Technical materials and serial number lists and updated status of the maintenance events. • Input: <StkH1 - EPP7> Information about the evolution of the ZynqU+ mechanical system are evaluated for identifying possible bottlenecks in the ZynaU+ design. Log files collected in Operation & Management phase of StkH 1 are propagated. • Output: <StkH2 - EPP1> Analysis performed in this phase will be used to update the requirements and trigger the continuous engineering loop. 2. *FMC A/D (A/D)*: • Input: <StkH1 - EPP7> Log files collected in Operation & Management phase of StkH 1 are propagated and analysed. • Output: <StkH2 - EPP1> Analysis performed in this phase will be used to update the requirements and trigger the continuous engineering loop. 3. *Relays (Relay)*: StkH2: identification of potential updates of existing IoT integration framework, of new releases with significant improvements, etc. No tools currently support this phase. • Input: <StkH1 - EPP7> StkH1 send specification of the OS and updates of

Engineering process phase	Addressed/Focus
	the infrastructure implemented to run the instance of the remote monitor system. Information about other updates relevant for the IoT framework are shared. • Output: <StkH2 - EPP1> technical documents of the evolved system are passed to the requirement team. • Output: <StkH4 - EPP7> evolved features of the IoT framework are transmitted to the StkH4 that will adapt the UA if necessary. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH4 will guarantee security and functional updates of the mobile applications for 10 years. • Input: <StkH2 - EPP7> technical document of the evolved IoT framework API. • Input: <StkH4 - EPP7> updated technical document of the deployed application. • Output: <StkH4 - EPP1> report on the new features to be considered in the requirement update that will trigger the development of a new version of the application. 5. *Remote reconfiguration of FPGA (RemCtrl)*: • Input: <StkH1 - UEP> engineering reports of the teams involved in the design of previous versions of boiler. • Input: <StkH1 - EEP6> Aggregated data to be analysed for improving the project of the product. • Output: <StkH1 - EPP1> report on the new features to be considered in the requirement update that will trigger the development of a new version of the system. • Output: <StkH2 - EPP7> collected data for evolving new versions of all the HW/SW components of the product. 6. *Accelerated digital design on multiple PCs (ParDesign)*: n.s.
Training & Education	**Activity** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: StkH2 and StkH3 will create the datasheet of ZynqU+ system and reference manual of low level SW API for A53 CPU running Debian embedded Linux OS. 2. *FMC A/D (A/D)*: StkH2 create the datasheet of A/D integration in ZynqU+ and reference manual of low level SW API for A53 CPU running Debian embedded Linux OS. 3. *Relays (Relay)*: StkH1 create the datasheet of Relay iand reference manual of low level SW API for STM32 MPU. 4. *FPGA Control Unit for Power/Load (FPGA)*: StkH4 and StkH5 will create the datasheet of FPGA system and reference manual of low level SW API for MicroBlazs soft core running in FPGA. 5. *Remote reconfiguration of FPGA (RemCtrl)*: StkH4 and StkH5 will create the datasheet of RemCtrl system and reference manual of low level communication protocol description. 6. *Accelerated digital design on multiple PCs (ParDesign)*: StkH3 will create the datasheet of Design acceleration flow and reference manual describing configuration of AH framework for ParDesign acceleration on multiple PCs in a local cloud for StkH 1. **Tools** 1. *Embedded Zynq Ultrascale+ Unit (ZynqU+)*: StkH2 create the datasheet of ZynaU+ and reference manual of low level API. • Input: <StkH2 - EPP2> functional description of the cyber physical system. • Input: <StkH2 - EPP3> technical document of the engineered HW design and SW firmware with datasheets and manuals. • Output: <StkH1 - EPP8> aggregated and corrected documents are provided to the ZynqU+ user (StkH1). 2. *FMC A/D (A/D)*: StkH2 will document the code to be shared with the StkH1. • Input: <StkH2 - EPP2> functional description of the A/D card. • Input: <StkH2 - EPP3> technical document of the engineered code and

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
used libraries.
• Output: <StkH1 - EPP8> aggregated and corrected documents are provided to the A/D user (StkH1).
3. Relays (Relay):
4. Input: <StkH2 - EPP3> documented Altium Designer schematic design and source code for the STM32 MCU used in the Relay unit.
• Output: <StkH1 - EPP8> aggregated and corrected documents are provided to the Relay vendor (StkH1).
• FPGA Control Unit for Power/Load (FPGA): StkH4 will develop video tutorial and interactive tutorial.
• Input: <StkH4 - EPP2> documentation about the model to be elaborated for producing the video and interactive tutorials.
• Input: <StkH4 - EPP3> documentation of the application with reference to the used IoT framework APIs.
• Output: <StkH1 - EPP8> video tutorial, interactive tutorial and aggregated documents are provided to the FPGA user (StkH1).
5. Remote reconfiguration of FPGA (RemCtrl): StkH1 will aggregate all the material and distribute develop video tutorial, interactive tutorial, user manuals, and installation manuals.
• Input: <StkH1 - EPP2> documentation about the measurement site model.
• Input: <StkH1 - EPP3> documentation of the measurement site components and schematics.
• Input: <StkH2 - EPP8> documentation of the measurement site and IoT framework components and schematics.
• Input: <StkH4 - EPP8> documentation of the FPGA component, video and interactive tutorial.
• Output: <StkH5 - EPP8> video tutorial, interactive tutorial, user manual, training session of installation and maintenance techniques, installation manual.
• Output: <StkH6 - EPP8> video tutorial, interactive tutorial and user manual.
6. Accelerated digital design on multiple PCs (ParDesign): StkH 2 will aggregate material and distribute develop video tutorial, interactive tutorial, user manuals, and installation manuals related to parallel accelerated HW compilation controlled by Arrowhead framework.
• Input: <StkH2 - EPP2> documentation about the parallel compilation synchronisation and scheduling.
• Input: <StkH2 - EPP8> documentation of the IoT framework components.
• Output: <StkH2 - EPP8> video tutorial, interactive tutorial, user manual, training session of installation and maintenance techniques, installation manual.
</td></tr>
</table>

10.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 56 UC-10 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	This objective will be matched in the use-case that solution will go from laboratory environment to deploy in Arcelik Industry environment and data will be collected in the field (phase 5 of StkH1 AHT-EPP). And the collected data will be evaluated by stakeholders for further improvement opportunities. In the evolution phase, the project input/output and the aims may be revised and feed into the requirements.
Obj. 2 - The move from single to	In this UC we represent each of the actors involved in the product life cycle as a stakeholder. All the engineering phases are intended for a single stakeholder (StkH

WP2 Objective	Focus & Planed actions
integrated multi stakeholder automation and digitalization	1) that will produce one component. All the EPs are integrated and connected between each other internally and externally (different StkH). Connections are not fully automated and several of them are manual. During the project we will automatize some of these connections by interfacing the modules with and without the AHT Framework.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Several phases of the EPs have multiple interactions with the other phases of the same EP and EPs from other stakeholders. Interactions, represented in the AHT-EP model with connection lines, are implemented by exchanging information between tools contained and operating in each of the AHT-EP phases. This exchange of information is made automatically for some phases which pass data files, as usually done between phases 1-2, or manually as in the case of many of the connections between phase 1-2 and 7-1. Each phase of the EP related to the StkH3 includes the toolchains for the five different components (Ahtool Framework Service Provider API, DAQ Board, RelayNetwork, ZynqModule and IoT Integration Framework)
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Each AHT-EP considers the development of documentation and reference manuals. The StkH 3, which will develop the user application, is in charge of producing application notes and user manuals for operators in the industry environment (StkH 1). Also, the operator (StkH 1) is in charge of the operation manual that describes the operation and the maintenance of the end product.

Table 57 UC-10 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	There are two parts of the use case that contribute to the reduction of the engineering costs: • Automatic electronic validation of power supply units. • Automated data collecting and reporting of test results
Obj. 2 - Interoperability for IoT and SoS engineering tools	We have reached compatibility with AH Framework 4.1.3 with secure communication (Zynq C/C++ SW clients).
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	We have reached compatibility with AH Framework 4.1.3 with secure communication (Zynq C/C++ SW clients). And locally we will use legacy connectivity (serial communication) which will communicate with the microcontroller based control and measurement devices/tools.
Obj. 4 - Integration platform interoperability with emerging digitalization and	The process of HW compilation will be using AH Framework to synchronize and schedule Vivado HW design Runs on multiple PCs on local cloud.

Project Objective	Focus & Planed actions
automation framework	
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	We have reached compatibility with AH Framework Services in Secure Mode.
Obj. 6 - Training material (HW and SW) for professional engineers	We will consider the internal application notes used by UTIA, EDI and ARCELIK as candidates for public training material.

10.4 Engineering Process analysis

Table 58 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-10.

AHT-EPP	**1 (R)**	**2 (FD)**	**3 (PE)**	**4 (DC)**	**5 (OM)**	**6 (M)**	**7 (E)**	**8 (TE)**	**Unknown**
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X	X		X	X	X	
OBJ-AHT #2		X	X						
OBJ-AHT #3		X	X	X					
OBJ-AHT #4			X	X	X				
OBJ-AHT #5			X	X	X	X			
OBJ-AHT #6		X	X	X				X	
OBJ-WP2 #1		X	X	X			X		
OBJ-WP2 #2		X	X	X				X	
OBJ-WP2 #3		X	X	X				X	
OBJ-WP2 #4		X	X	X				X	

11. UC-11 Configuration tool for autonomous provisioning of local clouds (DAC)

The use case aims at the simplification of the on boarding process for IoT devices, taking into account the distributed characteristics of the envisioned system under test and wireless communication for the data exchange.
Although the envisioned system is designed to be flexible in terms of areas of application, the final demonstration will be presented for a logistics use case where the deployment of new

sensors to existing local clouds will be maximally simplified and automatized, and the solution will be scalable at the same time.

At M0, there is no automation of the on boarding process and new devices have to be connected to a system and configured manually.

It should be noted that we differentiate two phases - the development of tools/components and the utilization of these tools in the use case execution/demonstration. Eclipse Arrowhead will be used to reduce the engineering costs of the use case execution only.

11.1 Overall description of the UC-EP

At M0, the envisioned system has to be integrated and configured manually by a qualified system integrator on site (e.g. in a factory or in a delivery truck). In case of any fault or failure of IoT devices, or in case a reconfiguration is needed, again the integrator is required on site. The envisioned use case at M36 (Figure 39) in terms of the engineering activities related to the use case execution could be summarized as below:

1. Configure and run core services on a field gateway (deploy local cloud).
2. Add a new AHT-compliant device (authorize) to the local cloud registry through
    a. the on boarding application and cloud management infrastructure OR
    b. through NFC of the field gateway.
3. Confirm the AHT-compliant device is trying to connect to correct AH Local Cloud and authenticate the device after it is physically connected or powered on.
4. Assess the performance of the local cloud.
5. Optionally deploy new devices to particular field gateways.

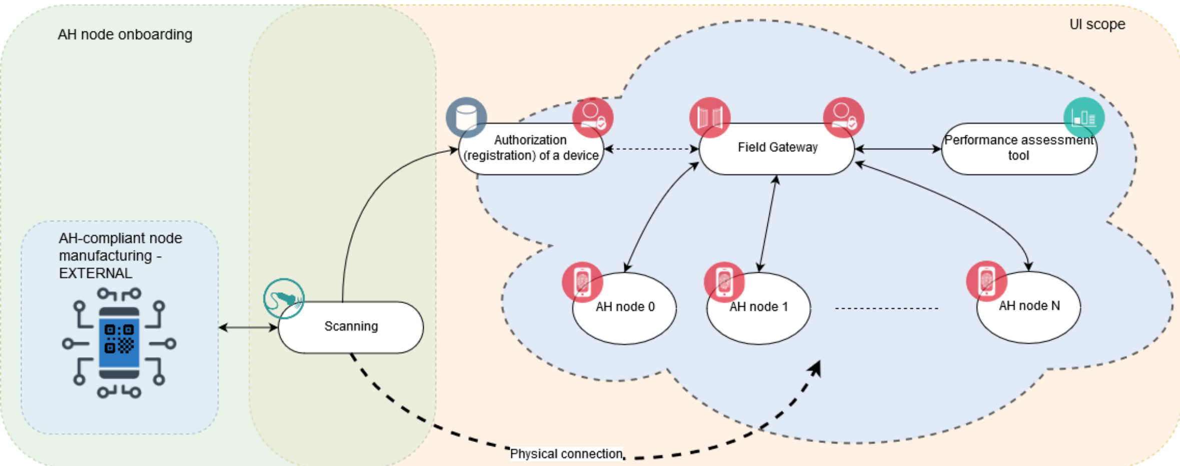


Figure 39 UC-11 the envisioned system at M36.

The Onboarding toolchain (M36) can be summarized as shown in Figure 40

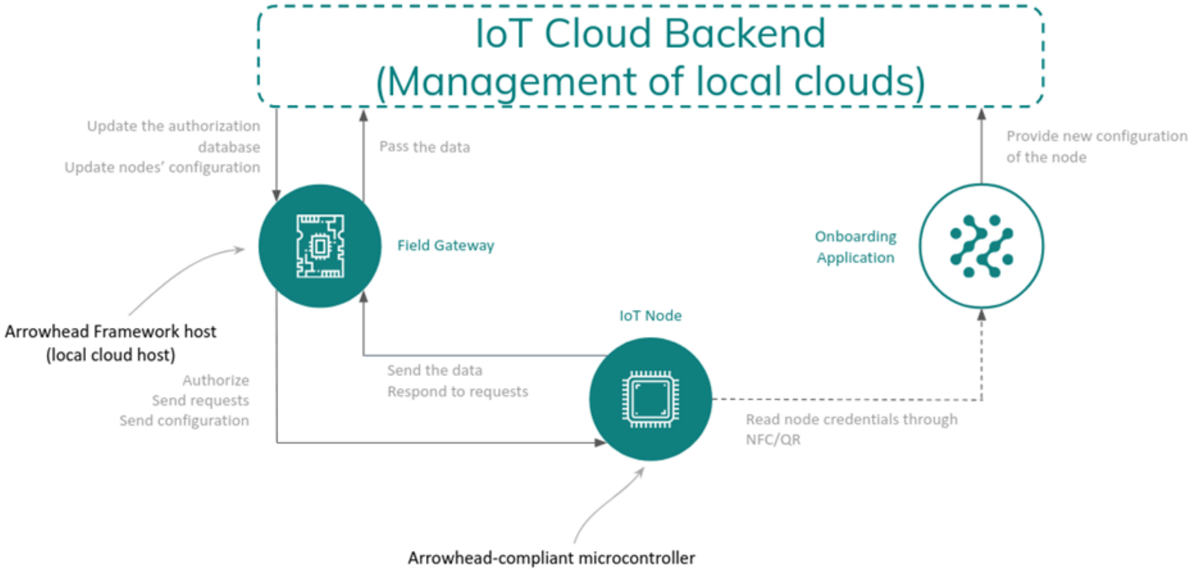


Figure 40 The on boarding toolchain at M36 of UC-11.

As a part of this use case, four stakeholders can be differentiated:

- SH1: IoT solution developer - responsible for the development of the complete solution - assuring that all the components will be compatible and will work together as planned. Responsible for continuous integration and development of the solution.
- SH2: IoT system integrator and maintainer - entity responsible for the on-site integration of the system and all the maintenance activities related to upkeeping the system. Note that there might be a situation where SH1 and SH2 are the same entity but it depends on the specification of the agreement between the developer and the client. The aim of this use case is to minimize the engineering activities that this stakeholder needs to perform in order to get the system up and running.
- SH3: Company that uses the system - the system is installed on-site and is used for the everyday operation of the company. At M0 the role of this stakeholder is limited to using the system in terms of providing the data about the current status of the process (stage of the process, conditions, etc.) and reading the data, while all the maintenance, extension and alterations to the system are performed by SH2.
- SH4: Final customer - optionally might be granted access to the system as a data consumer (read-only) with the right to see the current status of the engineering process of the ordered product or service. At M0 this stakeholder does not have any information about the status of the process/service.
- SH5: Supplier - optionally might be granted access to the system to provide the information about the planned delivery dates, warehouse stock status, availability etc. At M0 this information is not available and SH3 has to ask for the availability and the planned delivery dates on demand.

The above description of stakeholders is generic due to the fact that the envisioned Onboarding process might be applied in a number of different use cases that fit the above framework.

As an example a logistics use case might be provided, where SH3 is a delivery company, SH4 is a client who orders the package, and SH5 is the package provider (or a producer of the components for SH4). SH1 develops the system and SH2 installs it on trucks of SH3 in order to get the continuous monitoring of the transport conditions like the temperature or horizontal acceleration of the package (if there are some constraints). SH4 orders some components from SH5, whereas SH5 subcontracts SH3 to deliver the package due to the fact that the

delivery conditions might be continuously monitored, and the actual position of the trucks might be monitored - which helps SH4 plan the production. Due to this automated traceability of the delivery process, SH5 has proof that the quality of components used for the final product is high, which might be used for monitoring of SLA (service level agreement) execution between SH3 and SH5.
The following tools/components are envisioned for M36:

- Arrowhead-compliant small-footprint producer node (PN) - available at M0 just as a measurement module.
- Field gateway (FG) - at M0 passes the data from sensors to the management cloud.
- Cloud management infrastructure (CMI).
- Performance assessment tool (PAT; not available at M0).
- Onboarding application (OA; not available at M0).
- Arrowhead Local Cloud wireless security enhancement tool (SET).

11.2 Engineering Process Description

In this use case, represented in Figure 41, there are five stakeholders (SHs):

- SH1 => IoT solution developer
- SH2 => IoT solution integrator/maintainer
- SH3 => Company using the IoT solution
- SH4 => Final customer
- SH5 => Supplier

StkH1 and StkH3 cooperate during the design phases to develop the final architecture of the solution that meets the requirements. As a result, StkH1 develops training materials and documentation of the solution, which can be consumed by other stakeholders. StkH2, solution integrator, use onboarding toolchain to deploy the designed solution (EPP-4), which can be further used by other stakeholders through Cloud Management Infrastructure in EPP5. Also, tools supports EPP6 to examine the performance of local clouds, and to diagnose the systems. As a part of EPP7, the designed system might be further extended (which also addresses moving from the design-time to run-time engineering) using the onboarding toolchain.
Additionally, Security Enhancement Tool might be used to incorporate spatial information to the onboarding toolchain. The data being a part of EPP-5 in the Cloud Management Infrastructure can be used/shared with other stakeholders, opening the possibility to build a true multi-stakeholder SoSs.
The yellow lines denote the onboarding toolchain.

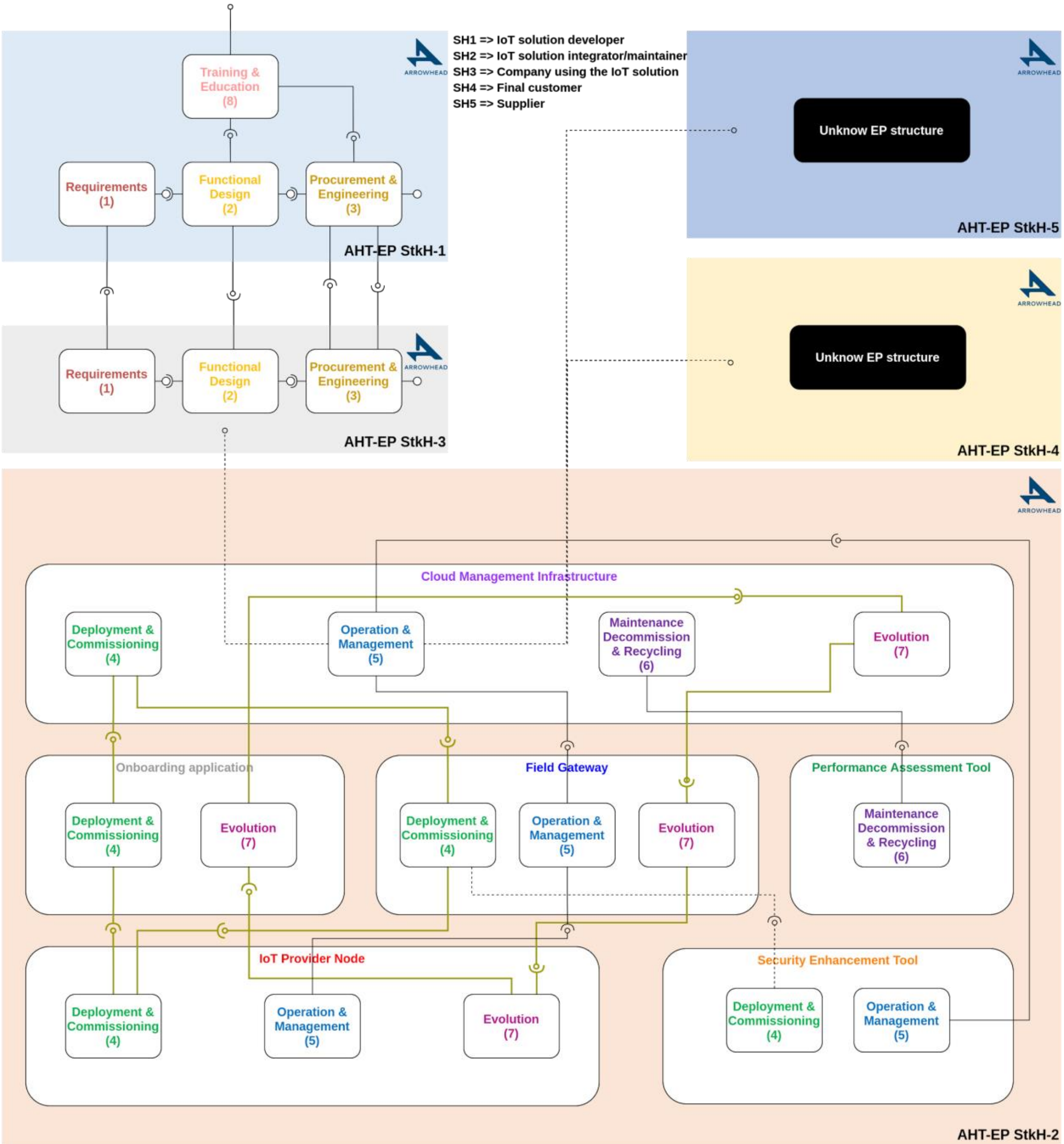


Figure 41 AHT-EP of use case UC-11

The development of the UC is maintained in the agile methodology framework which can be in turn mapped to the RAMI architecture. The envisioned AHT-EP is a more granular RAMI counterpart and thus the engineering process can be mapped to AHT-EP.
After integration of arrowhead, the services are being registered in the service registry and then orchestrated, which definitely can be attributed to use Arrowhead as both:

- run-time interoperability-enabling framework
- life cycle management tool, which currently allows to deploy new services, operate on them and manage them and finally - to evolve existing systems by dynamically attaching new service providers to local cloud.

There is no mismatch between UC-EP and AHT-EP. The engineering process of the UC can be fully mapped to AHT-EP.

For the use case execution first three phases are not considered. It results from the fact that the use case is focused mainly on the deployment of new nodes and the operation of the system to validate Arrowhead methodology and EP in terms of WP7 objective (i.e. reduction of the engineering costs).
It is assumed that the data required in the initial phases (requirements, functional design and procurement and engineering) are given upfront by the definition of the use case. However, the system modeling phase can be included in the general engineering process of the use case as an extension. For this, already available solutions can be used (e.g. SysML toolchain) to not reinvent the wheel.

Although onboarding application does not necessarily have to be developed with the SOA architecture and will not be Arrowhead compliant, it still fits AHT-EP in the Deployment & Commissioning phase as it takes an input (from both IoT node's NFC interface and user) and produces an output (passed further to the Cloud Management Interface).

The goal of implementation of the use case is automation of all the repeatable processes and configurations that should be made by the user. Thus, attachment of new nodes should be user friendly and easy, and at the same time scalable (potentially infinite). There is a certain number (to be determined) of IoT Nodes that can be attached to a single Field Gateway, but the number of Field Gateways attached to Cloud Management Infrastructure is easily scalable. Wireless Security Enhancement Tool and Performance Assessment Tool are easily scalable, as they are installed per AH Local Cloud.

The engineering process can be differentiated into the development and use case execution parts.

Table 59 AHT-EP Phase focus of UC-11

Engineering process phase	Addressed/Focus
Requirements	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Specification of the requirements for the final product; conceptual meetings of the technological, management and sales teams. 2. *Field gateway (FG)*: Requirements collection and specification, conceptual discussions with stakeholders, accounting for possible future extensions. 3. *Cloud management infrastructure (CMI)*: Specification of the requirements for the final product; conceptual meetings of the technological, management and sales teams. 4. *Performance assessment tool (PAT; not available at M0)*: Specification of the requirements, both functional and non-functional. 5. *Onboarding application (OA; not available at M0)*: Specification of the requirements, both functional and non-functional. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.
Functional design	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Device simulation and prototyping, first implementation. 2. *Field gateway (FG)*: Design & simulation with industry-standard CAD tools 3. *Cloud management infrastructure (CMI)*: Design of the architecture of this tool. 4. *Performance assessment tool (PAT; not available at M0)*: Selection of the technological stack.

Engineering process phase	Addressed/Focus
	5. *Onboarding application (OA; not available at M0)*: Selection of the technological stack. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.
Procurement & Engineering	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Optimization of the design. 2. *Field gateway (FG)*: Manufacturing outsourced, manufacturing of several prototypes until obtained final design. 3. *Cloud management infrastructure (CMI)*: Initial prototype in the development environment. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: Mock of the application. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.
Deployment & Commissioning	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: n.s. 2. *Field gateway (FG)*: Setup with correct firmware and configuration for respective use-case. 3. *Cloud management infrastructure (CMI)*: n.s. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s. **Activity Execution** 1. *Arrowhead-compliant small-footprint producer node (PN)*: When preparing a package, the producer node should be onboarded upfront to particular gateways using the onboarding application. 2. *Field gateway (FG)*: Field gateway should be deployed on each truck by StkH3 prior to the delivery. The envisioned onboarding process is designed to be highly automatized and simple so that a technician of StkH3 should be able to easily deploy new Field gateways as needed. Field gateway will host Eclipse Arrowhead as an interoperability framework. 3. *Cloud management infrastructure (CMI)*: When new field gateways are being deployed they are added to the database as available for further IoT nodes deployment. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: This tool can be used when onboarding new devices to include spatial data to the verification of the devices to be onboarded. It might be defined as e.g. maximum distance of the sensor from the corresponding field gateway in order to admit the node to the cloud.
Operations & Management	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Sales and marketing activities. 2. *Field gateway (FG)*: Regular firmware updates to ensure continued security of the Field Gateway. 3. *Cloud management infrastructure (CMI)*: n.s. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s. **Activity Execution** 1. *Arrowhead-compliant small-footprint producer node (PN)*: In the operation phase the IoT nodes serve as measurement providers. 2. *Field gateway (FG)*: Filed gateway will be used as a local cloud provider and a

Engineering process phase	Addressed/Focus
	proxy between the nodes and cloud management infrastructure. 3. *Cloud management infrastructure (CMI)*: Cloud management infrastructure gives the ability for all stakeholder to monitor the current status of the delivery, e.g. measured temperature or acceleration. Also, dynamic reconfiguration in terms of sensors configuration (like sampling time) could be dynamically changed. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: Security enhancement tool can be used to constantly verify the spatial information about devices connected to the local cloud and generate events (e.g. suggest de-authentication) if device leaves defined area.
Maintenance Decommissioning & Recycling	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Customer support, remote monitoring, potential end of support. 2. *Field gateway (FG)*: Removal/destruction of cryptographic keys and other confidential data, recycling electronic waste according to local regulations (e.g. WEEE directive). 3. *Cloud management infrastructure (CMI)*: Monitoring of the application. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s. **Activity Execution** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Reconfiguration of the system might be needed when it's operating, which should be possible from the cloud management infrastructure level. 2. *Field gateway (FG)*: n.s. 3. *Cloud management infrastructure (CMI)*: n.s. 4. *Performance assessment tool (PAT; not available at M0)*: This tool will be installed on a field gateway to provide the information about the performance of local cloud in terms of e.g. CPU usage, link quality. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.
Evolution	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Restructuring and reshaping the design, adding new functionalities. 2. *Field gateway (FG)*: Update firmware with new versions of software components, eventually redesign hardware to incorporate additional electronic interfaces. 3. *Cloud management infrastructure (CMI)*: n.s. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s. **Activity Execution** 1. *Arrowhead-compliant small-footprint producer node (PN)*: When extending the local clouds new nodes could be attached by scanning through NFC credentials of the node. Once the node is onboarded and then powered up, it pairs with an appropriate field gateway and starts providing measurements. 2. *Field gateway (FG)*: Field gateway implements automagic discovery procedure for paring nearby IoT nodes that were previously onboarded to this gateway. 3. *Cloud management infrastructure (CMI)*: When onboarding a new device, this tool passes the information about the deployed local clouds to the onboarding application, and once the device is being onboarded - passes the data to the selected field gateways to provide credentials of the nodes that should be admitted to the local cloud. 4. *Performance assessment tool (PAT; not available at M0)*: n.s.

Engineering process phase	Addressed/Focus
	5. *Onboarding application (OA; not available at M0)*: Credentials of the onboarded IoT node are scanned through NFC and the node is configured (e.g. type of sensor attached, field gateways to which the node should be admitted). The configuration is passed to the cloud management tool. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.
Training & Education	**Activity Development** 1. *Arrowhead-compliant small-footprint producer node (PN)*: Documentation of the node. 2. *Field gateway (FG)*: Train stakeholders how to install and use the field gateway. 3. *Cloud management infrastructure (CMI)*: n.s. 4. *Performance assessment tool (PAT; not available at M0)*: n.s. 5. *Onboarding application (OA; not available at M0)*: n.s. 6. *Arrowhead Local Cloud wireless security enhancement tool*: n.s.

11.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 60 UC-11 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The development of the onboarding process in the envisioned way aims at making extensions of the system easy and scalable. It also means that the system might be dynamically reshaped during its operation, which addresses the run-time engineering feature - which to date had to be done as a part of the design-time system pre-configuration procedure (prior to the installation).
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The information from IoT nodes is populated to the Cloud Management Infrastructure, being a central node for connecting different stakeholders - e.g. the final user and system's maintainer. Through an appropriate configuration of access rights, both may reconfigure or access the data from the system.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Use of kubernetes on the cloud along with its on-demand scalability makes the scalability of the entire system potentially limitless. On the hardware side Field Gateways should be able to handle approximately 20 IoT devices each, and when more nodes are needed - simply another Field Gateway should be added. A single gateway is a node in the Internet with its own IP, and the number of gateways is limited by the throughput of the Internet network and the number of available IP numbers.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	In the development phase the documentation is maintained as a part of the code repository. IT might also be included to the CI/CD pipeline for the automated documentation generation and update.

Table 61 UC-11 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	There are two parts of the use case that contribute to the reduction of the engineering costs: (i) automatic deployment of a local cloud through its dockerized version, and (ii) automated onboarding process of new devices connected to the local cloud along with appropriate UI for configuration of the new device. We can estimate that the time of onboarding and configuration will be reduced by approximately 40%.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Probably not applicable.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Probably not applicable.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	The onboarding tool will reduce the need for manual work done by the engineer onboarding a new device. Moreover, the device should find the field gateway and connect to the desired local cloud automatically.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The onboarding process will be automatized using the imprinted QR codes (with a public key of a device) in connection with the developed UI. As a consequence, the set-up time will be reduced.
Obj. 6 - Training material (HW and SW) for professional engineers	Probably not applicable.

11.4 Engineering Process analysis

Table 62 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-11.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	C

OBJ-AHT #1	X	X	X	X			X	X	
OBJ-AHT #2									
OBJ-AHT #3									
OBJ-AHT #4				X	X				
OBJ-AHT #5				X	X				
OBJ-AHT #6		X	X	X				X	
OBJ-WP2 #1				X			X		
OBJ-WP2 #2				X	X	X	X		
OBJ-WP2 #3				X	X		X		
OBJ-WP2 #4		X	X	X				X	

12. UC-12 Digital Twins and structural monitoring (NTNU)

The use case is to develop digital twin of crane using both online and offline sensor data to provide structural monitoring.

There is currently a considerable hype factor related to Internet of Things (IoT) and Digital Twin (DT) technologies, with corresponding inflated expectations. In the wake of this hype there is a plethora of technology and solution promises. Retrofitting existing assets with sensors is supposed to make them smart. Connecting them to existing workflows is supposed to make them intelligent. Furthermore, by combining with big data it is promising to have cognitive systems. IoT and Digital Twins are considered to have passed the first camel hump on the hype cycle by many, but the real challenge lies ahead of us. We can expect a Darwinist selection among fiercely competing players, converging towards those that can bring the technological opportunities to maturity. That is – how to go from connected devices to DT delivering insight and understanding that provide real and tangible value to end users.

The aim of this Use Case is to address the most critical R&D challenges in DT for predictive maintenance of cranes. These R&D topics are based on live DT experience performed by NTNU researchers and master students. These benchmarks represent the current state of the art in DT implementations for structural monitoring. Hexagon/MSC and Amazon are the market leaders in IoT and DT solutions and applications. However, DT for structural monitoring and predictive maintenance are still on the research stage and very few successful applications are reported. The Akselos Company, which provides Digital Twin solutions developed together with MIT, has reported 2 case studies. Their patented Akselos Integra software was used to create a condition-based Digital Twin of a FPSO ship and a Shiploader. Their solutions are based on a Reduced Basis Finite Element Analysis (RB-FEA) approach and a Decision Support system developed and licensed by MIT. The MIT data analytics works like a digital guardian that allows operators to not only monitor an asset's condition in real time, but helps them to see the future.

12.1 Overall description of the UC-EP

In this Use Case, we have determined to use the Palfinger crane PK 65002M mounted on NTNU research Gunnerus vessel as the testbed of the Use Case, as shown in Figure 42.

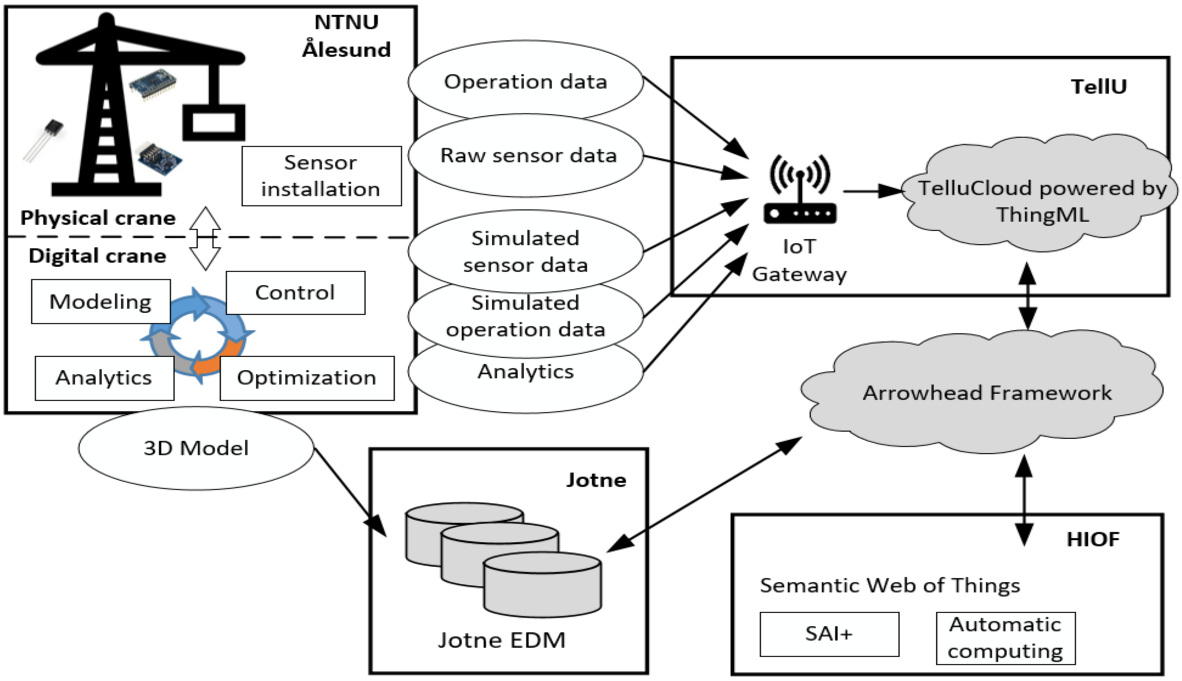


Figure 42 UC-12 Architecture

There will be four functional blocks in this Use Case:

1. Crane sensor installation; Crane model including dynamic model and 3D model will be developed; sensor data with simulated data and real data will be collected (NTNU Ålesund).
2. Data will be transferred through IoT gateway, and ThingML will be used to generate the tools that are compliant with the Arrowhead framework (TellU).
3. The models and the data will be presented in Jotne EDM system, to analyze and visualize the results (Jotne).
4. A sematic web of Things will be established to run autonomic computing and SAI+ (HIOF).

Some tools/tool chains will be used in the Use Case, such as FEDEM, 20sim, NX, TellU ThingML, TelluCloud, Jotne EDM, SAI+, etc.

We break down the use case into four phases:

1. Define problem: how to generate a digital model of a crane for remote monitoring the structure and provide meaningful information for real-time operation & maintenance.
2. Specify requirement: What sensors should be installed? How can we use the sensor data? Who will be the end user of the digital crane? And what function should be included?

3. Develop the solution: figure out the techniques needed for the use case; make possible sensor installation proposal; provide solid way for data transmission; realize the functionalities of IoT and web of Things.
4. Build a prototype: implement a prototype test of the crane on the Gunnerus ship (a research ship in NTNU); realize real-time data presentation and IoT related functions.

In particular, we are clear to use the Palfinger PK6500M crane as the testbed. It is the crane mounted on NTNU's research vessel -- Gunnerus vessel. We are now more or less at the end of phase 2. This procedure is close to the IEC 81346 from "Requirements" to "Operation & Management".

12.2 Engineering Process Description

Figure 43 defines the Engineering Process of the crane UC about the interactions of all the four stakeholders. In order to achieve digitalization of crane operation, in the "requirement" phase, necessary sensors with different types, brands, and working scope are investigated.
The result is the basis for the Functional Design phase, where we define fundamental functions of the UC.
Before we determine the sensor types and number to be installed on crane, we conduct a bike ride experiment for remote monitoring the bike status during riding. The aim is to investigate functionality in terms of data transmission, storage and visualization in the system. As long as these functions are determined, we start the order and the installation of sensors with the help of the crane manufacturer.
In parallel, we install the devices we used in the bike experiment near by the crane.
By proper configuring the devices, the function for remote monitoring the crane can be achieved.
The rest of the EP phase 6-8 are undergoing and will be updated in the next deliverables of the project.

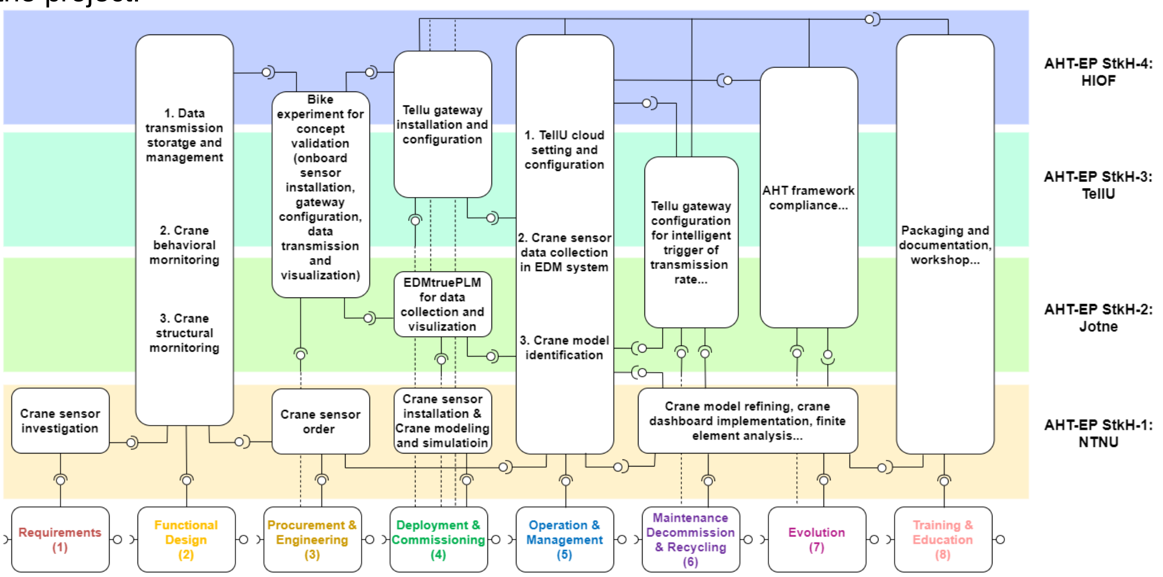


Figure 43 AHT-EP of use case UC-12

Now we are using teams for data/ model share for managing the engineering process.
We have used the tools FEDEM for data analysis; tools NX and 20 sim for model generation; tools for data transmission tested by TellU and Jotne in the bike experiment. These tools are matched to the phases of "functional design".

For building the crane 3D model, we have refined a model from the crane supplier. Now the resulting model is good enough for visualization. For the crane dynamic model, we have created the model that can simulate crane operation for crane motion analysis. For data transmission in AHT framework, we have succeeded to conduct the bike experiment. The result can be used in the crane data transmission in the use case.

The use case will take all the AHT-EP phases into account. Up to now, we have considered the first four AHT-EP phases. The res twill be considered in the near future.

In the engineering phase, we have succeeded to use Tellu gateway and TelluCloud service for data transmission in bike experiment. However, the experience cannot be directly transferred to the crane use case, since Tellu Gateway has to connect to an extra Pluto gateway for data transmission. Now we found a partner from Germany has experience to use Pluto gateway in AHT framework. We will contact the partner for help.

Table 63 AHT-EP Phase focus of UC-12

Engineering process phase	Addressed/Focus
Requirements	**Activity** 1. *Sensor investigation*: Has several meetings with crane supplier; discussed the possible sensors that can be installed on the crane; 2. *Sensor installation*: Sensor shipping; time schedule for both crane supplier and the available time for NTNU Gunnerus ship; electrical technician; 3. *Crane modelling and simulation*: Crane 3D models used for visualization; crane dynamic model used for crane dynamic simulation and control; Finite element analysis; 4. *Data transmission*: Design IoT sensor solution for data transmission, storage and visualization from crane to remote end; 5. *Crane behaviour and structural monitoring*: Investigate sensor frequency; propose to develop dynamic sensor data transmission based on crane operation status; **Tools** 1. *Sensor investigation*: • Input: Requirement from the use case, e.g., sensor type, dimension, positon. • Output: excel file that records possible sensors that are applicable to the crane. 2. *Sensor installation*: • Input: installation requirements from crane supplier. • Output: excel file that records the requirements. 3. *Crane modelling and simulation*: • Input: rough crane 3D model from crane supplier. • Output: figure out components collision within the 3D model. • Input: crane specification from crane supplier, like dimension, mass, material. • Output: understand the physical property like inertial of the crane. 4. *Data transmission*: • Input: IoT sensor solution idea. • Output: IoT sensor solution scheme using TellU cloud and Jotne EMD system. 5. *Crane behaviour and structural monitoring*: • Input: discuss data transmission efficiency. • Output: propose to realize dynamic data transmission based on crane operation status. • Input: current crane model has 8 links while the real crane has 7 link.

Engineering process phase	Addressed/Focus
	• Output: should update crane model.
Functional design	**Activity** 1. *Sensor investigation*: Tested sensor data transmission with TelluCloud; determined sensor type, and installation method; 2. *Sensor installation*: Determined shipping time and installation time; 3. *Crane modelling and simulation*: Crane 3D models from NX; crane dynamic model from 20sim; 4. *Data transmission*: Defined bike riding experiment, focusing on data transmission, storage and visualization; 5. *Crane behaviour and structural monitoring*: Update crane model (from 8 links to 7 links); **Tools** 1. *Sensor investigation*: • Input: excel about sensor information. • Output: determine the sensor type, number and position. 2. *Sensor installation*: • Input: excel about installation requirements. • Output: determine the installation method. 3. *Crane modelling and simulation*: • Input: crane specification PDF file. • Output: components properties for modelling. 4. *Data transmission*: • Input: bike riding experiment plan. • Output: define experiment target, including data transmission, storage, and visualization. 5. *Crane behaviour and structural monitoring*: • Input: 8-link crane model, FEDAM model. • Output: propose to simplify crane model with 7 links for visualization.
Procurement & Engineering	**Activity** 1. *Sensor investigation*: Got a quoto and accept the quoto; 2. *Sensor installation*: Received sensor and relevant components; 3. *Crane modelling and simulation*: Obtain rough crane 3D model from crane supplier developed crane dynamic model in 20sim; 4. *Data transmission*: Prepared bike ride experiment programmed for the experiment; 5. *Crane behaviour and structural monitoring*: Crane dashboard development; crane structural modelling in FEDAM; **Tools** 1. *Sensor investigation*: • Input: excel about sensor information. • Output: determine the sensor type, number and position. 2. *Sensor installation*: • Input: accept quoto from crane supplier. • Output: Receive sensors and relevant components for installation. 3. *Crane modelling and simulation*: • Input: rough crane model in 20sim. • Output: components with collisions in 20sim. 4. *Data transmission*: • Input: bike riding experiment plan. • Output: prepare hardware and software for the experiment. 5. *Crane behaviour and structural monitoring*: • Input: 7-link crane model. • Output: crane dashboard system for visualization.

Engineering process phase	Addressed/Focus
Deployment & Commissioning	**Activity** 1. *Sensor investigation*: The electrician came to check crane status; 2. *Sensor installation*: Installed sensors on the crane; has used wrong type of Pluto Gateway; need to replace it and update PLC; 3. *Crane modelling and simulation*: Tested crane model in 20sim; 4. *Data transmission*: Configured TellU gateway; configured Jotne EMD system; 5. *Crane behaviour and structural monitoring*: n.a. **Tools** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: • Input: personnel from crane supplier for sensor installation. • Output: Sensors succeeded installed on the crane. 3. *Crane modelling and simulation*: • Input: crane model in 20sim. • Output: simulation test of crane operation in 20sim. 4. *Data transmission*: • Input: hardware and software for the bike ride experiment. • Output: hardware and software configuration and programming. 5. *Crane behaviour and structural monitoring*: n.a.
Operations & Management	**Activity** 1. *Sensor investigation*: Simple operation test; 2. *Sensor installation*: Simple operation test; 3. *Crane modelling and simulation*: Took crane operation to get sensor data; 4. *Data transmission*: Performed the bike riding experiment; collected sensor data and visualized the result; 5. *Crane behaviour and structural monitoring*: Plan to perform onsite crane operation to get sensor data; plan to test FEDAM model for certain crane operation under environmental effects; **Tools** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: n.a. 3. *Crane modelling and simulation*: • Input: perform onsite crane operation. • Output: collect sensor data for the crane operation. 4. *Data transmission*: • Input: perform bike ride experiment. • Output: collected sensor data and visualize the result. 5. *Crane behaviour and structural monitoring*: • Input: collected sensor data from crane operation. • Output: a simple demo of the dashboard system.
Maintenance Decommissioning & Recycling	**Activity** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: Received the new Pluto Gateway; 3. *Crane modelling and simulation*: n.a. 4. *Data transmission*: Re-run the bike riding experiment due to unstable GPS signal; 5. *Crane behaviour and structural monitoring*: Compare onsite crane operation with simulated crane operation; perform FEA; **Tools** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: n.a. 3. *Crane modelling and simulation*: n.a.

Engineering process phase	Addressed/Focus
	4. *Data transmission*: • Input: sensor data collection. • Output: identify unstable signal problem. 5. *Crane behaviour and structural monitoring*: • Input: behaviour comparison between onsite test and dashboard system. • Output: list of behaviour differences from comparison.
Evolution	**Activity** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: Deployed the new Gateway on the Gunnerus ship; 3. *Crane modelling and simulation*: Refined the crane 3D model; 4. *Data transmission*: Fixed unstable signal problem; 5. *Crane behaviour and structural monitoring*: Repeat phase 4-6 to refine the simulation results; **Tools** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: n.a. 3. *Crane modelling and simulation*: • Input: simulation result and onsite test result. • Output: refined crane model in 20sim. 4. *Data transmission*: • Input: unstable signal problem. • Output: solve the problem. 5. *Crane behaviour and structural monitoring*: • Input: list of behaviour difference. • Output: improve dashboard system based on the behaviour difference.
Training & Education	**Activity** 1. *Sensor investigation*: Documentation of sensor installation plan; 2. *Sensor installation*: Documentation of sensor types and features and their location on the crane; 3. *Crane modelling and simulation*: Version control of crane 3D model; 4. *Data transmission*: documented the bike riding experiment; confirmed the IoT sensor solution for the UC; 5. *Crane behaviour and structural monitoring*: Documentation of the simulation results; internal workshop to present the use of the simulation and results. **Tools** 1. *Sensor investigation*: n.a. 2. *Sensor installation*: • Input: sensor installation information. • Output: documentation of sensor installation information. 3. *Crane modelling and simulation*: • Input: several version of crane model in 20sim. • Output: version control document of crane models. 4. *Data transmission*: • Input: bike ride experimental results. • Output: documentation of the bike ride experiment. 5. *Crane behaviour and structural monitoring*: • Input: dashboard results. • Output: documentation of dashboard results.

12.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 64 UC-12 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	From the AHT-EP, we determine the test-bed crane sensor type in the requirement phase and identify the functions of the UC in “Functional Design” phase. The change from design time to run time engineering first happens in the “Procurement & Engineering” phase, from which we start to investigate the functionality of the crane system to be implemented by means of a bike riding experiment. Thus, the successful experience from the bike riding experiment can be applied to the development of the crane monitoring system.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Different stakeholders are connected in the AHT-EP, from the “Functional design” phase to the “Evolution” phase. For example, in the bike ride experiment, three stakeholder in the UC evaluate the devices and validate the function for data transmission, storage and visualization. Although stakeholders have their own focus in the UC, their work are integrated in the EPs, either by manual connection or automatic connection.
Obj. 3 - Handling of substantially increased number of I/O’s due to much more fine grained automation	Taking advantages of the AHT-EP, the stakeholders have the ability to have multi-interactions with each other either in a manual manner, e.g., passing the installation specification of crane sensor in “deployment” phase, or in an automatic manner, like the data transmission from Tellu gateway to Jotne EDM system. This is particular beneficial in phase 5-7 in the UC. For example, the “evolve” phase enables the information exchange more flexible, thus we can update the crane model, make FEA based on the inputs from historical operation data in multiple loops.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Digital learning and training activities are designed as “packaging, documentation and workshop” in the “Training” phase in the AHT-EP. These activities mainly serve for phase 4-7, to support the development of documentation, video demo, and reference manual for the crane monitoring system.

Table 65 UC-12 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	In order to save costs, more efforts will be put on “Requirements” and “Functional design” phases at the beginning. For this use case, we will identify the “Requirements”, such as, what types of sensors data should be and can be installed to realize structural monitoring and decrease maintenance frequency. These requirements determine the “Functional design”, including hardware & software selection and IoT solutions. The result can save the engineering cost by 20-50%. Currently, we have finished the IoT scheme and the sensor data can be transmitted to remote end.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Today, there exists no standards for connecting systems data to IoT data. The Norwegian use case aims to foster new solutions by creating the backbone for IoT and CAD/PLM/MBSE integrations.

Project Objective	Focus & Planed actions
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	By using ISO 10303 (STEP) any legacy tools can be connected assuming they have such adaptors available, like most CAD tools have and the major PLM tools like PTC/Windchill, SAP and more. Now we will use the standard in EMD system for visualization of the crane operation.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	The digitalization of the crane platform will comply with some standard of protocols, encodings and semantics for data transmission, access and storage. In the IoT scheme for the UC, we use Modbus TCP for communication between Pluto gateway and TellU gateway.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Arrowhead Framework authorization system will be investigated.
Obj. 6 - Training material (HW and SW) for professional engineers	The use of the digital twin will be documented as a manual for training new users. For example, a new user can be trained to analyze operation data, evaluate risk and estimate maintenance from the remote terminal. We have documented the sensor installation for the use of other stakeholders.

12.4 Engineering Process analysis

Table 66 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-12

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X						
OBJ-AHT #2		X	X						
OBJ-AHT #3		X	X		X				
OBJ-AHT #4			X		X				
OBJ-AHT #5			X						
OBJ-AHT #6		X	X	X	X	X	X	X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2		X	X	X	X	X	X		

OBJ-WP2 #3				X	X	X	X		
OBJ-WP2 #4				X	X	X	X	X	

13. UC-13 Deployment engine for production related sensor data (BOLIDEN)

The legacy way of working for production-data based data integration has been handled with either point to point integration or with the same data integration technology as traditional integration such as from and to Financial systems. Point to point integration means closed vendor implementation with high risk to lack reusability, standardization and fulfilment of internal quality and security requirements. The assessment is that the traditional integration approach based on IBM web sphere will not be able to scale with the need for sensor integration and fulfil requirements such as a distributed setup.

Main purpose of this activity is to enable for Boliden fast, secure and reliable integration of sensor data to enable further processing (monitoring, analysis, etc.), it is called internally "Boliden Integration Box" (BIB). This use case on interoperability level but has direct and indirect connections with the Arrowhead integration layer.

The baseline is the traditional approach of integration for new sensor related data and can be measured against the BIB integration approach.

13.1 Overall description of the UC-EP

In the traditional approach the exchanges have been done by project bases and solved by vendor methodology / lock-in integration or using the Boliden integration platform service teams process to implement data exchange which has not full competence in production related data and technology is assessed to not fit requirements in mid-term. In both cases waterfall project models are used where requirement for integration are implemented and operated by project team or Boliden integration service team.

The new approach with BIB the process will largely remain the same but the BIB enables reuse of data exchange pattern with an iterative development.

The Stakeholders involved are the BIB team, a source system responsible and a destination system responsible (in case of bi-directional exchange the both aspects have to consider for the system responsible). The destination system responsible, usually the one in need for data is requesting an integration and the BIB team implements the data exchange based on existing or newly built adapters. Setup, deployment and document is automated as much as possible. Figure 44 illustrates the preliminary architecture of the use case.

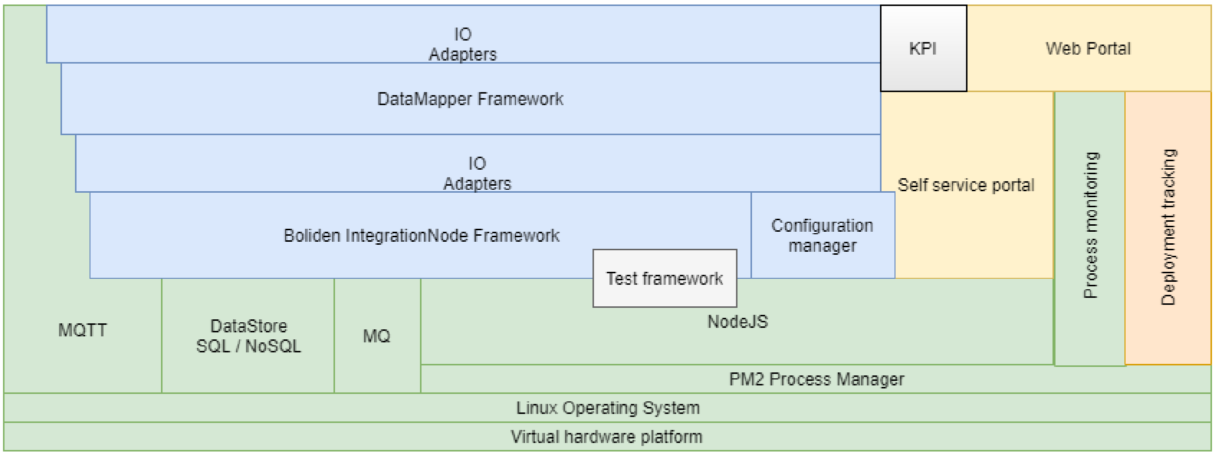


Figure 44 Use case 13 architecture.

Boliden is based on site by site operation and has the requirement for site survivability meaning without internet line the production has to run. This matches the distributed setup of the BIB where on per site can be setup.
Figure 45 illustrate the integration approach that is adopted.

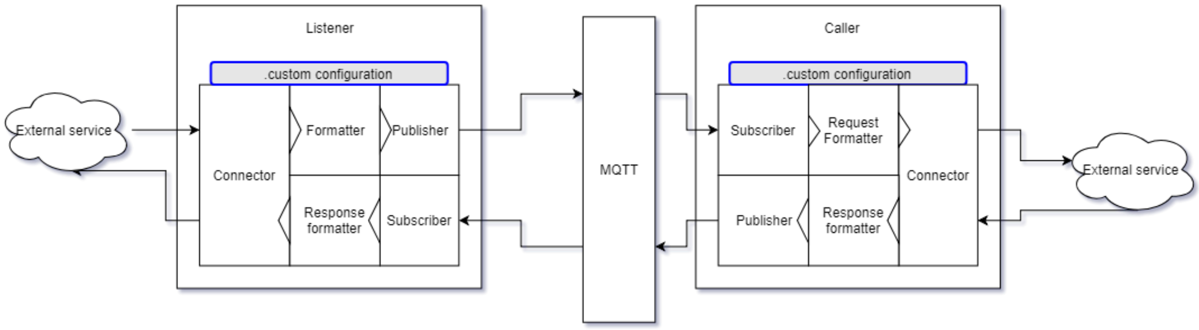


Figure 45 UC-13 integration approach.

Base on business priority the activities on building BIB have been started before the start of the AHT project and the focus during the project will be on the latter phases in the UC engineering process from data exchange request to implementation.
In requirement phase the data exchange requirements are collected from destination system responsible (Stakeholder 1) and assessed by the BIB team (Stakeholder 2). In case an adaptor is already present translation / transformation definitions are reviewed with destination system responsible (SH1). If the adaptor is not implement, the adaptor has to be considered as well, both in requirements and functional design. This is then also reviewed with the source system responsible (SH3). The BIB team (SH2) design the data integration in functional design phase. The procurement & engineering is internally done by the BIB team (SH2). The same applies for deployment & commissioning. During this process the source and destination system responsible are involved to ensure right quality of data. Operation & Management as well as Maintenance, decommissioning & recycling is handled by the BIB team (Stakeholder 2). The evolution is handled based on feedback from system responsible and BIB roadmap, event data and problem tickets. Training & education is mainly internally in the BIB team where a wiki all relevant documents is placed.

The technology use and adopted during engineering phase consist of the BIB components as outlined in the architecture above as well as the wiki for documentation.

13.2 Engineering Process Description

In Figure 46 the UC-13 AHT-EP schema. In general, two components are looked at in this use case:

1. Boliden Integration Box (BIB) backend which is the deployment in a production site where "local survivability" is needed in case the internet line goes down;
2. the BIB adapter which is a specific interoperability implementation between two systems.

This specification from the requestor and which is typically also the destination data source owner as well as from the source system owner. In case an adapter is already available for the source system, this can be reused then.
The setup of backend and configuration is done by the Boliden Integration Box Team internally. Also operation is with the BIB Team.
Interaction is needed with data source and destination responsible in maintenance and evolution phase.

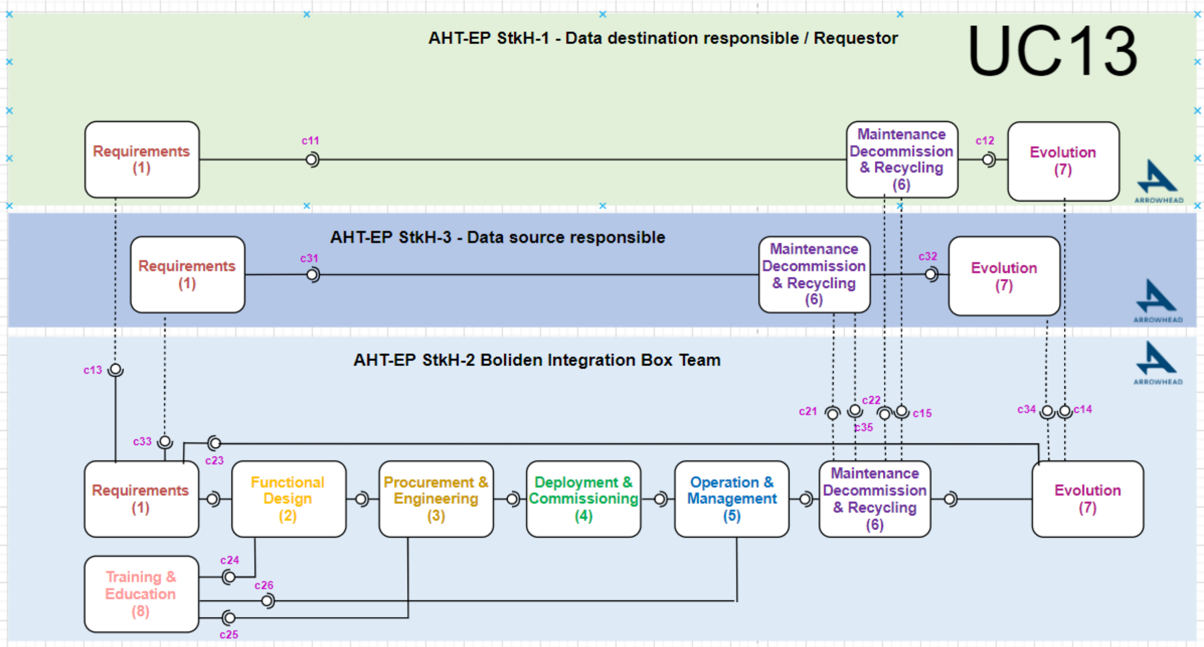


Figure 46 AHT-EP of use case UC-13

The legacy approaches used waterfall methodology usually driven by the external partner chosen. There was limited integration and learnings happening between activities.
The BIB uses an iterative / agile approach a) develop new adapters b) create new data integration. Depending on the use-case this can be within a business project (waterfall drive) or handled as an activity with the BIB team. The process ensures a full lifecycle management with defined standards. The process maps to the AH-EP.
All phases makes sense and support the use-case.
No lack of technology identified yet.

Compared to before the mapping of the AHT-EP support the standardization approach compared to before where any EP could be used (usually vendor driven).

Areas for scalability that are considered by the BIB is the reusability for data source and destination owners for data exchanges. The use-case provides scalability with a fast deployment of BIB instances including adapters for needed data exchanges in a standardized way. The operation reliable does not scale at this point in time as required components are outside the scope of the use-case.

No particular standards yet implemented next to the incorporation of AH-EP which makes compliance to RAMI 4.0 for some parts.

Table 67 AHT-EP Phase focus of UC-13

Engineering process phase	Addressed/Focus
Requirements	**Activity** SH1 requirement for new data needed in local or global cloud, this also includes transformation / translation requirements potentially needed. Other integration meta data is capture such as frequency, security and other service level related requirements. *1. Boliden Integration Box backend (BIBB):* SH1 requirement for data exchange in a local cloud, e.g. production site where no SIB instance is in place. 2. *Boliden Integration Box adapter (BIBA):* SH1 requirement for data exchange where for either source or destination an adapter is missing, SH3 involved if source system is affected. **Tools** 1. *Boliden Integration Box backend (BIBB):* SH1 request setup of site instance for BIBB from SH2. • Input: <c13> SH1 request data integration which can triggers a local deployment if not already a SIB backend is present at site. • Output: <SH2 - EPP2> Request details on needed integrations and requirements to determine e.g. sizing of local deployment 1. *Boliden Integration Box adapter (BIBA)*: SH1 request data integration with BIBA from SH2. • Input: <c13> SH1 request data integration from data source to data destination. SH1 also gives details on integration end-point such as API details and expected data format • Input: <c33> SH2 request details from SH3 on data source access, format and other specifications • Output: <SH2 - EPP2> Request details in standard format for needed data integration.
Functional design	**Activity** SH2 designs the translation / transformation in alignment with SH1 and SH3. 1. *Boliden Integration Box backend (BIBB*): n.a. 2. *Boliden Integration Box adapter (BIBA):* SH2 design adaptation for either source or destination system. **Tools** 1. *Boliden Integration Box backend (BIBB*): n.a. 2. *Boliden Integration Box adapter (BIBA)*: SH2 designs adapter. • Input: <SH2-EPP1> Standardized requirements form for data integration • Output: <SH2 - EPP3> Adapter design, data mapping needs, integration meta data such as frequency.
Procurement & Engineering	**Activity** SH2 implements provisioned data.

Engineering process phase	Addressed/Focus
	1. *Boliden Integration Box backend (BIBB)*: Internal SIB team plans and prepare deployment. 2. *Boliden Integration Box adapter (BIBA)*: Internal SIB team plans and prepare deployment. **Tools** 1. *Boliden Integration Box backend (BIBB)*: n.s. 2. *Boliden Integration Box adapter (BIBA)*: SH2 develops adapter. • Input: <SH2-EPP2> Functional specification for adapter and its setup • Output: <SH2 - EPP4> Developed and tested adapter with source and destination data.
Deployment & Commissioning	**Activity** SH2 implements and SH1, SH3 test fulfilment in delivery according to specification. 1. *Boliden Integration Box backend (BIBB)*: Internal SIB team plans and prepare deployment. 2. *Boliden Integration Box adapter (BIBA)*: Internal SIB team plans and prepare deployment. **Tools** 1. *Boliden Integration Box backend (BIBB)*: SH2 deploys BIBB. • Input: <SH2-EPP3> Deployment scripts for BIBB, destination server ready for BIBB installation. • Output: <SH2 - EPP5> Local instance deployed. 2. *Boliden Integration Box adapter (BIBA)*: SH2 deploys adapter BIBA. • Input: <SH2-EPP3> Developed adapter and specification for setup parameters. • Output: <SH2 - EPP5> Adapter operation and recurring activities scheduled / event driven integration configured.
Operations & Management	**Activity** 1. *Boliden Integration Box backend (BIBB)*: Handled by SH2. 2. *Boliden Integration Box adapter (BIBA)*: Handled by SH2. **Tools** 1. *Boliden Integration Box backend (BIBB)*: SH2 operated BIBB. • Input: <SH2-EPP4> deployed BIBA. • Input: <c26> training information needed to operate BIBB stored in wiki. • Output: <SH2 - EPP6> ticket and monitoring information as input for maintenance. Also decommission request in case not active integration are present anymore. 2. *Boliden Integration Box adapter (BIBA)*: SH2 operated BIBA. • Input: <SH2-EPP4> deployed BIBA. • Input: <c26> training information needed to operate BIBA stored in wiki.

Engineering process phase	Addressed/Focus
	• Output: <SH2 - EPP6> ticket and monitoring information as input for maintenance. Also decommission request in case integration is note needed anymore.
Maintenance Decommissioning & Recycling	**Activity** 1. *Boliden Integration Box backend (BIBB)*: Handled by SH2. Regular review for usage performed. 2. *Boliden Integration Box adapter (BIBA)*: Handled by SH2. Regular review for usage performed. **Tools** 1. *Boliden Integration Box backend (BIBB)*: SH2 manages issues for BIBB and interacts with SH1 and SH3 for all adapters for maintenance and decommissioning. • Input: <SH2-EPP4> deployed BIBA. • Output <c21, c22 >: Issue and maintenance plans for data destination affecting data integration for SH1 and SH3. • Output: <SH2 - EPP7> collected input for evolution needs. 2. *Boliden Integration Box adapter (BIBA)*: SH2 manages issues for SIBB and interacts with SH1 and SH3 for maintenance and decommissioning. • Input: <SH2-EPP4> deployed BIBA. • Input / Output <c15, c35, c21, c22 >: Issue and maintenance plans for data destination affecting data integration between SH1, 2, 3. • Output: <SH2 - EPP7> collected input for evolution needs.
Evolution	**Activity** Handled by SH2 in alignment with feedback from SH3 and SH1 (for changes coming from data sources). 3. *Boliden Integration Box backend (BIBB)*: Handled by SH2 in alignment with evolution of SIB components and feedback from all SH1 and SH3 (all instances). 4. *Boliden Integration Box adapter (BIBA)*: Handled by SH2 in alignment with evolution of SIB components and feedback from all SH1 and SH3 (all instances). **Tools** 1. *Boliden Integration Box backend (BIBB)*: SH2 aggregates evolution demands. • Input <SH2-EPP6>: Improvement needs from maintenance. • Input <c14, c34>: Evolution needs by SH1 on data integration, e.g. sizing of BIBB. • Output <SH2-EPP1> Improvement requirements for BIBB. 2. *Boliden Integration Box adapter (BIBA)*: SH2 aggregates evolution demands from SH1 and SH3. • Input <SH2-EPP6>: Improvement needs from maintenance. • Input <c14>: Evolution needs by SH1 on data integration, e.g. adaptation for destination system upgrades affecting data end-points. • Input <c34>: Evolution needs by SH3, e.g. upgrade of source system affecting data end-points. • Output <SH2-EPP1> Improvement requirements for BIBA.

Engineering process phase	Addressed/Focus
Training & Education	**Activity** SH2 educated data users on data usage possibility, limitation and duties. 1. *Boliden Integration Box backend (BIBB)*: SH2 creates needed documentation for instance deployment. 2. *Boliden Integration Box adapter (BIBA)*: SH2 creates needed documentation for instance deployment. **Tools** 1. *Boliden Integration Box backend (BIBB)*: SH2 documents backbends information for EPP-5 in Wiki. • Input <c24>: Design documentation and information. • Input <c25>: Engineering documentation but also deployment, operation, maintenance and improvement processes. • Output <c26> Relevant documentation and training for SH2 in operations & management all stored in Wiki. 2. *Boliden Integration Box adapter (BIBA)*: SH2 documents adapter information for EPP-5 in Wiki. • Input <c24>: Design documentation and information. • Input <c25>: Engineering documentation but also deployment, operation, maintenance and improvement processes. • Output <c26> Relevant documentation and training for SH2 in operations & management all stored in Wiki.

13.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 68 UC-13 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The objective matches partly has transformation/translation is defined at runtime and can be change at run-time in the EP. Also the evolution phase for all incorporated stakeholder makes it possible to support this objective.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	This objective matches as in the legacy approach the work was completed in verticals with limited to one area. With a new stakeholder as the BIB team and the data source and destination responsible, standardization and automation can be achieved and supports this objective.
Obj. 3 - Handling of substantially increased number of I/O's due to much	The objective matches as the BIB platform does not only allow for scalability in I/Os but also with fast BIB deployments in local clouds.

WP2 Objective	Focus & Planed actions
more fine grained automation	
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The objective is supported as consolidated material is placed in a wiki available for all team members and part of all engineering phases.

Table 69 UC-13 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	n.s.
Obj. 2 - Interoperability for IoT and SoS engineering tools	n.s.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	n.s.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	n.s.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	n.s.
Obj. 6 - Training material (HW and SW) for	n.s.

Project Objective	Focus & Planed actions
professional engineers	

13.4 Engineering Process analysis

Table 70 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-13.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X	X		X	X	X	
OBJ-AHT #2		X	X	X	X				
OBJ-AHT #3			X	X					
OBJ-AHT #4			X			X	X		
OBJ-AHT #5			X						
OBJ-AHT #6	X	X	X	X	X	X	X	X	
OBJ-WP2 #1		X	X	X		X	X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3			X	X					
OBJ-WP2 #4	X	X	X	X	X	X	X	X	

14. UC-14 Smart Diagnostic Environment for Contactless Module Testers (IFAT)

This Use Case have been combined with UC-16.5. Thus will be described in the UC-16.5 section 16.5.

15. UC-15 Smart Kitting to Manage High Diversity (VTC)

This use case will focus on the kitting process that is a well-established internal logistics process for suppling assembly lines. Kitting is to ensure that material is delivered and presented to the assembly operator at the main production line in an optimal way. The work of this use case consists of developing concepts for smart and automated kitting operations and the required toolchains.
Initially, specific focus is on the engineering process related to preparation of kitting. Large number of parts, and even higher number of possible combinations imply logistic and engineering challenges. Decreasing the workload on the kitting engineers by automating layout planning for the kits is desirable in order to reduce processing times and thus cost.
Plans regarding the increase of the diversity of assembled products require optimizations in the manufacturing order processing pipeline.

Figure 47 shows the complete process, where kit engineering is shown in the top middle. All kit planning is today manual. Digital support is missing.

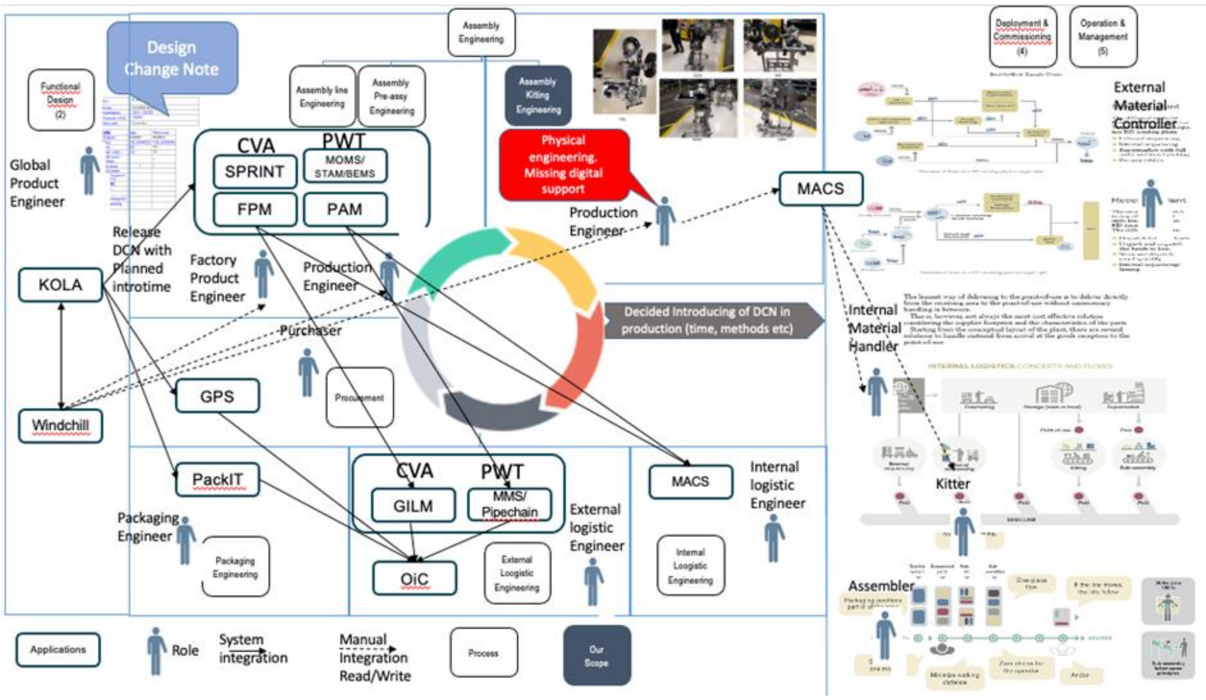


Figure 47 Complete engineering process for kitting.

Figure 48 expands the current situation of manual kit engineering in more detail.

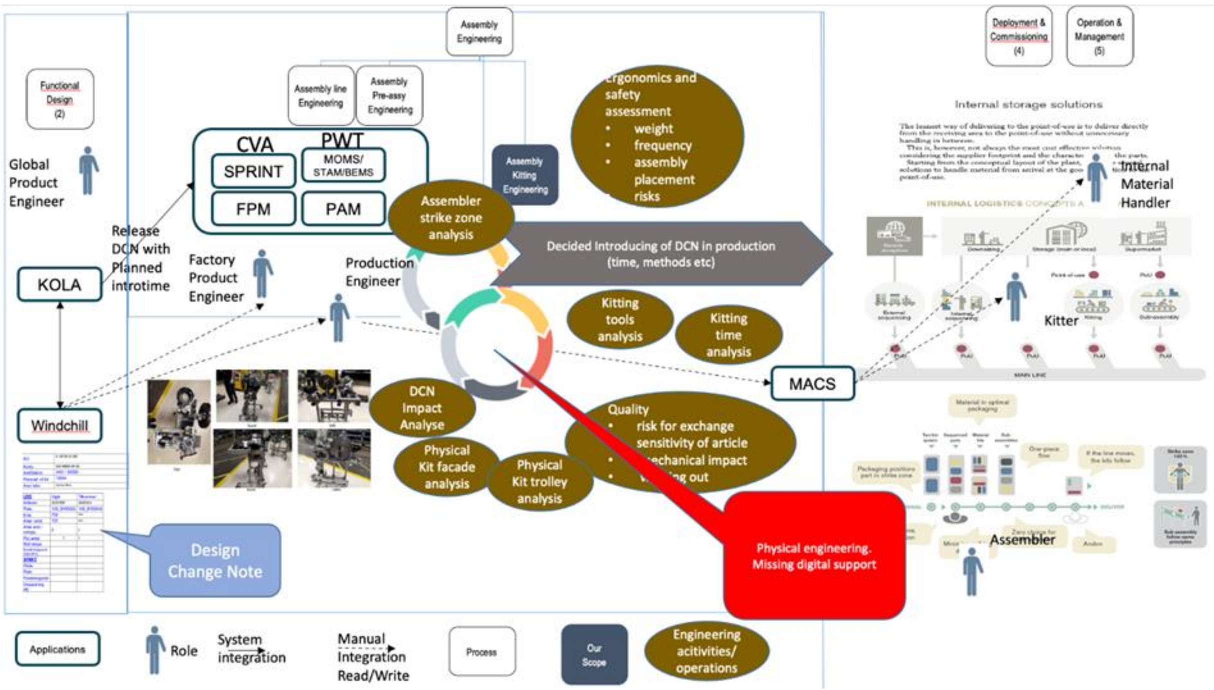


Figure 48 Visualization of Manual work of kitting Engineering.

15.1 Overall description of the UC-EP

The Smart Kitting System (represented in Figure 49) integrates every part of the kitting process and creates a feedback loop in order to efficiently manage the whole operation. We focus our work mainly on the Layout engineering part, as that is the core of the system, most of the complexity relies there.

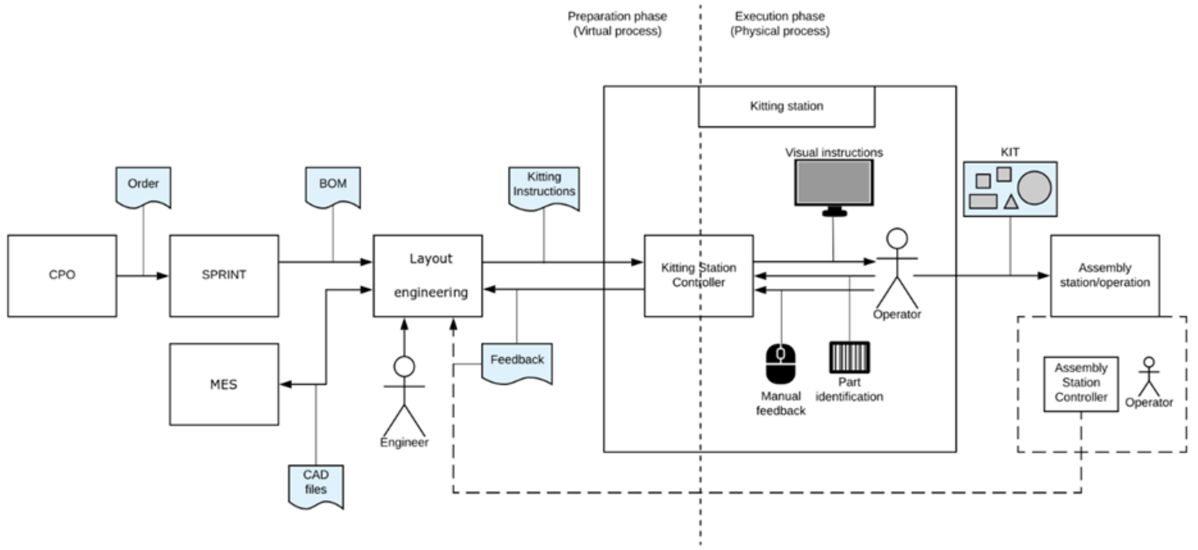


Figure 49 Use case 15 architecture.

The system of systems is based on the Eclipse Arrowhead framework and in future milestones aimed to integrate with the MES system already used in Volvo's plants via APIs. The first milestone however relies on data exported from the proprietary MES databases to more easily consumable generic data formats, like CSV and JSON.

While the main goal is to automate the layout planning, manual intervention, correction possibility shall be offered to the engineer. In later phases of the project, we may also implement a feedback interface for the operators, to increase efficiency by reducing the time required for error handling.

Figure 50 provides an overview of the connection between the different components which build up the smart kitting system.

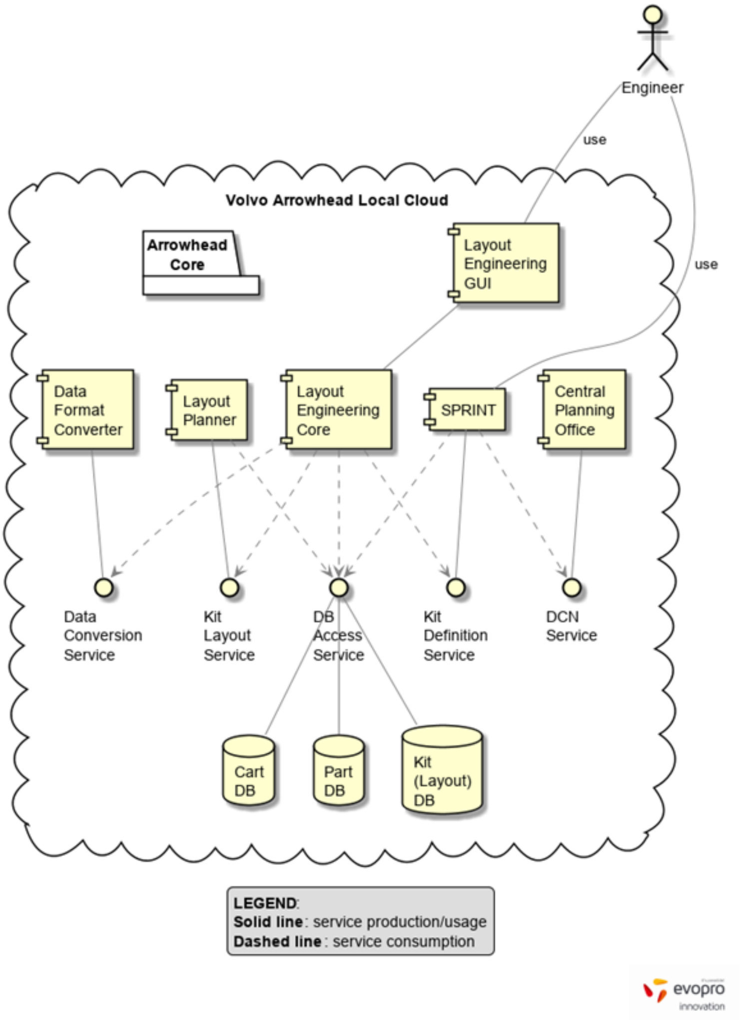


Figure 50 Overview of the connection between the different components in smarting kitting system.

The Smart Kitting solution is implemented as a standalone tool, which is intended to support logistic engineers at Volvo to manage and automate the process of kit-engineering. The layout planner is implemented as a plugin for the application. The Smart Kitting application relies on a Service-Oriented-Architecture. It implements Arrowhead-compliant services and open standard data formats. This enables easy integration with legacy tools used by Volvo, and so to create a toolchain covering the entire EP of the use case.
Additional work is needed here to implement Arrowhead adapters to the legacy engineering tools. In Figure 51 the techniques applied in engineering phases. Whereas, in Figure 52 the Volvo use case related engineering process.

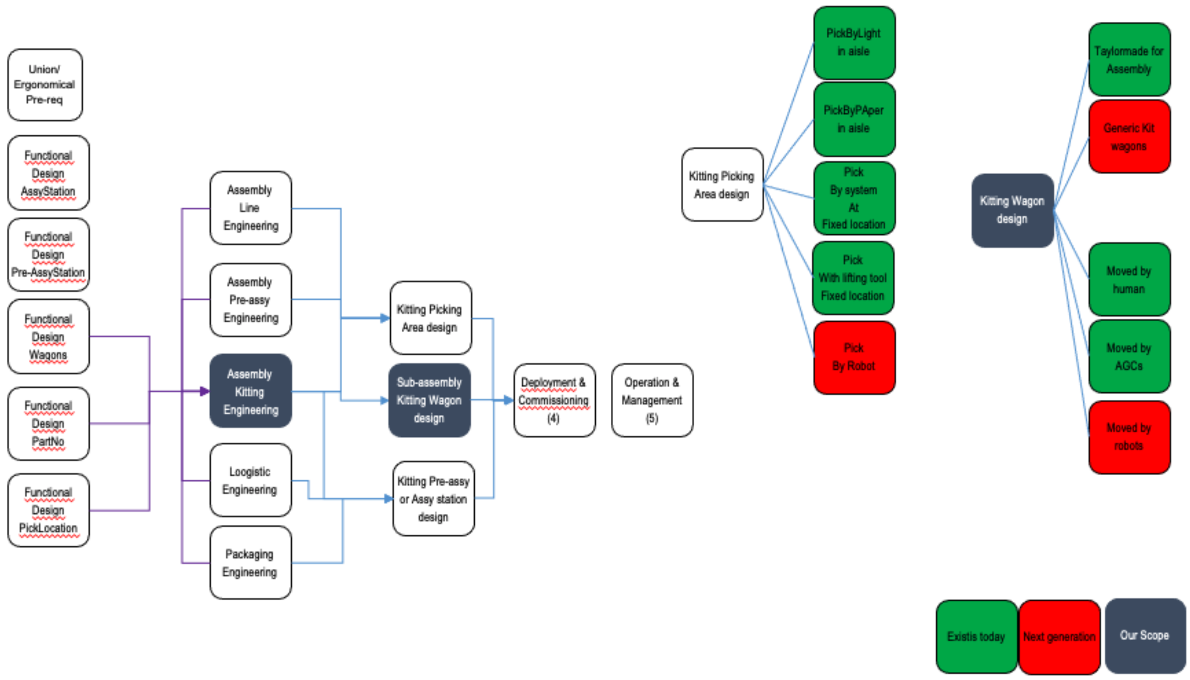


Figure 51 The techniques applied in engineering phases.

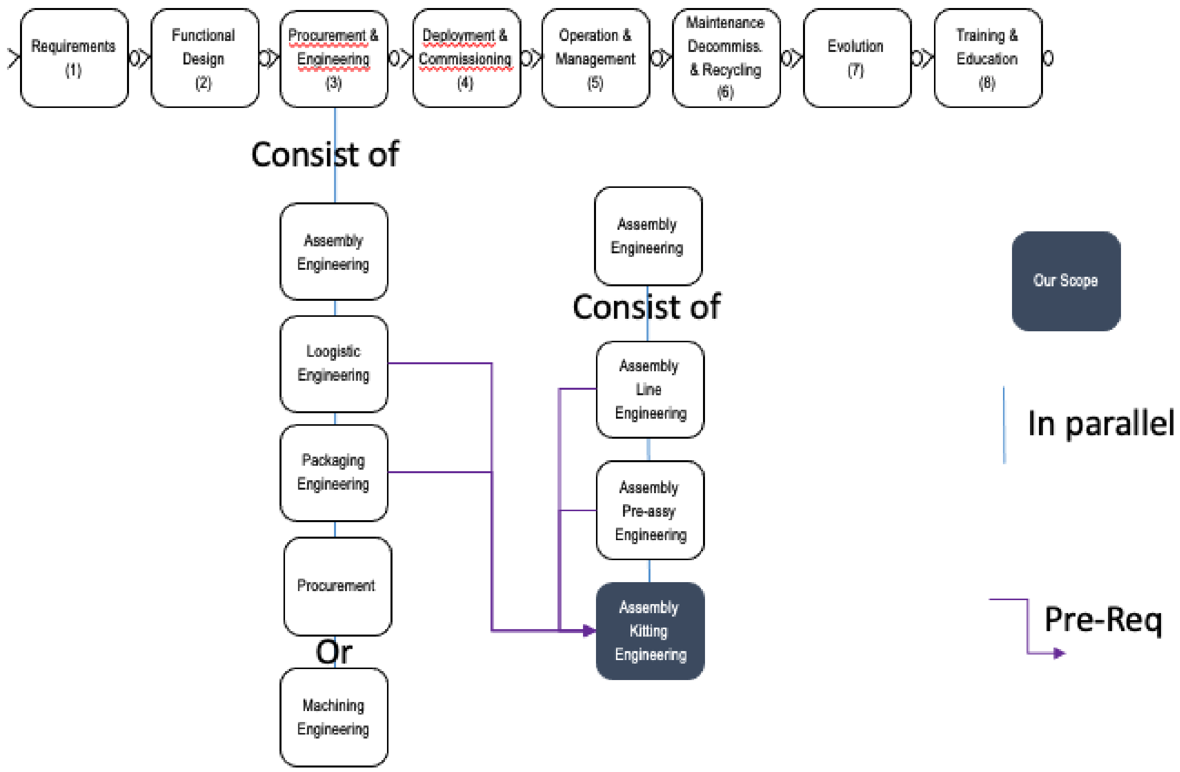


Figure 52 the Volvo use case related engineering process.

15.2 Engineering Process Description

In Figure 53 the AHT-EP of use case 15 which is composed by two stakeholders:

- Volvo Trucks Engineering: the engineering process covering the entire lifecycle of a truck.
- Assembly Kitting Engineering: the engineering process covering the lifecycle of a kit.

In the following some of the main description of the activities performed in the AHT-EP phases.
Requirements: the requirements set up by assembly, logistics and packaging engineering for kitting.
Functional Design: the functional design of the kit including kitting wagon design.
Procurement & Engineering: the physical preparation of the kit.
Deployment & Commissioning: the delivery of the kit to the assembly line.
Operation & Management: the application of the kit at the assembly line.
Evolution: Feedback from the assembly line and from the kit preparation to improve kit design.

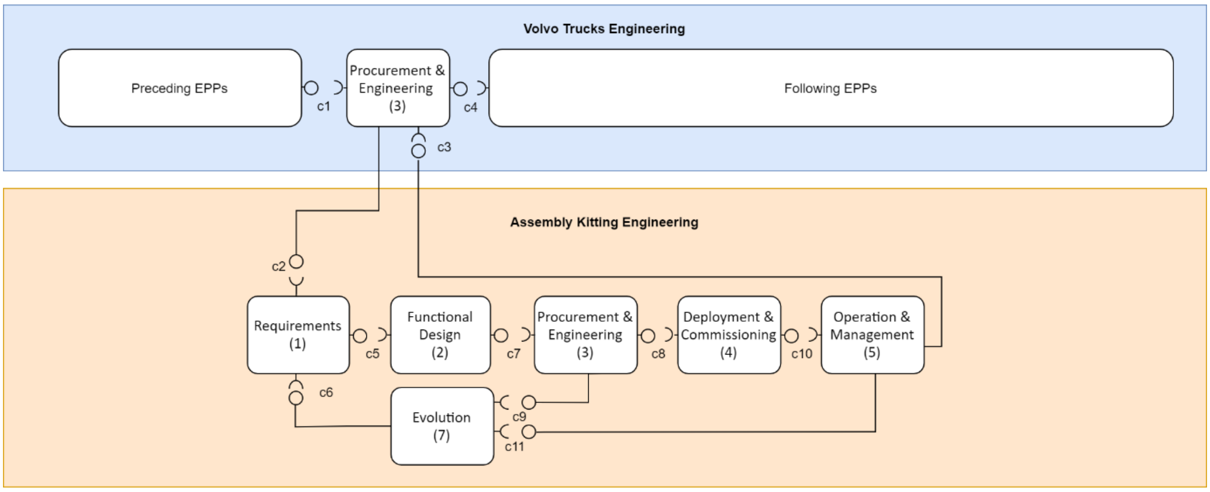


Figure 53 AHT-EP of use case UC-15.

The total process is shown in Figure 54.

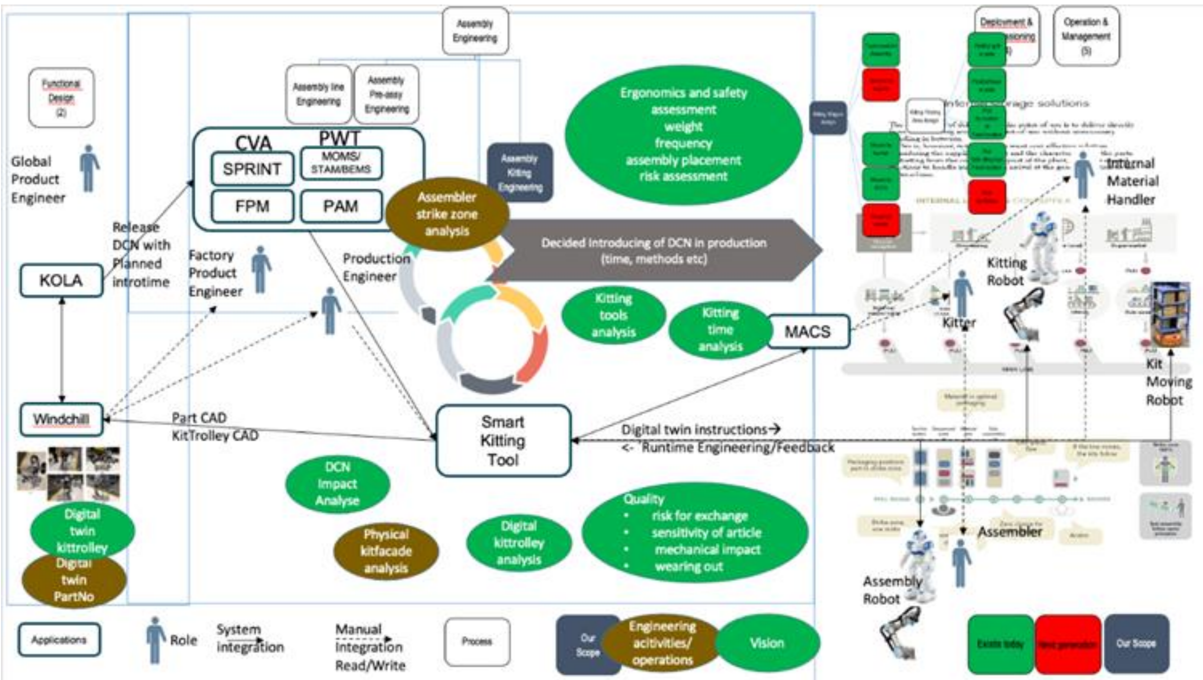


Figure 54 Future engineering process (vision).

Smart Kitting Tool: second iteration of the Proof of Concept is released. Integration with legacy tools used by Volvo is in preliminary stage. Experiments of automated data conversion are carried out. Performance evaluation of the layout planning algorithm has been performed.

There are several envisioned new systems, including a digital twin for the kitting operations that will use the resulting kitting layout provided by the smart kitting tool. Hence, the smart kitting tool needs to provide a service including that information.
Windchill will be the system providing information to the smart kitting tool.
Phases 6 and 8 are not covered as of now.
The use case team has not adopted any Engineering Process standards beside the AHT-EP model.

Before the project, all kitting activities were carried out manually. The process was broken down into sub-steps, but only in a functional manner (e.g. Kitting Picking Area Design, Kitting Wagon Design). Several legacy tools was used to assist the engineering work, but the information transport from one to the other required manual export/import.
By applying the AHT-EP, the kitting process was turned into a lifecycle management flow, which helped us to identify missing tools and interoperability issues. We created the Smart Kitting Tool to cover EPP2 and EPP7. EPP1 is assisted by legacy tools used already in production at Volvo, and we put significant effort into the automation of data transfer between these tools (from EPP1 to EPP2).

There is a lack of technology in the procurement and engineering step 3 of the complete truck assembly process. The lacking technologies include the smart kitting tool which is in focus for this use case. In addition, other systems are missing, e.g., a digital twin system.

In this use case we adopted the AHT-EP to Assembly Kitting Engineering, which is a sub-process of the life-cycle management of Volvo trucks. As seen in the AHT-EP diagram Figure 53, we focused on EPP3 of Volvo Trucks Engineering, and adopted the AHT-EP on a sub-process level by defining EPPs to the lifecycle of Kits, which can be considered as a temporary product of logistics engineering consumed by the assembly line.

Table 71 AHT-EP Phase focus of UC-15

Engineering process phase	Addressed/Focus
Requirements	n.s.
Functional design	n.s.
Procurement & Engineering	**Activity** Procurement & Engineering is connected to changes on the products done by the Functional Design. ONE change on the products is handled in a DesignChangeNote (DCN). A DCN includes changes on part-level form 1 up to thousands of part-numbers. From small changes on one small assembly to changes affecting Truck and components (engine, gearbox, etc.) at the same time in many different production plants. A proposed time for introduction of DCN is included. A global DCN coordinator (if needed) is appointed. Within the different plants The DCN is handed by DCN Coordinator (Normally the Local Product Engineer) that will gather information and keeping the progress of the

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
DCN content. The goal is that every stakeholder need to agree WHEN this change could be introduced into the plant. The when EVERYONE is ready (last time steer) that is set as REAL introduction date of the DCN.
The goal of this DCN is to introduces this with the minimal cost including purchasing, logistics internally and externally (transportation, packaging, downsizing, Supply to line methods, Assembler Strike Zone and balancing of the line (where to produce on line)
It is also the goal to secure tools and equipment's for the total change exists and is possible to use.
Included processes in the DCN process are:

Procurement: The Purchaser is responsible for the Procurement. They decide supplier and negotiate the price and secure the quality of the part according specifications. Agree with the supplier regarding packaging.
Packaging Engineering: The Packaging Engineer take care of the packaging of the part from supplier via external and internal logistics to the line side. Pallets, plastic boxes, paper boxes according Volvo standard or EDIFACT standard is used. I also includes the volume per total pallet and per package on the pallet in what "emballage" the supplier should pack it in. Taking the logistic cost in consideration
Logistic Engineering: The Logistic Engineer take care of the logistic of the part from supplier via external and internal logistics to the line side. It start with the proposed packaging at the supplier from the Packaging Engineer to from the request from the Production Engineer how the part will be handed over to Assembler. Pallet, smallbox, Kitted, sequenced as example.

The External Logistic Engineer focus on the logistic flow from Supplier to Goods Receiving.
The Internal Logistic Engineer focus on the logistic flow from Goods Receiving to Point if Us.

Assembly line Engineering: The Production Engineer is responsible for balancing the line meaning to secure that each station has same/similar assembly content in time to get the line moving as quick as possible (bottleneck calculations). They are also responsible to make EACH station as efficient as possible to decrease the assembly time. According the Workplace design principals.

In their tool according how to feed the station the uses different part supply methods:

Pallet delivery
Smallbox deliveries
Sequence part according Production Sequence on lineside
Pre-Assemblies of line feeding line with various part methods (pallet, smallbox, kitting)
Material kitting for Production Order (Our Scope)

The Production Engineer design the kitting process. In process he need to think about:

Kitting location(s)
Kitting wagon design
Kitting Pre-Assy
Assembly station design

</td></tr>
</table>

Engineering process phase	Addressed/Focus
Deployment & Commissioning	n.s.
Operations & Management	n.s.
Maintenance Decommissioning & Recycling	n.s.
Evolution	n.s.
Training & Education	n.s.

15.3 How the AHT-EP allows to match the Project and WP2 objectives

Smart Kitting Tool: this tool will assist engineers by managing kit iterations and layouts of part numbers on the kitting wagon. (EPP2) It will also provide possibilities for operators to give feedback on proposed kit instructions and use that information to evolve the kit layouts (EPP7) Layout Planner Algorithm.
Adapters for the integration of legacy tools: adapters are required to facilitate automation of data transport from legacy tools, especially from EPP1 to EPP2.

Table 72 UC-15 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The kitting: By adopting the Arrowhead Framework design principles, we created the Smart Kitting solution based on a Service-Oriented-Architecture. This implies that existing services can be replaced more easily during run time, which will be automatically discovered by the Smart Kitting Tool via the installed Arrowhead Local Cloud. For example, the Layout Planning tool can be upgraded independently from the Smart Kitting frontend and the new version can be deployed in run time.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The production process already in the baseline contains several stake holders: Volvo internal stakeholders, external stakeholders (e.g., PTC providing the Windchill product). After the introduction of the smart kitting tool into the engineering process, EvoPro is a new stakeholder. There will be additional internal stakeholders at Volvo as shown in Figure 54. The development of the smart kitting tool: By declaring the use case and inviting other parties to develop an automated solution to the Kitting Process, Volvo invited to multi-stakeholder cooperation. Each stakeholder creates Arrowhead-compliant services, which are used to exchange information via the Arrowhead Local Cloud.
Obj. 3 - Handling of substantially increased number of I/O's due to much	Although there are a few more stakeholders, there is no substantial increase in I/O.

WP2 Objective	Focus & Planed actions
more fine grained automation	
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The smart kitting tool is documented so that it can be used for training of the kitting engineering personnel. The GUI is refined as part of functional design, which connects to the training activities. The training activities will not be substantial and will be postponed to after the project.

Table 73 UC-01 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	n.s.
Obj. 2 - Interoperability for IoT and SoS engineering tools	n.s.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	n.s.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	n.s.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	n.s.
Obj. 6 - Training material (HW and SW) for	n.s.

Project Objective	Focus & Planed actions
professional engineers	

15.4 Engineering Process analysis

Table 74 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-15.

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	U	C	C	C
OBJ-AHT #1			X				X	X	
OBJ-AHT #2			X		X				
OBJ-AHT #3			X						
OBJ-AHT #4			X						
OBJ-AHT #5			X		X				
OBJ-AHT #6		X	X					X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2	X		X		X				
OBJ-WP2 #3	X		X		X		X		
OBJ-WP2 #4		X	X					X	

16. UC-16 Production Support, Energy Efficiency, Task Management, Data Analytics and Smart Maintenance (IFD)

This Use Case is subdivided in five sub use cases which are discussed in the following five subsections.

16.1 UC-16.1 Equipment and Energy Data Management (IFD)

Most of the equipment in the Semiconductor Industry is complex and requires detailed monitoring to ensure smooth production processes. In many cases the equipment provides already a good set of values, which can be used for monitoring.
Collecting those values provides a vast number of possibilities to learn and streamline monitoring application to even perform predicative maintenance applications.
However, some values are not provided by the equipment out of the box but need to be measured in their environment. One example is the power consumption.
Additionally some older equipment used in manufacturing only provide a minimal set of events for monitoring the production processes. Those events can be generated by using external sensors mounted to the equipment.

This leads to the problem of accurately relate the data provided by the machine and the data observed in the environment, storing them in an easily accessible and processable way, and relay events to the equipment integration.

16.1.1 Overall description of the UC-EP

The Architecture for the Use Case can be visualized in two views.
In the first view, the hardware view, the involved systems and there functionality is presented (Figure 55).

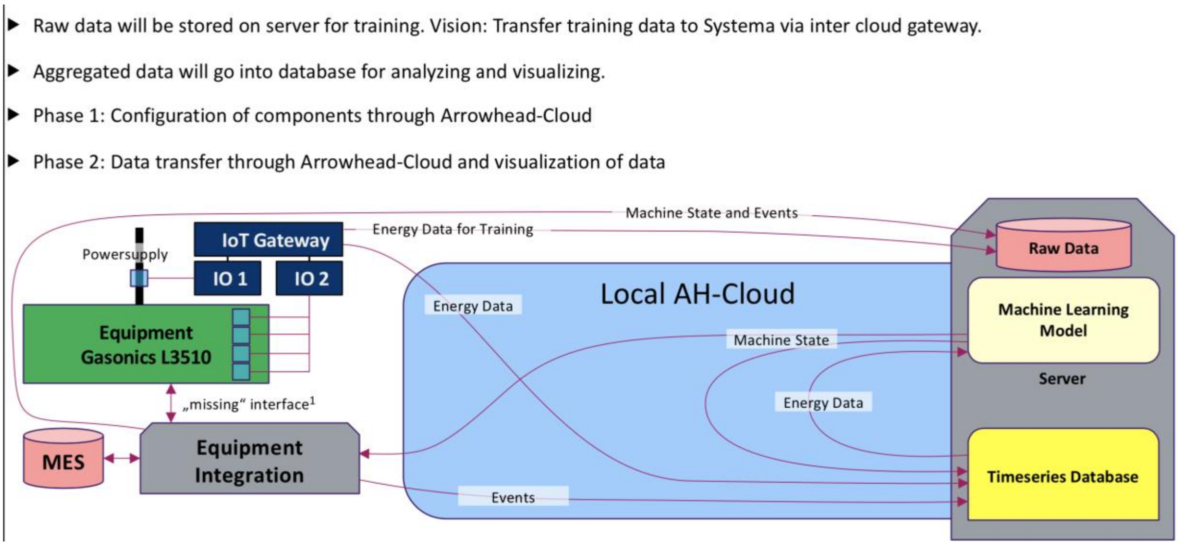


Figure 55 The hardware view of UC-16.1

In the second view, the deployment view, the involved software systems and their interaction is presented (Figure 56).

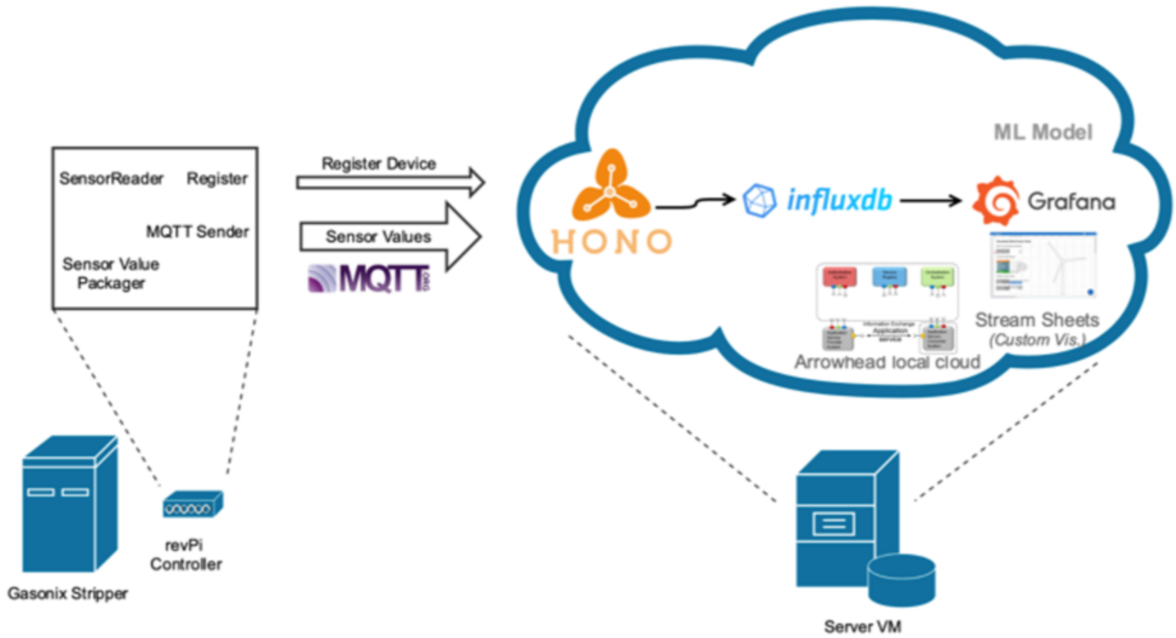


Figure 56 the deployment view of UC-16.1.

The Energy Data use case is about fetching data from energy sensors attached to a semiconductor production machine.
The idea is to attach a revPi controller to it which collects the sensor data and acts as a gateway to send the sensor data to an Eclipse Arrowhead local cloud backend.
In the arrowhead local cloud, the sensor data is received by an instance of Eclipse Hono, an IoT messaging hub. This hub translates the incoming MQTT messages to a common AMQP 1.0 message, which in turn is forwarded to a time series database that stores this data for later analysis.
A first use of the data is realized with a dashboard that visualizes some key parameters of the incoming data.
The backend is orchestrated with Arrowhead mandatory core services.
Use Arrowhead Environment for system integration to establish a monitoring application.
In the following the goals of Backend Architecture:

- Store telemetry data within a unified database or data lake
- Leverage a scalable messaging infrastructure to cope with high volume of messages
- Extend the number of supported messaging protocols if necessary
- Provide a device update management to augment the functionality of the controllers if necessary

The architecture presented in the section above is planned to be described by means of SysML and the Arrowhead protocol.

The use case focuses on the integration of the Gasonics stripper component. Data provided is provider dependent. The use case facilitates several components of the Eclipse Arrowhead framework and uses MQTT ans communication protocol and json for internal data representation. In the backend the Arrowhead integrations as well as selected tools from the Eclipse IoT stack is used. Data ingest is realized with Eclipse Hono, whereas persistence is planned to be realized with InfluxDB. From a toolchain point of view is it planned to facilitate the SysML profile for Arrowhead to replay the initial engineering steps already done with established tooling.

16.1.2 Engineering Process Description

The Figure 57 shows the whole engineering phases for UC16.1 (Task 9.4.1 “Equipment Data) which is one sub use case concerning UC16. Below AHT-EP structure of the UC Equipment Data pictures the different roles of the stakeholders.
The results of the verification and validation of the AHT-EP for the UC Equipment Data is that we have two different phases within the UC.
The first phase describes the engineering model of the demonstrator.
The second phase describes the roll out of the demonstrator in a realistic semiconductor environment. This process is typical within the semiconductor industry.

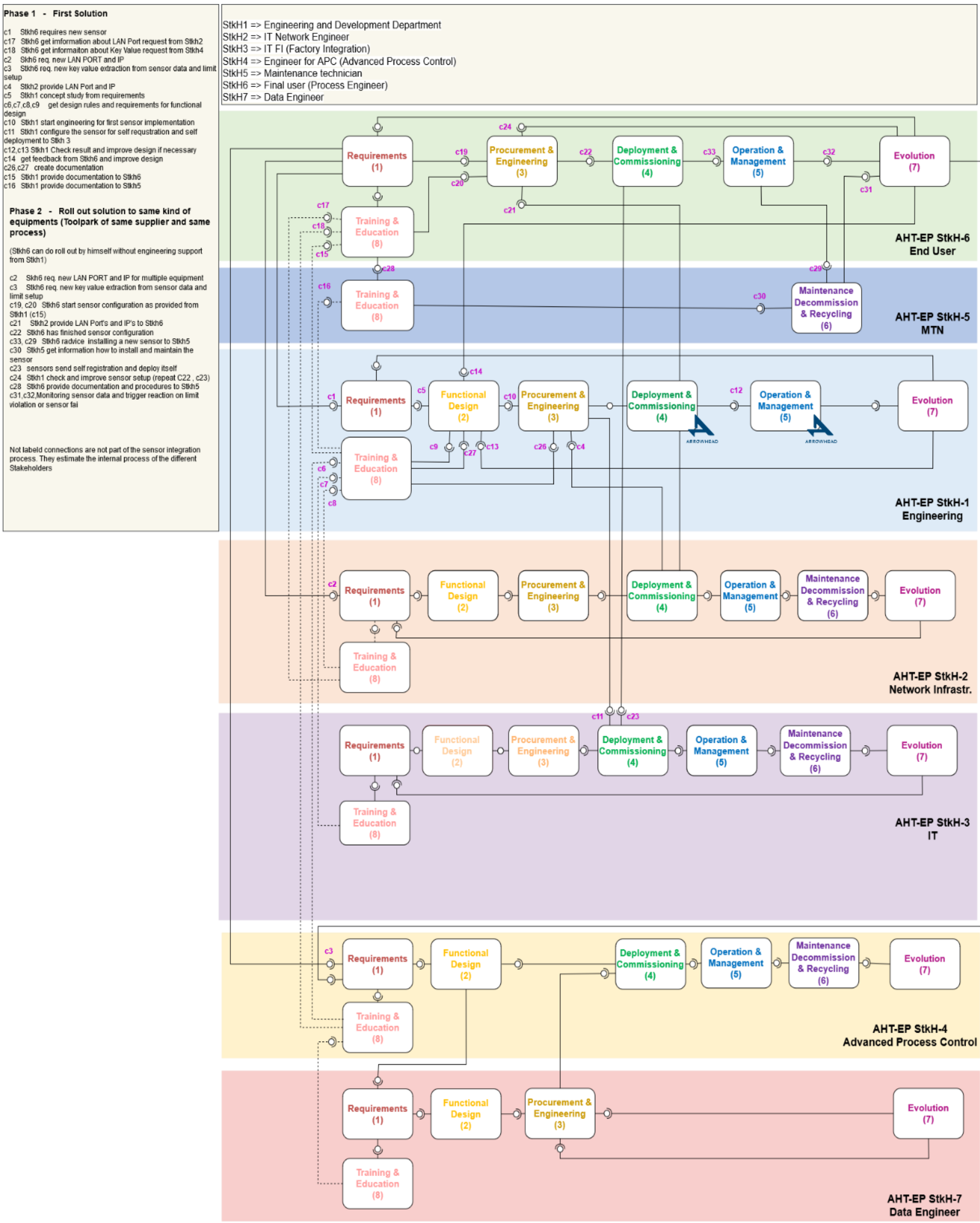


Figure 57 AHT-EP of use case UC-16.1

Currently, there is no framework in use to manage the engineering process for sensor data ingestion pipelines. We roughly follow the Unified Process as lived in the Eclipse Process Framework (https://www.eclipse.org/epf/) for the development of the backend services. However, a holistic approach for the development and rollout of the whole system is not available so far.

The Use Case execution within the Arrowhead Tools project mainly touches the phases Requirements, Functional Design, Procurement & Engineering, Operation & Management and Training & Education, whereas the other phases will only be touched. The first three phases are executed within the Unified Process and ordinary MS Office tools are used for documentation and communication. Training is performed via power point slide decks as well as face to face trainings, which are recorded and provided for later use.

Our code is meant to integrate with the Arrowhead tools framework by providing and consuming services via its service registry and the mandatory core services.

The power supply measurement devices are not providing its data in a SoA manner. Also the health and runtime data of the Gasonics stripper device does not provide its data as a Service registered in an Arrowhead service registry.

Integration of Message Gateway for scalable data ingestion. We are currently developing an integration code for the Eclipse Hono project that allows an integation with the Arrowhead framework.
Configuration System: We integrate the Eclipse Hawkbit project. Integration code is already available, which is currently about to be contributed to the Eclipse Arrowhead project.
Digital Twin Device Abstraction: We are developing an integration of the Eclipse Ditto digital twin solution within the arrowhead framework. This is still work in progress.
We are working on an integration of the SysML Arrowhead tools profile with Eclipse Vorto for using device abstractions of the Vorto repository and leveraging its code generator capabilities.

Table 75 AHT-EP Phase focus of UC-16.1

Engineering process phase	Addressed/Focus
Requirements	**Activity** The requirements elicitation process started in collaboration with WP1 for general requirements for the framework and toolchain and a use case specific requirements elicitation resulting in a set of required parameters and functionality. A requirements definition file in Excel is created which is passed to the next phase. 1. *Gateway Software Components*: n.a. 2. *Hardware Components*: Define physical signal(s) to measure and the required sampling rate for specific use case (Current@0.5kHz-1kHz, Voltage@1Hz). 3. *Backend Software Components*: Requirements are collected in Excel sheets and detailed in accompanied word and power point documents and tracked In issue tracker (Atlassian JIRA). As some of the components are open source, additional issues are created in external issue tracking systems (github.com) to facilitate development there. 4. *Equipment Integration (EI)*: n.a.
Functional design	**Activity** The results of the requirements elicitation phase were used to come to an initial architecture depicted in A.b of the WP12410_survey. Part of this phase was also an analysis and evaluation of the Arrowhead Framework and its various integrations in order to decide about the tools to be used in the use case realization. The requirements in the Excel sheet are taken and transferred to a specification document, which details the requirements. 1. *Gateway Software Components*: Check sensors for supported data formats. Convert format of data to required format. (JSON, CSV) Aggregation of data.

Engineering process phase	Addressed/Focus
	2. *Hardware Components*: Selecting sensor(s) able to measure required physical signals with the required frequency. (Advantech WISE-4010, PicoScope XXXX) Select Hardware for processing and conversion of data formats (Kunbus RevPi). 3. *Backend Software Components*: For the functional design word documents as well as diagrams in UML and non-standardized notations are used. 4. *Equipment Integration (EI)*: n.a.
Procurement & Engineering	**Activity** A procurement process for some hardware components is initiated and orders are placed for some software components. The engineering for the hardware extensions as well as for the backend systems is initiated. The document is transferred to the procurement department, which uses the document to send out requests for quotations. Results are quotations, of which one is selected and an order is placed. The result of the order is an implemented sub component. However, this phase runs in cycles in which several words documents and emails are exchanged which detail the requirements and spec document. 1. *Gateway Software Components*: n.a. 2. *Hardware Components*: Hardware is supplied by external partner specific to use case. 3. *Backend Software Components*: Procurement is done with SAP and internal procurement systems. The engineering for the backend is done with usual software development tooling such as an integrated development environment (InteliJ IDEA), revision control systems (git, stash), builds are automated with Jenkins build server. 4. *Equipment Integration (EI)*: n.a.
Deployment & Commissioning	**Activity** The implemented sub component is packaged as a tar.gz package, which is copied on a central ftp server, from which the component is copied to the actual system on which the component runs. 1. *Gateway Software Components*: Software on the PoC is deployed by external partner using their standard process (SSH, VPN). 2. *Hardware Components*: Configure static DHCP for devices. 3. *Backend Software Components*: ssh, ftp, kubernetes, kubectl. 4. *Equipment Integration (EI)*: Webservice deployed in EI environment for communication with AH environment.
Operations & Management	**Activity** The running sub component generates log files in text format and the running servers generate reports on health status. Both are in text format and written to the file system. 1. *Gateway Software Components*: TBD 2. *Hardware Components*: TBD 3. *Backend Software Components*: Top, syslog, kubectl, Prometheus. 4. *Equipment Integration (EI)*: TBD

Engineering process phase	Addressed/Focus
Maintenance Decommissioning & Recycling	**Activity** If the sub component needs a bug fix an order is again placed and the process is rerun as described above. 1. *Gateway Software Components*: Maintenance for PoC done by engineering and external partners. 2. *Hardware Components*: Maintenance for PoC done by engineering and exernal partners. 3. *Backend Software Components*: As dev cycle, + cve databases (NVD etc.). 4. *Equipment Integration (EI)*: n.a.
Evolution	**Activity** If the sub component needs a new features an order is placed. 1. *Gateway Software Components*: TBD 2. *Hardware Components*: TBD 3. *Backend Software Components*: As dev cycle. 4. *Equipment Integration (EI)*: Events from IoT Gateway sent to EI and in reverse direction.
Training & Education	**Activity** Training material for new sub components are part of the initial order. It is mostly provided as word documents and accompanied by a power point slide deck. For initial ramp up a training by the solution provider is done. 1. *Gateway Software Components*: n.a. 2. *Hardware Components*: n.a. 3. *Backend Software Components*: Word, power point, internal streaming tooling. 4. *Equipment Integration (EI)*: n.a.

16.1.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 76 UC-16.1 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	In the use case we plan to model the system of systems of the use case with SysML and apply tooling to generate parts out of the models. So we move some parts of the engineering to the run-time. **Planned actions** • AHT-EPP 1: Some requirements defined. • AHT-EPP 2: SysML modelling. • AHT-EPP 3: Demonstrator for this in work together with colleagues from Bosch.IO. • AHT-EPP 4: Model execution and local cloud update.

WP2 Objective	Focus & Planed actions
	• AHT-EPP 5: Will be done within next phases of AHT.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	By applying the Eclipse Vorto-SysML integration we reuse device models already available in the public Vorto repository and take into account publishing some of our device models into the repository as well. **Planned actions** • AHT-EPP 2: Research of available device models, upload designed models To Vorto repository. • AHT-EPP 3: Demonstrator in Work. • AHT-EPP 4: Need decision from IT Department which IoT Frameworks will meet the requirements. • AHT-EPP 5: Will be done within next phases of AHT.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Via an automated data collection and storage in a backend we have the possibility to process more data points with a bigger number of parameters as this was possible before. **Planned actions** • AHT-EPP 2: Required services and tools are addressed. • AHT-EPP 3: Will be tested. • AHT-EPP 4: Concept for IoT Framework depend on demonstrator result and the defined requirements. • AHT-EPP 5: Data made available via standardized APIs to allow integration into data processing pipelines.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Access and processing of the data retrieved from the sensors will be described in some training material, which is planned to make available as online training material for later reference of the technicians. **Planned actions** • AHT-EPP 1: Not part of the use case, New IoT Framework project ist upcoming. • AHT-EPP 4: Onboarding of technicians to the newly available data sources. • AHT-EPP 8: Providing online course material for later reference.

Table 77 UC-16.1 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Use of available Open Source Eclipse IoT components and Eclipse Arrowhead framework. Apply Open Source Tooling for Modelling tasks, where possible. Publically share and consume modelling artefacts (Eclipse Vorto Models).
Obj. 2 - Interoperability for IoT and SoS engineering tools	Apply integration components of Eclipse IoT into Eclipse Arrowhead framework. Apply SysML for modelling and linking of engineering artifacts.

Project Objective	Focus & Planed actions
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Generate integration code to Eclipse Hono messaging component via Eclipse vorto models (integrated into Arrowhead framework). Implement custom data converters to ingest data into backend.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Probably not applicable.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Use of proven Security Systems such as Keycloak for user and system data management in the backend. Application of Secure transport for all communication protocols.
Obj. 6 - Training material (HW and SW) for professional engineers	Parallel to development activities create training material and tailor and improve it constantly so that in the end a training as well as its material is available for technicians.

16.1.4 Engineering Process analysis

Table 78 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-16.1

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X		X			X	
OBJ-AHT #2		X	X						
OBJ-AHT #3			X		X				
OBJ-AHT #4									
OBJ-AHT #5			X		X				
OBJ-AHT #6	X	X	X	X				X	
OBJ-WP2 #1	X	X	X	X	X		X		
OBJ-WP2 #2		X	X	X	X				

OBJ-WP2 #3		X	X	X	X				
OBJ-WP2 #4	X	X	X	X				X	

16.2 UC-16.2 I/O Link SC sensors (IFD)

The Semiconductor Industry has been using the SEMI Equipment Communication Standard / Generic Equipment Model (SECS / GEM) for quite some time. This is the most advanced standard so far. Almost every equipment in our wafer producing fabs are speaking this standard protocol. Equipment or also called Tools are controlled remotely by Infineons Equipment Automation Framework (EAF).
Beside the basic functionalities also physical sensor streams will be transferred using this protocol to the Advanced Process Control (APC) and the Statistical Process Control (SPC) databases. Additionally a Run to Run (R2R) Framework is even adjusting process parameters during the wafer production process in-situ. Going back 20 years this was the most advanced protocol existing. Now it lacks in the possibility adding IoT Sensors around the production tools. To close this cap the EAF framework became able to speak some few other protocols like ModbusTCP, direct Siemens PLC Communication, reading from files and receiving ASCii data from TCP Sockets.
In the past years other digital sensor protocols have been pushed forward.
The latest most important Sensor Interfaces are:

- Actuator-Sensor Interface - AS-i (First fieldbus for sensor integration).
- Single-drop digital communication interface for small sensors and actuators (SDCI) better known by the name - IO-Link.
- Message Queue Telemetry Transport – MQTT.
- OLE for Process Control (OPC)- Unified Architecture - OPC-UA (OLE= Object Linking and Embedding).
- Sensor Measurement List – SenML.
- REpresential State Transfer – REST Interface.

These standards are further developments of more than 30 existing field bus communication protocols. Well known protocols are for example: Profibus, ProfiNET, CAN, CAN-Open, MODBUS , DeviceNET, EtherCAT, INTERBUS, LON and ARCNET.

Goal of the use case is to minimize sensor integration effort by transforming sensor signals from different protocols in a self-describing sensor language using arrowhead as translator.

In the semiconductor industry there are already a lot of standards. They are published by or can be bought from the SEMI (Semiconductor Equipment and Materials International) under http://www.semi.org/eu/ and http://www.semi.org/standards.
Semiconductor Equipment and Materials International (SEMI) is a global trade organization of manufacturers of equipment and materials used in the fabrication of semiconductor devices such as integrated circuits, transistors, diodes, and thyristors. Among other activities, SEMI acts as a clearinghouse for the generation of standards specific to the industry and the generation of long-range plans for the industry.
In the following, the reference of some of the most important standards:

- SEMI E5-0709E SEMI Equipment Communications Standard 2 Message Content (SECS-II)
- SEMI E30-0307E2 Generic Model for Communications and Control of Manufacturing Equipment (GEM)

- SEMI E37-1109 High-Speed SECS Message Services (HSMS) Generic Services
- SEMI E116-0707E Specification for Equipment Performance Tracking
- SEMI E120-0310 Specification for the Common Equipment Model (CEM)
- SEMI E125-0710 Specification for Equipment Self Description (EqSD)
- SEMI E132-0310E2 Specification for Equipment Client Authentication and Authorization
- SEMI E134-0710 Specification for Data Collection Management

16.2.1 Overall description of the UC-EP

In Figure 58 the functional blocks of the use case that can be described as follows:

1. sensor selection and installation (In our case IFM AL1350 IO-Link Master + IO-Link TV7105 temperature Sensor, + IO-Link Vibration Sensor VVB001).
2. Web based app registers the sensor and generates a hash id.
3. Web based app configures the AL1350 Sensor interface.
4. AL1350 publishes temperature and vibration data to the MQTT broker using the hash id as topic.
5. EAF registers itself and ask for hash id's for a certain tool.
6. EAF subscribes to all relevant has id's to get all sensor data.

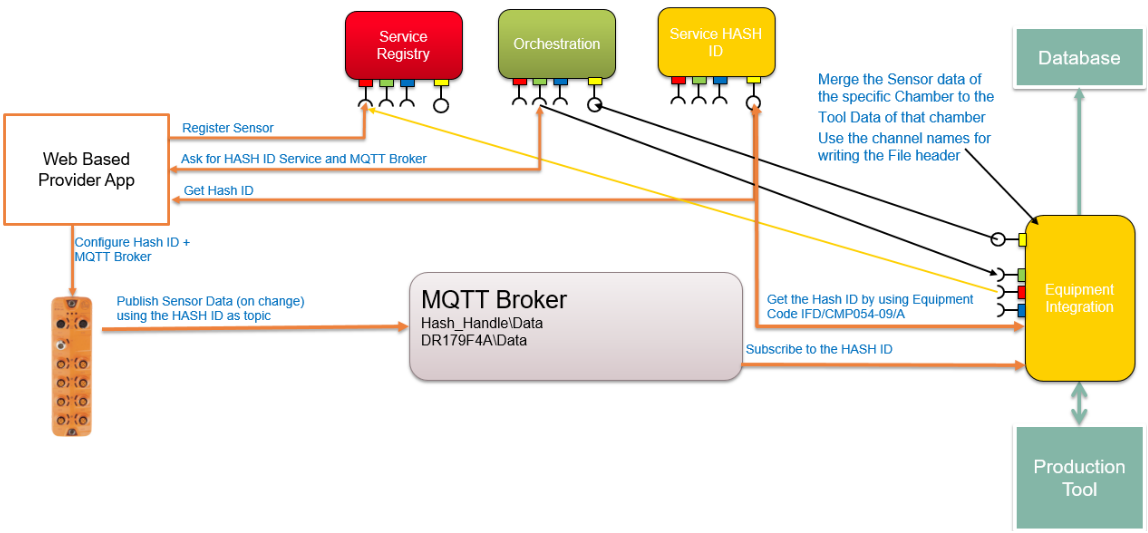


Figure 58 Functional blocks of the UC-16.2

If the Sensor cannot provide data in SenML format translator apps will be required.
For example the functional blocks of the use case looks as follows:

1. sensor selection and installation (In our case IFM AL1350 IO-Link Master + IO-Link TV7105 temperature Sensor, + IO-Link Vibration Sensor VVB001).
2. Web based app registers the sensor and generates a hash id.
3. Web based app store sensor name, Signal name, hash id's , tool name, chamber name, protocol name, location data (Building , floor,) an other metadata into a database.
4. Web based app configures the AL1350 Sensor interface to send the data to the translator app / alternatively the AL1350 can be configured manually.
5. translator app is asking the "web based app" for the metadata.

6. Translator app (which is already registered within the arrow head framework) receives sensor data and publishes the temperature and vibration data (using the hash id from the Web based app) in SenML format.
7. EAF registers itself and ask for hash id's for a certain tool.
8. EAF subscribes to all relevant has id's to get all sensor data.

The use case engineering process starts with collecting the different sensor protocols used within the fab floor. And getting an overview of the equipment automation framework and its components. As well as collecting relevant databases to store the data and provide it to higher levels of data analyses (Machine learning….).
Then, appropriate sensors speaking IoT sensor protocols are selected, e.g. IO-Link and MQTT. Next Step is to prove available functionalities within the Eclipse Arrowhead framework (e.g. databases, Sensor protocols) The next step is diagram preparation including required data converters. The last step, is to prove adaption of data to Semi Standards. E5, E30, E37, E125.

16.2.2 Engineering Process Description

The Figure 59 shows the whole engineering phases for UC16.2 (I/O-link sub use case) which is the main driving sub use case concerning UC16.
In this use case we have seven stakeholders:
StkH1 => Engineering and Development Department
StkH2 => IT Network Engineer
StkH3 => IT FI (Factory Integration)
StkH4 => Engineer for APC (Advanced Process Control)
StkH5 => Maintenance technician
StkH6 => Final user (Process Engineer)
StkH7 => Data Engineer

Below the different roles of the stakeholders are described within two phases of the project.

Phase 1 - First Solution
c1 Stkh6 requires new sensor
c17 Stkh6 get information about LAN Port request from Stkh2
c18 Stkh6 get information about Key Value request from Stkh4
c2 Skh6 req. new LAN PORT and IP
c3 Stkh6 req. new key value extraction from sensor data and limit setup
c4 Stkh2 provide LAN Port and IP
c5 Stkh1 concept study from requirements
c6, c7, c8, c9 get design rules and requirements for functional design
c10 Stkh1 start engineering for first sensor implementation
c11 Stkh1 configure the sensor for self requstration and self-deployment to Stkh 3
c12, c13 Stkh1 Check result and improve design if necessary
c14 get feedback from Stkh6 and improve design
c26, c27 create documentation
c15 Stkh1 provide documentation to Stkh6
c16 Stkh1 provide documentation to Stkh5

Phase 2 - Roll out solution to same kind of equipment (Toolpark of same supplier and same process)
(Stkh6 can do roll out by himself without engineering support from Stkh1)
c2 Skh6 req. new LAN PORT and IP for multiple equipment

c3 Stkh6 req. new key value extraction from sensor data and limit setup
c19, c20 Stkh6 start sensor configuration as provided from Stkh1 (c15)
c21 Stkh2 provide LAN Port's and IP's to Stkh6
c22 Stkh6 has finished sensor configuration
c33, c29 Stkh6 advice installing a new sensor to Stkh5
c30 Stkh5 get information how to install and maintain the sensor
c23 sensors send self-registration and deploy itself
c24 Stkh1 check and improve sensor setup (repeat c22, c23)
c25 Stkh1 provide new requirements to Stkh4 (e.g. change limit setup, new key value calculation)
c28 Stkh6 provide documentation and procedures to Stkh5
c31, c32 Monitoring sensor data and trigger reaction on limit violation or sensor fai

Not labeled connections are not part of the sensor integration process. They estimate the internal process of the different Stakeholders.

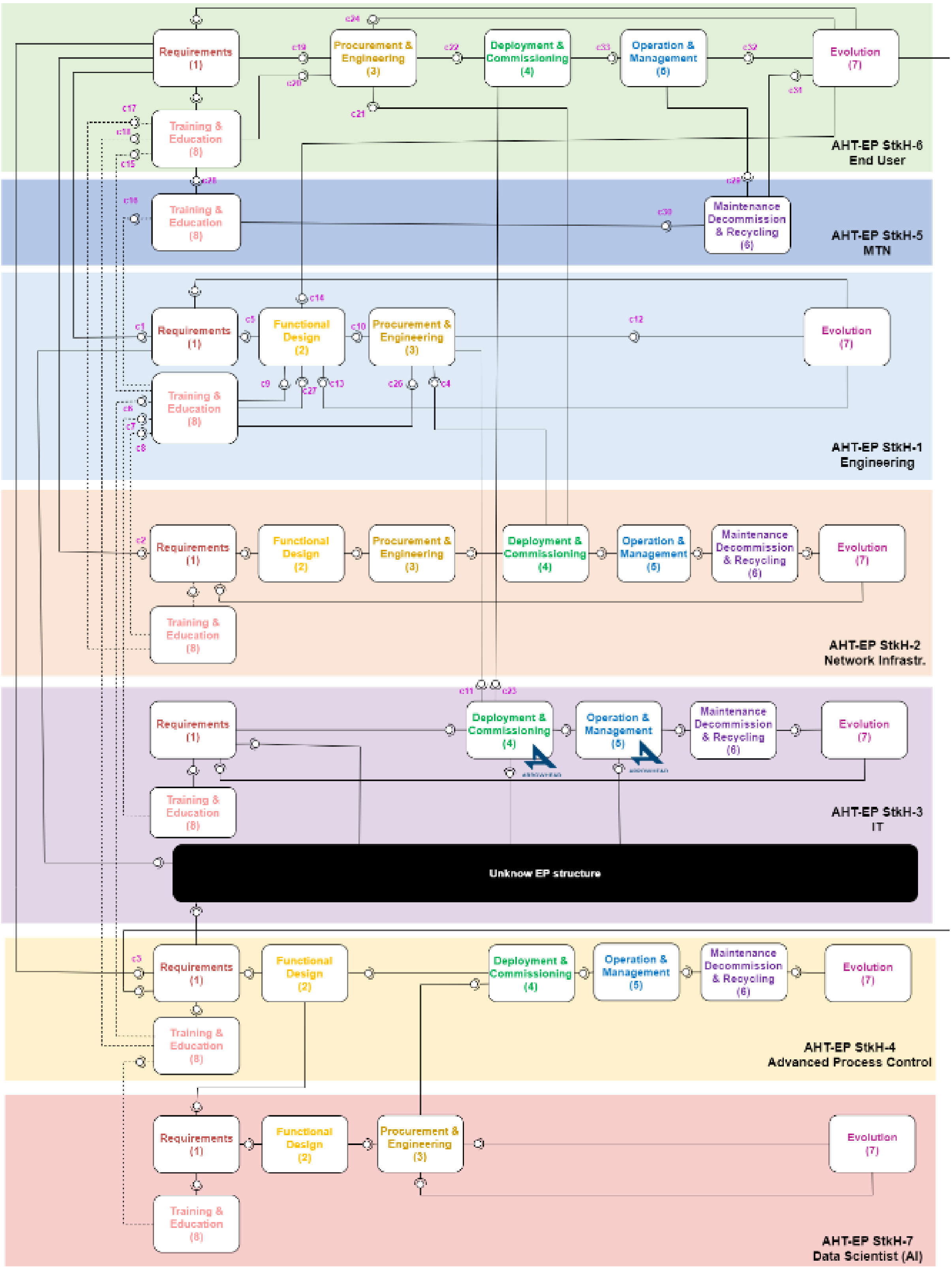


Figure 59 AHT-EP of use case UC-16.2

Currently, there is no framework in use to manage the engineering process for sensor data implementation. The best existing interface seems to be SenML. This data format provides most relevant fields like sensor value, name, unit and more.

See: https://tools.ietf.org/html/rfc8428. For PLC programming IEC61131-3 is used.

The PnP sensor integration approach mainly spans across the domains of Procurement & Engineering, i.e., the selection and acquisition of the required measurement hardware, the Deployment & Commissioning, i.e., the configuration and installation of the measurement hard- and software, the Operation & Management phase, i.e., the continuous data acquisition, processing and presentation (machine diagnosis), Maintenance and Evolution (hard- and software support and improvement) as well as training and education, e.g., groups of experts for sensor integration over all Infineon sites.

Sensor protocol translating services has to be developed. The query service for metadata to find matching sensor channels on the MQTT Broker is in work by HTW Dresden. Looking for partners to develop missing services or check if they are exist already. Additionally a requirement is the C# programming language to maintenance reasons. Services except the core services need to be built in C#.

The PnP sensor use case aims to shorten the engineering time and effort. So The Main focus is on EP3. EP1 and EP2 need to be done manually to find the best solution for the very particular sensor use cases. EP6 is not considered for AHT due to there is another system in place.

SECS/GEM or Interface A Communication protocols. (SEMI E5, E30, E37, E125) InterfaceA (SEMI E125) has the possibility of self-description. But no other Industrial sensor beside Semiconductor Equipment speaks this protocol. As result the sensor integration lacks on modern dynamic ways of integration. The result is high integration effort in terms of configuration, installation and data access for analyses and further more. To provide data analysts with more and better information the goal is to reduce the configuration effort to one point, to the sensor. The sensor describes where it belong, what its provide and where data has to go. To add an additional channel / sensor it's just some clicks less than a minute.

Table 79 AHT-EP Phase focus of UC-16.2

Engineering process phase	Addressed/Focus
Requirements	**Activity** Collected specific details for the sensor use case. Which physical signal to measure? i.e. Voltage, Current, Pressure, Flow Temperature, Vibration a.s.o, Data Rates, Output Frequency, Where is the Sensor belonging to, Where to store the data, algorithms, purpose of the additional sensor signal. *Manual Work*: • Collect required Sensor Signals • Understanding purpose of the sensor (goal) • Check ROI for Sensor Installation
Functional design	**Activity** Match Sensor selection with available sensor hardware and interfaces / protocols. i.e. Hardware: µController Box or PLC (Siemens S71200, Beckhoff, ABB AC500) Interfaces/Protocols: Siemens API, ModbusTCP, OPC DA, ASCii data (General TCP).

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>Other Software (Putty, APC Tracker etc., Softing OPC Toolbox):
Test Sensor Data Output (Putty, MQTT Broker etc.)
Manual Work:
Select Interfaces and Protocolls for POC
Create Service Order for new LAN Port and IP
APC Trent Suite RunToRun Framework (R2R):
Request for new Sensor Integration
New key numbers
Sensor Software or Configuration Tools /IDE’s (Arduino, ABB Automation Builder, DAVE4 / Eclipse, IFM LineRecorder Device):
programming µControllers or PLC if signal processing or conversion is necessary
Equipment Automation Framework (EAF):
Request for new data interfaces if not existing (TRAIL Request)</td></tr>
<tr><td>Procurement & Engineering</td><td>Activity
This Block includes the ordering of hardware as well as ordering all in house services from different departments (E.g. IT Software, Network Connections, Data key number configuration) Also it includes programming and configuring the sensor hardware or controllers. Sensor Installation and start-up. Request al necessary software changes i.e. new protocols for EAF if the sensor could not be ordered with an existing one. Request LAN Port and IP Address or Name if DHCP is possible. Request for APC Suite (Advanced Process Control) configuration. Selecting Input streams and create calculated key numbers per stream i.e. Mean, min max, trend etc., Installation of the sensor on the first tool, test dataflow and improve functionality as needed.
Manual Work:
Install Hardware, Set IP addresses etc.
APC Trent Suite RunToRun Framework (R2R):
Configure data channels and key number calculations (manually by using UI).
Sensor Software or Configuration Tools /IDE’s (Arduino, ABB Automation Builder, DAVE4 / Eclipse, IFM LineRecorder Device):
Configure sensor hardware i.e. IFM AL1342
Equipment Automation Framework (EAF):
Build missing interfaces / functionalities as requested. Depends on priority of the project
Configure data interface (Done by IT manually)</td></tr>
<tr><td>Deployment & Commissioning</td><td>Activity
Ordering and installing hard and software for roll out the solution to multiple equipment/tools. Request for IT to rollout the tested setup to the desired equipment. Request APC Suite configuration for implementation of new sensor streams. Setup an operational concept.
Other Software (Putty, APC Tracker etc., Softing OPC Toolbox):
Setup APC Tracker to monitor sensor data input continuously for data gaps</td></tr>
</table>

Engineering process phase	Addressed/Focus
	Manual Work: • Install additional hardware by "copy exactly" *APC Trent Suite RunToRun Framework (R2R)*: • Configure limits and messaging for calculated key numbers *Sensor Software or Configuration Tools /IDE's (Arduino, ABB Automation Builder, DAVE4 / Eclipse, IFM LineRecorder Device)*: • Copy Software to Sensor (Bootloader) *Equipment Automation Framework (EAF)*: • Subscribe to sensor data by selecting the topic (Site/Name of ProductionTool)
Operations & Management	**Activity** Continuous data input is monitored by APC Checker software. Messaging per E-Mail and SMS if limit violations occur. Stability issues will be automatically highlighted by the system. *Other Software (Putty, APC Tracker etc., Softing OPC Toolbox)*: • Monitoring the Data availability, incidents, errors (APC Checker) *Manual Work*: • Select data to be collected for training *APC Trent Suite RunToRun Framework (R2R)*: • Automatic messaging on limit violation • Run to Run control. Change process recipe slightly on base of sensor data *Equipment Automation Framework (EAF)*: • Data merging with tool datas and logistical metadata, Data storage into databases
Maintenance Decommissioning & Recycling	**Activity** *Manual Work*: • Maintenance Engineers are able to change hardware and do sensor configuration *APC Trent Suite RunToRun Framework (R2R)*: • 24*7 Hotline available *Equipment Automation Framework (EAF)*: • 24*7 Hotline
Evolution	**Activity** Regularly review sensor data and limits my maintenance and process engineers. Improve key number calculation, limit setting and Run to Run Controllers. (R2R = Algorithms which control equipment parameter automatically (process recipes) in a given window based on sensor data. *APC Trent Suite RunToRun Framework (R2R)*: • Review data (manually) • Improve Limit setting (manually)

Engineering process phase	Addressed/Focus
	• Improve R2R algorithms *Sensor Software or Configuration Tools /IDE's (Arduino, ABB Automation Builder, DAVE4 / Eclipse, IFM LineRecorder Device)*: • Upload AI Algorithm to Sensor / Edge Device if possible
Training & Education	**Activity** Create training material in form of a presentation or a Wiki page. Inform relevant groups i.e. Maintenance and process engineers of the production department. Users that have to do with sensor integration, Databases or Equipment integration. Train 24*7 service. Train maintenance technicians in shift. *Manual Work*: • Setup training material and rules for Sensor integration, Create Wiki page and provide information to all relevant teams

16.2.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 80 UC-16.2 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	If user friendly tools for the runtime management of sensor nets (sensing services) and service modules) are available, smart sensor nodes could be configured on-the-fly, which would make the integration of new sensors or data processing modules much easier and possible during run-time.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Prove of interoperability for the defined use cases is still ongoing.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Splitting (mostly) monolithic applications, like the advanced machine diagnosis platform, into several modules (services) may allow for a higher scalability (e.g., multiple data analysis modules in parallel) and robustness (multiple databases in parallel). To be decided once the FOUP-AMHS vibration measurement demonstrator (testbed) is ready for evaluation – same for PnP sensor integration.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	By improving the automated machine diagnosis methods, much less expert knowledge is required to interpret the results. Smart displays on sensor nodes and –hubs could also provide digital and interactive guidelines for the installation and positioning, which also reduces the effort within the Deployment & Commissioning phase - same for PnP Sensor integration.

Table 81 UC-16.2 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Install MQTT Broker with load balancer, Install test Sensors for IO-Link, ModbusTCP, µController Solution (Arduino) , PLC Solution , Setup missing AHT Services and test Sensor installation behaviour.
Obj. 2 - Interoperability for IoT and SoS engineering tools	The unified sensor communication protocol SenML reduces the engineering time spend for Sensor integration and also for IT (EAF).
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Integration of Sensor without REST / JSON interface, i.e. ModbusTCP and OPC DA / UA. Transform sensor data on SenML (Unified Interface for our IT) This reduces Changes requests to existing software landscape.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Sensors are mostly selected by the functionality or measurement quality or other criteria but they need to be compatible with the existing IT infrastructure. In addition AI algorithms on local clouds need to access the data easily. If appropriate modules are provided, the AH Framework could serve as a gateway between this data providers (µController sensor device transfers data in SenML format as compressed plain text via WLAN).
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Growing sensor networks require beside the automated sensor integration a management system to provide analyses about the installed base. (Number of devices per department. Data traffic, location – coordinates IP addresses a.s.o., This information have to be provided securely. Also wired LAN Ports are more and more expensive compared to the growing sensor installations. The secure WLAN integration comes into the focus to lower installation costs. Only a few industrial sensors provide secure WLAN connections without PSK (Pree Shared Key) EAP – TLS with SSID MDE is needed. So another way would be the sensor integration within an “unsafe” WLAN but with a save authentication service.
Obj. 6 - Training material (HW and SW) for professional engineers	Training activities for maintenance experts and end users regarding the correct usage of the smart maintenance platform, e.g., how to correctly input kinematic data, will be planned and executed. Guidelines regarding the correct installation of measurement equipment and sensors to retrofit additional machines will be compiled and provisioned. The further automation of the diagnosis process including automated maintenance suggestions will reduce the required Training & Education effort.

16.2.4 Engineering Process analysis

Table 82 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-16.2

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	C
OBJ-AHT #1	X	X	X	X	X		X	X	
OBJ-AHT #2		X	X	X					
OBJ-AHT #3		X	X	X					
OBJ-AHT #4		X	X		X				
OBJ-AHT #5			X		X			X	
OBJ-AHT #6	X	X			X	X		X	
OBJ-WP2 #1	X	X	X	X	X	X	X	X	
OBJ-WP2 #2	X	X	X	X	X	X		X	
OBJ-WP2 #3		X	X	X	X	X		X	
OBJ-WP2 #4	X	X			X	X		X	

16.3 UC-16.3 SC Data Management (IFD)

Within an industrial factory, sensors, actuators and control systems provide information about the current state of the production system. In order to implement predictive control strategies and exploration of cause-effect-relations it is necessary to store time series of measured values.
As shown in Figure 60, the operators (StkH1) of the factory equipment require information about how to control the equipment with regard to higher efficiency or/and higher performance. A data analyst (StkH2) is contracted to find out the necessary information. The role of StkH2 can be taken over by an experienced operator, a special department, an external service company or even contracted researchers of a university. Automation experts (StkH3) have to provide the necessary data base and to schedule and route the sensor values from the automation equipment into the data base.
As Eclipse Arrowhead framework is the communication framework of choice, an additional core service of the framework, the Extended Historian Service (EHS), is needed to achieve the following objectives:

- StkH2 (analyst) should be able to easily configure the data sources regarding the scheduling of the data acquisition and in best case the source data end points.
- StkH2 (analyst) should be able to use the tools of his choice to evaluate the data.
- StkH3 (automation expert) should be able to connect systems like fieldbus systems or complete programmable logic controllers (PLC) to the EHS and to assign system internal addresses to data channels, which are selectable by StkH2.
- StkH1 (operator) is not directly supported by the EHS, but his requirements are met in a shorter time than possible today.

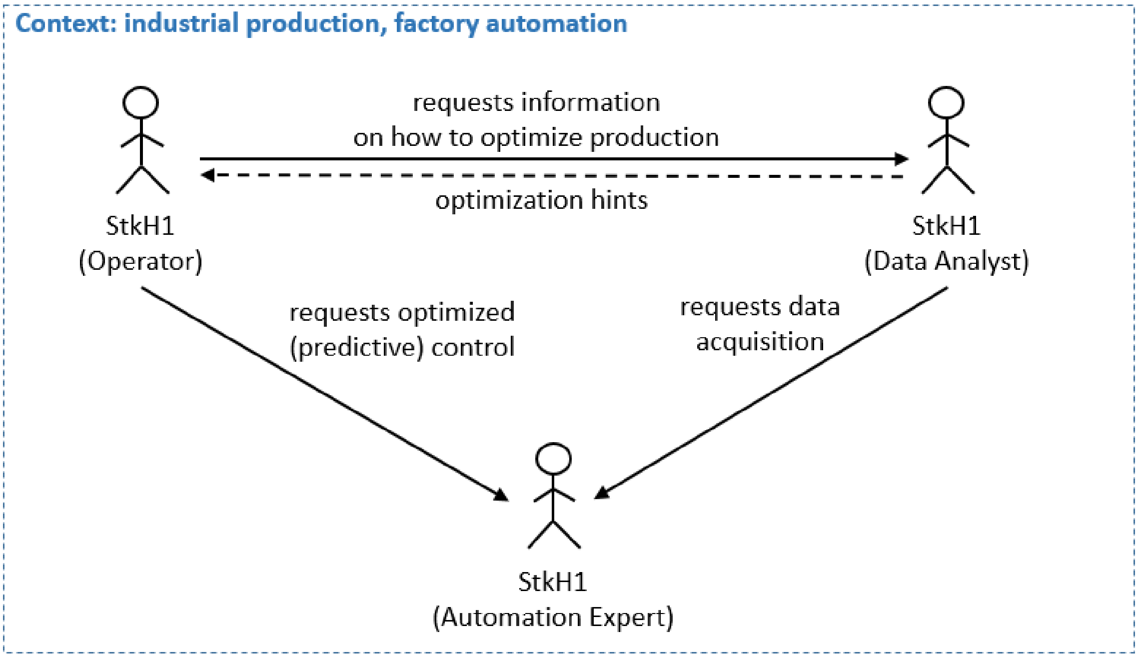


Figure 60 Interactions between stakeholders of UC-16.3

The second part of the use case (Task 9.4.4) is entitled "Automatic Decision Making by Predictive Maintenance and Machine Learning". Systems for predictive maintenance are typically developed as specific applications tailored to the specific needs of a specific maintenance problem. I.e., a new maintenance problem requires in general a new application. The main goal of this UC is to provide a generic software environment which allows the specification of any kind of a prediction model (for engineered systems) in a declarative manner and then to run (i.e. execute) this model in a runtime environment. This generic software suite consists of two core components. A development component called Delphi allows to build such prediction models taking advantage from a broad range of pre-defined concepts and constructs from symbolic and sub symbolic AI. The prediction models is executed by a runtime component called Pythia which transforms the prediction model into a predictive maintenance process.

The philosophy of this prediction suite relies on concepts from symbolic knowledge processing as failure structures, fuzzy logic & reasoning and case based reasoning as well as concepts and principles from Machine Learning, especially Deep Learning and Data Science. That's why we call such a prediction model a hybrid model.

Concrete subject of predictive maintenance are the a.m. implanters. The use case deals primarily with the implantation of ions on wafers. One problem is to determine the lifetime of the ion source and to initiate appropriate maintenance actions automatically due to the state of the ion source and the running implantation process.

The work is carried out in close cooperation with the project partner IFD and within the infrastructure of Infineon. Therefore, all interactions with machinery and equipment are determined by the means and by the functionality of the infrastructure at Infineon Technologies Dresden.

This UC is specific in the sense that access to equipment, sensor and process data is provided via existing infrastructure means at Infineon as the Yoda communication framework, data lake and data base technology (and not using AHT up to now). The main focus and emphasis is about the architecture of hybrid AI systems as well as the integration & cooperation of reasoning strategies and of symbolic and sub symbolic AI technologies. This is in its entirety rather untypical for AHT projects.

From the viewpoint of the final maintenance application (for the ion source) the components Delphi & Pythia are considered to be tools which are used to build a predictive maintenance application – despite the fact that Delphi & Pythia are the main subject of our work which happen to be applied to the ion source problem which represents a sub subject of our work. This fact makes the following discussion from time to time rather difficult in some respect.
Software used: Eclipse & Java, Spyder Anaconda, Python, Tensorflow, Keras & R, Scrum, Confluence, Jira & Yoda (Infineon's communication framework).

16.3.1 Overall description of the UC-EP

In Figure 61 some examples of data analytics software in UC-16.3
Roles of the three stakeholders involved in the UC (highlighted red)

- StkH1 = operator of production system (factory)
- StkH2 = data analyst (company internal department or external service provider)
- StkH3 = automation expert (electrical or maintenance department or external service)

Functional Blocks:

- Extended Historian service (EHS): a new Arrowhead core service
- Configuration system for the EHS
- Time-series (historian) database (integrated in EHS)
- Job scheduler (integrated in EHS)
- Data source adapters: provide interfaces to the data analytics software and using remote (most likely gRPC) interfaces of the EHS to access data
- Data sink adapters: provide access to sensors over fieldbus systems or similar and using remote interfaces (most likely gRPC) of the EHS to feed the data into the historian database
- Arrowhead Framework core services (registry, authorization, orchestration)
- Sensors as built-in the production systems

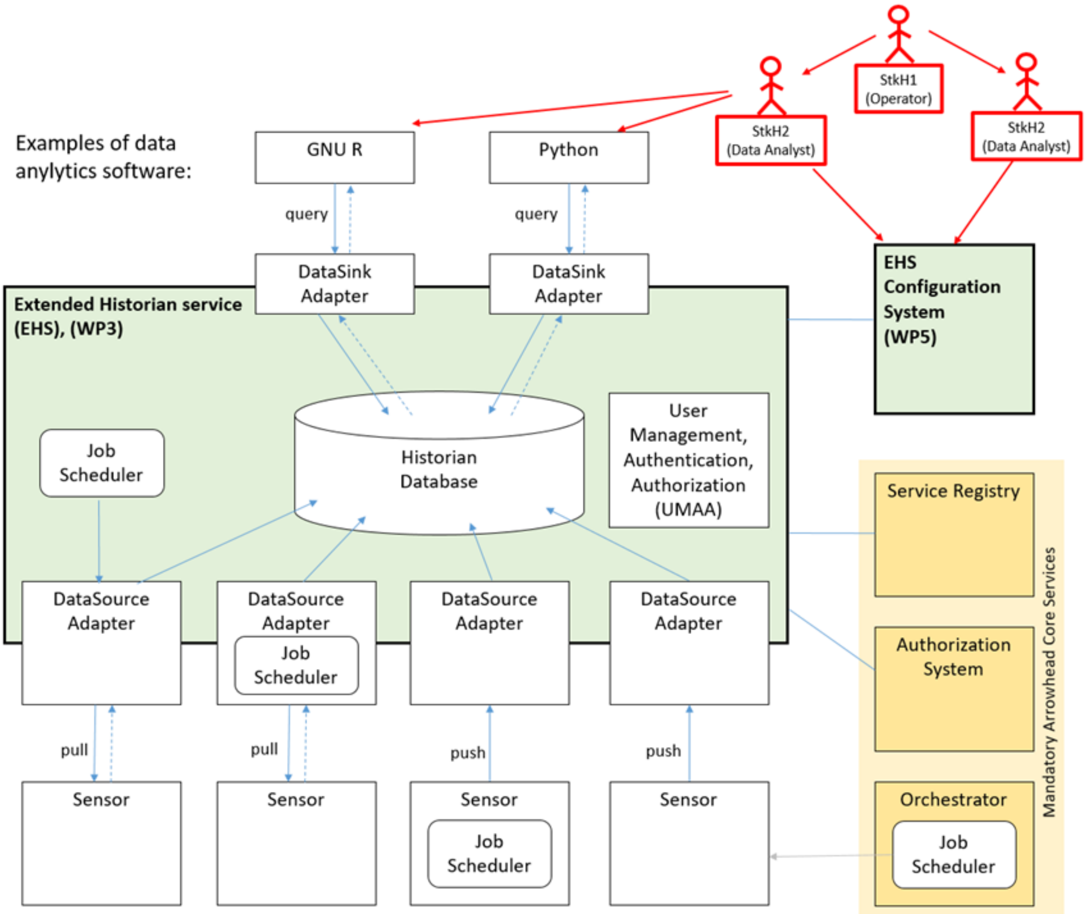


Figure 61 Examples of data analytics software in UC-16.3

Old engineering approach for (acyclic) on-demand request for optimization information.

- StkH1 requests optimization information from StkH2
- StkH2 specifies, which data is to be used and requests StkH3
- StkH3 fiddles with automation systems, sets up a data base and grants StkH2 access to the data
- StkH2 works to access the data from the data analysis software of his choice and develops necessary algorithms
- StkH2 hands over the wanted information to StkH1

As a variant cyclic periodic request for predicted conditions of the production system could be requested, which needs StkH2 to implement a scheduling system too and to feed a database, which has to be accessed by programs developed by StkH3.

Problems of the old engineering approach:

- Hand-written schedulers of StkH3 and in the variant of cyclically provided predictions also of StkH2 increase development efforts.
- High investment cost for a time-series database beside the control and communication systems
- Inflexibility: StkH2 has to request StkH3 for new data channels and for changes in data acquisition schedules.

New engineering approach:

- StkH1 requests optimization information from StkH2

- StkH2 specifies, which communication and control systems are involved and requests StkH3
- StkH3 sets up and configures pre-built data source adapters for data access
- StkH2 uses pre-build data sink adapters to directly access time-series values from the data analysis software of his choice and develops necessary algorithms
- StkH2 hands over the wanted information to StkH1

Benefits of the new engineering approach:
- StkH3 doesn't need to install a database. It is integrated in the communication framework (Arrowhead Framework)
- StkH2 doesn't need to request StkH3 to get access to new sensor data, since he has access to whole communication systems
- StkH2 can directly change the scheduling of data acquisition, which had in the old approach to be organized and programmed by StkH3 after conversations with StkH2
- StkH2 doesn't need general purpose database drivers to access the data (even if that is possible too), but he gets a driver for his data analysis software for a comfortable access to time-series of sensor values

The overall intention of the second part of the UC 16.3 "Automatic Decision Making by Predictive Maintenance and Machine Learning" is to automate the process of the determination of the lifetime of the ion source of an implanter and to initiate and automate the maintenance process in order to replace the ion source. This process may be extended to other components of an implanter as e.g. the plasma flood gun or others.

The functional blocks shown in Figure 62 may be described as follows:
- Create a prediction model (via Delphi)
- Run the prediction model (via Pythia)
- Get equipment & process data (from a data base)

(The data base is fed with equipment & process data by Infineon internal procedures, i.e. by existing Infineon Infrastructure)
Data are processed and analyzed by Pythia's & the Pred. Model's internal AI-logic
Depending on the results of the analysis messages, emails and maintenance orders are automatically created for actions to be taken.

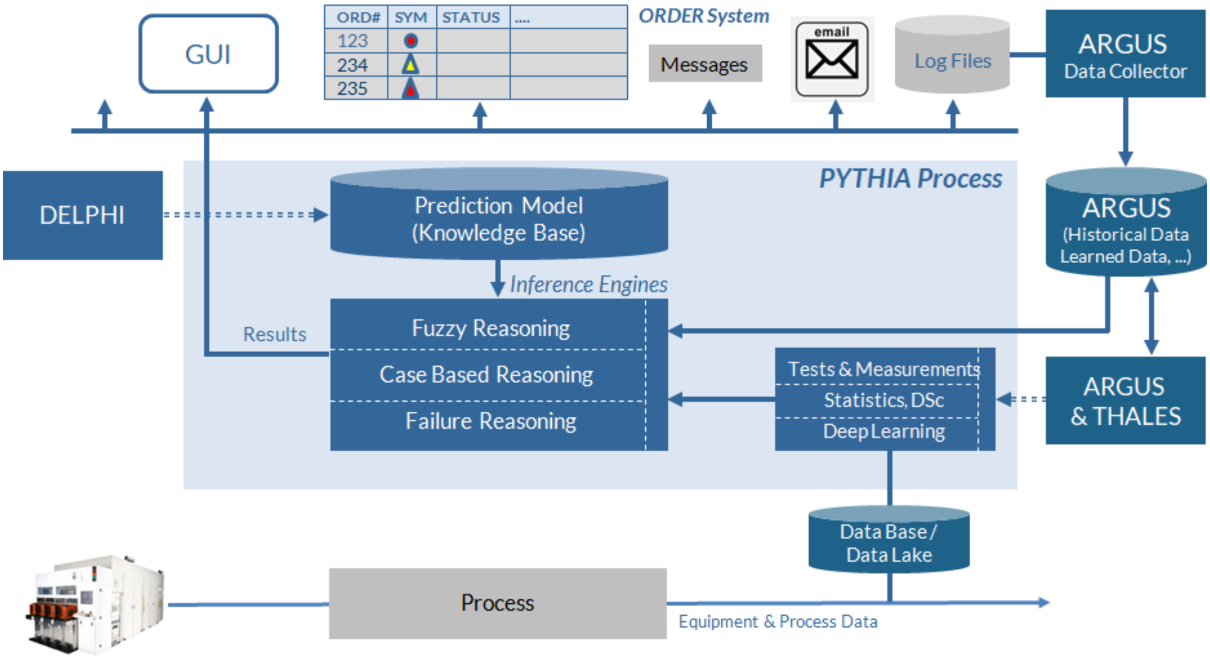


Figure 62 functional blocks of Automatic Decision Making by Predictive Maintenance and Machine Learning

The Figure 63 illustrates the idea of a hybrid prediction model.

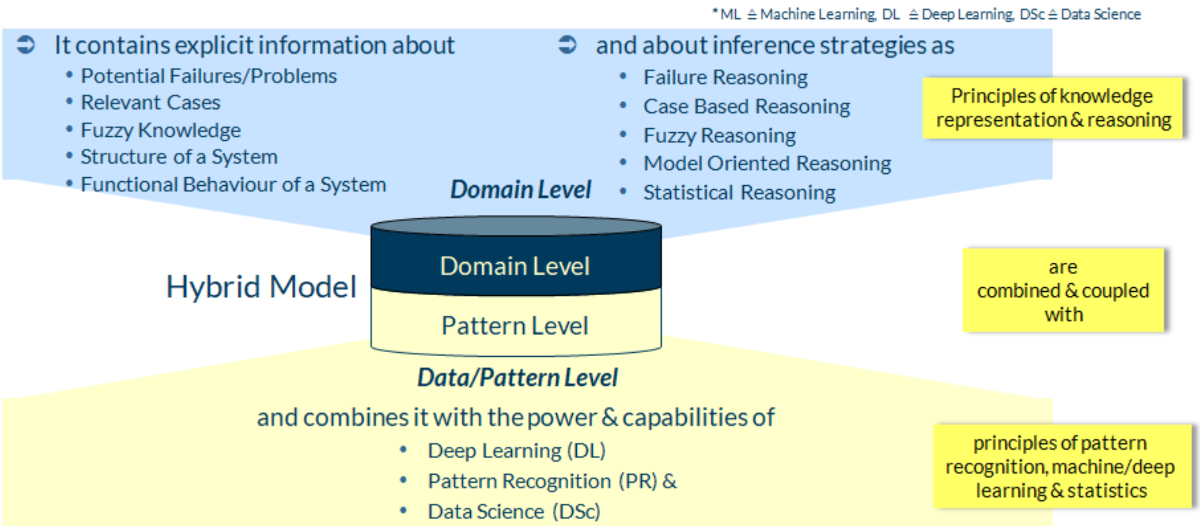


Figure 63 Hybrid prediction model

This second part of the use case is carried out in close cooperation with the project partner Infineon (IFD) and deals primarily with the ion implantation process. The engineering process starts with the analysis and collection of relevant equipment and process data to assess the remaining lifetime of the ion source. Therefore complete production data of the past years is used. For this assessment of lifetime a deep neural network is used which is implemented in Python, Tensorflow and Keras. Furthermore, additional process data are needed to determine the state of the ion source and the implantation process. These data are stored timely by Infineon-specific routines in a data base from where they are taken from at runtime. Based on a hybrid AI-based analysis the state of the ion source and the implantation process is determined. Depending on the results of this analysis and the seriousness of the current situation staggered actions are triggered as e.g. creating and sending messages, e-mails etc. up to the automatic creation of maintenance orders for subsequent processing (see also section A).

16.3.2 Engineering Process Description

In Figure 64 the engineering process of the "Automatic Decision Making by Predictive Maintenance and Machine Learning" part of the use case.
StkH-1 Semantis is responsible for the design and implementation of the predictive maintenance suite, consisting of the authoring tool for prediction models DELPHI and the runtime component PYTHIA to execute prediction models. DELPHI & PYTHIA are used as tools to develop, maintain and run a prediction model in order to monitor and assess the ion source and other components of an implanter. After all and from a functional point of view the prediction model (which is an equivalent for a predictive application) is the final goal of the use case despite the fact that the development of the maintenance suite represents the lion's share of the work. The system is in principle ready for operational use after a training of the responsible staff. StkH-2 IFD IT integrates the resulting toolchain within the fab’s IT landscape based on the training received. StkH-3 IFD Operation/Maintenance uses the procedures for operation based on the training received and maintains the implanters. Furthermore, it collects feedback and suggestions which provide the basis for maintenance and further refinement of the requirements for StkH-1.

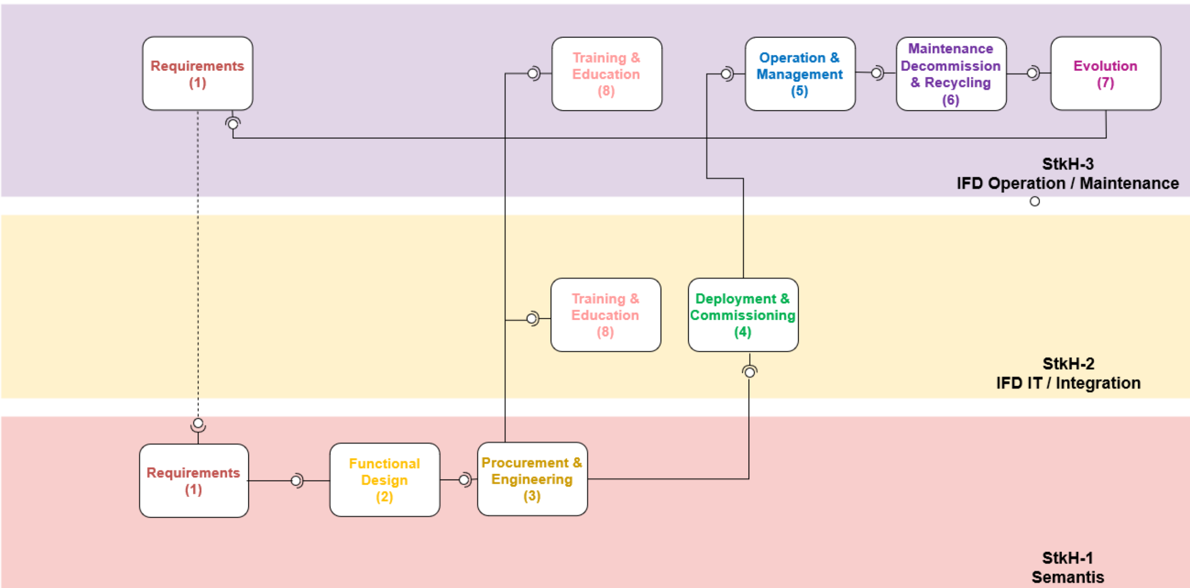


Figure 64 AHT-EP of use case UC-16.3

The Delphi/Pythia-suite - currently under development - represents a comprehensive framework to develop and run predictive maintenance applications for engineered systems.

While all phases are applicable in principle for our UC ion implantation, emphasis is laid on the "requirement, construction & design" steps especially to EP1 to EP3. This is due to the specific challenge of our UC not to code a prediction program in a classical manner by programming in the narrow sense but to specify and model i.e. to "describe & document" them as hybrid knowledge bases.

Due to the current infrastructure at Infineon Dresden, the majority of equipment and process data needed for prediction purposes of implanters is organized in a data lake/data base. The YODA communication framework is an essential part of Infineon's infrastructure and serves for the general exchange of data. Therefore a communication interface has been developed and is in use. But there is also a need for a big variety of further information from e.g Deep Learning or Machine Learning or statistical routines. Most of them are written in Python and/or R. In order to provide information to and obtaining results from such routines and component ML_comm is under development to trigger the execution of such routines and to exchange information between Delphi/Pythias and such routines.

The traditional way of performing data analytics in production systems is to install a separate data acquisition systems including limited data analytics facilities beside the control system of the factory. A communication framework integrating data acquisition and providing easy access for external data analysis software as we provide it in scope of the AHT project is unknown.

The engineering process adopts the view point aspects of the IIRA and uses features like black-box descriptions and de-composition defined by the V-Model XT.

Table 83 AHT-EP Phase focus of UC-16.3

Engineering process phase	**Addressed/Focus**
Requirements	**Activity** *Extended Historian Service (EHS)*: Requirements are collected in WP9 and WP3 of the AHT project. *EHS Configuration system (EHS-CS)*: Requirements are collected in WP9 and WP5 of the AHT project. *Production System*: StkH1 defines the need for optimization of the production system in order to reach goals regarding several optimization criteria like minimization of material, energy and the like or performance enhancements. *Data Analysis Tools*: n.s. *EHS Adapters (source and sink adapters)*: Requirements are collected in WP9 and WP3 of the AHT project. Regarding the second part of the UC, an assumption for Predictive Maintenance for the ion source of an implanter is a basic understanding of the ion implantation process and the data needed for that purpose. This step includes the specification of the expected overall functionality. It turned out that all equipment and process data needed were available within a data lake and/or a data base. *Design & Development of a Generic Suite for Predictive Maintenance (pdM)*: Identify and specify generic concepts for typical maintenance problems, generic prediction models and corresponding apps for operational use. *pdM / Condition Monitoring (for Ion Implantation)*: Identify and organize relevant data for assessing the ion source lifetime and state. **Tools** *Extended Historian Service (EHS)*: • Input: general requirements for data acquisition • Input: specific requirements for EHS (from data analysis) • Output: requirements for EHS

Engineering process phase	Addressed/Focus
	EHS Configuration system (EHS-CS): • Input: general requirements for data acquisition • Output: requirements for EHS-CS including GUI face plates *Production System*: • Input: general requirements for data acquisition • Output: specific optimization requirements for the production system *Data Analysis Tools*: • Input: general requirements for data acquisition • Input: specific requirements for data analysis • Output: requirements for data analysis • Output: specific requirements for EHS *EHS Adapters (source and sink adapters)*: • Input: general requirements for data acquisition • Output: requirements for EHS adapters
Functional design	**Activity** *Extended Historian Service (EHS)*: The functional design is done in WP3 of AHT. There is a specification document. *EHS Configuration system (EHS-CS)*: The functional design is done in WP5 of AHT. A specification document is work in progress. *Production System*: StkH3 defines needed access to Arrowhead application services, fieldbus systems and control systems. *Data Analysis Tools*: StkH2 analyses the production system structure and specifies influencing parameters stemming from the automation systems. He additionally designs the necessary algorithms as mathematical expressions. *EHS Adapters (source and sink adapters)*: The functional design is done in WP3 of AHT. There is a specification document. Regarding the second part of the UC, the functional design of the Delphi/Pythia maintenance suite was/is the most challenging step. It follows a hybrid AI approach combining explicit (symbolic) knowledge processing, Machine Learning & (sub symbolic) Deep Learning. The knowledge based components incorporate the following concepts among others: functional modelling of the equipment under test (model-based approach), failure models, fuzzy logic & reasoning and cases & case based reasoning. *Design & Development of a Generic Suite for Predictive Maintenance (pdM)*: Develop an architecture for a generic smart & knowledge based authoring & runtime suite for generic pre-diction models (Confluence etc.). *pdM / Condition Monitoring (for Ion Implantation)*: Design of the toolchain to obtain data from implanters for prediction purposes. **Tools** *Extended Historian Service (EHS)*: • Input: requirements for EHS • Output: documented decomposition of the EHS • Output: interface description of EHS components *EHS Configuration system (EHS-CS)*: • Input: requirements for EHS-CS • Output: documented decomposition of the EHS-CS

Engineering process phase	Addressed/Focus
	• Output: interface description of EHS-CS components *Production System*: • Input: specific optimization requirements for the production system • Output: specific requirements for data analysis *Data Analysis Tools*: • Input: requirements for data analysis • Output: mathematical description of data analytics algorithms *EHS Adapters (source and sink adapters)*: • Input: requirements for EHS adapters • Output: documented decomposition of the EHS adapters • Output: interface description of EHS adapters components
Procurement & Engineering	**Activity** *Extended Historian Service (EHS)*: The procurement and engineering is done in AHT WP3. *EHS Configuration system (EHS-CS)*: The procurement and engineering is done in AHT WP5. *Production System*: StkH3 changes programs in a manner that cyclically generated information from the data analysis tool (e.g. predictions) will lead to an optimized production process. Since the creation of optimization information will be less reliable, there will be default values in case of absence of the optimization information. *Data Analysis Tools*: StkH2 develops the algorithms defined in AHT-EPP 2 as source code for the data analysis tool. *EHS Adapters (source and sink adapters)*: StkH3 exports variable lists out of the communication or control configurations. This may be a network configuration file or a PLC variable or I/O list. Those lists are transformed by StkH3 into adapter configuration files. Regarding the second part of the UC, this phase is less relevant in this use case. E.g., no procurement activities were needed. *Design & Development of a Generic Suite for Predictive Maintenance (pdM)*: Check for missing features and reusability specification. *pdM / Condition Monitoring (for Ion Implantation)*: Check for appropriateness of available data and prediction results. **Tools** *Extended Historian Service (EHS)*: • Input: functional design description of EHS • Output: documented source code of EHS • Output: compiled binaries of EHS *EHS Configuration system (EHS-CS)*: • Input: functional design description of EHS-CS • Output: documented source code of EHS-CS • Output: compiled binaries of EHS-CS *Production System*: • Input: functional design description of production system • Output: documented source code of production system (controller code) • Output: compiled binaries of production system controller code

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>Data Analysis Tools:
Input: mathematical description of data analytics algorithms
Output: documented data analytics code
Output: in case of compiled language extension (e.g. C code extensions): compiled binaries of data analytics code
EHS Adapters (source and sink adapters):
Input: functional design description of EHS adapters
Output: documented source code of EHS adapters
Output: compiled binaries of EHS adapters</td></tr>
<tr><td>Deployment & Commissioning</td><td>Activity
Extended Historian Service (EHS): StkH2 or StkH3 deploy the EHS together with the Arrowhead Framework on a computing resource within the Intranet of the producing company.
EHS Configuration system (EHS-CS): StkH2 or StkH3 deploy the EHS-CS on any computing resource within the Intranet of the producing company.
Production System: StkH3 installs the changed programs into the control systems.
Data Analysis Tools: StkH2 deploys the data analysis software on any computing resource within the Intranet of the producing company. The algorithm source codes are put on the right places.
EHS Adapters (source and sink adapters): StkH3 deploys the source adapters on any computing resource within the Intranet of the producing company. Then the configuration files are put in the right place. StkH2 installs the right sink adapter in the data analysis tool.
Regarding the second part of the UC, the resulting ion source application will at first be tested for specific equipment, i.e. implanters. Later on this block requires to setup an operational concept.
Design & Development of a Generic Suite for Predictive Maintenance (pdM): Detailed planning to meet the requirements of operational use.
pdM / Condition Monitoring (for Ion Implantation): not relevant here.
Tools
Extended Historian Service (EHS):
Input: requirements on the EHS run-time system
Input: compiled binaries of EHS
Output: installed run-time system
Output: installed compiled binaries of EHS
EHS Configuration system (EHS-CS):
Input: requirements on the EHS-CS run-time system
Input: compiled binaries of EHS-CS
Output: installed run-time system
Output: installed compiled binaries of EHS-CS
Production System:
Input: requirements on the production system run-time system
Input: compiled binaries of production system controller code
Output: installed run-time system
Output: installed compiled binaries of production system controller code
Data Analysis Tools:
Input: requirements on the data analytics run-time system</td></tr>
</table>

Engineering process phase	Addressed/Focus
	• Input: compiled binaries of data analytics code • Output: installed run-time system • Output: installed data analytics code *EHS Adapters (source and sink adapters)*: • Input: requirements on the EHS adapters run-time system • Input: compiled binaries of EHS adapters • Output: installed run-time system • Output: installed compiled binaries of EHS adapters
Operations & Management	**Activity** *Extended Historian Service (EHS)*: StkH2 or StkH3 start the EHS together with the Arrowhead Framework. *EHS Configuration system (EHS-CS)*: StkH2 or StkH3 start the EHS-CS. *Production System*: StkH1 changes the operation procedures based on the information got by StkH2. Cyclically generated optimization information will influence the control systems automatically. *Data Analysis Tools*: StkH2 produces the information on how to change the operation procedures in order to optimize the production process or he starts the cycle of information generation. *EHS Adapters (source and sink adapters)*: StkH3 starts the source adapters. Sink adapters are started in scope of the data analysis software. Regarding the second part of the UC, operation and management lead usually to new information about relevant aspects of the equipment to be monitored. This information can be used to improve the software. *Design & Development of a Generic Suite for Predictive Maintenance (pdM)*: Improve & extend functionality and robustness of the maintenance suite. *pdM / Condition Monitoring (for Ion Implantation)*: Improve & extend the reusability of the prediction apps. **Tools** *Extended Historian Service (EHS)*: • Input: installed run-time system and EHS binaries • Output: the running EHS *EHS Configuration system (EHS-CS)*: • Input: installed run-time system and EHS-CS binaries • Output: the running EHS-CS *Production System*: • Input: installed run-time system and production system binaries • Output: the running production system *Data Analysis Tools*: • Input: installed run-time system and data analytics code • Output: the running data analytics code *EHS Adapters (source and sink adapters)*: • Input: installed run-time system and EHS adapters binaries • Output: the running EHS adapters

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td>Maintenance Decommissioning & Recycling</td><td>

Activity

Extended Historian Service (EHS): n.s.

EHS Configuration system (EHS-CS): StkH2 or StkH3 can use the EHS-CS to monitor problems in the EHS or the adapters.

Production System: n.s.

Data Analysis Tools: StkH2 uses the debugging facilities of the data analysis software to repair failures.

EHS Adapters (source and sink adapters): n.s.

Regarding the second part of the UC, improvements based on the experiences from operational tests.

Tools

Extended Historian Service (EHS):

- Input: the running EHS
- Input: bug-reports
- Output: bug-fixed version of EHS

EHS Configuration system (EHS-CS):

- Input: the running EHS-CS
- Input: bug-reports
- Output: bug-fixed version of EHS-CS

Production System:

- Input: the running production system
- Input: bug-reports
- Output: bug-fixed version of production system controller code

Data Analysis Tools:

- Input: the running data analytics code
- Input: bug-reports
- Output: bug-fixed version of data analytics code

EHS Adapters (source and sink adapters):

- Input: the running EHS adapters
- Input: bug-reports
- Output: bug-fixed version of EHS adapters

</td></tr>
<tr><td>Evolution</td><td>

Activity

Extended Historian Service (EHS): Feature requests are handled over the software development resource (probably GitHub).

EHS Configuration system (EHS-CS): Feature requests are handled over the software development resource (probably GitHub).

Production System: Evolution starts by changed requirements, so StkH1 will define new requirements and initiate further development of all necessary components (EHS and adapters).

Data Analysis Tools: Evolution of data analysis software is out of scope of AHT.

EHS Adapters (source and sink adapters): Feature requests are handled over the software development resource (probably GitHub).

Regarding the second part of the UC:

Design & Development of a Generic Suite for Predictive Maintenance (pdM): Enrich

</td></tr>
</table>

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td>
& improve modelling concepts and constructs for prediction models.

pdM / Condition Monitoring (for Ion Implantation): Extend the features to obtain data via multiple channels.

Tools

Extended Historian Service (EHS):
• Input: the running EHS
• Input: feature requests
• Output: feature enriched version of EHS

EHS Configuration system (EHS-CS):
• Input: the running EHS-CS
• Input: feature requests
• Output: feature enriched version of EHS-CS

Production System:
• Input: the running production system
• Input: feature requests
• Output: feature enriched version (optimized) of production system controller code

Data Analysis Tools:
• Input: the running data analytics code
• Input: feature requests
• Output: feature enriched version of data analytics code

EHS Adapters (source and sink adapters):
• Input: the running EHS adapters
• Input: feature requests
• Output: feature enriched version of EHS adapters
</td></tr>
<tr><td>Training & Education</td><td>
Activity

Extended Historian Service (EHS): The software will be provided including installation and usage manuals. The code will be in-line documented. Missing information can be requested as bug-report in the software development resource (probably GitHub).

EHS Configuration system (EHS-CS): The software will be provided including installation and usage manuals. The code will be in-line documented. Missing information can be requested as bug-report in the software development resource (probably GitHub).

Production System: n.s.

Data Analysis Tools: Training regarding data analysis software is out of scope of AHT. There are typically various videos available on-line via video-sharing platforms like YouTube.

EHS Adapters (source and sink adapters): The software will be provided including installation and usage manuals. The code will be in-line documented. Missing information can be requested as bug-report in the software development resource (probably GitHub).

Regarding the second part of the UC, create training material in appropriate form and keep relevant groups, maintenance personnel and process engineers informed. Train maintenance technicians in shift.

Design & Development of a Generic Suite for Predictive Maintenance (pdM): Provide appropriate information for maintenance personnel.

pdM / Condition Monitoring (for Ion Implantation): Provide appropriate information
</td></tr>
</table>

Engineering process phase	Addressed/Focus
	for maintenance personnel. **Tools** *Extended Historian Service (EHS)*: • Input: documented source code • Input: README.md descriptions • Output: visually appealing form of the documentation *EHS Configuration system (EHS-CS)*: • Input: documented source code • Input: README.md descriptions • Output: visually appealing form of the documentation *Production System*: • Input: documented source code • Output: print out of the documented source code (or PDF) *Data Analysis Tools*: • Input: documented data analytics code • Output: visually appealing form of the data analytics code *EHS Adapters (source and sink adapters)*: • Input: documented source code • Input: README.md descriptions • Output: visually appealing form of the documentation

16.3.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 84 UC-16.3 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	In the evolution phase feedback information is being evaluated from usage in the field. This information is collected and pre-analysed by StkH 3 and forwarded to StkH-1 (Semantis). StkH-1 analyses the information to drive the update process to adjust and update the requirements to trigger the update process. Thus, the objective is matched.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The objective has been matched. Three stakeholders are involved in the entire development process which exchange information between each other. The exchange happens to be manual mostly.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Some phases of the entire EP have interactions with other phases of the same EP and EPs from other stakeholders. However, due to the specifics of the UC do not require a substantially increased number of I/Os but offers room for such if needed.

WP2 Objective	Focus & Planed actions
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	StkH-1 will develop and provide the training and reference material for the end user. Ideally, the material can be provided in form of a video or in an interactive form StkH-2 is responsible for provide appropriate material for the installation process.

Table 85 UC-16.3 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Once the EHS and EHS-CS is existing and several EHS adapters are developed, then data analysis is only a matter of configuration. Since the EHS comes built-in with the communication system (Arrowhead Framework), there will be no need to buy, learn and set-up an additional data acquisition system. Actions: within the AHT project basic versions of EHS, EHS-CS and EHS adapters are created. By providing with Delphi & Pythia a generic hybrid Toolsuite for predictive maintenance a reduction of costs by at least 50% is expected with regard to the development of predictive application. Furthermore, all new applications will come along with a rich set of functionality and a high degree of standardization.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Due to the current IT infrastructure at project partner this could not sufficiently addressed up to now.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The EHS can be equipped with several EHS source adapters. While one adapter can be attached as consumer to an Arrowhead local cloud, there can be other source adapters attached providing data from e.g. MODBUS TCP or MQTT. The data analysis can access the integrated data of all those data sources. Actions: Within the AHT project basic versions of 2 data source adapters are developed, one of them acting as an Arrowhead consumer service, while the other adapts the EHS to a legacy automation system. Within the AHT project, a basic version of an EHS send adapter will be created in order to provide the possibility to run data analytics scenarios based on both data sources. Although the automatic machine diagnostic approach will be mainly implemented and evaluated using our Delphi/Pythias suite, the integration of selected sub-components, e.g., data acquisition from different standard interfaces, pre-processing, and caching and data storage of this platform could be accomplished using the Arrowhead Framework in the future.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	The EHS source and sink adapter concept fits also emerging digitalization and automation frameworks, but within the project there are no further activities planned to implement such adapters. Some use cases may require the integration of advanced data collection and integration frameworks. If appropriate modules are provided, the AH Framework could serve as a gateway.

Project Objective	Focus & Planed actions
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The data acquisition and evaluation for the automatic machine diagnostic approach will be executed on premises of the involved partner IFD. Due to the existing IT infrastructure at IFD these aspects have currently a lower priority for our UC.
Obj. 6 - Training material (HW and SW) for professional engineers	The developed software components (EHS, EHS-CS and EHS adapters) will be provided with usual installation and usage manuals. The engineering already includes the training material in terms of installation and usage manuals. If that material is not enough or lacking information, then this should be treated as bug and is covered by maintenance activities. Training activities for maintenance experts and end users regarding the correct usage of authoring and runtime components have to be provided. But the ongoing degree of automation for diagnostic processes will reduce the effort for training and education.

16.3.4 Engineering Process analysis

Table 86 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-16.3

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X					X	
OBJ-AHT #2									
OBJ-AHT #3			X	X	X				
OBJ-AHT #4			X		X				
OBJ-AHT #5			X		X	X			
OBJ-AHT #6		X	X		X			X	
OBJ-WP2 #1	X	X	X	X			X		
OBJ-WP2 #2	X	X		X	X	X	X	X	
OBJ-WP2 #3			X	X	X		X	X	
OBJ-WP2 #4		X	X					X	

16.4 UC-16.4 Smart Maintenance (IFD)

The use-case involves the application of smart monitoring and predictive maintenance in the facility- and automated transportation systems (AMHS) within semiconductor manufacturing. Different kinds of predictive maintenance exist and are applied in practice as they all have their advantages and disadvantages. There is reactive maintenance if it is enough to wait for

problems and solve them when they occur. In contrast, periodic maintenance is applied if unexpected breakdowns lead to high costs due to production standstill or consequential damages. Predictive maintenance tries to compute the ideal time instances for maintenance actions by applying a wear model in combination with operational data in order to avoid both unexpected standstills and unnecessarily early replacements etc. with the drawback of the modelling effort and the continuous preparation of operational data.
Different software exists that can support the planning of maintenance actions. Examples are the IDS Innomaster, Prüftechnik Omnitrend, Conimon WebApp, the Matlab Predictive Maintenance toolbox, and Bachmann WebLog Expert. Some vendors provide also specialized hardware, especially for vibration measurement, e.g. Beckhoff EL3632, Bachmann AICxx, and Delphin Expert Vibro.
Currently within the use case, the measurements are all manual steps.

- Within the AMHS, a measurement FOUP is dispatched manually, the resulting vibration measurements are generated as plain CSV files that have to be manually evaluated.
- In facility, measurements on the selected demonstrator machinery (exhaust fan) are sporadically conducted with a handheld device, only a few simple KPIs (e.g., RMS) are assessed by hand.

16.4.1 Overall description of the UC-EP

For the use case "predictive maintenance in facility", the manual measurement shall be replaced with an automated, continuous measurement and automated machine diagnosis of the bearings built in the fan
The functional blocks of the use case shown in Figure 65 can be described as follows:

1. modelling of the kinematics of the monitored machine
2. sensor selection and installation
3. data acquisition including set up of data transfer and data storage
4. diagram preparation from measured data including filtering of relevant parts
5. feature extraction from diagrams (e.g. narrow band peaks, harmonic families, rms values)
6. pattern recognition combining features from different diagrams.

The last step is currently not a standard step in practice but is recommended by standards and guidelines like VDI3832.

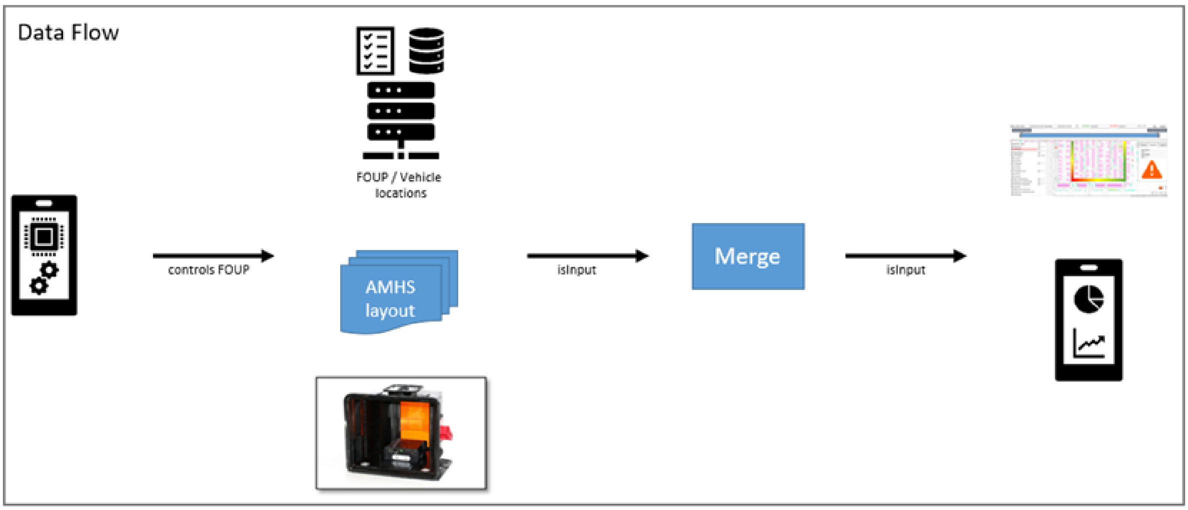


Figure 65 UC-16.4 Data Flow

As for the vibration monitoring within the AMHS, the process shall be automated and integrated into the existing IT landscape of the semiconductor production fab as outlined in the following picture. In detail, the triggering of measurements shall be automated and the resulting vibration data automatically merged with information about the layout of the transport system and the Vehicle positions (i.e., measurement FOUP location over time). The resulting data shall then be available for visualization within the IT landscape of the fab.

The use case "predictive maintenance in facility" engineering process is done in close cooperation of the machine diagnosis contractor and the end user (customer) and starts with the modelling of wear relevant aspects of the machine, especially regarding kinematic properties of rotating elements. Then, appropriate sensors are selected, e.g. vibration sensors with a suitable acceleration range and frequency range. The next step, diagram preparation including data filters, is not done by all providers as many of them only work with aggregated KPIs like RMS of unfiltered time signals. The last step, pattern recognition based on several diagrams, is recommended in expert literature and standards (e.g. VDI3832) but very rarely supported by software vendors for predictive maintenance. Regarding the vibration severity measurement within the AMHS, the maintenance engineer should be able to start a measurement run after the identification of critical or suspicious passages within the AMHS. The merging of data will be done using services within the AH Tools framework and transferred to visualization tools for investigation and remedy assessment by the AMHS experts.

16.4.2 Engineering Process Description

In Figure 66 the AHT-EP of the UC 16.4.
StkH-1: TU Dresden; responsible for the design and implementation of the measurement FOUP toolchain as well as for the provision of the required services, additional soft- and hardware. Once an iteration of development is complete, StkH-2 and StkH-3 are provided with manuals and training how to configure the tools, execute measurements and interpret the data (Stkh-3) and what is required to enable the integration of the toolchain within the fabs IT infrastructure (Stkh-2), e.g., configure WLAN Access, preparation of external tools and data sources.
Stkh-2: Infineon Dresden (IFD) IT integrates (deploys) the finished toolchain within the fabs IT landscape based on the training.
Stkh-3: IFD Maintenance and Automation department; uses the toolchain for measurement and maintains the HW systems (e.g., battery charge) based on the training, collects feedback and enhancement suggestions (evolution) which is the basis for the refinement of the requirements as input for Stkh-1.

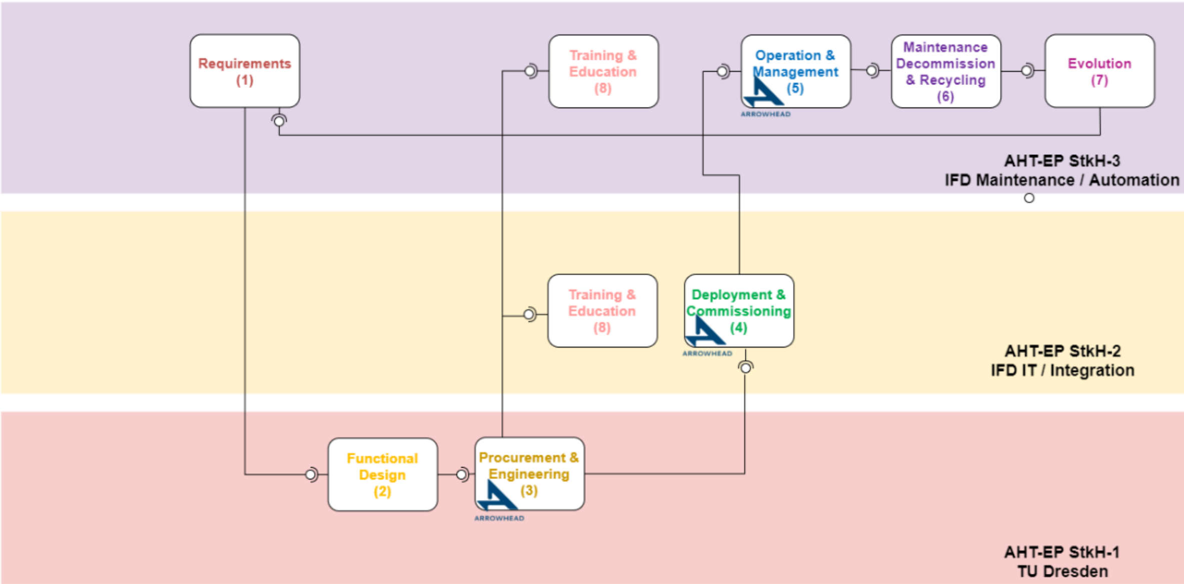


Figure 66 AHT-EP of use case UC-16.4

By using the AHT-EP model within the use-case of FOUP vibration measurement within the AMHS, it is intended to be compliant with some of the dimensions represented in the RAMI 4.0 model.
In the original version of the FOUP-AMHS vibration measurement, the maintenance engineers of the semiconductor manufacturing plant had to manually measure and process the data, which was a tedious task. Since the new demonstrator, incorporating several AH framework services, is ready for evaluation and deployment, these tasks are mostly automated and the data is automatically merged with core data from different areas of the fab. This will greatly enhance the productivity of the tool. The same holds true for the use-case of vibration measurement within the facility area. Before the project, the measurements were conducted manually and stored in a legacy DB. This tedious data acquisition and evaluation is eliminated by the usage of the Conimon measurement system and WebApp with its automated machine diagnostics possibilities. The adoption of the AHT-EP also provided a guideline to methodically plan and execute the necessary activities of the use-cases from the design phase up until the provision of training and education material

The machine diagnostics approach in facility is based on waterfall process model and a variety of Standards and norms, most prominently, for the use case of rolling bearing diagnosis as it is used within the facility monitoring use-case, the norms VDI 3832 and ISO 10816. For the use-case “predictive maintenance in AMHS”, an AHT-EP compliant PoC is currently implemented.

The automatic machine diagnostic approach mainly spans across the domains of procurement & engineering, i.e., the selection and acquisition of the required measurement hardware, the deployment and commissioning, i.e., the configuration and installation of the measurement hard- and software, the operation and management phase, i.e., the continuous data acquisition, processing and presentation (machine diagnosis), maintenance and evolution (hard- and software support and improvement) as well as training and education, e.g., training maintenance operators to correctly interpret the KPIs and results.
Legend:

- StkH1->Semiconductor Fab;
- StkH2->Machine diagnostics service company;
- StkH3->Software integration Service Company.

For the facility monitoring use-case, no tools are developed besides the Conimon platform at the moment, which is in turn extended to support use-cases within the semiconductor industry; potential further development to incorporate the AH framework is evaluated later in year 2. However, within the AMHS monitoring use case (Measurement FOUP), several services are developed which are similar in scope (vibration data acquisition, enrichment with core- and auxiliary data and visualization) to evaluate and demonstrate the usability of the AH framework for such use cases. If successful, parts of the findings can be evaluated for implementation within the diagnosis platform. The demonstrator is currently actively developed on the basis of embedded computers.

Since the use-cases of monitoring of machine heath status (AMHS and facility) are quite singular as a whole, the developed services as planned of now are described instead, since they promise a certain re-usability within the automation of future AHT-Eps (External: Provided as Data sources or –Sinks by the manufacturing plant, Tools serve as gateways or SW-adapters).

In case of the automatic machine diagnostic approach (AMHS and facility), all phases are applicable, although the functional design to a lesser degree for the Machine diagnostics in facility as outlined above (standardized EPP in VDI3832).

The used Diagnosis Web Platform is currently tightly coupled, at least between the data input, which provides a generic interface to sensors or industrial controls, and the visualization of the diagnosis results. Depending on the assessment of the AH Framework during the work on the use-case “vibration analysis within semiconductor AMHS”, it could be decoupled into several modules (services) in future, e.g., data pre-processing, storage, automated diagnosis, visualization. As mentioned, this will be evaluated later in year 2.

Since the FOUP-AMHS vibration measurement demonstrator is ready for evaluation, the evaluation of the usage of the AHT EP and the AH framework can be summarized as follows: The automation of the measurement FOUP achieved using the Arrowhead IoT framework has undoubtedly made work easier and saved engineering efforts, time and money. The strengths and advantages of the framework compared to a proprietary software solution adapted to the specific application are particularly evident when a large number of services are integrated. Arrowhead standardizes the coupling and orchestration of the services and thus simplifies subsequent changes. A reuse of already developed components is easy. However, if the measurement FOUP data is to be used only for the original task, i.e., only to be displayed with the visualization tool, a solution without IoT Framework would be easier to implement. In this case, framework specific services can be omitted, since it is a one to one assignment of data source and data sink. With fewer intermediate stations, data transfer would be faster and less error-prone. In the case of measurement FOUP, the use of the Arrowhead Framework is only reasonable if additional sensors or services are integrated into the IoT framework. Only the use of several providers and consumers in the existing framework outweighs the additional effort required to install the Arrowhead Framework. Therefore, it will be evaluated in future if the IFD FAC use-case (facility area vibration measurement) can be integrated into the local cloud, for instance, sensor to monitor the machine health of production- or facility equipment.

Table 87 AHT-EP Phase focus of UC-16.4

Engineering process phase	Addressed/Focus
Requirements	**Activity** Relevant aspects for predictive maintenance are machine type, kinematic properties and production meta data. Further typical damages and faults of the machines under consideration from historical experience are helpful. *PDM / condition monitoring in Semicond. Facility*: Collect machine type, kinematic properties and production meta data. *Measurement FOUP Integration / Vibra. Analysis in AMHS*: Requirements are collected regarding measurement frequency, necessary data to merge, desired visualisation. **Tools** *PDM / condition monitoring in Semicond. Facility*: • Input: StkH1-EPP1; Requirements (machines to instrument, frequency and extend of the monitoring), machine kinematics and production meta data • Output: StkH2-EPP2; required hardware • Output: StkH2-EPP3; set of required diagnosis approaches (diagram types, KPIs etc.) *Measurement FOUP Integration / Vibra. Analysis in AMHS*: • Input: StkH1-EPP3; requirements are collected regarding measurement frequency, necessary data to merge, desired visualisation • Output: StkH3-EPP1; collection of functional specifications of the toolchain
Functional design	**Activity** This block is less relevant in this use case, it can be used to test and evaluate additional data analysis methods. *PDM / condition monitoring in Semicond. Facility*: Matlab will be used to test and evaluate new algorithms. *Measurement FOUP Integration / Vibra. Analysis in AMHS*: Design of the toolchain for automated vibration measurement on simulated environments (SBCs). **Tools** *PDM / condition monitoring in Semicond. Facility*: • Input: StkH2-EPP3 • Output: StkH2-EPP4; tested diagnosis workflow of interlinked chains of algorithms *Measurement FOUP Integration / Vibra. Analysis in AMHS*: • Input: StkH3-EPP1 • Output: StkH3-EPP2; design of the toolchain for automated vibration measurement on simulated environments
Procurement & Engineering	**Activity** This step includes the selection of diagram types based on the concrete machine including properties as rotation frequency and external noise sources. Additionally, feature evaluation methods have to be selected, i.e. KPIs that are relevant for condition estimation of the machine type. *PDM / condition monitoring in Semicond. Facility*: selection of hardware and diagnostic diagram types based on the concrete machine.

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td> Measurement FOUP Integration / Vibra. Analysis in AMHS: Missing features of the measurement toolchain have to be retrofitted, e.g., WLAN functionality for the meas. FOUP. Tools PDM / condition monitoring in Semicond. Facility: Input: StkH2-EPP2;
Output: StkH2-EPP5; selection of hardware Measurement FOUP Integration / Vibra. Analysis in AMHS: Input: StkH3-EPP2
Output: StkH3-EPP3; implement missing features of the measurement toolchain, e.g., WLAN functionality for the meas. FOUP</td></tr>
<tr><td>Deployment & Commissioning</td><td> Activity The hypothesis consolidation rules have to be adapted to the use case, i.e. the rules of pattern recognition for creating final hypotheses from diagram-specific primary hypotheses. In addition to that, the hypothesis generators have to be parameterized. PDM / condition monitoring in Semicond. Facility: Parameterization and adaption of diagnosis hypothesis generator. Measurement FOUP Integration / Vibra. Analysis in AMHS: The toolchain will be transferred from PoC phase to the server environment of the semiconductor manufacturing plant and tested in different conditions, e.g., in areas sparsely covered with WLAN. Tools PDM / condition monitoring in Semicond. Facility: Input: StkH2-EPP4; StkH2-EPP5
Output: StkH2-EPP6; measurement system with parameterized and adapted diagnosis hypothesis generator Measurement FOUP Integration / Vibra. Analysis in AMHS: Input: StkH3-EPP2
Input: StkH3-EPP3
Output: StkH3-EPP4; transfer from PoC phase to the server environment of the semiconductor manufacturing plant</td></tr>
<tr><td>Operations & Management</td><td> Activity Based on breakdowns typical vibration patterns preceding the breakdown can be learnt for improving the prediction quality of the wear model. PDM / condition monitoring in Semicond. Facility: improving the prediction quality by observing typ. Vibration patterns. Measurement FOUP Integration / Vibra. Analysis in AMHS: Will be performed by maintenance engineers and AMHS experts of the fab. Tools PDM / condition monitoring in Semicond. Facility: Input: StkH2-EPP6;
Input: StkH1-EPP2; feedback about system performance
Output: StkH2-EPP4(a); improving the prediction quality by adapting the diagnosis workflow Measurement FOUP Integration / Vibra. Analysis in AMHS: </td></tr>
</table>

Engineering process phase	Addressed/Focus
	• Input: StkH3-EPP4 • Output: StkH1-EPP4; test in different conditions, e.g., in areas sparsely covered with WLAN -> additional requirements or inadequacy of the tool chain will be collected
Maintenance Decommissioning & Recycling	**Activity** The algorithms and platform have regularly be brought to the latest revision. The vibration sensors have to be checked regularly. After changes of the machine, changes of the model properties may be necessary. *PDM / condition monitoring in Semicond. Facility*: Check vib. Sensors regularly, adopt model properties to changed operating conditions. *Measurement FOUP Integration / Vibra. Analysis in AMHS*: Will be performed by maintenance engineers and AMHS experts of the fab. **Tools** *PDM / condition monitoring in Semicond. Facility*: • Input: StkH2-EPP6; • Input: StkH1-EPP2; • Output: StkH2-EPP4(b); adopt model properties to changed operating conditions *Measurement FOUP Integration / Vibra. Analysis in AMHS*: • Input: StkH3-EPP4 • Output: StkH1-EPP5; usability tests and benefit assessment -> additional requirements or inadequacy of the tool chain will be collected
Evolution	**Activity** Operation and management leads usually to new information about wear relevant aspects of the machine. This information can be used to improve the analysis software. *PDM / condition monitoring in Semicond. Facility*: Incorporate new information relevant to wear into analysis algorithms. *Measurement FOUP Integration / Vibra. Analysis in AMHS*: Additional requirements or inadequacy of the tool chain will be collected and subsequently remedied. **Tools** *PDM / condition monitoring in Semicond. Facility*: • Input: StkH2-EPP6; • Input: StkH1-EPP2; • Output: StkH2-EPP4(c); incorporate new information relevant to wear into analysis algorithms *Measurement FOUP Integration / Vibra. Analysis in AMHS*: • Input: StkH1-EPP4 • Input: StkH1-EPP5 • Output: StkH3-EPP4(a); additional requirements or inadequacy of the tool chain will be subsequently remedied
Training & Education	**Activity** Different stakeholders have to cope with the software, reaching from operation personal over diagnosis experts to managers. This requires training material for each kind of user.

Engineering process phase	Addressed/Focus
	PDM / condition monitoring in Semicond. Facility: Provide for different stakeholders, e.g., operation and maintenance personnel. *Measurement FOUP Integration / Vibra. Analysis in AMHS*: Trainings and training material on the usage of the AMHS vibration monitoring tool chain will be provided. **Tools** *PDM / condition monitoring in Semicond. Facility*: • Input: StkH2-EPP6; • Input: StkH2-EPP4(c); • Output: StkH2-EPP7; provide for different stakeholders, e.g., operation and maintenance personnel *Measurement FOUP Integration / Vibra. Analysis in AMHS*: • Input: StkH3-EPP4(a) • Output: StkH3-EPP5; Trainings and training material on the usage of the AMHS vibration monitoring tool chain will be provided

16.4.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 88 UC-16.4 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	If user friendly tools for the runtime management of sensor nets (sensing services) and service modules) are available, smart sensor nodes could be configured on-the-fly, which would make the integration of new sensors or data processing modules much easier and possible during run-time. The analysis of the solutions for both of the predictive maintenance and machine diagnostic use-cases in the evolution phase leads to the development of new and enhanced releases of the products.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Single stakeholder: Previously, the maintenance engineers of the semiconductor manufacturing plant had to manually measure and process the data, which was a tedious task. The same holds true for the use-case of vibration measurement within the facility area. Before the project, the measurements were conducted manually and stored in a legacy DB. Transition to multi stakeholder: Once the new demonstrator, incorporating several AH framework services, is ready for evaluation and deployments, these tasks are mostly automated and the data is automatically merged with core data from different areas of the fab. This will greatly enhance the productivity of the tool. The tedious data acquisition and evaluation within the facility machine diagnostic use-case is eliminated by the usage of the Conimon measurement system and WebApp with its automated machine diagnostics possibilities.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Splitting (mostly) monolithic applications, like the advanced machine diagnosis platform, into several modules (services) may allow for a higher scalability (e.g., multiple data analysis modules in parallel) and robustness (multiple databases in parallel). To be decided once the FOUP-AMHS vibration measurement demonstrator (testbed) is ready for evaluation.
Obj. 4 - Address digital learning	By improving the automated machine diagnosis methods, much less expert knowledge is required to interpret the results. Smart displays on sensor nodes and –

WP2 Objective	Focus & Planed actions
and training activities as an integral part of the engineering cycle	hubs could also provide digital and interactive guidelines for the installation and positioning, which also reduces the effort within the Deployment & Commissioning phase. By adopting the AH-EP, a policy is in place to enforce an efficient and exhaustive user training and education for the new, automated products.

Table 89 UC-16.4 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Since an automatic machine diagnostic solution is currently non-existent within the industrial domain considered in UC 16, it is primarily aimed at reducing the maintenance efforts and improving monitoring coverage and machinery uptime. In addition, the smart machine diagnostic software platform will be enhanced to ease the creation of maintenance routes, asset management and measurement hardware configuration to further reduce the integration effort during the roll out to further machinery, i.e., mainly targeting the education & training as well as the deployment and commissioning costs as described previously.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Probably not applicable in case of the automatic machine diagnostic approach.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Although the automatic machine diagnostic approach will be mainly implemented and evaluated using an existing smart maintenance software platform, the integration of selected sub-components, e.g., data acquisition from different standard interfaces, e.g., IEPE, SECS/GEM or InterfaceA (EDA; Equipment Data Acquisition), pre-processing, caching and data storage of this platform could be accomplished using the Arrowhead Framework in the future, especially in case of success and the possible roll out / scaling to further machinery.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Some identified use-cases, e.g. measuring the vibration introduced to transported goods by automated material handling systems, require the integration of advanced data collection and integration frameworks, e.g., the software stack of elastic search, to be compatible with the existing IT infrastructure. If appropriate modules are provided, the AH Framework could serve as a gateway between this software stack and the raw data provider (Measurement FOUP, data transferred as compressed plain text via WLAN).
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Since the data acquisition and evaluation for the automatic machine diagnostic approach will be executed on premise of the involved partners, these aspects are of low priority for this particular aspect.
Obj. 6 - Training material (HW and SW) for	Training activities for maintenance experts and end users regarding the correct usage of the smart maintenance platform, e.g., how to correctly input kinematic data, will be planned and executed. Guidelines regarding the correct installation of measurement

Project Objective	Focus & Planed actions
professional engineers	equipment and sensors to retrofit additional machines will be compiled and provisioned. The further automation of the diagnosis process including automated maintenance suggestions will reduce the required training & education effort.

16.4.4 Engineering Process analysis

Table 90 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-16-4

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1			X		X	X	X	X	
OBJ-AHT #2									
OBJ-AHT #3			X			X			
OBJ-AHT #4					X	X			
OBJ-AHT #5			X						
OBJ-AHT #6			X	X	X	X	X	X	
OBJ-WP2 #1		X	X	X	X		X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3			X	X	X	X	X	X	
OBJ-WP2 #4			X			X		X	

16.5 UC-16.5 Tester Integration (IFD)

Existing semi-conductor tools do not support interfaces for digitalization. Upgrading these systems with replacing new ones in order to take pace with the emerging technologies is not a cost-effective solution. These systems are built as a monolithic architecture based on SECS/GEM standard for special/singe purposes. Performing any changes in their structure causes large amount of cost and time. In the use case of Infieon Regensburg in Germany various data from PLC and sensors of the production environment will be collected. The collected values will be transferred using SECS/GEM protocol between machines and a central database system. These data will be stored in database for further purposes e.g., predictive maintenance.

16.5.1 Overall description of the UC-EP

This system consists of a PLC and monitoring system which collects machine data in the production. The second part of the system is a database which stores the production data from various machines and monitoring sensors. There are also a group of industrial PCs which perform some aggregation on the data before sending them to the database. Both options are illustrated in Figure 67.

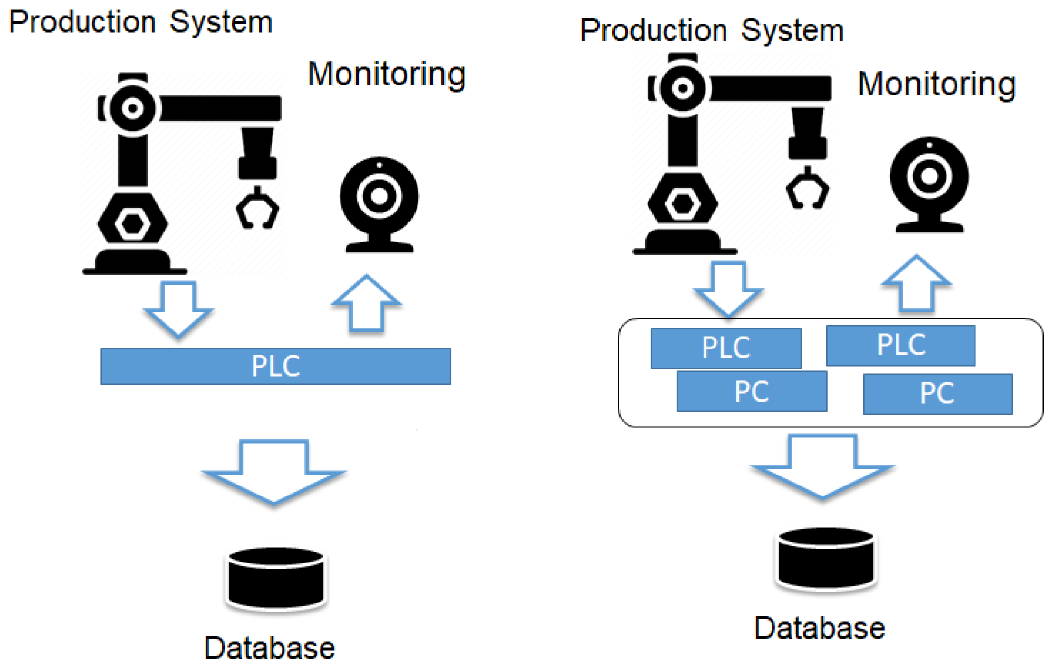


Figure 67 Two options of production systems

The engineering process consists of:
1. Decomposition of the system into subsystems and sub functionalities.
2. Generating a shadow system which simulates the current process in software level.
3. Validation of the software simulation with comparison of actual system with software values.
4. Replacing the parts of the actual system with separated sub systems and monitor their behavior through test phases.
5. Migrating the entire system to a separated system based on service oriented architecture concepts.

16.5.2 Engineering Process Description

In the Engineering process of use case 16.5. In this UC we have three stakeholders:
Stkh1: University of Luebeck
StkH2: Process engineering team from Infineon Regensburg
StkH3: IT team from Infineon Regensburg

C1: StkH1 plans to integrate AHF into their production environment. They ask the StkH2 to provide the implementation of AH Core and clients.
C2: StkH2 analyze the system and the possibility of deployment of AH Clients into the systems
C3 and C6: StkH2 provides the implementation of AH Clients for the PLC and Data monitoring system
C5: StkH1 generates material for training and installation guides for reproducibility of the engineering process
C7: StkH2 uses the training material
C8: StkH2 decides to launch the provided AH Cleints beside the existing system.

C9: StkH2 uses the tutorials provided by StkH1
C11, C10: StkH3 requires the details of the implementation and also the recommendations from Stkh3 as well as the tutorials for applying the implementation into the Production system of Infineon
C12: StkH3 provides feedbacks of integration and also the interface to StkH2. It makes it possible to monitor and evaluate the generated evolution in the production systems using AHF
C13: StkH2 reflects the feedback of the integration as new requirements for further development of AHF and AHT.

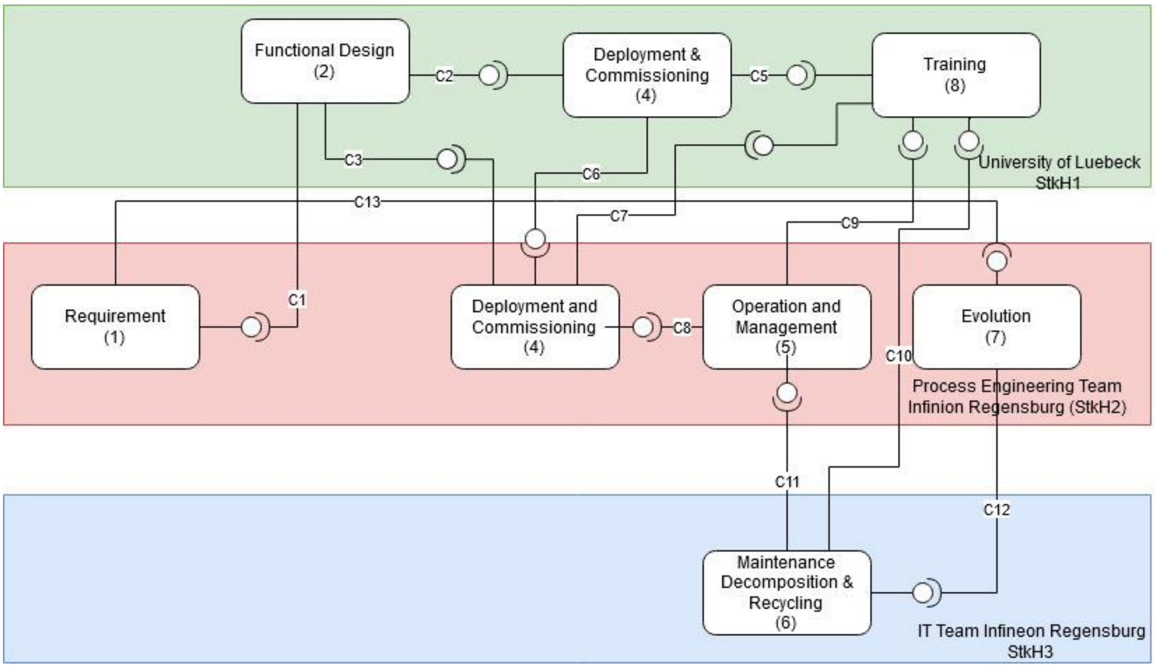


Figure 68 AHT-EP of use case UC-16.5

No framework is currently in use for the management of the engineering process.
We are using weekly meetups within the development team and biweekly meetings with the Infineon team. We use the iterative project management methodology. In order to track the tasks and description of the internal and external team meetings we use MS OneNote.

Since our use case consists of two simple services, we do not invest much on scalability.

The information is passed through emails, meetings and protocols. For managing the meetings we used webex and captured the meetings with OBS studio. Microsoft word and Microsoft excel is used to capture the information as meeting protocols.

Table 91 AHT-EP Phase focus of UC-16.5

Engineering process phase	Addressed/Focus
Requirements	Requirements engineering and analysis of current state of the system.
Functional design	Making preliminary design and development of architectural models

Engineering process phase	Addressed/Focus
Procurement & Engineering	Making the first iteration of system patterns
Deployment & Commissioning	n.s.
Operations & Management	n.s.
Maintenance Decommissioning & Recycling	n.s.
Evolution	n.s.
Training & Education	Some quick setup guidelines and documentations are missing. It is not specified in the documentation how someone can build a simple setup of arrowhead framework with a simple demonstrator. Everything is described details which make it hard to get a fast overview on AH basic functionalities. The configuration of provider is very simple but the configuration of the consumer requires a lot of effort.

16.5.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 92 UC-16.5 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	Separation of system into functionalities makes this objective possible.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Changing the monolithic systems into modular systems increases the chance of higher level of automation and digitalization.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	We should consider now more than one system. Each of these separated systems requires its own code base, documentation management, test environment.
Obj. 4 - Address digital learning and training activities as an integral part of	In order to be able to communicate with external team members (from Infineon) we need more learning and training material in each level to be able to describe why we should integrate AHF and how should we work further.

WP2 Objective	Focus & Planed actions
the engineering cycle	

Table 93 UC-16.5 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Sharing and Reusing the existing experiences from other use cases.
Obj. 2 - Interoperability for IoT and SoS engineering tools	Develop the service interfaces and integrate them to the existing system.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	There is no legacy automation tools in our project.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Migration to the service oriented architecture with well described service interfaces. Automatization of on-boarding process for service consumers of AHF.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	On the one hand, generating certificates for AHF, Service Provider and service consumer is not so simple. On the other hand, running AH Core without security mechanisms is not.
Obj. 6 - Training material (HW and SW) for professional engineers	Publishing white papers / videos and tutorials which makes it easier to reuse the engineering processes.

16.5.4 Engineering Process analysis

Table 94 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-16.5

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	P	C	C	C	C	C	N
OBJ-AHT #1	X	X	X	X					
OBJ-AHT #2		X	X						
OBJ-AHT #3									
OBJ-AHT #4			X	X	X	X			
OBJ-AHT #5			X						
OBJ-AHT #6		X	X		X	X		X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2	X	X	X	X			X	X	
OBJ-WP2 #3			X		X			X	
OBJ-WP2 #4		X	X		X	X		X	

17. UC-17 Linking a Building Simulator to a Physical Building in Real-Time (AEE INTEC)

The goal of the use case is to optimize the performance of a building – including the HVAC system and controls – in an automated way during operation, and even if the building use or user behaviour changes.

Larger buildings such as office buildings or industrial buildings are usually equipped with a building automation system (BAS). This system uses a variety of sensors installed to automatically control the room conditions (temperature, in some cases humidity) to the values desired by the building operator or users. In some cases, shading systems or other equipment is also automatically (or semi-automatically) operated.

In most cases, the operation of a building and its HVAC system is currently not optimized at all after commissioning. If optimization takes place, in a first step the building and building services are equipped with an additional monitoring system (often only in conjunction with a research project). After monitoring for a year, the results are analyzed and problems are detected. Afterwards, the control system can be adjusted manually to increase thermal comfort and reduce energy consumption. Continuous monitoring would be necessary to evaluate the benefits and ensure optimized operation.

Therefore, the baseline (M0) of the use case is a standard setup of a building automation system (BAS) with a variety of sensors and a monitoring campaign of 1-2 years to ensure energy efficient operation of the system.

17.1 Overall description of the UC-EP

In the baseline scenario, there are three stakeholders:

- The installer/planner of building services and building automation system (StkH2)
- The facility management (StkH3)
- A research institute conducting the monitoring campaign to optimize the performance of the building services (StkH4)

The main stakeholder in the improved (automated) optimization process (the developer of the simulation program and building tracker, StkH1) does not play a role in the baseline scenario. The building and system simulation software developed by StkH1 may be used by StkH2 in the design phase of the building services, but that is not always the case. Standard office buildings are often designed without the help of a simulation tool.
Figure 69 shows schematically the process of optimization in the baseline scenario. The building equipped with monitoring sensors is connected to a data logging system. Monitored data is fed into a data base or other analysis tools. Then, the data has to be analyzed manually and improved control settings or other possible improvements of the building services are identified manually. Adjustments to the control system of the building are also done manually.

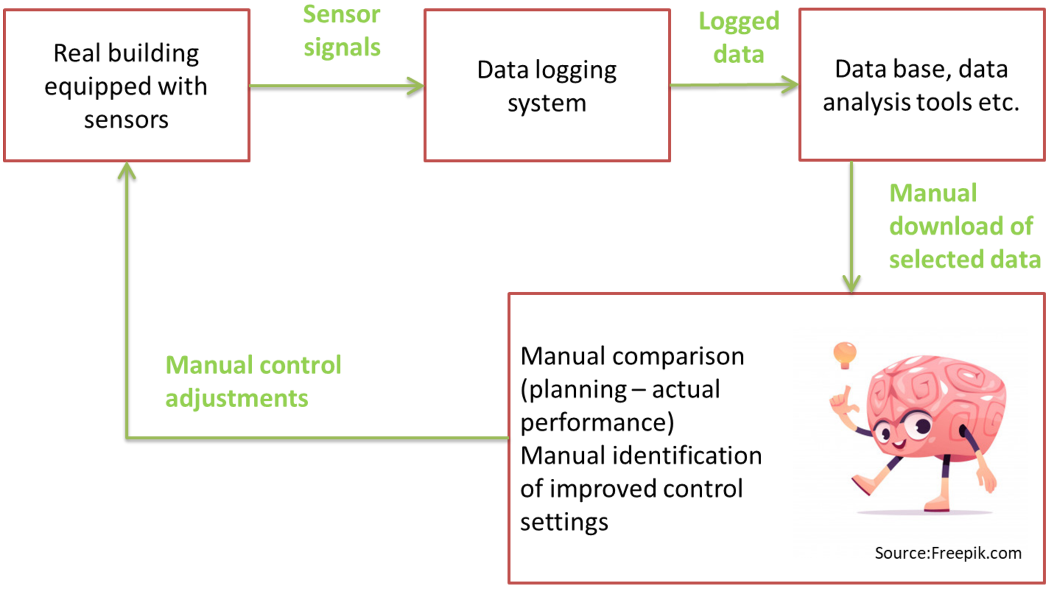


Figure 69 Process of optimization in the baseline scenario of the UC

Optimization of the performance of a building and its building services is rarely done after commissioning of the building.
If an optimization takes place, this cannot be done using the existing building automation system. Instead, a monitoring campaign is carried out (often in the framework of a research project). In these cases, a research institute plans and installs a number of sensors, logs the data for e.g. one or two years, analyses the data and makes suggestions to improve the building services. This procedure is expensive and while some software tools may be used, a lot of it has to be done manually.
There is no standardized engineering process. Automation may be available for certain steps within the engineering process.
A monitoring campaign includes all engineering phases up to maintenance/ decommissioning. There may even be evolution evolved assuming that the concept may be improved and used for future projects (different buildings).

In the main blocks of the use case where:

- Building Tracker (includes IDA modeler and IDA solver): The building tracker is to a large extent ready for commissioning to provide virtual sensing capabilities. Development and refinements are ongoing with respect to integration with the AHF and for automatic parameter tuning. The details of the services to be provided and consumed through the AHF remain to be specified.
- Occupancy and behavior tracker: The tool is in active development and will be Arrowhead compliant. Measurement data gathering has started for development and validation purposes, and a first prototype is expected to be available this year. Also here the detailed specification of the services consumed and provided remain to be written.
- OPC UA server: The tool itself is not developed within the AHT project, but an integration for the AHF is provided by WP3(?) including the service definitions and specifications.
- IDA simulation environment: Building and system simulation tool already on the market.
- Onboarding and startup tool: Is in active development within the project.

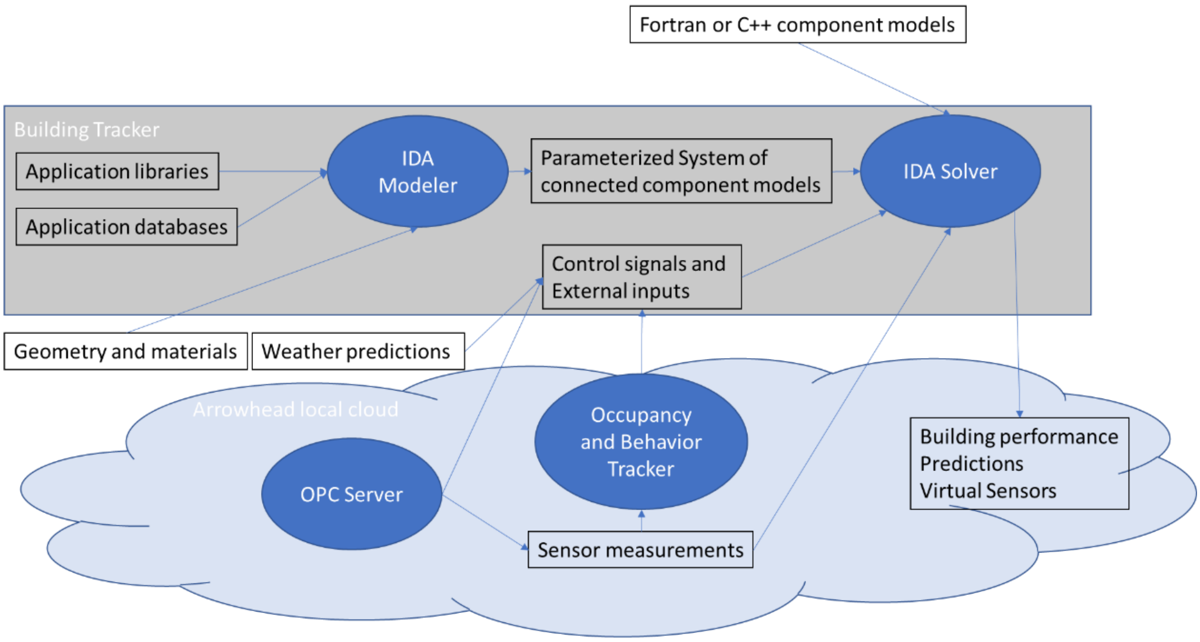


Figure 70 Main blocks of the UC

17.2 Engineering Process Description

The following schematic shows the engineering process for the optimization process for a specific building.
Three stakeholders are involved: The developer of the building tracker (StkH1), the installer and/or planner of the building services (StkH2) and the facility manager (StKH3) who is responsible for the operation of the building.
All three stakeholders work together to provide requirements for the planned optimization system based on their knowledge about the building and its services (c7, c8 and c9).
The functional design is then performed by StkH1 who designs the toolchain that will be realized in the building (depending on the available BAS) and as well as specifies the parameters and algorithms for the optimization process. This is done in cooperation with the planner of the building services (StkH-2, c10 and c11).

Procurement and engineering is also carried out by StkH1. Once the optimization system is designed and engineered it can be deployed and commissioned by StkH2 (Installer).
The operation of the system is the task of StkH3 (Facility Management). The preparation of training material and conduction of training courses is done by STkH1 who trains the other stakeholders (c14 and c15).
The development of training materials is based on information from phases functional design (c12) and procurement and engineering (c13).
An evolution of the optimization system based on the experiences and feedback from users can then be done by STkH1 which will then lead to new requirements (c6). This will start a new engineering process which may lead to updated algorithms or updated software for the building tracking system of that specific building. StkH3 is responsible for maintenance.

The use case concentrates on the operation phase of the provided service (optimization of building operation) and develops a toolchain and some of the required tools to accomplish this service. The engineering phases (requirements, Functional design, procurement and engineering and deployment and commissioning) are executed for the setup of the service in the demonstration building but are not part of the toolchain.

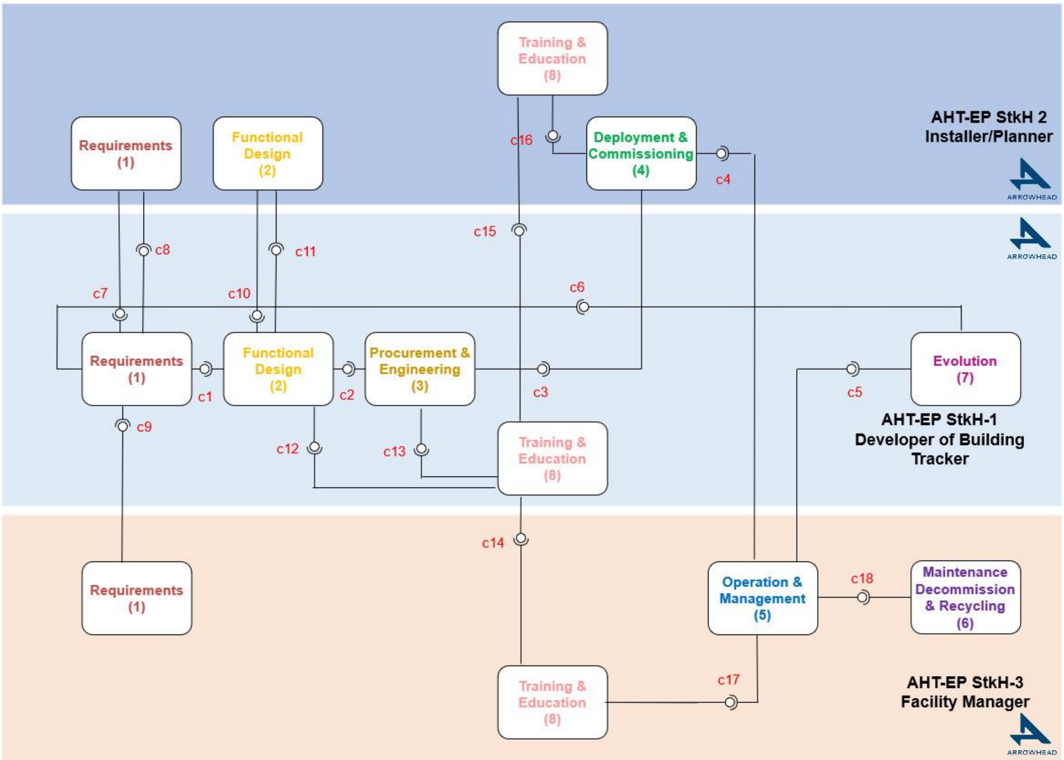


Figure 71 AHT-EP of use case UC-17

Base line scenario uses completely different approach to solve the optimization problem.

In the current setup, all sensors in the building are registered in the arrowhead framework as a package (via the building management system) and not separately. For the future, it could be envisaged to use smart sensors that are able to connect directly to the local cloud. That would mean each of them could provide a separate service that could be consumed by the building tracking system.

The engineering phases can also be applied to the development of the tools (building tracker and occupancy and behavior tracker) and the service (optimization of the building performance) that is being provided.
In that case the responsibility for all engineering phases is with StkH 1. Inputs regarding the requirements may come from StkH2 and StkH3. StkH1 will provide training for the other stakeholders so they can take over the Deployment & Commissioning, Operation & Management and Maintenance, Decommissioning & Recycling at the end of the lifetime.

Table 95 AHT-EP Phase focus of UC-17

Engineering process phase	Addressed/Focus
Requirements	**Activity** StkH3 and StkH2 communicate specifications of the building and its building services (including the type and brand of BAS) to StkH1 as an input to the building and system model and the configuration of the OPC UA server.
Functional design	**Activity** Definition of the setup of the toolchain according to the used BAS type and brand in cooperation between StkH1 and StkH2. The IDA Simulation environment is used for the design of the building and system model.
Procurement & Engineering	**Activity** The components for the building tracking system are purchased by StkH1. OPC UA server and building tracker (building and system model in the IDA Simulation environment) are configured with the list of available sensors in the BAS by StkH1/StkH2.
Deployment & Commissioning	**Activity** Installation and commissioning of the entire building tracking system by StkH1 (Occupancy and behaviour tracker, building tracker, on boarding and start-up tool).
Operations & Management	**Activity** OPC UA server, building tracker, occupancy and behaviour tracker are used for continuous optimization of control parameters and for fault detection by StkH3.
Maintenance Decommissioning & Recycling	**Activity** Outputs of building tracker can be used for fault detection/maintenance purposes (of the building services) by StkH4.
Evolution	**Activity** Collected information from the operation and management phase will be used for future updates of the building tracking system.
Training & Education	**Activity** Training material to be developed by StkH1 to train StkH2 and StkH3

17.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 96 UC-17 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	There is the possibility to update the building tracking software and its algorithms during the lifetime of the system. The experiences during the operation can be used to develop new or improved algorithms for the building tracker. This process can be called run time engineering.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	The baseline scenario, includes mainly one stakeholder (in many cases a research institute) that conducts a monitoring campaign and suggests improvements of the building services and its controls. The building tracker scenario includes a few stakeholders that are directly involved in the design and operation of the system.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	While the baseline scenario was not automated at all, the building tracking scenario allows to track multiple sensor signals including the possibility of fault detection and the possibility to feed optimized control signals back in the BAS.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Training activities are in integral part of the use case because it is important to transfer the know-how of the building tracking system from the developer of the building tracker to the other stakeholders. In the future (after a number of buildings has been successfully been in equipped with a building tracking system), some of the engineering phases may even be transferred to planners or installers of BAS. This may make sense for the functional design phase and the procurement and procurement and engineering phase. The building tracker developer could than only be the vendor of the building tracking software as well as of various algorithm options for different building configurations.

Table 97 UC-17 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	The developed building tracking system will reduce energy costs due to optimization of the control strategies for building services by implementation of advanced models (occupants sensing, weather forecast, fault detection, etc.). On the other hand, the costs can be reduced due to the use of virtual sensors and also because the building tracking system replaces manual data analysis and optimization.
Obj. 2 - Interoperability for IoT and SoS engineering tools	n.s.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead	n.s.

Project Objective	Focus & Planed actions
Framework integration platform	
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Interoperability of the different components in the newly developed building tracking system will be investigated in a laboratory setup. Possibility of implementation in the demonstration building will be evaluated.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	It is investigated how the Arrowhead Framework can be used to ensure security for the data transfer between sensors and building tracking system.
Obj. 6 - Training material (HW and SW) for professional engineers	Use case will prepare training materials for the developed building tracking system.

17.4 Engineering Process analysis

Table 98 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-17

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X		X		X		
OBJ-AHT #2									
OBJ-AHT #3									
OBJ-AHT #4		X	X				X		
OBJ-AHT #5			X		X				
OBJ-AHT #6		X	X	X	X			X	
OBJ-WP2 #1	X	X	X				X		
OBJ-WP2 #2		X	X					X	
OBJ-WP2 #3	X	X	X	X	X	X	X	X	
OBJ-WP2 #4		X	X	X	X			X	

18. UC-18 Secure sharing of IoT generated data with partner ecosystem (Boliden)

Data in mining industry is a key component and currency for further and also disruptive increase in operational efficiency and safety. SCADA's, PLC's, machines, vehicles, any equipment used in production process generates valuable data and from may sides vendors / companies are pushing into this space to secure position to build further/new business or optimize current operations. Boliden has the position to own all data generated by equipment in operation for Boliden. Main purpose of this activity is to enable Boliden to have a secure way to make relevant data available via an open platform approach (does not have to be Boliden internal) to securely share data within the partner ecosystem including tracking use rights, traceability of usage and possibility to invalidate data.
Today, data access is handled by existing tools and following issues exist: 1. Potential legal issues arise, as legacy systems and current approaches do not provide possibility to security segment data. 2. Concept for data tagging is not available as standard 3. Possibility to trace where Boliden data is going and possibility of invalidation is not existing today. If addressed, this can not only prevent potential penalties but also provide a secure and efficient way to share data with Boliden ecosystem to optimize not only Boliden's production but also helps out partners to improve their products.
There are two baseline for this tool to look at

1. The secure data sharing without the platform and security concept in place. The baseline consist of legacy way of sharing data in tight quality and properly secured without the common platform in place including the initial security concept. The aspect for improvement are efficiency in data access from request to delivery but also security and data quality topics.
2. The implemented platform with initial security concept in place, under investigation. This baseline we want to use for potential comparison with improved access control mechanism based on LTU research work based on NGAC.

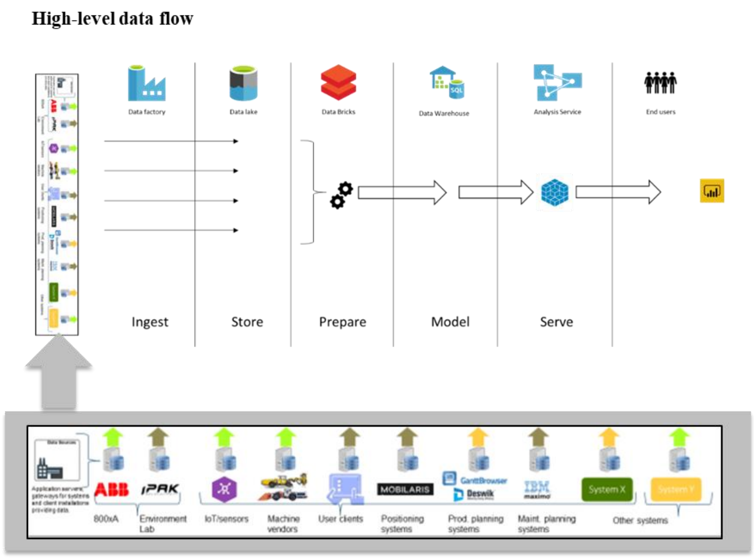
For more details about the use case description see Deliverable D9.1 (Task 9.6).

18.1 Overall description of the UC-EP

The initial approach prior to having a common data platform in place was individual activities on demand where the entire chain from data source to cleanings / quality assurance, enrichment with asset information, aggregation and consolidation was project by project driven and many decisions often with external parties. With respect to security the individual awareness was a determine factor and due to project pressure always at risk.

The new design includes a data platform where security is “built-in” and considered alongside the usage from generation in a sensor to visualization in consolidated reports or in data set for analysis. With the platform concept we separate 1) the inclusion of data sources and 2) the provisioning of data to enable a proper setup and inclusion of data and to increase speed to delivery for needed data based on business needs. A heavy focus is on security where for each included data source the sensitivity has to be reflected in meta-data and on the other hand user / role access to be mapped to access right when data is needed.

The high-level architecture is described below in Figure 72.



Description for picture – use case operates more on operability level but is based on the IoT data sources. To address the need to e.g ensure partners do not see their each others R&D machine data, security has to start at the source and be managed in a efficient way when transferring, sharing or for aggregated reporting. On the left and bottom are the data sources in scope and many already implemented, the data flow is high-level and shows the data handling on alongside the data sharing lifecycle.
Tools we use MS Azure based and is fed from with real-time data directly or from Osisoft PI solution but also other data like production scheduling

Interesting things:
Machine engineering companies are often not great software companies…
Even same machine different model can have different data quality issues and interpretations of values, e.g. missing value or "0"
suppliers are trying to find business model for their data and as we need the data for overall production optimization this is a hurdle, e.g. willingness to share how data should be interpreted or contractual tropics on usage etc.

Figure 72 UC-18 high-level architecture

The platform with implemented security concept as indicated above three separate items can be considered. The engineering process of the security approach, the engineering process of including new data sources into the data platform and the engineering process to make new data sources available. The focus will here be on the security approach.
Risk areas and key requirements will be defined in requirements phase by the IT platform team together with business representation from mining department including business data owners and financial department. The functional will be developed by the platform team and validated again existing data sources and known cases for data sharing. Functional design will be adopted as part of the evolution phase based on real-world case implementation and learning of improvements with LTU. The platform development is performed together with preferred supplier for data & analytics area and deployed to the platform following change process with change review by the platforms governance team. The security solution will be operated and maintained as a component in the data platform used in real-life cases to validate and evolve. The evolution includes feedback from rea-life cases that the platform team collects and if possible implements and also in corporation with use-case partners, e.g. NGAC based access management and possible automation. Training and education is done within the platform team and across needed colleagues outside. One key requirements is to ensure data ownership and the definition of the owners for attributes related to security. This has to be included in communication and change management as well.

18.2 Engineering Process Description

In the engineering process of use case 18. In general, three components are looked at in this use case:

1. data integration of a data source into the data platform requested by StkH3 and implemented with StkH1,2,4;
2. security setup based on StkH1 requirements and details from StkH3
3. and the data provision to StkH3 based on preparation by StkH2& 4.

The three components can be executed together or timely separated, e.g. if data source are integrated before actual usage.

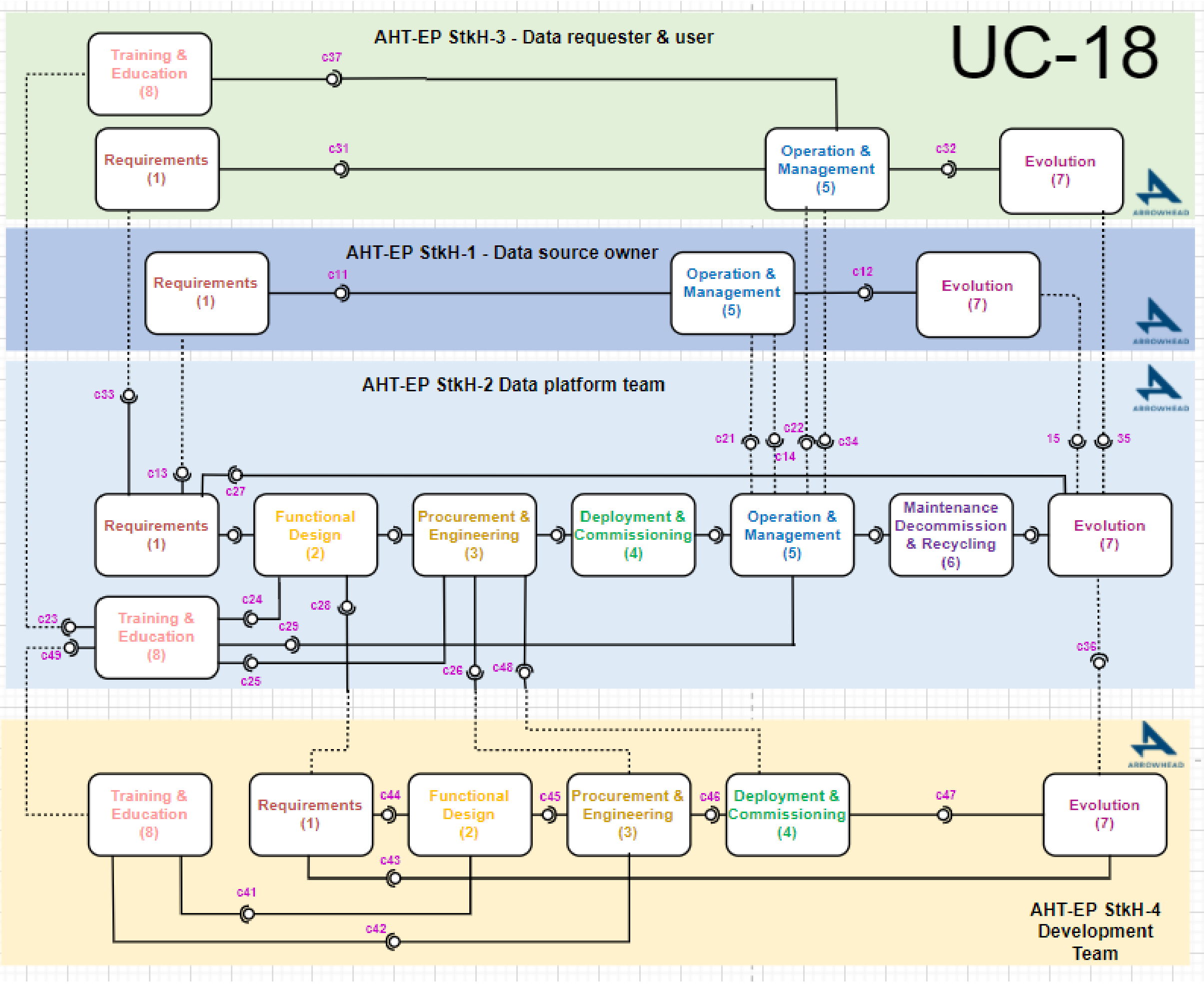


Figure 73 AHT-EP of use case UC-18

The legacy approaches used waterfall methodology usually driven by the external partner. There has been limited integration and learnings happening between activities. Security design and implementation has been on-offs involvements during projects
The new platform approach implements an iterative approach to a) include new data sources in sprints and b) make data available in use cases. Depending on the use-case this can be part of a business project (waterfall drive) or handled as an activity with the platform team (iterative). The process ensures a full lifecycle management and with focus on security: the right ownership and an attribute based approach to managed data access.

The phases of implementation match the AH-EP and are split up in two cycles one the integration of data sources and then the provisioning of actual data for a specific business need.

The tools developed are the finalization of the platform itself, the integration of few data sources, and the implementation of the security concept from sensor data to data provisioning and the validation and improvement of the security approach. This touches all 5 phases with respect to the security approach.

Testing as a phase is incorporate separate but an important part especially with our use-case. For our mapping we incorporated the testing in the procurement and engineering phase. All phases are incorporated.

Areas for scalability regarding life-cycle management are 1) number of data sources 2) number of data request. On data owner side and data user part the scalability is given. The legacy approach would allow for scalability in verticals with a huge risk to great duplicated work, complexity, duplication and security risk. With the new approach new data request for existing data triggers a fast provision process providing scalability for on the giving side (probably mainly seen as part of operation of the platform). The integration of new data sources is done pro-actively for major data sources and triggered by specific projects to create basis for scalability. Here, there is a bottle neck for scalability especially with the manageability of access right.

Microsoft Azure data & analytics reference architecture; NGAC as part of LTU investigation for access management and potentially included in use-case. Based on the AHT-EP mode we are compliant with some of the dimensions represented in the RAMI 4.0 model.

The difference between the original EP and the AHT-EP is the development that a new stakeholder, the platform team has been established to allow for iterative development alongside the engineering process. This has not been the case before and did lead to vertical development and separated solutions. As stakeholders in projects and data user are different the platform team allows for learnings, reuse, etc.

Table 99 AHT-EP Phase focus of UC-18

Engineering process phase	Addressed/Focus
Requirements	**Activity** *Data source integration into data platform*: Prioritized list of data sources to be included is created and maintained by SH2./ XX also project driven. *Security setup for data source*: For each data source in scope the SH2 is aligning security requirements with SH1. SH2 checks if prerequisites are available to set security in data platform. *Data provisioning*: Requirements for data provisioning are described by SH3. SH2 assesses need and decides on appropriate way to make the data set available. **Tools** *Data source integration into data platform*: SH3 requests DSI from SH2 • Input: <c33> SH3 request for prioritization and inclusion of data source into data platform and / or access to production related data • Input: <c27+A2> from Evolution from data users, data owner or data platform team to include data source based on prioritized list • Output: <SH2 - EPP2> Request details and requirements for integration needs and relevant parameters in enough quality to proceed to design phase, e.g. frequency, cleansing requirements *Security setup for data source*: SH2 request access security (Sec) requirements from SH1 • Input: <c33> SH2 request details on access and other parameters such as confidentiality from SH1. SH2 makes own assessment based on known security areas. • Input: <c27> from Evolution - from data users, owners and platform team to optimize access setup to data source, e.g. granularity, simplification • Output: <SH2 - EPP2> Requirements to allow for attribute based access

Engineering process phase	Addressed/Focus
	management setup *Data provisioning*: SH3 requests DP from SH2 • Input: <c33> SH3 data request access with details for access, e.g. user / group and usage: e.g. self-service analytics • Input: <c27> from Evolution - from Data platform team suggestions for additional data interesting in context for SH3 • Output: : <SH2 - EPP2> Requirements for specific access to data source for further usage / processing
Functional design	**Activity** *Data source integration into data platform*: SH2 iterates with SH1 the technical and functional details to load data into platform. *Security setup for data source*: SH2 applies security concept to defined parameters. If needed the data is enriched to ensure the efficient provision of data. *Data provisioning*: SH2 designs the provisioning of the data set. **Tools** *Data source integration into data platform*: SH2 designs DSI • Input:<SH2 - EPP1> SH2 specification on data source and relevant parameters • Output: <SH2 - EPP3> Design for integration into data platform or <c28> in case of external development by SH4 • Output: <c24> Documentation for internal SH2 *Security setup for data source*: SH2 design SEC access control setup • Input: <SH2 - EPP1>: SH2 consolidated requirements for security access design • Output: <SH2 - EPP3> Security design for implementation or <c28> in case of external development by SH4 • Output: <c24> Documentation for internal SH2 *Data provisioning*: SH2 uses SEC and DSI to design data set for provisioning (DP) • Input: <SH2 - EPP1>: SH2 consolidated requirements for data set access • Output: <SH2 - EPP3>design for data set provisioning or <c28> in case of external development by SH4 • Output: <c24> Documentation for internal SH2
Procurement & Engineering	**Activity** *Data source integration into data platform*: Based on parameters the data source is integrated by SH2. *Security setup for data source*: Handled as part of data source integration into data platform. *Data provisioning*: SH2 implements provisioned data. **Tools** *Data source integration into data platform*: SH2 requests development of DSI by SH4 or implements with own resources • Input: <SH2 - EPP2> SH2 Design specifications, depending on involvement of SH4 this is handed over to SH4 for development or handled withing the SH2 • Input: <c48>: SH4 integrated data source according to specification • Output: <c26> and also <c48>: SH4 interaction on setup of data source integration, e.g Azure components prepared by SH2, integration code

Engineering process phase	Addressed/Focus
	developed by SH4 • Output:<SH2 - EPP4> Data source integrated and tested • Output: <c25> Training information SH2, SH1 for integrated data source *Security setup for data source*: SH2 requests development of DSI by SH4 or implements with own resources • Input: <SH2 - EPP2> Access security pacifications, depending on involvement of SH4 this is handed over to SH4 for development or handled within the SH2 • Input: <c48>: SH4 implemented security concept ready for SH2 • Output: <SH2 - EPP4> Attribute based security settings implemented and tested • Output: <c25> Training information SH2, SH1 on security access processes and usage *Data provisioning*: SH2 sets up DP • Input: <SH2 - EPP2> SH2 or SH4 prepares data access based on specific security parameters • Input: <c48>: SH4 make data set ready for SH2 • Output: <SH2 - EPP2> Data source available for usage. This triggers as well needed trainings • Output: <c25> Training information SH2, SH3 on data set usage
Deployment & Commissioning	**Activity** *Data source integration into data platform*: The deployment & commissioning is made SH2. Verification of data quality by SH1. *Security setup for data source*: Handled as part of data source integration into data platform. *Data provisioning*: SH2 implements and test fulfilment in delivery according to spec, quality and security. **Tools** *Data source integration into data platform*: SH2 deploys data source after testing for completeness, correctness • Input <SH2-EPP3>: SH2 as above • Output <SH2-EPP5>: DSI in operation and handed over using created training material and documentation *Security setup for data source*: SH2 deployment of security setup • Input <SH2-EPP3>: SH2 as above • Output <SH2-EPP5>: SEC in operation and handed over using created training material and documentation *Data provisioning*: SH2 makes DP available for users • Input <SH2-EPP3>: SH2 as above • Output <SH2-EPP5>: DP in operation and handed over using created training material and documentation
Operations & Management	**Activity** *Data source integration into data platform*: Handled by SH2 as part of ongoing service management, Boliden has a model similar to the PM3 system support model. *Security setup for data source*: Handled as part of data source integration into data platform. *Data provisioning*: Handled by SH2.

Engineering process phase	Addressed/Focus
	Tools *Data source integration into data platform*: SH2 operates data source integration; • Input <SH2-EPP4>: deployed integrated data source • Input <c14>: data source owner information on data source, e.g. downtime, logical changes • Output <c21>: interaction on maintenance of data platform and usage reporting • Output <SH2-EPP6> maintenance needs, notification for decommissioning *Security setup for data source*: SH1 is involved in potential approval request • Input <SH2-EPP4>: Implemented SEC • Input <c14>: data source owner information on data source, e.g. data access changes, approval requests Output <c21>: Interaction on access on DSI • Input <c34>: changes on needed access for users or groups • Output <SH2-EPP6> maintenance needs, notification for decommissioning *Data provisioning*: SH1 uses DP in approved context • Input <SH2-EPP4>: Implemented DP • Input <c34>: data user operational request management • Output <c22>: Information on operations and maintenance • Output <SH2-EPP6> maintenance needs, notification for decommissioning
Maintenance Decommissioning & Recycling	**Activity** *Data source integration into data platform*: Handled by SH2 as part of ongoing service management, Boliden has a model similar to the PM3 system support model. *Security setup for data source*: Handled as part of data source integration into data platform. *Data provisioning*: Handled by SH2. Regular review for usage performed. **Tools** *Data source integration into data platform*: SH2 takes lead in case issues and manages decommissioning • Input <SH2-EPP5>: Operations data, issue reports, SH1,3 feedback • Output <SH2-EPP7> Improvement needs outside maintenance *Security setup for data source*: SH2 takes lead in case issues and manages decommissioning • Input <SH2-EPP5>: access data, issue reports, SH1,3 feedback • Output <SH2-EPP7> Improvement needs outside maintenance *Data provisioning*: SH2 takes lead in case issues and manages decommissioning • Input <SH2-EPP5>: Operations data, issue reports, SH1,3 feedback • Output <SH2-EPP7> Improvement needs outside maintenance
Evolution	**Activity** *Data source integration into data platform*: Handled by SH2 in alignment with feedback from SH1 and SH3. *Security setup for data source*: Security setup itself and changed security requirements are managed in regular service management meeting and included in release if needed. *Data provisioning*: Handled by SH2 in alignment with feedback from SH3 and SH1

Engineering process phase	Addressed/Focus
	(for changes coming from data sources). **Tools** *Data source integration into data platform*: SH2 consolidated feedback from SH1 and SH2 on DSI • Input <SH2-EPP6>: Improvement needs from maintenance • Input <c36>: Evolution ideas from development team on integration setup • Input <c35>: Data user feedback for evolvement • Input <c15>: Data owner feedback on data source integration evolvement • Output <SH2-EPP1> Improvement requirements details for DSI, e.g. quality of data, change of integration mechanism, change of aggregation, scheduling, etc. *Security setup for data source*: SH2 consolidated feedback from SH1 and SH2 on SEC • Input <SH2-EPP6>: Improvement needs from maintenance • Input <c36>: Evolution ideas from development team on integration setup • Input <c35>: Data user feedback for evolvement • Input <c15>: Data owner feedback on evolvement • Output <SH2-EPP1> Improvement requirements details for SEC e.g. granularity, access attributes, model, etc. *Data provisioning*: SH2 consolidated feedback from SH1 and SH2 on DP • Input <SH2-EPP6>: Improvement needs from maintenance • Input <c36>: Evolution ideas from development team on integration setup • Input <c35>: Data user feedback for evolvement • Input <c15>: Data owner feedback on evolvement • Output <SH2-EPP1> Improvement requirements details for DP e.g. usage reporting, performance, etc.
Training & Education	**Activity** *Data source integration into data platform*: SH2 created needed documentation for data sources, meta data and interpretations if needed. Education for SH1 is done data source by data source. *Security setup for data source*: Handled as part of data source integration into data platform. *Data provisioning*: SH2 educated data users on data usage possibility, limitation and duties. **Tools** *Data source integration into data platform*: SH2 aggregates material needed from development for operation of DSI • Input <c24>:Design documentation and information • Input <c25>: Engineering documentation for SH1,SH2 for documentation but also deployment, operation, maintenance and improvement processes • Input <c25>: Engineering documentation from SH4 • Input <41,42>: in case SH4 is used for development, this input aggregates needed documentation and training for SH2 • Output <c29> Relevant documentation and training for SH2 in operations & management *Security setup for data source*: SH2 aggregates material needed from development for operation of SEC • Input <c24>:Design documentation and information • Input <c25>: Engineering documentation for SH1,SH2,SH3 for

Engineering process phase	Addressed/Focus
	documentation but also deployment, operation, maintenance and improvement processes • Input <c25>: Engineering documentation from SH4 • Input <41,42>: in case SH4 is used for development, this input aggregates needed documentation and training for SH2 • Output <c29> Relevant documentation and training for SH2 in operations & management *Data provisioning*: SH2 aggregates material needed by SH4 • Input <c24>:Design documentation and information • Input <c25>: Engineering documentation for SH2,SH3 for documentation but also deployment, operation, maintenance and improvement processes • Input <c25>: Engineering documentation from SH4 • Input <41,42>: in case SH4 is used for development, this input aggregates needed documentation and training for SH2 • Output <c29> Relevant documentation and training for SH2 in operations & management • Output <c37>: training and documentation for data users including security rules

18.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 100 UC-18 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The objective matches as with the integration of data sources in the platform with appropriate security the provisioning can happing in system operation and not as separate projects with separate fixed design phases. The incorporation of the evolution phase was not present before and also enables this objective.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	This objective is achieved as in the legacy approach the work was completed in verticals with limited to one area (but with multiple stakeholders). With a new stakeholder as the platform, team standardization and automation can be achieved and all phases are connected.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Objective is met on data integration and data access level as the approach enables separation of inclusion of data sets with right security. information for security is connected in the EP with the help of the platform team and with attribute based access management data provision with right data access can be done efficiently.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The objective fits as the platform development and improvement process is orchestrated by the platform team and new developments such as the security concept validated. This Team also handles documentation, training and learning needs.

Table 101 UC-18 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	The reduction of the engineering costs is achieved directly and indirectly. With respect to secure sharing: Direct reduction: • Not a case by case discussion but standard for security management of data sources including governance leads to less time per data source integration • If data sources are included then reduction from request to delivery is dramatically improved as no separate project is needed. Indirect reduction: • No rework due to quality issued in data • Adds clarity to data users and data owners and reduces confusion
Obj. 2 - Interoperability for IoT and SoS engineering tools	The use-case is on interoperability level and builds on the ability to feed time-based data with correct tagging into the data platform.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The use-case is on interoperability level and allows for integration of legacy and AHF integration platforms.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	The use-case is on interoperability level and allows for integration of legacy and AHF integration platforms.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The use-case is on interoperability level and requires unique ID's per data sources to be able to enrich the information with asset information and use / access rights. Based on the defined concept this puts less requirements integration frameworks and allows for digitalization and automation.
Obj. 6 - Training material (HW and SW) for professional engineers	Probably not applicable.

18.4 Engineering Process analysis

Table 102 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-18

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	C	C	C	C	N
OBJ-AHT #1	X	X	X					X	
OBJ-AHT #2		X	X						
OBJ-AHT #3			X						
OBJ-AHT #4			X	X	X				
OBJ-AHT #5			X	X	X	X			
OBJ-AHT #6		X	X					X	
OBJ-WP2 #1		X	X	X	X		X		
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3	X	X	X	X	X	X	X	X	
OBJ-WP2 #4		X	X	X	X	X	X	X	

19. UC-19 Deployment and configuration (3E)

SynaptiQ Solar is the trusted B2B performance monitoring solution of key reference players, steadily growing in connected capacity with a CAGR +30%.
Customer can model their photovoltaic plants and use any kind of logger and file formats to send the data to SynaptiQ. The freedow of choice has a cost in configuration where each details must be configured (modules types, loggers types, file formats, etc.).

19.1 Overall description of the UC-EP

SynaptiQ Solar is the trusted B2B performance monitoring solution of key reference players, steadily growing in connected capacity with a CAGR +30%.
Description of the main steps of the use case engineering process represented in Figure 74:

- SynaptiQ Configurator Module: defines complete model of the photovoltaic plant (all the devices, loggers type and brand). All the configuration is manually done in a friendly-user interface.
- SynaptiQ Mediation Module: handles the heterogeneous data inflow on-site devices. The configuration is manually done in a XML file by our engineer operational team.
- SynaptiQ Data Engine: processes the inflow of information.
- SynaptiQ Operational: handles the visualization of the aggregated data and alarms.

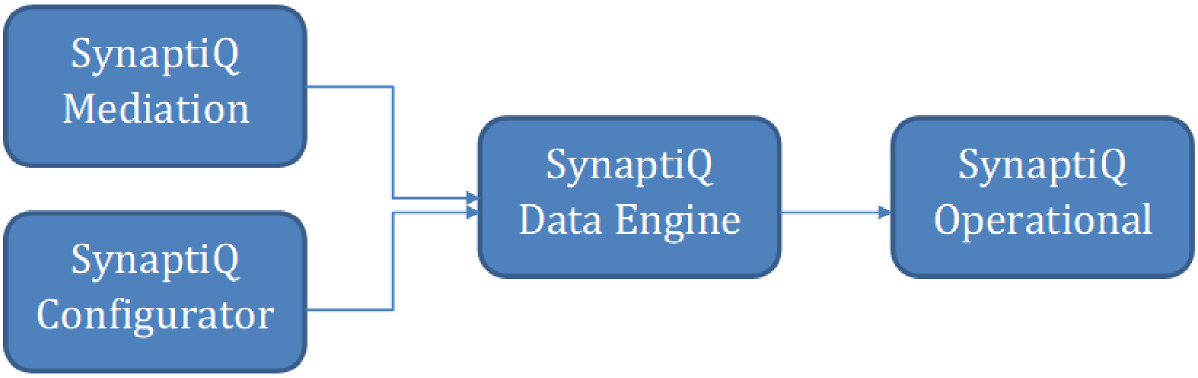


Figure 74 The main steps of the use case engineering

Customers manually configure their plant in the SynaptiQ configurator and validate it by publishing to the SynaptiQ Data Engine.
Our operational engineer team configure the Synaptiq Mediation Module based on the information the customer gave us or entered in the SynaptiQ Configurator.
Once bot Synaptiq Mediation Module and SynaptiQ Configurator configured, the data acquisition process can start and data are processed in the SynaptiQ Data Engine.
The data are shown in the SynaptiQ Operational Module with the devices and hierarchy of devices configured in the SynaptiQ Configurator.

In case of modification on site (replacement of an inverter for example), the modifications have to be mirrored in the SynaptiQ configurator and in the SynaptiQ Mediation Module.

The current process is fully manual. The customers configure their plants in the SynaptiQ Configurator or they provide the information to our engineer operational team via a ticketing system to configure the plant.
After that, the plant is published to our SynpatiQ Data Engine to prepare the acquisition of the data. This part is semi-automated, our engineering team have to trigger the validation.
The SynaptiQ Mediation Module is then configured manually, pushed to a repository and deploy to production. This step is fully manual.

19.2 Engineering Process Description

In the engineering process of the use case 19.
Only Deployment & Commissioning and Maintenance Decommissioning & Recycling phases are considered.
The main goal of our modules is to mirror the PV plant in a virtual model. Based on this model the raw data received will be parsed, stored and aggregated. Hence only the deployment (first configuration of the model) and the maintenance (update of the model) are relevant in our use case.

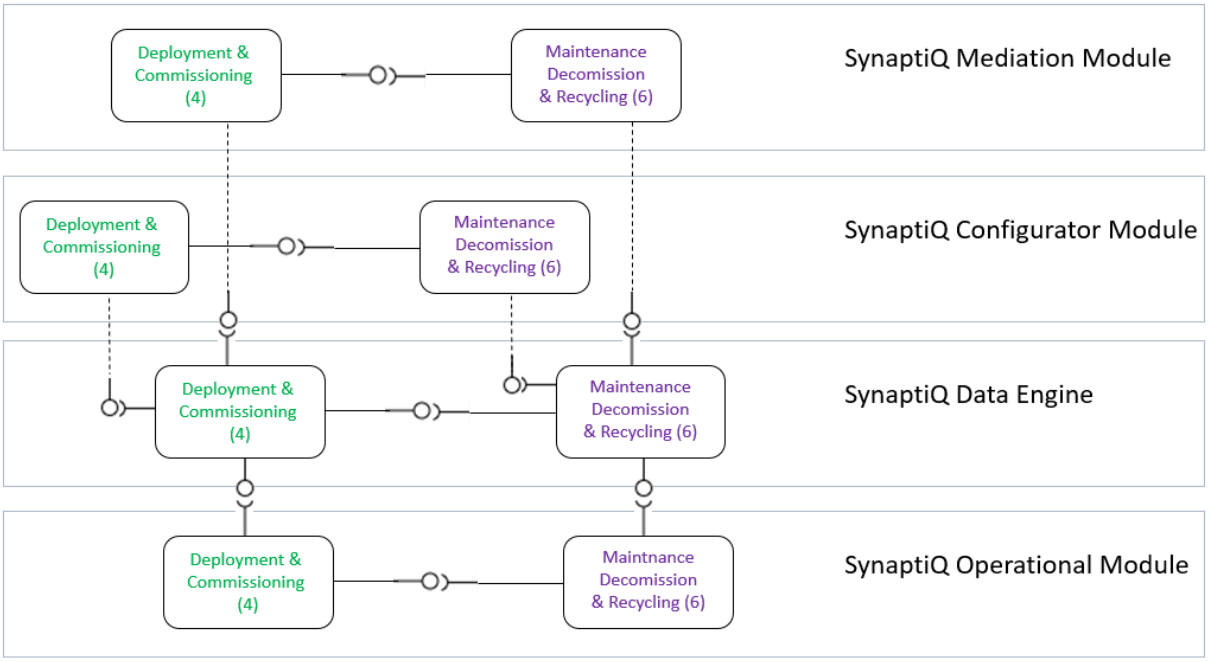


Figure 75 AHT-EP of use case UC-19

Currently, the customers have access to the SynaptiQ Configurator Module to model their PV plants or to adapt the model based on modifications on site. The SynaptiQ Mediation Module has to be configured by our engineer operational team.
With the development of the Mediation Module Configurator (MMC), the configuration of the SynaptiQ Mediation Module will be automatic based on the model configured by the customer. The anomaly detection module (developed by Sirris) will help to detect changes in the model based on the raw data and will allow an automatic reconfiguration of PV plant model.

With the AHT-EP adoption, it will be easier to define the steps that can easily be automated. We have currently a lot a manual steps that prone errors.

There is no scalability per say. A customer can create as many PV plant models of any size as he wants.

Table 103 AHT-EP Phase focus of UC-19

Engineering process phase	Addressed/Focus
Requirements	n.s.
Functional design	n.s.
Procurement & Engineering	n.s.
Deployment & Commissioning	**Activity** *SynaptiQ Configurator*: Customer configure its PV plant Model in this module.

<table>
<tr><th>Engineering process phase</th><th>Addressed/Focus</th></tr>
<tr><td></td><td> Synaptiq Mediation: Based on file format and logger type/brand, the SynaptiQ Mediation Module XML file is updated.
SynaptiQ Data Engine: Automatic deployment based on configuration of SynpatiQ Configurator trigger by our engineer operational team.
SynaptiQ Operational: Fully automatic based on model in SynaptiQ Data Engine.
Tools
SynaptiQ Configurator:
Input: customer or engineering team directly configures the PV plant.
Output: XML snapshot of the model
Synaptiq Mediation:
Input: engineering team directly configures the logger type and brand.
SynaptiQ Data Engine:
Input: XML snapshot of the PV plant model from the SynaptiQ Configurator.
Output: full model and aggregation design in MySQL DB.
SynaptiQ Operational:
Input: access to SynaptiQ Data Engine Database</td></tr>
<tr><td>Operations & Management</td><td>n.s.</td></tr>
<tr><td>Maintenance Decommissioning & Recycling</td><td> Activity
SynaptiQ Configurator: In case of changes on site, the customer applies the same changes in the PV plant model.
Synaptiq Mediation: Based on file format and logger type/brand, the SynaptiQ Mediation Module XML file is updated.
SynaptiQ Data Engine: Automatic re-deployment based on configuration of SynpatiQ Configurator trigger by our engineer operational team.
SynaptiQ Operational: Fully automatic based on model in SynaptiQ Data Engine.
Tools
SynaptiQ Configurator:
Input: customer or engineering team directly updates the PV plant.
Output: XML snapshot of the model
Synaptiq Mediation:
Input: engineering team directly updates the logger type and brand
SynaptiQ Data Engine:
Input: XML snapshot of the PV plant model from the SynaptiQ Configurator.
Output: full model and aggregation design in MySQL DB.
SynaptiQ Operational:
Input: access to SynaptiQ Data Engine Database</td></tr>
<tr><td>Evolution</td><td>n.s.</td></tr>
<tr><td>Training & Education</td><td>n.s.</td></tr>
</table>

19.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 104 UC-19 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	We plan to have less step in the configurations by automating the communication between the modules. We also plan to have an anomaly detection to detect changes in an already configured PV plant or to detect configuration errors in a newly created model.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Currently, there are different configuration modules to use. We plan to reduce it to only the SynaptiQ Configurator Module and have fully automated exchange with the other SynaptiQ modules.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Probably not applicable.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Trainings are already given on the SynaptiQ Configurator Module, the only module that we will have to interact with.

Table 105 UC-19 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	The adoption of AHT shall help to reduce engineering and maintenance cost. This will lead to cost reduction for activation and maintenance of new equipment and this bring market growth.
Obj. 2 - Interoperability for IoT and SoS engineering tools	We foresee to develop a community of hardware and service providers in the section that align to the Arrowhead Tools framework proposed by 3E. The electrical data logger would be able to communicate with our gateway.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools	The UC will use the automated configuration of the Arrowhead project (TRL 4-5).

Project Objective	Focus & Planed actions
to the Arrowhead Framework integration platform	
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	We plan to integrate the automation framework in order to have an auto-configuration of power electronic devices. We aim to put in place a full auto-configuration of the services using meta-data provided through the gateway services but that could also originate from service provided by system design software tools.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Probably not applicable.
Obj. 6 - Training material (HW and SW) for professional engineers	We will push via sector organisations and standardisation organisations the promotion of the technology towards other equipment and service providers.

19.4 Engineering Process analysis

Table 106 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-19

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	N	N	P	C	P	C	N	P	N
OBJ-AHT #1			X	X		X			
OBJ-AHT #2			X	X	X				
OBJ-AHT #3			X	X					
OBJ-AHT #4			X	X	X	X			
OBJ-AHT #5									
OBJ-AHT #6				X				X	
OBJ-WP2 #1				X	X	X			
OBJ-WP2 #2				X		X			
OBJ-WP2 #3									
OBJ-WP2 #4				X				X	

20. UC-20 Elastic Data Acquisition System (FAGORR)

The aim is to develop an Elastic Data Acquisition System that implements different services that will help in Deployment & Commissioning, Operation and Management and Maintenance engineering processes.
In Figure 76, the overview of the baseline of the system is shown. As shown in the figure the system is composed of different elements:

- PLC: Siemens, Beckhoff
- DAS: .NET based custom software
- BBDD: SQL Server / Redis
- Custom: Custom applications developed by engineers or Data Scientists.
- Format: .NET based software prepared to format the data.
- Dispatch: .NET based software prepared to send the data.
- Cloud: Fagor Arrasate´s IoT Platform

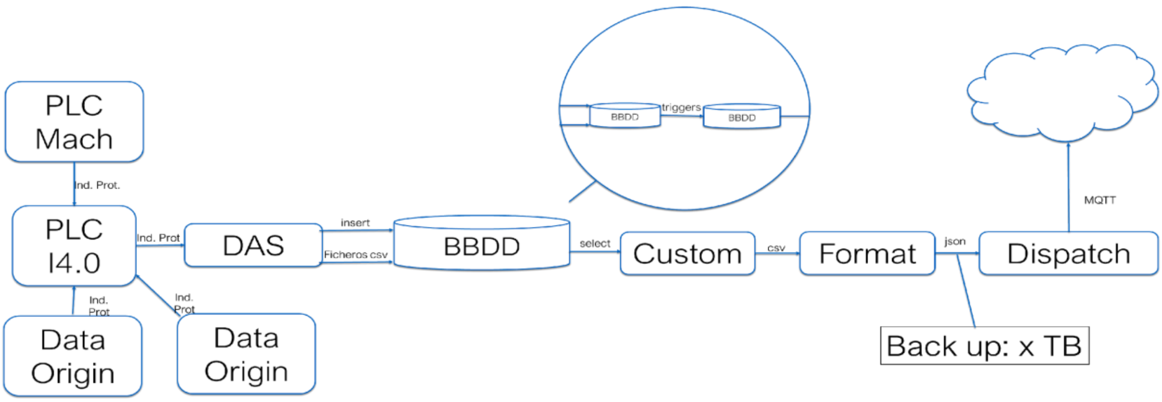


Figure 76 Overview of the baseline of the system

With the DAS is possible to: 1) define the PLC variables that are going to be monitored and 2) choose the protocol each variable is going to be captured. Then automatically the systems starts monitoring those variables and these ones areas introduced in a database (BBDD). The platform supports the deployment of Custom applications if needed such as machine learning applications, custom dashboards and so on. The function of the format is to convert the csv file to a json file with the PPMP specification. Finally the data is sent by the dispatcher with the correct format to the Cloud.

Thus, with the whole system is possible to define the PLC variables that are going to be uploaded to the cloud. To do this, several parameters must be specified, e.g. the sampling frequency for each variable, the industrial protocol to use, the final cloud endpoint to upload the variable, etc.

20.1 Overall description of the UC-EP

Thus, with the whole system is possible to define the PLC variables that are going to be uploaded to the cloud. To do this, several parameters must be specified, e.g. the sampling frequency for each variable and the industrial protocol to use or the final cloud endpoint to upload the variable.

In Figure 77, we show the four different functional blocks identified in the use case:

- Deployment and commissioning phase
  - *DAS configuration*: Defines the parameters and the communication protocol in order to start monitoring them.
  - *Data Dispatcher configuration*: Chooses the communication protocols and the number of threads to be used by the system to increase performance.
- Operation & Management phase
  - *Status monitoring & diagnosis*: Diagnosis the full system in order to know the status of each part and the automatic configuration of the resources used by them if needed.
- Maintenance phase
  - *Application updates*: Automatic update and new deploys of the system applications.

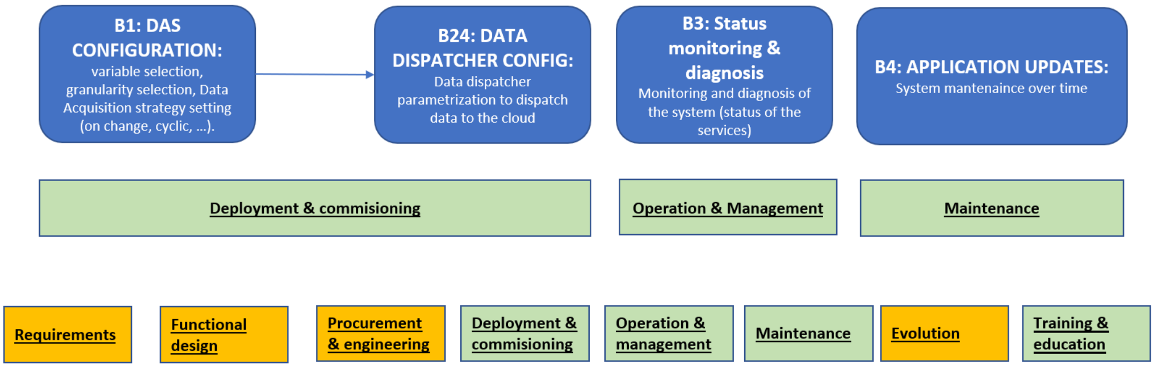


Figure 77 the four different functional blocks of the UC-20

The planned improvement of the toolchain potentially impact on the following engineering process by reducing time in:

- Deployment phase
- Operation phase (monitoring of the remote system)
- Problem resolution. Currently the system software update is error-prone, and consumes a lot of time, thus, the automatization of this will improve the system.

Available technology are enough for the phases addressed in the use case.

The tools developed within this use case will be used by the Machine Manufacturer, thus they are designed for and by them. The machine manufacturer doesn't want to sell these tools, but to use them in order to provide new business models based on data.

Deployment & commissioning adopt the two following standards:

- OPCUA: For the configuration of the plant level data acquisition.
- PPMP: For data interoperability with the cloud.

20.2 Engineering Process Description

Relation between engineering phases:

Connection/relation between engineering phases of Deployment & Commissioning and Operation & Management: When the PLC variables and the communication protocols are defined the system will start monitoring the machine. In that moment it is necessary to make the diagnosis of the status of the system, thus, there is a linear connection between both of them (Deployment & Commissioning and Operation & Management). In the same way, if a new variable is added in the Deployment & Commissioning phase, this one needs to be analyzed in the Operation & Management phase.
Something similar occurs with the Maintenance phase, every issue detected on the Operation & Management makes to change the Deployment & Commissioning phase. Thus, all the engineering phases somehow are connected between them.
In Figure 78 it can be seen how the relationship between the manufacturer and the final user is carried out by the Operation & Management phase, both of the machine and also of the applications and platforms deployed on it.
In the Manufacturer stakeholder, mainly the deployment and commissioning, and maintenance phases are present, but the support on the operation & management phase is also ver relevant.
Between the manufacturer and the final user, the installation technician stakeholder is present, which shares the same phases as the manufacturer stakeholder but the Operation & Management gains importance in comparison to the Manufacturer stakeholder, as this actor can perform as consultor during the Deployment & Commissioning phase and also in the future to the final user.

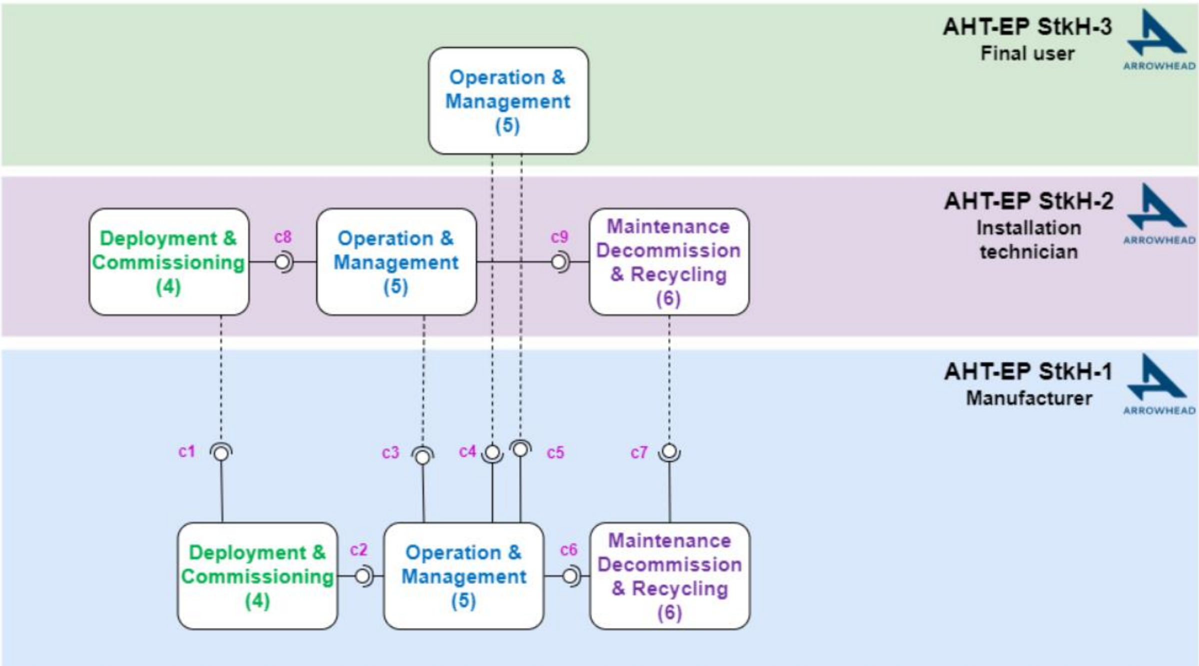


Figure 78 AHT-EP of use case UC-20

The Toolchains are developed using Ikerlan's Framework. The Framework simulates Arrowhead framework. Now we are migrating it to Arrowhead Framework.
At the current state we did not identify any lack of technology for our multi stakeholder use case.
Our Use Case uses a Middleware which communicates with Arrowhead framework. Thus, new tools were developed.

These AHT-EPPs are not applicable in this UC because the scope of the UC is the deployment of a product on the client's premises, and does not cover the following EPs.

- Requirements.
- Functional Design.
- Procurement & Engineering.
- Evolution.
- Training & Education.

The Tools from the toolchain of status monitoring and diagnosis are finished. This Tools are integrated into de Ikerlan's framework and now migrating to Arrowhead framework. The rest of the tools will be developed later.

Table 107 AHT-EP Phase focus of UC-20

Engineering process phase	Addressed/Focus
Requirements	n.s.
Functional design	n.s.
Procurement & Engineering	n.s.
Deployment & Commissioning	**Activity** This is a process carried out by a system engineer during the project set-up. Normally, this phase starts on Fagor Arrasate's premises, but it can't be fully validated since the project is delivered to the customer. The variables, granularities, database configuration and so on is done by a system engineer on Fagor Arrasates premises, prior to sending the machine to the client plant. Once there, the set-up engineer is responsible for the final set-up and project reception, so the final configurations are done directly on the client factory. It is necessary to manually and on-site configure the variables of the PLC that need to be monitor. It is necessary to manually and on-site configure the variables, protocols and endpoints to be sent the data. *DAS Configuration Toolchain*: Fagor and Koniker develop, test, compile and install the Toolchain. Ikerlan develops a Middleware to integrate the tools with Arrowhead Framework. Within this phase, Fagors technicians customises the configuration of the DAS (Data Acquisition Software).
Operations & Management	**Activity** The monitorization of the system is carried out manually. Every day, a system engineer is in charge of monitoring the whole system, detecting anomalies and solving them, if possible. The monitoring is currently being done automatically, shipping logs from the different elements of the value-chain to the cloud. These logs can be accessed through different dashboards in our application. Whenever something fails, the engineers needs to do a manual and one-to-one diagnosis to detect the problem. *Monitoring and Operation Toolchain*: Fagor and Koniker develop, test, compile and install the Toolchain. Ikerlan tests the developed Middleware. Within this phase, an

Engineering process phase	Addressed/Focus
	agent is deployed in the IPC in conjunction with the DAS system on the final client premises and Fagors technicians are able to monitor the behaviour of the system.
Maintenance Decommissioning & Recycling	**Activity** It is necessary to maintain the application deployed in the industrial domain, but currently the software updates are manual and face-to-face. Nowadays, the system engineer manually deploys the applications and does the needed configuration for them to work. He is also responsible for updating the deployed versions. It is necessary to manually and on-site deploy and update the applications. *Deployment Toolchain*: Ikerlan develops the toolchain for the remote-updates. Within this phase, Fagors technicians are responsible for deploying new versions of the DAS system on the customer premises.
Evolution	n.s.
Training & Education	n.s.

20.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 108 UC-20 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	The objective will be reached as the Monitoring Operation tool will provide information from the runtime environment, usually final client stakeholder, to design time environment on the technical and manufacturer stakeholders.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	We are developing tools which can be integrated in different systems. The tools are multi stakeholder and the interaction between them is quite manual nowadays. Thanks to the AHT framework, we will work on the automatization of these toolchains.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	The use of digital tools and frameworks as the ones used in the AHT project make easier the handling of the connections between the EP and Tool Chains.
Obj. 4 - Address digital learning and training activities as an integral part of	n.s.

WP2 Objective	Focus & Planed actions
the engineering cycle	

Table 109 UC-20 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	By the Deployment & Commissioning, Operation & Management and Maintenance EP based tools, these cost reductions will be reached. This will be done by the Arrowhead Tools project execution. More precisely, with the tool of remote set-up and automatic deployment. Nowadays, the set-up of the whole platform lasts from 4 to 5 days. Within the tools described in this document, it is expected to reduce this time to 2-3 days. The new toolchain aims to reduce the deployment time need in order to update or introduce a new software in the industrial domain. To do so, a toolchain that will be composed by different tools will be able to analyze the status of the already deployed systems and then the toolchain will be able to verify if any update in needed. If so, the toolchain will atomically deploy the new version.
Obj. 2 - Interoperability for IoT and SoS engineering tools	This will be achieved by the use of standards in data semantics, formats and protocols.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	The tools on this use case are very specific and private.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	Fagor can benefit from Arrowhead framework ecosystem, thus it will be ahead on this kind of technology integration.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	Only authorized people will be able to use the services and the tools.
Obj. 6 - Training material (HW and SW) for	Improvement of the current Data Acquisition simulation tools.

Project Objective	Focus & Planed actions
professional engineers	

20.4 Engineering Process analysis

Table 110 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-20

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	N	N	N	C	C	C	N	P	N
OBJ-AHT #1				X	X	X			
OBJ-AHT #2				X	X	X			
OBJ-AHT #3				X	X				
OBJ-AHT #4				X	X	X			
OBJ-AHT #5					X	X			
OBJ-AHT #6					X				
OBJ-WP2 #1				X					
OBJ-WP2 #2					X				
OBJ-WP2 #3				X	X	X			
OBJ-WP2 #4									

21. UC-21 Data-based digital twin for electrical machine condition monitoring (ABB)

The use case will develop data-based digital twin for electrical machine condition monitoring. The aim is that in the project, engineering data will be better integrated organically with electrical motor operation, operation/performance management and maintenance phases. At the moment this is done mostly manually/separately based on the adopted practices and related stakeholders. The information type consists of documentation, simulation and testing data matrixes, services (operation management and maintenance) and file transfer. e.g. *.csv, *.mat, *.json files, HTTP.

21.1 Overall description of the UC-EP

The UC will utilize different analysis methods for electrical machine performance evaluation and condition monitoring consisting functional design, engineering, operation management, maintenance and evolution phases.
The functional blocks of the use case, shown in Figure 79 will consists of:

- Electrical machine FEM digital twin
- Updated FEM digital twin
- Machine-learning (ML) tools

- Dynamically optimal parameters for electrical drives
- Smart predictive maintenance tools
- Digital twins for virtual performance testing
- SoA development for electrical motor services offered as automatic cloud services
- Concept creation for condition monitoring and operation parameter optimization

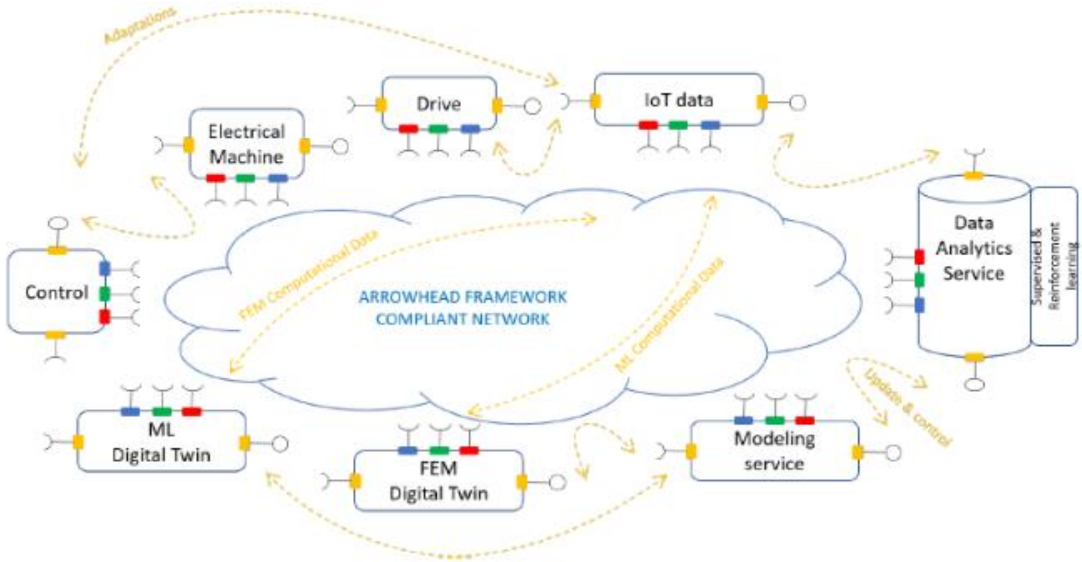


Figure 79 functional blocks of the use case 21

There are several test systems available. The planned main one consist of 2 motors and 2 drivers at ABB. A variety of existing hardware and software tools are utilized. Software tools in use e.g. FEM, ELMER pre- and post-processing tools, Python, Matlab, etc. for processing and modelling. Arrowhead framework will be used for service managements. SySML tools can be used for the planning of functional, performance, and interface requirements and orchestration. IoT-Ticket for digital services.

The tools completed during the first project year:

- Noise analysis (Noiselator) digital twin: Routines and REST API for audible video files
- System and service definitions and support for Arrowhead Framework instance
- ML surrogate model for a permanent magnet motor utilizing Elmer FEM as solver
- REST-API interface for FEM simulations in the workflow at Rahti container cloud

Development continues in the UC for wider utilization of machine-learning (ML) digital twins of electrical machines and drivers (including both induction and permanent magnet motors) able to learn both from FEM computation results and from the IoT data.

Planned improvements of the toolchain will makes easier to manage decision making in operation and maintenance management phases.
Although there are a lot of low cost hardware and software solutions available, industrially functioning, both robust and reliable low cost, light way digital hardware and software solutions are less, if none available.
The UC AHT_EP order of blocks is the following: Designing, procurement, modelling, simulation & testing for operation management and maintenance.
The AHT-EP is partly scalable within product families, outside that more work needed.
Basic guidelines of the standards of the field, e.g. ISO 13374 for condition monitoring part of the operation and maintenance phases.

21.2 Engineering Process Description

The objective in UC21 is to configure and deploy analytic and light weight machine learning tools (StkH-3 and 4), and further deploy them as digital twins for electrical machines (StkH-1) operation and maintenance monitoring (StkH-4 and 5) through IoT -services (StkH-2). From the engineering procedure point of view this can be presented as below.

In Figure 80, in the blue background are presented the core stakeholders:

Electrical machine and its Test bench (StkH 1) used during the project for the electrical machines. Other stakeholders are built around them. In the procedure Electrical machine and Test bench provide Training and Education (8) date/information to subscribing instances:

- Elmer/High Power computing (StkH 3),
- Machine learning and Condition Based Maintenance (StkH 4 & 5),
- IoT (StkH 2).

In addition, Electrical machine and Test bench set (provide) Requirements (1) respective to Elmer/High Power computing (StkH 3), Machine learning and Condition Based Maintenance (StkH 4 & 5) and IoT (StkH 2).

Furthermore, Electrical machine and Test bench in Operation and Management (5) & Maintenance (6) process phases, are mutually both providing to and subscribing from Condition Based Maintenance and Machine Learning (Stkh 4 & 5).

The evolution phase represents natural improvements of the systems and their characteristics, and training and education are in practice documentation of the actions as well as lessons learned kind of activates targeting further development.

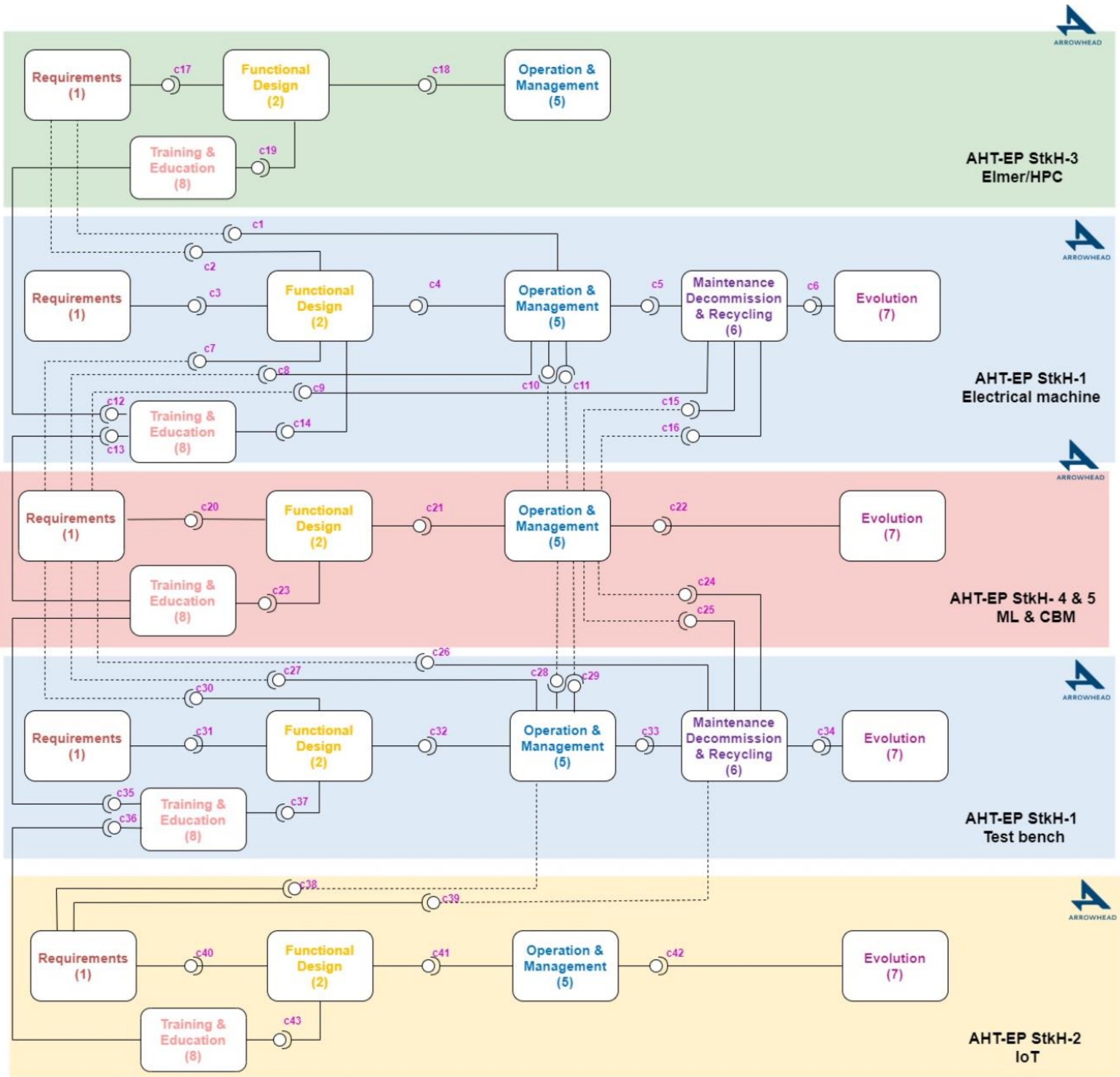


Figure 80 AHT-EP of use case UC-21

Related Arrowhead instance for electrical motor digital twin functional design, operation and maintenance managements is presented in Figure 81.

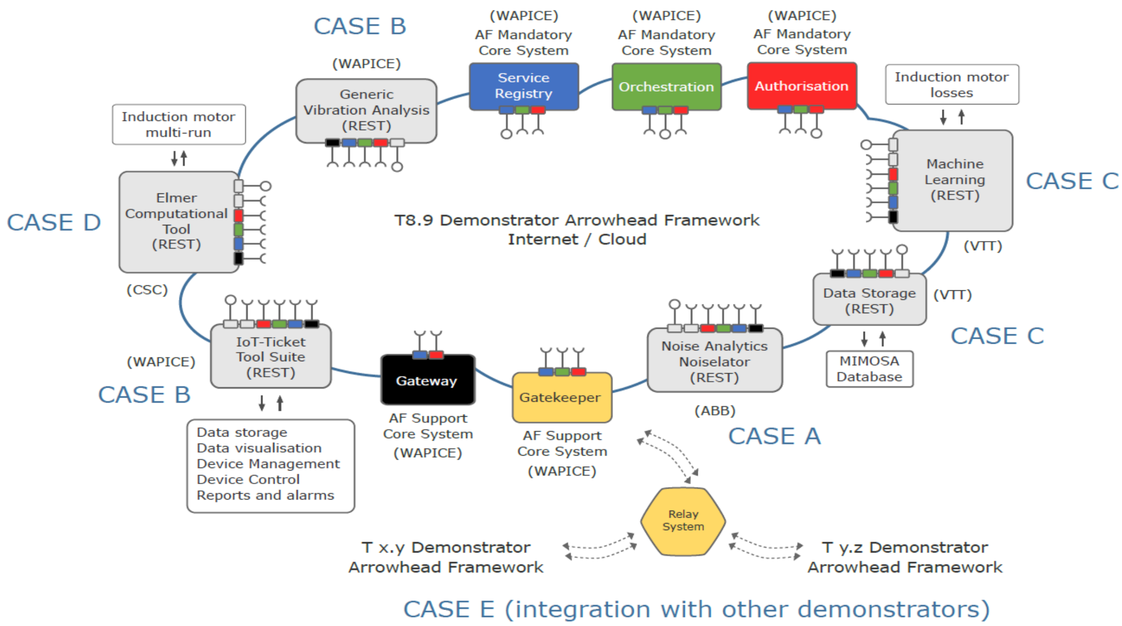


Figure 81 Arrowhead instance for electrical motor digital twin functional design

Typically electrical motors can be monitored for life-cycle purposes based on automation, operation data and measurement responses (e.g. temperature, vibrations). Mechanically, the most critical component in electrical motors are the bearings, and the condition of the bearings can be evaluated based on temperature and vibration measurements, and separate analytics of them. The targeted digital twin based electrical motor management system will be based on Arrowhead framework compliant services when suitable. New aspect is to utilize digital twins for electrical motor operation and maintenance management, and e.g. noise analytics for condition evaluations and product life-cycle management.

Physical electrical motor and driver/inverter, distributed control system, ELMER etc. simulator, machine learning and analytical tools, dynamic operational measurements, digital twin modelling, analytic and model based motor performance evaluation, failure prediction are matched e.g. functional design, operation and maintenance management phases within AHT-EP.

Basic guidelines of the standards of the field, e.g. ISO 13374 for condition monitoring part of the operation and maintenance phases within the EP.

UC is partly scalable within utilized electrical motor product families, outside more work needed. Hybrid modelling approach planned to be used to boost digital twins utilization over wider range of applications.

Different EP elements have been integrated. Previously they were considered separately with a narrow perspective. E.g. the designed models are not updated with the installation data or updated during the lifetime evolution. Wider view over the AHT-EP provides a possibility to automatically keep the parameters, computing, services and digital twins up-to-date throughout electrical machine product total life-cycle. Provides new opportunities to keep the processes in optimal shape and use. In addition, this makes easier to manage decision making in operation and maintenance management phases.

Table 111 AHT-EP Phase focus of UC-21

Engineering process phase	Addressed/Focus
Requirements	**Activity** *Induction motor*: StkH 1 define requirements electromechanical design. *Permanent magnet*: StkH 1 define requirements for electromechanical design. *Test bench*: StkH 1 define requirements for electromechanical design.
Functional design	**Activity** *Induction motor*: StkH 1 does the design. *Permanent magnet*: StkH 1 does the design. *Test bench*: StkH 1 does the design.
Procurement & Engineering	**Activity** *Induction motor*: StkH 1 does the engineering design and 2D and 3D models. *Permanent magnet*: StkH 1 does the engineering design and 2D and 3D models. *Test bench*: StkH 1 does the bench engineering and procurement.
Deployment & Commissioning	**Activity** *Induction motor*: n.s. *Permanent magnet*: n.s. *Test bench*: StkH 1 does the commissioning and share the design for all the other StkHs.
Operations & Management	**Activity** *Induction motor*: StkH 1 does videoing of the machines for bearing/noise analytics. Data (*.MOV and *.MP4 files) delivered to StkH5. *Permanent magnet*: n.s. *Test bench*: n.s.
Maintenance Decommissioning & Recycling	**Activity** *Induction motor*: n.s. *Permanent magnet*: n.s. *Test bench*: n.s.
Evolution	**Activity** *Induction motor*: Models from StkH 1 are processes by StkH 4 for modelling the losses with ML tools. Video data analyzed by StkH 5 for noise based analytics. *Permanent magnet*: Models from StkH 1 are processes by StkH 4 for modelling the losses with ML tools (NN & gradient boosting). StkH4 uses the models together with StkH 3 for torque estimation. In parallel StkH1 and StkH6 develop analytical model for hybrid analytics. *Test bench*: Based on design data StkH 2 design service architecture and dashboard for IoT, and implement it based on IoT-Ticket. StkH 2 deliver WRM24/7+ edge and connection device to StkH1. StkH 1 & 4 start planning for advanced orchestration of required services.

Engineering process phase	Addressed/Focus
Training & Education	**Activity** *Induction motor*: n.s. *Permanent magnet*: n.s. *Test bench*: n.s.

21.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 112 UC-21 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to run time engineering	Identify the requirements from the use case. Define the test instances/cased for electrical motors; Model automatic build-up. Parameter evolution follow-up for the EP phases 5 and 6.
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	Addressed through Arrowhead framework compliant services authorizing and orchestrating required local and inter-cloud connections between the IoT and data analytics devices, electrical machines, drives, and controls accompanied with the modelling analysis services, and the digital twins.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Each of the provided and consumed service will be Arrowhead framework compliant. This makes the system transformable and expandable, and easy to update when needed.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	Use case training activities will focus mainly use case (UC-21) stakeholders. The training materials will be documentations and recommendations on separate matters.

Table 113 UC-21 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Noise analysis digital twin and digital twin model automatic build up via Rest API are expected to yield significant cost reduction compared to the currently applied, original processes in the use case environment. Services through the Arrowhead Framework is planned to simplify the design and development of the final user application, resulting in a reduction of the 50% of design and development time.

Project Objective	Focus & Planed actions
Obj. 2 - Interoperability for IoT and SoS engineering tools	The Arrowhead Framework utilized at local and inter-cloud connections between the IoT and data analytics devices, electrical machines, drives, and controls accompanied with the modelling analysis services, and the digital twins.
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	Each of the provided and consumed service is Arrowhead framework compliant. This makes the system transformable and expandable, and easy to update when needed. Legacy tools (e.g. FEM, IoT-Ticket) will be planned to share information through services exposed on the Arrowhead Framework.
Obj. 4 - Integration platform interoperability with emerging digitalization and automation framework	One part of the IoT platform is based on IoT-Ticket enabling digital services and improvement of operational performance. The platform will be integrated with the Arrowhead Framework.
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	The use case is based on the services published on the Arrowhead Framework and used by the different IoT components. The security of these is managed through AF, and certified connections over the local cloud.
Obj. 6 - Training material (HW and SW) for professional engineers	The documentation of the code, of its functionalities, of the related services are handled following the traditional process: new manuals will be provided for the administrator of the IoT integration framework, for the system operator, for the maintenance operator, and other relevant stakeholders.

21.4 Engineering Process analysis

Table 114 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-21

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	P	P	C	C	C	C	N
OBJ-AHT #1	X	X	X	X				X	
OBJ-AHT #2		X	X	X	X				
OBJ-AHT #3			X		X		X		
OBJ-AHT #4			X		X				
OBJ-AHT #5			X		X				

OBJ-AHT #6		X	X	X	X	X		X	
OBJ-WP2 #1		X	X	X	X	X			
OBJ-WP2 #2	X	X	X	X	X	X	X	X	
OBJ-WP2 #3	X	X	X		X	X			
OBJ-WP2 #4		X			X	X		X	

22. UC-22 Arrowhead Framework training tool (STM)

The use case will develop a generic learning tool for STM32 boards for younger students and beginners in microcontroller programming.
Scratch is a visual programming language and online community targeted primarily at children. Using Scratch, users create their own interactive stories, games and animations, then share and discuss their creations with one another.
Scratch Extensions make it possible to connect Scratch projects with external hardware (such as LEGO WeDo or PicoBoard) or sources of information on the web. They add a collection of building blocks that can be used to interact with a particular device or data source. The Scratch blocks library already features a number of existing blocks for creating algorithms with data and control structures and event. It also includes components to plug-in actuators (motion) and data acquisition (\data").
The target is the ARROWHEAD STUDIO4EDUCATION tool.
Verification Trials will be conducted with professionals from different partners within Arrowhead Tools. Trial will also be made with students at several partner Universities e.g. LTU and BME. Trials will further be conducted in French schools to encourage students to develop IoT Arrowhead-based projects, all using the ARROWHEAD STUDIO4EDUCATION tool.
In the use case will be used the STM32 Discovery kits, a cheap and complete solution for the evaluation of the outstanding capabilities of STM32 MCUs. They carry the necessary infrastructure for demonstration of specific device characteristics, a HAL library and comprehensive software examples allow to fully benefit from the devices features and added values. Extension connectors give access to most of the device's I/Os and make the connection of add-on hardware possible. With the integrated debugger/programmer the discovery kits are ideal for prototyping.

22.1 Overall description of the UC-EP

The STudio4Education use case consists of three tools:

- Papyrus: requirements collection and functional modelling using the SysML Arrowhead Profile
- STudio4Education: Blockly framework based for development of the application (the visual language)
- CLI STM32Duino compiler and the STM32 hardware including sensors / actuators for executing the application (thanks to STM32Duino project).

The three tools are communicating via the Arrowhead framework as shown in Figure 82.

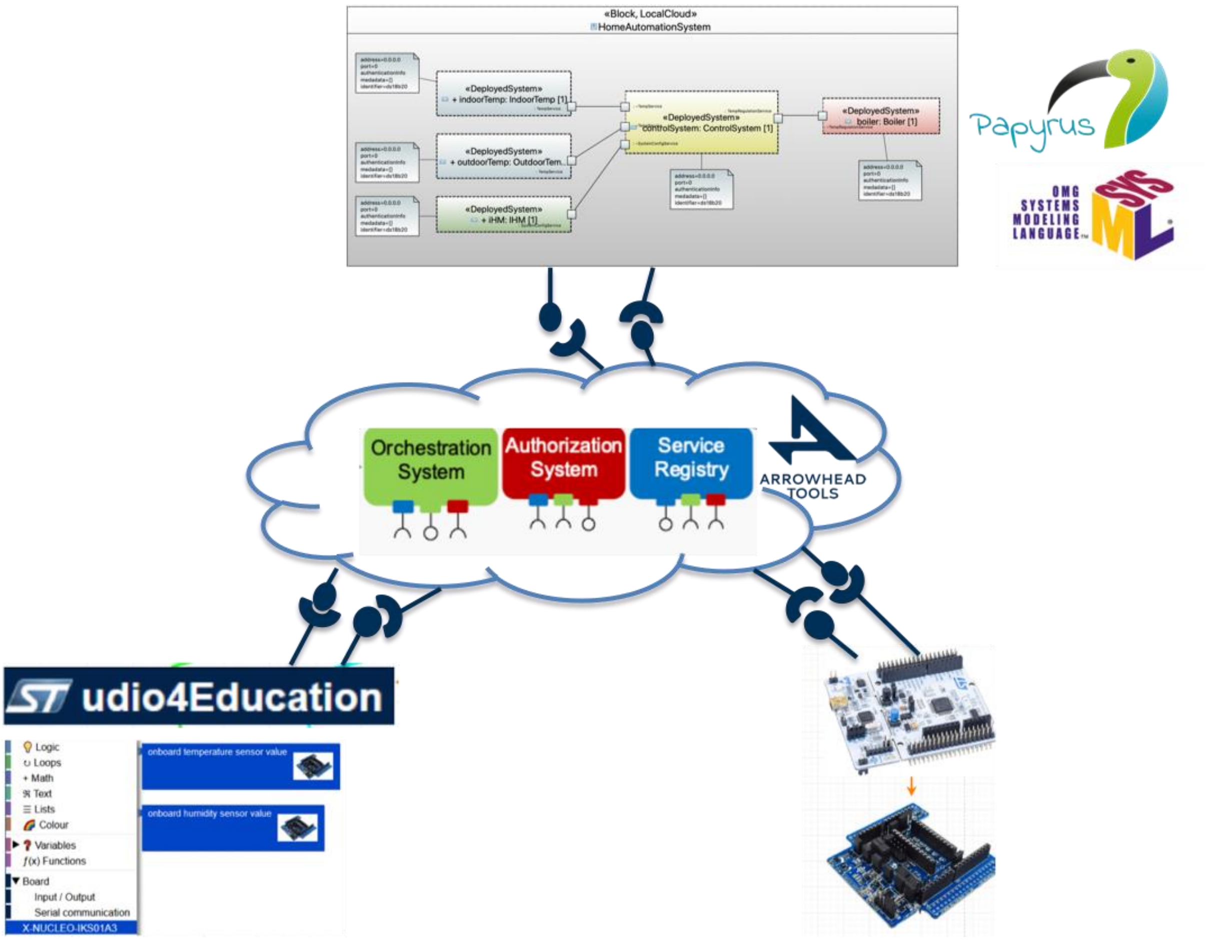


Figure 82 tools composing STudio4Education

In Figure 83 the mapping of the tools on the AHT-EPPs. The teacher models the IoT use case thanks to Papyrus.

Students consume the use case model (as a provided service by the teacher) via the Arrowhead framework: Studio4Education, thanks to NodeJS, consumes the list of actuators & sensors from Papyrus model. It writes back a configuration file and launches STudio4Education, so students on local computer only gets in toolbox menu the needed categories for behavior programming. Local STudio4Education uploads C++ code directly to the local connected board or via Arrowhead.

Students local computer provides services (sensors, actuators data) via the Arrowhead Framework, for someone or specifically for teacher tracking. System can be used by students at home, COVID homework for example.

Sensors data are sent via serial USB communication, or via Arrowhead to STudio4Education. Data can be logged, graphed, visualized. Sensors and actuators data are provided to distant consumers via Arrowhead. This way data can be verified by teachers, logged to control an autonomous project, or allow distant use of STudio4Education.

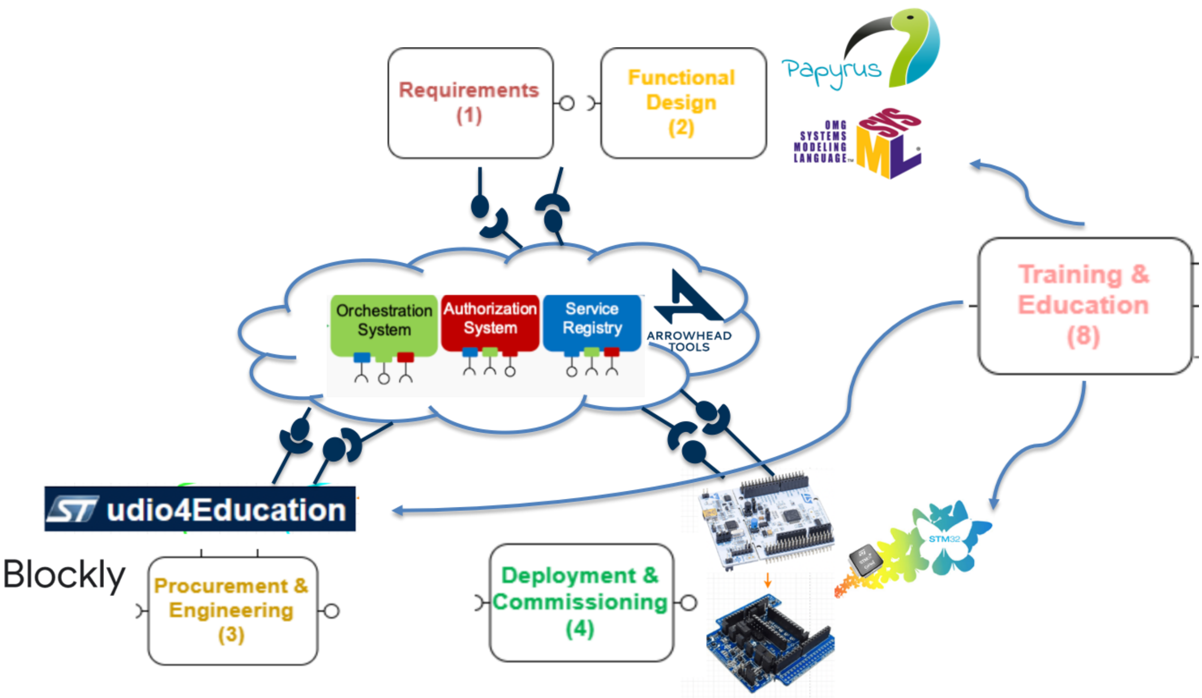


Figure 83 tools composing STudio4Education mapped on the AHT-EPPs

The planned testbed consist of an STM32 MCU combined with humidity and temperature sensors. The Software tools used are Studio4education where the development is ongoing as well as the CLIduino compiler. Arrowhead framework will be used for service managements. SySML tools from CEA can be used for generating specific menus accordingly to the underline hardware.

22.2 Engineering Process Description

In studio4education use case, there are three stakeholders:

- The professor that prepares the training material and provide it to the students (c6).
- The student stakeholder that receives the training material from the professor, develop a project and sends back the results and feedback to the professor (c9, c8).
- The industrial stakeholder that sends new requirements on the educational platforms to the professors (c13).

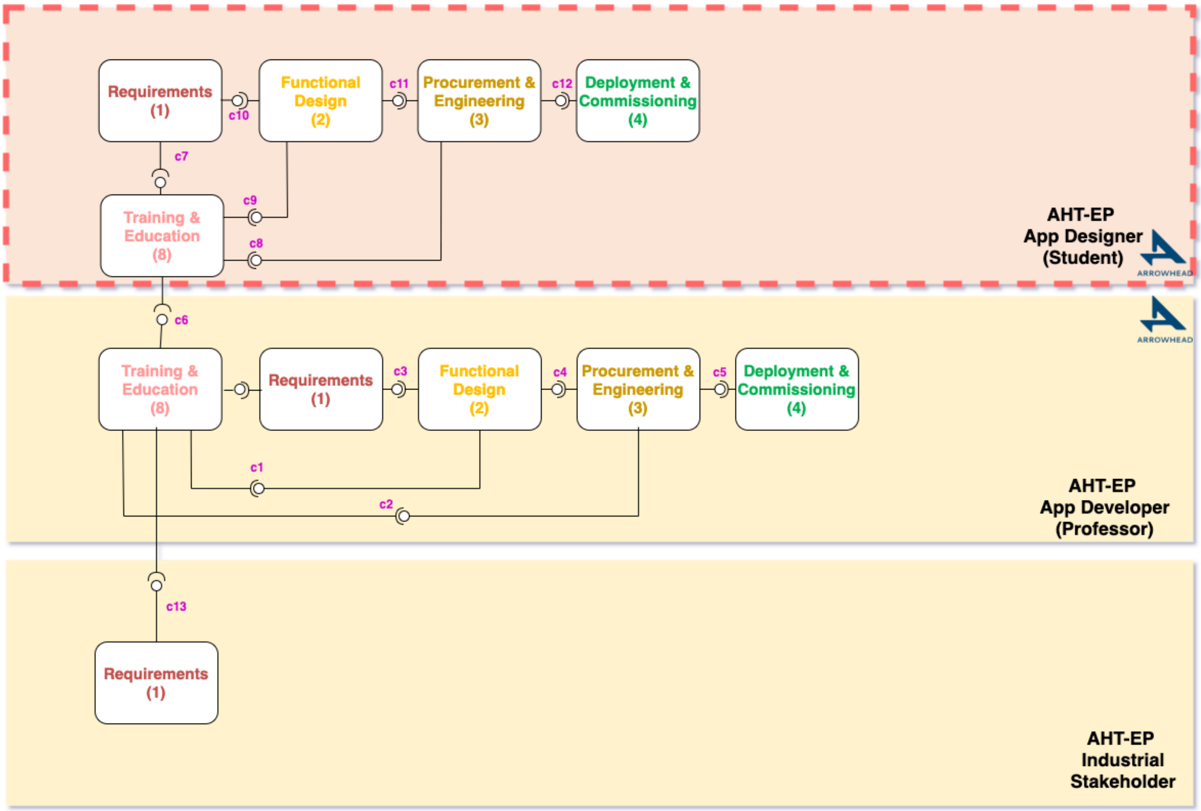


Figure 84 AHT-EP of use case UC-22

Before the adoption of the AHT-EP, the tools involved in the use case were communicating manually. The output of each tool needed a manual adaption to be integrated in another tool. After the adoption of the AHT-EP, the tools are now seamlessly communicating using the Arrowhead framework.

The STudio4Education use case is covering the phases 1, 2, 3, 4 and 8.
Papyrus is used for Requirements (1) and Functional Design (2).
STudio4Education is used Procurement & Engineering (3) and Deployment & Commissioning (4).
The whole use case is also covering the Training & Education phase 8.

Papyrus provides a list of actuators and sensors (json format) to STudio4Education. STudio4Education exports a C code to the STM32 MCU and STM32Duino boards. Actuators and sensors of the STM32 boards send real time data to STudio4Education which can send it back as a service through AHF.

The tools completed during the first project year is STudio4Education. Papyrus and STudio4Education are communicating via code generation. Next development steps will cover the fully integration with the Eclipse Arrowhead framework: all the tools of the use case will communicate via the Eclipse Arrowhead framework

Table 115 AHT-EP Phase focus of UC-22

Engineering process phase	Addressed/Focus
Requirements	**Activity** Requirements are modelled in Papyrus using SysML requirements diagrams. Requirements models are connected to functional design models using traceability links.
Functional design	**Activity** The functional Design of the application is modelled in Papyrus using the Arrowhead SysML profile. Functional design models are connected to the engineering tool through automatic code generation.
Procurement & Engineering	**Activity** The development of the application is done graphically in Studio4Education/Blockly tool. Studio4Education is configured by the json files exported from Papyrus. Studio4Education automatically generates C code including STM32 libraries
Deployment & Commissioning	**Activity** The deployment of the code to the embedded STM32 cards is done using STM32 MCU and STM32Duino development environments. C Code provided by Studio4Education is uploaded to the STM32 cards. During the application execution, actuators and sensors connected to STM32 boards send real time data to Studio4Education.
Operations & Management	Probably not applicable.
Maintenance Decommissioning & Recycling	Probably not applicable.
Evolution	Probably not applicable.
Training & Education	**Activity** Training materials are mainly documentations.

22.3 How the AHT-EP allows to match the Project and WP2 objectives

Table 116 UC-22 WP2 objectives

WP2 Objective	Focus & Planed actions
Obj. 1 - The change from design time to	The objective will be matched since the change from design time to run-time will be automatic. We automatically generate executable code from design-time models.

WP2 Objective	Focus & Planed actions
run time engineering	
Obj. 2 - The move from single to integrated multi stakeholder automation and digitalization	In this use case, we introduced three stakeholders that will interact with each other using the Eclipse Arrowhead framework. The objective will be then matched.
Obj. 3 - Handling of substantially increased number of I/O's due to much more fine grained automation	Each of the provided and consumed services will be Arrowhead framework compliant.
Obj. 4 - Address digital learning and training activities as an integral part of the engineering cycle	The main purpose of STudio4Education is to address the learning and training activity especially nowadays with COVID19.

Table 117 UC-22 Project objectives

Project Objective	Focus & Planed actions
Obj. 1 - Reduction of solution engineering costs by 20-50%	Automating the communication between the different tools (Papyrus, Blockly, STM32 IDE).
Obj. 2 - Interoperability for IoT and SoS engineering tools	Automating the communication between Papyrus (Functional Design tool), Blockly (engineering tool) and STM32 IDE (deployment tool).
Obj. 3 - Interoperability and integration of data from legacy automation engineering tools to the Arrowhead Framework integration platform	n.s.
Obj. 4 - Integration platform interoperability with emerging digitalization and	n.s.

Project Objective	Focus & Planed actions
automation framework	
Obj. 5 - Flexible, interoperable and manageable security for digitalisation and automation solutions	n.s.
Obj. 6 - Training material (HW and SW) for professional engineers	The studio4education use case is providing training material for high school students. These training material could be adapted to professional engineers.

22.4 Engineering Process analysis

Table 118 Matching of the Project and WP2 objectives in each AHT-EP phase of UC-22

AHT-EPP	1 (R)	2 (FD)	3 (PE)	4 (DC)	5 (OM)	6 (M)	7 (E)	8 (TE)	Unknown
UC-EP match	C	C	C	C	N	N	N	C	N
OBJ-AHT #1		X	X	X				X	
OBJ-AHT #2		X	X	X					
OBJ-AHT #3									
OBJ-AHT #4									
OBJ-AHT #5									
OBJ-AHT #6	X	X	X	X				X	
OBJ-WP2 #1		X	X						
OBJ-WP2 #2	X	X	X						
OBJ-WP2 #3	X	X	X	X					
OBJ-WP2 #4	X	X	X					X	

Bibliography

[1] G. Urgese, P. Azzoni e F. Montori, «WP12410_survey,» 2020.
[2] P. Azzoni, G. Urgese e F. Montori, «WP124 Use Cases survey,» 2019.

List of abbreviations

Abbreviation	Meaning
AHT	Arrowhead Tools
EAf	Eclipse Arrowhead framework
SoS	System of Systems
SOA	Service Oriented Architecture
DoA	Declaration of Agreement
UC	Use Case
UC-EP	Use Case Engineering Process
AHT-EP	Arrowhead Tools Engineering Process
n.s.	Not Specified
n.a.	Not Available

Revision history

Contributing and reviewing partners

Contributions	Reviews	Participants	Representing partner
X	X	Gianvito Urgese	POLITO
X	X	Jan van Deventer	LTU
X	X	Paolo Azzoni	ETH
x		Lukáš Maršík	CAMEA
x		Tomas Vojnar	BUT
x		Frans Rosbak	PHC
x		Peter van der Meulen	PHC
x		Jose María Alvarez Rodríguez	UMLA
x		Odei Ayastuy Lizarralde	UMLA
x		Oskar Berreteaga	UMLA
x		Ramon Schiffelers	ASML

x		Koen van Wijk	ICTG
x		Anja Zernig	KAI
x		Lars Oscarsson	LIND
x		Carlos Rodriguez de Yurre	FAUT
x		Mikel Carrasco	FARR
x		Sara Bocchio	ST-I
x		Gerry Nigro	REPLY
x		Maurizio Griva	REPLY
x		Davide Brunelli	IUNET
x		Federico Montori	IUNET
x		Edoardo Patti	POLITO
x		Marco Castangia	POLITO
x		Enrico Macii	POLITO
x		Elisa Londero	ETH
x		Antonio Lionetto	ST-I
x		Giusy Tomarchio	ST-I
x		Jose Luis Buron Martinez	ACCIONA
x		Mustafa Kucukkuru	ARCELIK
x		Alper Özel	ARCELIK
x		Çağlar Henden	ARCELIK
x		Marek Tatara	DAC
x		Kjell Bengtsson	NTNU
x		Guoyuan Li	NTNU
x		Markus Frank	BOLIDEN
x		Wang Zhiping	VTC
x		Mikael Bjorn	VTC
x		Germar Schneider	IFD

x		Jürgen Kühnle	BOSCH
x		Valentin Fetscher	BOSCH
x		Gunter Welde	IFD
x		Matthias Fehr	IFD
x		Norbert Waleschkowski	SEMANTIS
x		Thomas Wagner	TUD
x		Daniel Broxtermann	IFD
x		Dagmar Jähnig	AEE
x		Tom Tourwé	3E
x		Jon Rodriguez	FARR
x		Aitor Agirre	FARR
x		Jan Westerlund	ABB
x		Tullio Salmon Cinotti	UNIBO
x		Marcello Coppola	STM
x		Saadia Dhouib	CEA
x		Emmanuel Vaumorin	MAGILLEM

Amendments

No.	Date	Version	Subject of Amendments	Author
1	2021-04-10	0.1	First Draft	Gianvito Urgese
1	2021-05-24	0.4	Second Draft	Gianvito Urgese

Quality assurance

No	Date	Version	Approved by
1	2021-05-26	4.0	Jerker Delsing