



Artificial Intelligence in Manufacturing leading to Sustainability and Industry5.0

Project Acronym:

AIMS5.0

Grant agreement no: 101112089

Deliverable no. and title	**D3.1 – Preliminary micro service requirements and first architecture concepts**	
Work package	WP3	Cyber architecture for AI supported sustainable production
Task	T3.1-3	Requirements and interaction with other work packages; Micro service-based platform enabling AI integration at edge level
Subtasks involved	ST3.2.1-4	
Lead contractor	Infineon Technologies AG Thomas Gutt, mailto:thomas.gutt@infineon.com	
Deliverable responsible	AITIA International Zrt. Attila Frankó - afranko@aitia.ai / franko@tmit.bme.hu	
Version number	v1.2	
Date	31/05/2024	
Status	Final	
Dissemination level	Public (PU)	

Copyright: AIMS5.0 Project Consortium, 2024

Authors

Participant no.	Participant short name	Author name	Chapter(s)
20	TIAG	Anna Ryabokon, Francesca Flamigni, Tamara Terbul-Jarmer, Markus Tauber	3
20	TCAG	Silviu Graciunas	3
14	AIT	Oliver Zendel, Christoph Schmittner	3
18	PYT	Ralph Baldrian	3
19	RSA	Belal Abu Naim, Yasin Ghafourian	3
46	GUT	Tristan Wójciak, Mateusz Groth	3
33, 34	AITIA, BME	Attila Frankó	1,2,3
33	AITIA	Tamás Bordi	3
47,48,49,50	UC3	Olov Schelén et al.	3
15	CISC	Ralph Weissnegger	3
	FMX	Karl-Benedikt Reith	3
40, 41, 44	PCL, RUG, TU/e	Ahmet Gozutok, Alex Latipov, Ilayda Celenk, Thomas Vergutz, Michel Medema	3
45	DAC	Marek S. Tatara	3
48	LTU	Jerker Delsing	1,2,3,4
3	BMW	Daniel Jess	3

Document History

Version	Date	Author name	Reason
v0.1	04.09.2023	Attila Frankó	Initial Version
v1.0	08.04.2024	Attila Frankó	First version with contributions
V1.1	25.04.2024	Attila Frankó	Revised version based on internal feedback
v1.2	31.05.2024	Jerker Delsing, Attila Frankó	Restructuring, revisioning and finalizing the document

Publishable Executive Summary

D3.1 “Preliminary micro service requirements and first cyber architecture concepts” of the AIMS5.0 EU project presents the first-year findings and achievements of WP3.

A set of 45 requirements for cyber architecture has been gathered from both a basic technology viewpoint and an application use case viewpoint. These requirements will guide the further development of the cyber architecture, which will build on our baseline of the Eclipse Arrowhead architecture and reference implementation v4.6.1.

AIMS5.0's ambition is to enable efficient engineering integration of AI functionality into production automation solutions. For this purpose, further investigations and development will be concentrated on the following topics:

- Fundamental SOA property support: Look-up, Late binding, and Loose coupling
- Security support
- Interoperability support
- System of Sytems Integration support
- Basic engineering and operations support
- Model-based engineering support

These areas have been addressed by scientifically investigation and as a Proof-of-Concept carefully defining the next major release of Eclipse Arrowhead targeting:

- Operational robustness and security
- Real-time support
- Relaxation of hard requirements simplifying the integration of the cyber architecture into legacy production automation environments
- Support for model-based engineering and operational deployment and management.

This document highlights how the cyber architecture is designed, fundamentally validated and intended to be used in application use cases, and their validation and verification of feasibility and fulfillment of requirements are provided. Model based AIGym concept will be also introduced shortly as a infrastructure to build AI-enabled SOA/Micro-service architectures for industry.

Finally, contributions to standards like Power of Attorney (PoA) through IETF are briefly discussed.

Part of the work technical achievements performed in this work package and in the reporting period was published as international, peer-reviewed conference papers: [BAN1, SW1, GD1, SVS1, HG1, PV1, HCS1][1].

[1] [BAN1] Belal Abu Naim et al., „A Generic Framework for Resource-Limited Microcontrollers Deployment in I-IoT Systems“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.
[SW1] "Real-time Container Orchestration based on Time Utility Functions", S. Walser, J. Ruh, S. Craciunas, 20th IEEE Int. Conference on Factory Communication Systems, IEEE, 2024
[GD1] G. Dimirakopoulos et al., "On the Way to Realize the 5th Industrial Revolution: Achievements, Challenges and Research Areas," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-6, doi: 10.1109/NOMS56928.2023.10154446.
[SVS1] S. Vattaparambil Sudarsan, O. Schelén and U. Bodin, "Multilevel Subgranting by Power of Attorney and OAuth Authorization Server in Cyber–Physical Systems," in IEEE Internet of Things Journal, vol. 10, no. 17, pp. 15266-15282, 1 Sept.1, 2023, doi: 10.1109/JIOT.2023.3265407.
[HG1] Gergely L. Hollósi et al., „AIMS5.0 AI Toolbox: Enabling Efficient Knowledge Sharing for Industrial AI“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.
[PV1] Pal Varga et al., „MLOps in CPS - a use-case for image recognition in changing industrial settings“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.

Table of contents

1. Introduction 6
2. Microservice architecture requirements 7
3. Initial cyber architecture concepts 8
3.1 Robustness, interoperablility and composability to legacy OT and IT automation installations .10
3.2 WP3 Questionnaire about SOA and micro service competence 11
3.3 Real-time virtualization and orchestration platforms for reliable industrial production systems 15
3.4 Robust real-time networking for AI applications 20
3.5 Model based AI Gym concept 22
4. Use case integration of the common cyber architecture 23
4.1 AI-supported Industrial IoT utilizing the common cyber architecture 23
4.1.1 A prototype for evaluating investigated aspects of the common architecture 26
4.1.2 Architecture for integrating Micro services at the edge level 26
4.2 AI-Enhanced Value Chain 28
4.2.1 System Modelling 29
4.2.2 Modelling Production Processes as Business Processes 30
4.2.3 ML Modelling between process parameters and part quality prediction 32
4.2.4 Production Planning / LLM Supported Production Planning 33
4.2.4.1 Production Planning using LLMs 33
4.2.4.2 Automated Planning Framework 34
4.3 Servitization and service delivery enabled by Zero administration 35
4.4 AI techniques at different layers in Machine Tool Domain 38
4.5 Automated Architecture Compliance Assessment 40
4.6 Architecture for MLOps solutions 42
4.7 Deployment of new AI approaches/services on the edge 43
4.8 Micro service based platform enabling AI integration at edge level 45
4.8.1 Detailed description of the work plan 45
4.8.2 Research outline 45
4.8.3 Objectives 46
4.8.4 Initial results 46
4.9 Collecting requirements and relation to the common architecture 46
4.9.1 Software and hardware realted requirements 46
4.9.2 Requirements for Digital Twin systems 48
4.10 ML algorithms for workflow optimization 49
4.11 Gate Boarding of Sensors 49
4.12 Improved security and reliability measures in communication systems 50
4.13 Autonomous, reactive and teachable surveillance network (system of systems) 51
4.13.1 Objectives 51
4.13.2 Architecture 51
4.13.3 Microservice Environment 53
5. Conclusions 55

[HCS1] Cs. Hegedus et al., „Tailoring MLOps Techniques for Industry 5.0 Needs“, CNSM 2023-2023 IEEE/AnServApp Network Operations. In press.

6. Publications 56
7. References 57

List of figures

Figure 1: The microservice cyber architecture will provide an SOA architecture infrastructure supporting fundamental service-oriented properties like Look-up, Late binding, and Loose coupling, plus a number of support functionalities to build working solutions. 6
Figure 2: Eclipse Arrowhead 4.6.1 core and support microsystems 7
Figure 3: Questionnaire results I. 8
Figure 4: Questionnaire results II. 10
Figure 5: Questionnaire results III. 11
Figure 6: Architecture Diagram Online Phase 15
Figure 7: TTTech Industrial Nerve Architecture 20
Figure 8: TTTech Industrial Management System 22
Figure 9: UC 11 AI-supported Industrial IoT for Indoor Food Production: System Architecture (TIAG, ZelosPlant, AIT, RSA, GUT) 23
Figure 10: The Hydroponic Prototype 23
Figure 11: Architecture for integrating Micro services at the edge level 26
Figure 12: Use Case 2 System Architecture 26
Figure 13: System Modelling Steps 28
Figure 14: Integrating Camunda into the Eclipse Arrowhead framework allows Camunda to use the Orchestrator core service to look up service endpoints at runtime. 29
Figure 15: Abstract representation of a production process, where each step is defined by a transfer function that maps the input and process parameters to an output and a part quality prediction. 30
Figure 16: Capabilities outline 32
Figure 17: Interaction between servitization parties 33
Figure 18: Conceptual architecture (technology agnostic) 33
Figure 19: Preliminary Arrowhead-based architecture diagram 34
Figure 20: General architecture 35
Figure 21: ThreatGet Architecture 38
Figure 22: ThreatGet AI-enhanced process 40
Figure 23: MLOps platform components 42
Figure 24: Research topics 43
Figure 25: Basic architecture. 45
Figure 26: Conceptual architecture with relation to other system 45
Figure 27: Architecture of Sentinel system 45
Figure 28: Publish-subscribe approach within Sentinel 46

1. Introduction

This WP3 deliverable addresses the cyber architecture requirements as collected from use cases and other technology work packages. The first architectural concepts that address these requirements have been defined and fundamentally validated in use-cases. The architectural concepts are intended to build upon previous work in the stepping-stone projects like Arrowhead Tools and Productive5.0, which both have made significant contributions to the open-source project Eclipse Arrowhead. The intention of AIMS5.0 and WP3 is to continue such contributions to the Eclipse Arrowhead, further building its usefulness in real industrial environments.

Thus, this report holds the following major sections:

- Microservice architecture requirements
- Initial cyber architecture concepts
- Use-case integration of the common architercture
- Conclusions of this document

2. Microservice architecture requirements

In Task 3.1, requirements for the microservice-based cyber architecture and its reference implementation have been collected from two primary types of sources

- Technology work packages 1,2 and 3
- Use-case work packages 5 and 6 and their use cases

These two sources give two different types of requirements. The technology work packages provide, in general, more technology fundamental requirements, while the use case provides more application-oriented requirements. These differences need to be considered when evaluating the fulfillment of the requirements.

All requirements collected so far are listed in Appendix – Requirements.
For each requirement, the following information is collected.

- Submitter name
- Submitter email
- Providing task
- Category
- Integration platform, Engineering tools, Engineering tools chain integration, Training material, Governance, Verification use case
- Objective
- Requirement description
- Rationale
- Y1 fulfillment
- What's missing
- Evaluation (text)

The requirement Appendix is a living internal document to which further requirements will be added, and the fulfillment evaluation will be added for Y2 and Y3. Thus, a clear picture of the advancement of cyber architecture and its reference implementation will become visible.

Currently, there are 45 requirements collected. For a smaller number of these requirements, an analysis of the Y1 fulfillment has been possible and is thus as well commented in the Appendix. One division of the appendix is public and released as a part of this document (D3.1), while the other division is confidential and will be included in D3.4.

3. Initial cyber architecture concepts

The cyber architecture is fundamentally based on a microservice/SOA paradigm. This paradigm has been used in previous projects like Productive5.0 and Arrowhead Tools. All previous results have been transferred to the open-source project Eclipse Arrowhead, http://www.arrowhead.eu and http://github.com/eclipse-arrwohead.

The work on the cyber architecture within AIMS5.0 is based on the current status of Eclipse Arrowhead and is further guided by the collected requirements and the project DoA.

The cyber architecture and its implementation in real-world use cases can be described with the help of .

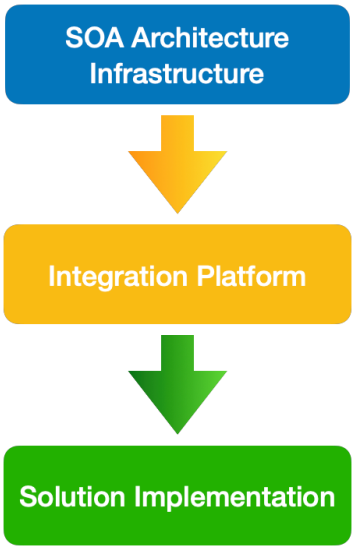


Figure 1: The microservice cyber architecture will provide an SOA architecture infrastructure supporting fundamental service-oriented properties like Look-up, Late binding, and Loose coupling, plus a number of support functionalities to build working solutions.

The SOA architecture infrastructure will provide:

- Fundamental SOA property support: Look-up, Late binding and Loose coupling
- Security support
- Interoperability support
- System of Sytems Integration support
- Basic engineering and operations support
- Model-based engineering support

To evaluate the investigated topics, the project architecture base line was chosen as the Eclipse v4.6.1. The released candidate and prototype microsystem are depicted in Figure 2 below.

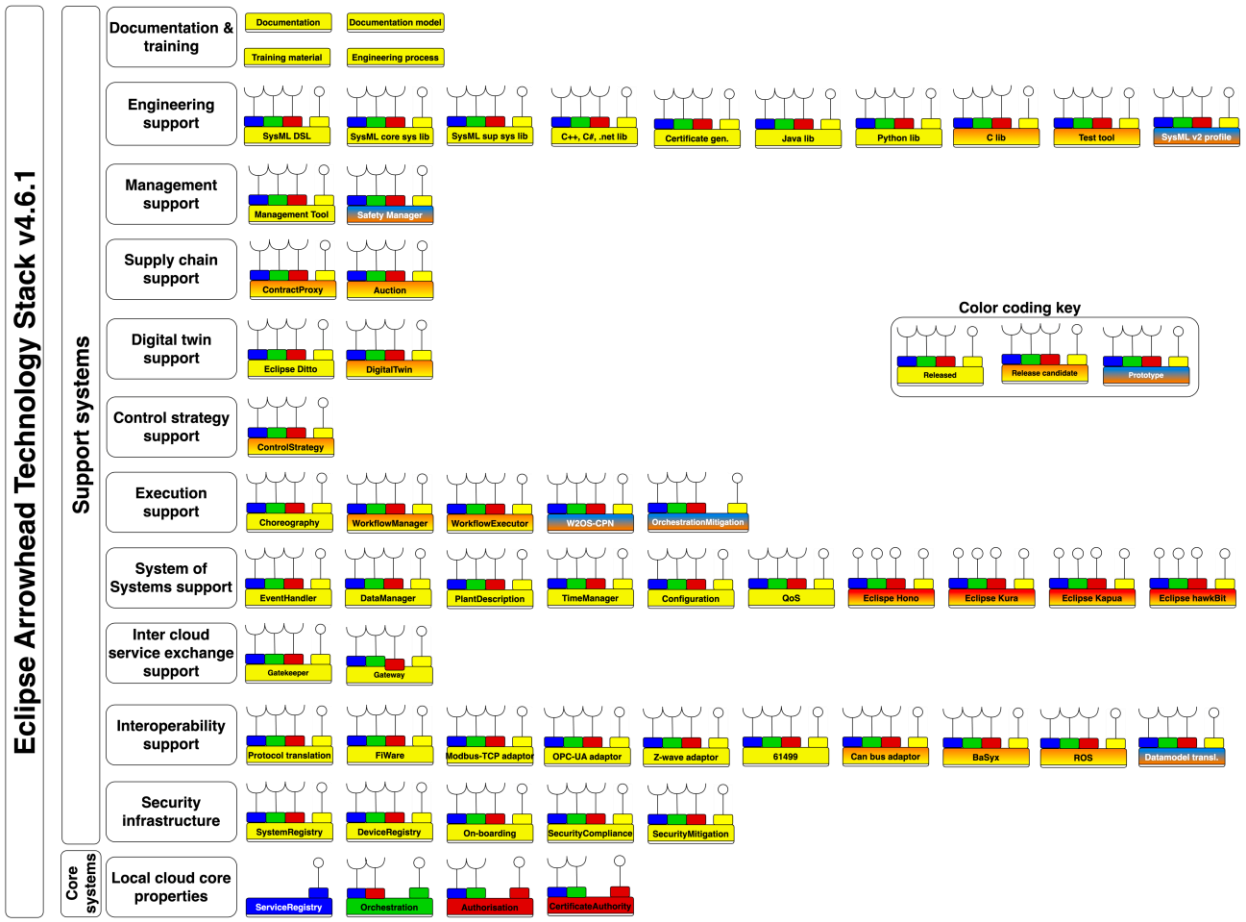


Figure 2: Eclipse Arrowhead 4.6.1 core and support microsystems

The WP3 ambition is to mature and extend this baseline. The overall ambition is to make Eclipse Arrowhead the major open-source microservice/SOA architecture and reference implementation that is to be deployed on heterogeneous devices from deep edge to fog with interoperability to cloud servers.

WP3 AIMS5.0 targets the following additional improvements and extensions to the Eclipse Arrowhead architecture and its reference implementations:

- Robustness, interoperability, and composability to legacy OT and IT automation installations.
- System of systems composability for AI functionality
- Software deployment and management for AI functionality
- Safety and security in AI-supported production automation
- Robust real-time networking for AI applications
- The AI-gym, AI functionality as services for rapid integration, testing, and V&V

Compared to our baseline, the following advancements have been addressed and achieved within year 1 of AIMS5.0.

3.1 Robustness, interoperablility and composability to legacy OT and IT automation installations

This is addressed through the update to the Eclipse Arrowhead to v5.0. This update is by AIMS5.0 in cooperation with Arrowhead fPVN and some other EU projects, e.g., Minie.IO and AI-REDGIO. A Roadmap WG group, which gathers some 20-40 people at each telco meeting, has developed the Generic System of Systems Description, GSoSD, document. Which was presented at the Lulea workshop in Jan 2024. The document has been slightly updated and has had some minor editing since then. The GSoSD can be found at
https://github.com/eclipse-arrowhead/roadmap/tree/main/5.0 Draft/
along with a set of other specification documents. Those who want to have a look into the discussion resulting in the v5.0 GSoS can follow that via issues tagged with 5.0, https://github.com/eclipse-arrowhead/roadmap/issues.

During this period, AITIA has worked on three related tasks – as a part of investigating the defined research topics and evaluating them with Arrowhead Framework:

- Maintaining the 4th generation of Eclipse Arrowhead framework
- Contributing to the design and architecture of the 5th generation of Eclipse Arrowhead framework
- Developing an industrial system (named Sentinel Surveillance System), where both AI integration and the advanced usage of Arrowhead are presented
- Maintaining the 4th generation of Eclipse Arrowhead framework

Two patch releases (4.6.1, 4.6.2) have been published during this period, containing various bug fixes, security updates, and, most importantly, some new core services (in the Configuration and Choreographer Supporting Systems) based on the experiences from the Sentinel Project.

Major changes from the v4.6.1 are:

- The scope of Eclipse Arrowhead is now phrased as:
- **The Eclipse Arrowhead project's focus is to provide microservices that produce and consume microservices at the edge.**
- **Eclipse Arrowhead provides a microservice SOA architecture and reference implementation to be deployed on heterogeneous devices from deep edge to fog with interoperability to cloud servers.**
- Vocabulary is more clearly defined. For example, Microsystem replaces System; Microservice replaces Service.
- Open for multiple authentication and authorization mechanisms. Thus, the Authorization core system is split into an Authorization and Authentication system.
- Some basic requirements are relaxed to simplify integration with legacy solutions through the relaxation of shall definitions and usage to recommended usage.
    - Supporting integration to existing brownfield technology and company conventions!
    - Special notes on those deviations from recommendation will render less architectural support for desired properties, such as
    - real time, security, safety, …

Currently, the definition of the SysD (microsystem description) and SD (microservice description) documents for the core Arrowhead microsystems is ongoing. The documents and associated discussion can be found at https://github.com/eclipse-

arrowhead/roadmap/tree/main/5.0 Draft, and the associated specification discussion for each document can be followed by issues in Github.

Eclipse Arrowhead roadmap meetings are held fairly regularly, about every 3:ed weeks, with some 20-40 participants. The work is led by Professor Jerker Delsing, with strong contributions from SME companies like e.g., Sinetiq, AITIA, Thingwave.

The goal of the 5th generation is to enable the usage of Arrowhead within a wider range of use cases. In this spirit the framework will not contain mandatory components (only "recommended") and the components will have independent working modes as well, will allow to apply a wider range of security related solutions (separated authentication and authorization features), will have independently deployable service orchestration strategies and will use more sophisticated interface representation with the possibility of automatic translation.

A rough timeline for the first v5.0 release is Q4 2024.

3.2 WP3 Questionnaire about SOA and micro service competence

In alignment with the mission and goals of WP3, a questionnaire has been crafted, consisting of six questions aimed at discerning the general knowledge of partners about SOA, microservices and the related composability questions and their applications via Arrowhead. This is not closely related to collecting requirements as it was performed in a different manner. The goal of the survey is to provide clarity regarding the success of dissemination and adaptation of microservice architecture and Arrowhead Framework in the previous projects.

Responses have been gathered from 13 partners, namely CISC, EDI, Fabmatics GmbH, Gdansk University of Technology, IBERMATICA, IFD, Lfoundry, NXP, RSA, Semaku, Stream Analyze Sweden AB, TIAG, TU Dresden, Universidad Politécnica de Madrid. Below is a summary of their answers.

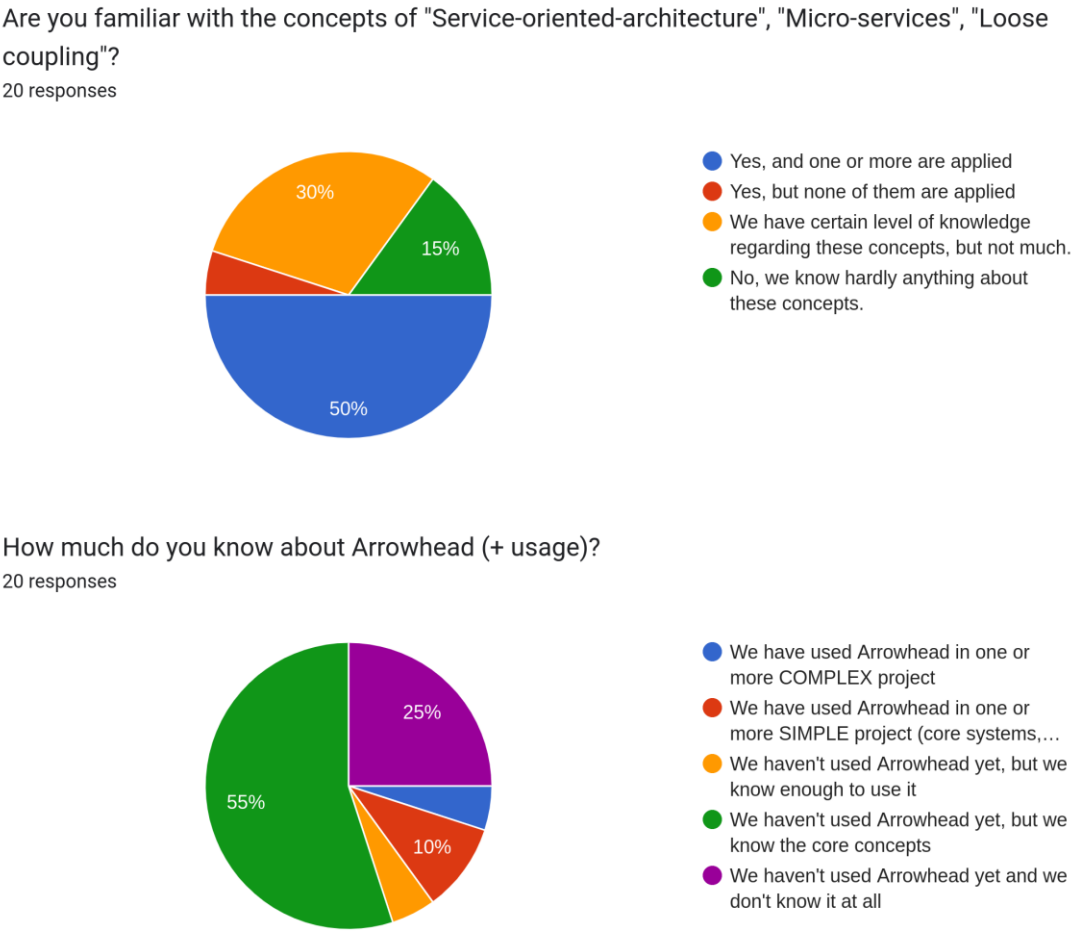


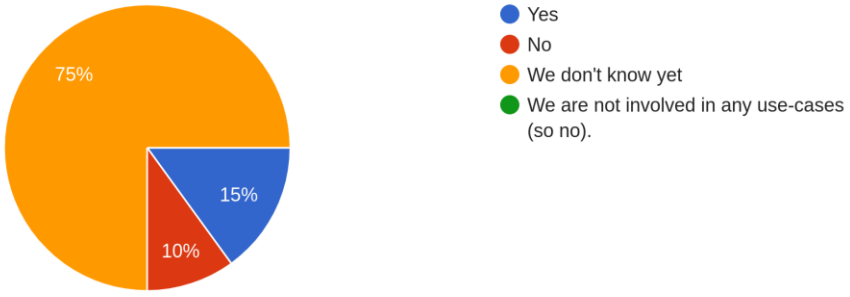
Figure 3: Questionnaire results I.

Please list (or shortly describe) that project in which you used Arrowhead!

- AH was tested within the IFD factory and AH Tools are applied (e.g. IO-Link)
- Wireless onboarding of sensors in Arrowhead Tools
- We used it as a part of Wireless Security Enhancement Tool demonstrator in Arrowhead Tools

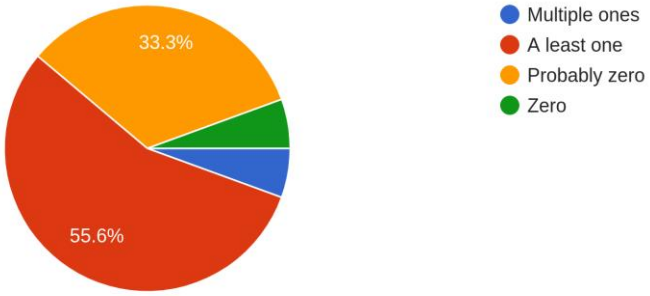
Do you plan to use Arrowhead in one or more use-cases in which you're involved?

20 responses



How many partners with Arrowhead competence do you have in your use-cases?

18 responses



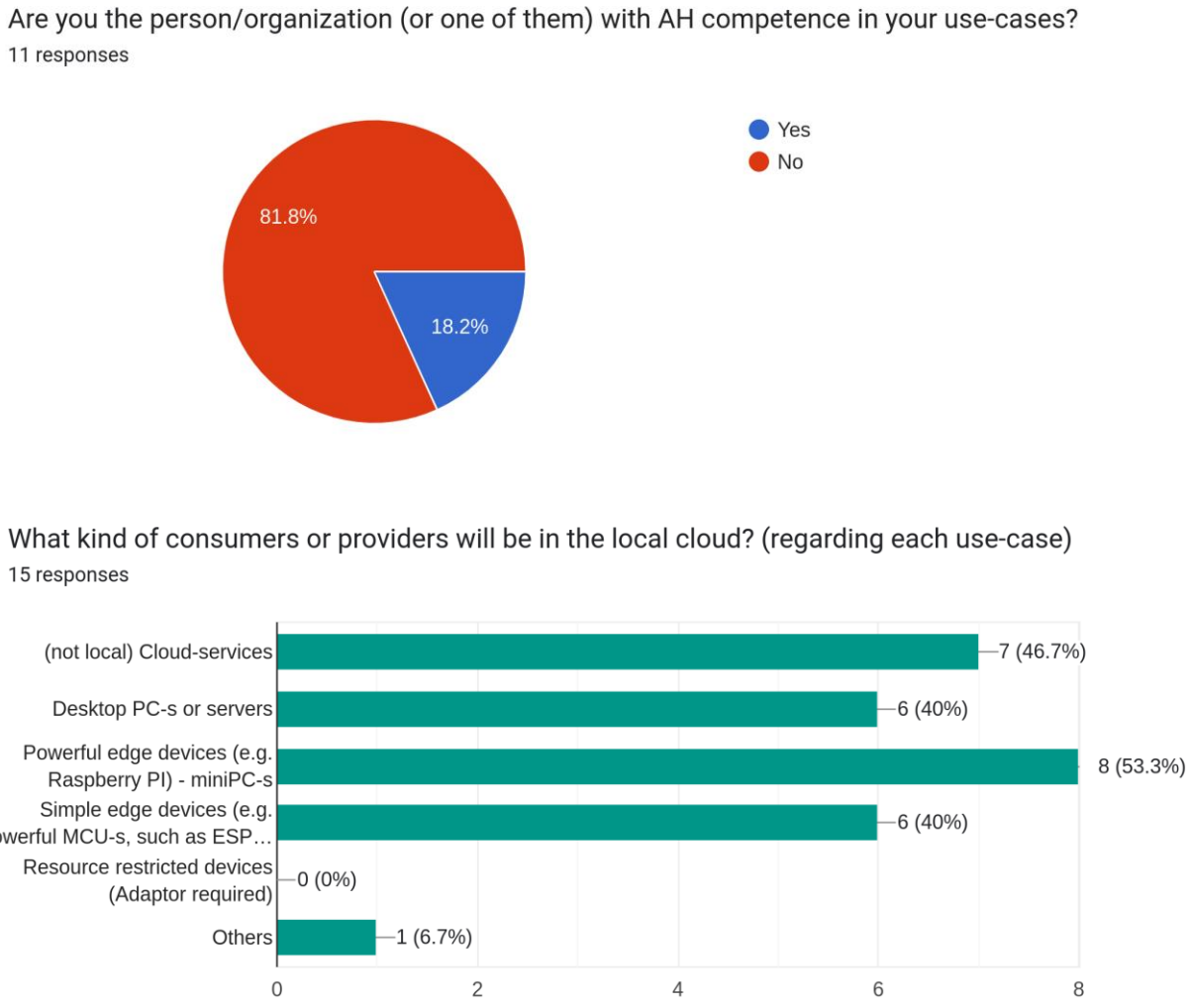


Figure 4: Questionnaire results II.

You can specify the devices if you want/can (including the specific use-cases):

- Wireless sensors / edge gateways
- this is to be defined
- Edge computing platform and additional UC partner devices (e.g.sensors)

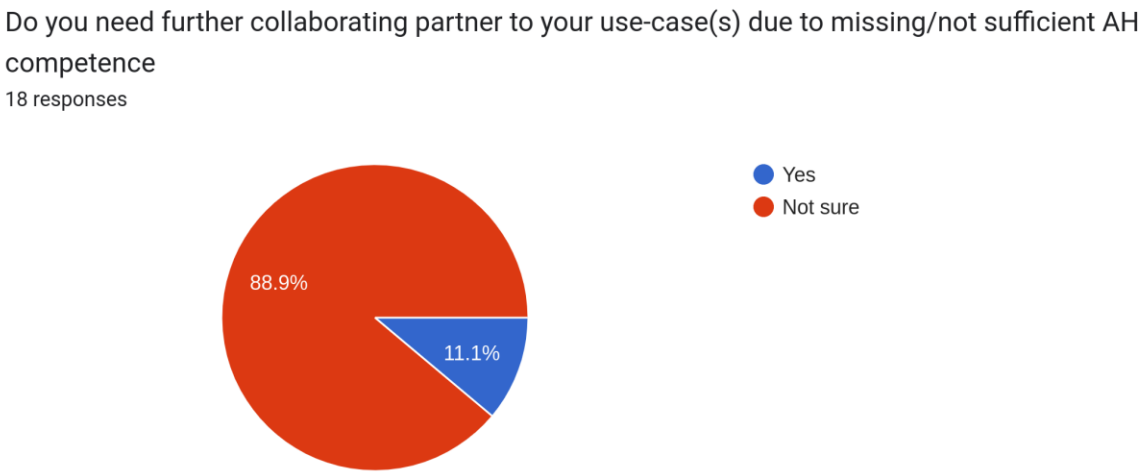


Figure 5: Questionnaire results III.

According to the responses, most partners are not just familiar with the concept of microservices and SoA, but they also know the Arrowhead framework as well. However, most partners have limited practical experience in using Arrowhead as a foundation in the use case. Consequently, the dissemination in the previous project can be considered successful as the partner adopted the microservices-based approach as well as the Arrowhead Framework. Therefore, in WP3, the dissemination of Arrowhead should be continued, and extensive help should be provided for partners to further adopt the framework into their systems in a more advanced way.

3.3 Real-time virtualization and orchestration platforms for reliable industrial production systems

The research and development of TTTech Industrial Automation (TIAG) and TTTech Computertechnik (TCAG) in this work package are dedicated to extending current software virtualization platforms such as containers or hypervisors, adding (hard) real-time guarantees needed for reliable data communication or control in industrial and production systems. The implementation of the virtualization method will allow for easier deployment, software packaging, and cloud-based management. For example, the AI-supported security algorithms and models developed by Gdansk University of Technology (GUT) will be deployed using the TTTech containerization approach on the TIAG Nerve edge computing platform. They will be executed in a safe and secure way, guaranteeing freedom from interference during the data communication tasks of industrial systems such as, e.g., automated indoor food production (UC11, see more details about the UC11 system architecture in the corresponding section of this deliverable).

Container-based virtualization is a lightweight deployment solution within heterogeneous Fog and Edge Computing (FEC) systems. With large data sets available from the sensors, e.g., in automated production systems, the need for resource distribution between Edge, Fog, and Cloud is essential to deal with highly complex real-world industrial systems. Efficient service orchestration across those levels in real-time is a huge challenge. When used in real-time FEC applications, orchestration is needed to adapt to variations in system behavior, transient overload scenarios, and changes in resource availability. Container-based orchestration can efficiently and automatically deploy, redimension, and relocate containers, each executing an “own” service (e.g., a variable controlling energy consumption on a module or for the

whole system, or material usage, etc.) according to the measured real-time performance of critical applications.

In the reporting period M1-M12, TIAG with TTTech focused on the development and evaluation of a highly configurable orchestration framework for real-time containers that improves the scalability and expressiveness of state-of-the-art approaches. We designed three different heuristics for the offline placement and dimensioning of containers on nodes that scale with the size of modern industrial systems and enhance the decision-making of the online hierarchical controller to include time-utility functions which can model more accurately the usefulness of results in the case of deadline misses. We evaluated our framework's performance and efficiency using synthetic test cases and a real-world test bed.

FEC combines the advantages of cloud computing with the ability to transfer low-latency control functions to edge nodes. This helps reduce the communication latency for critical messages, which is crucial for real-time applications, e.g., in industrial automation [Salman2019], [Shaik2020].

Virtualization in FEC improves the deployment and adaptation of services based on the current state and capabilities of the nodes [Morabito2017]. Container-based virtualization has the benefits of smaller image sizes and faster startup times [Xavier2016], offering a lightweight deployment within heterogeneous cloud or fog environments with less overhead than full virtualization. As a result, containerization has become the preferred way to package and deploy applications in FEC [Struhár2023], [Vaucher2018]. Container-based virtualization facilitates FEC because additional instances can be quickly started and migrated between server instances as needed depending on an application's demand, and resources can be optimized by removing unnecessary instances [Tosatto2015].

Hence, in industrial automation, container-based virtualization and orchestration have a significant role in fulfilling Industry 4.0 and 5.0 requirements for flexibility and resource efficiency [Struhár2023], [Gkamas2022], [Lasi2014], [Salman2019] but can also be beneficial for other mixed-criticality domains like, e.g., mobile machinery or energy applications. GNU/Linux operating systems realize containers using namespaces and control groups (cgroups), isolating and providing access limitations to applications (tasks) when accessing resources such as CPU, memory, or IO [Abeni2019]. However, containerization cannot guarantee meeting critical deadlines in real-time applications. Abeni et al. [Abeni2019] extended the scheduling feature of the Linux kernel to enable the hierarchical scheduling of containers so that containers are scheduled using earliest-deadline first (EDF), and tasks within the containers use a fixed-priority (FP) scheduler.

S-Julán et al. [S-Julián2023] reviewed automated management in cloud-based architectures aimed at improving network latency, power consumption, and overheads. Struhár et al. [Struhár2023], [Struhár2021] present an orchestration framework that uses an offline SMT-based approach to create an initial dimensioning of containers on nodes and propose a hierarchical online control approach for redimensioning and moving containers based on the runtime performance and system load. However, the approach in [Struhár2023], [Struhár2021] suffers from the downside that the offline phase does not scale for real industrial systems and that the online adaptation of containers is based on simple metrics that do not capture the utility of computation results after a deadline miss. TTTech implemented a highly configurable orchestration framework that uses a similar control hierarchy as in [Struhár2023], with the following main improvements:

- redesign, reimplement, and optimize the controller hierarchy from [Struhár2023];
- propose three (3) different heuristics for the offline placement and dimensioning phase that scale with the size of modern industrial systems: using Simulated Annealing (SA), Greedy Algorithm (Greedy), Compositional SMT (C-SMT)
- enhance the decision-making of the online hierarchical controller by including time-utility functions that can model more accurately the usefulness of results in the case of deadline misses.

Our work builds upon Struhár et al. [Struhár2023], who present a hierarchical resource orchestration framework that calculates initial container placements in an offline-phase via an SMT-based approach and relocates containers in the online-phase based on simple metrics. We have not only introduced a more scalable heuristic-based offline step and extended the online control decisions using time-utility functions (see below) but also redesigned and rebuilt the online phase while keeping a similar hierarchical control structure.

The approach in [Struhár2023] is entirely implemented in the Kernel, and thus, it cannot be extended easily and is less portable. We kept the changes to the Kernel to a minimum, adding only three hook functions, and implemented the logic in a loadable kernel module. Furthermore, instead of having a separate Container-Level-Controller that adapts the budget of a container, this functionality is implemented in the Node-Level-Controller (NLC), which is running in userspace as opposed to the kernel level implementation in [Struhár2023].

There are two phases in the proposed approach; the offline phase aims to use the three mentioned heuristics to find an initial placement of containers to nodes and a correct dimensioning in terms of budget. In the online phase, a time-utility function (TUF) that assigns each containerized task a semantic value capturing the degree of temporal error is used. TUF assigns a utility to the result of a task depending on its completion time, even after the deadline has expired (the TUFs commonly used in real-world applications yield 100.0 if the task finishes before its deadline). A diagram showing the architecture of the online orchestration phase is shown in Figure 6 utilizes the following components: the Cluster-Level-Controller (CLC), the Node-Level-Controller (NLC), and the Linux kernel module (KM), containing the logic of the online orchestration phase.

The communication between the CLC and the NLC is implemented via bidirectional HTTP requests. The CLC configures registered NLCs and can issue to start, to stop, and to relocate a container on registered nodes by communicating with the respective NLCs. Initially, the CLC starts containers by sending start commands to the NLCs according to an allocation found in the offline phase. In the online phase, or during runtime, we have implemented a simple relocation policy based on the current CPU utilization in the cluster. When a NLC demands relocation of a container due to a diminished time-utility, the CLC requests the current CPU utilization from NLCs running on registered nodes. The CLC decides to relocate the container with degraded performance to the node with the lowest CPU utilization and stops the container on the node it was previously running on. If there is no node with lower CPU utilization than the node the container is currently assigned to, the CLC aborts the relocation.

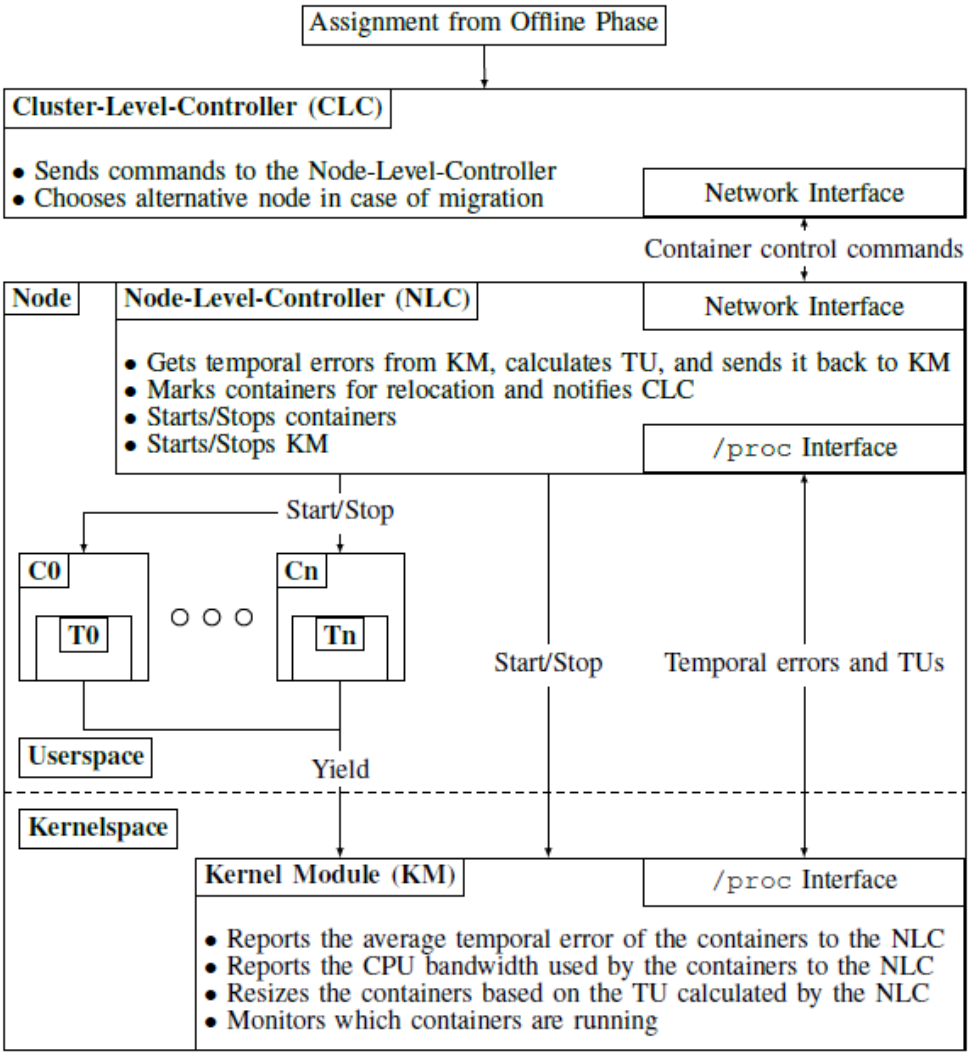


Figure 6: Architecture Diagram Online Phase

After starting up, the NLC awaits the reception of its configuration from the CLC containing, e.g., KM parameters, the relocation interval, or the update interval between the NLC and the KM. The relocation interval determines an NLC's periodic checks if a container needs relocation, i.e., if a container's time-utility dropped below a threshold. We can pick the behavior of the relocation interval from three available options (fixed, minimum, and maximum). The function of the KM is twofold. Firstly, the KM keeps track of hosted containers and monitors temporal errors for each container's tasks. Secondly, the KM adapts each container CPU quota when receiving tasks' updated TUs from the NLC via the /proc filesystem. We performed separate experiments to evaluate the offline phase and the online phase, and the results were presented at the 20th IEEE International Conference on Factory Communication Systems (WFCS 2024) in Toulouse. The proposed approach demonstrates the scalability and performance of the TTTech framework using synthetic test cases and a real-world test bed. Using this approach, the advantages of software containerizations and virtualization, such as easier deployment, software packaging, and cloud-based management, will be transferred to such systems with guarantees known from classical real-time systems.

We performed an architecture analysis on how to combine our approach with the Arrowhead framework and envision possible integration dimensions of our real-time orchestration framework with Arrowhead as follows:

a. **Enhancing the real-time data handling of Arrowhead with real-time execution control of our framework.** We can use Arrowhead as a microservice architecture in which self-contained local real-time clouds feature nodes running our orchestration framework in the underlying OS. These critical local real-time clouds have nodes with real-time execution capabilities, i.e., OSes with real-time scheduling methods that provide real-time execution (e.g., fulfilling deadline requirements of tasks) as well as a time-sensitive communication backbone connecting the nodes. On these nodes, the applications (services) that run have real-time requirements in terms of, e.g., fulfilling deadlines locally but also end-to-end latencies when they are part of dependency chains across nodes that are located within the same local real-time cloud or even feature inter-cloud communication and execution. Within the nodes of these local real-time clouds, our framework is integrated into the OS, controlling the CPU bandwidth available to each container running on the nodes such that the real-time behavior is maintained (i.e., deadline misses are minimized) and the real-time utility (expressed with the help of time-utility functions) is maximized. Our framework acts, therefore, as an enhancer for Arrowhead capabilities, augmenting its real-time communication capabilities with the real-time execution and orchestration of services on the local nodes of local real-time clouds.

b. **Orchestration support for our offline dimensioning.** Our orchestration framework is computing an optimal dimensioning of containers (running for, e.g., microservices) depending on their real-time requirements and the capabilities and load of the respective nodes. For this, we compute both an allocation of containers to nodes and a limit and rate of access to the computing resources of the assigned node. This is done via schedulability tests for real-time tasks under hierarchical scheduling on periodic resource abstractions [Lipari2003] under the consideration of optimization criteria, for example, reducing the system overhead and/or reducing the overhead of the periodic resource abstraction. However, our framework is unaware of service providers and consumers. Arrowhead can be responsible for registering and discovering services within the local cloud, and the Arrowhead Orchestrator can find a proper provider for a service consumer. However, it may be that the provider still needs to be assigned to run on a node or that the increase in load due to servicing the new consumer requires more computing resources than are available on the node where the service provider is currently running. In these cases, the service provider needs to be (re)allocated to a node in the local cloud depending on its resource requirements and the resource availability of the possible nodes. Our method can be used to compute both the allocation and the optimal dimensioning of the real-time container, which includes this service provider.

c. **Arrowhead as an interoperability, security, and safety provider.** At this stage, our solution needs further improvements to fulfill interoperability, security, and scalability requirements of Industry 4.0. Hence, the Arrowhead capabilities concerning authorization, authentication, and interoperability can be used orthogonally to our approach. For example, our framework uses simple and unsafe Server-Sent Events (SSE), defined in the HTML specification, as a communication protocol to communicate to the cluster-level controller the state of the containers on each node. This is not scalable since it relies on a particular communication protocol, nor is it safe since it does not have any authentication and authorization mechanisms. Hence, our framework can be subject to attacks and suffers from poor scalability and portability. These aspects could be handled using the communication (e.g., autonomous protocol translators) and security (Safety and Security Managers) capabilities within Arrowhead. Moreover, the Data Manager, Gatekeeper, Gateway, and Time Manager services of Arrowhead can be used to, e.g., collect and exchange the necessary data for the

cluster-level controller (i.e., state of resource consumption and current resource demands) and synchronize time between local clouds, thus enabling our framework to go beyond local clouds and provide inter-cloud real-time execution control.

As the next step regarding the security mechanism, as a part of ST2.3.3, the GUT architectural security mechanisms will also be analyzed and integrated into the overall edge computing architecture of UC 11. Currently, the design of the UC11 system architecture is ongoing. See below a preliminary simplified architecture explaining the roles of the UC partners and their contributions described in this document for more details. GUT's smart antenna, enhanced with AI-supported mechanisms, will potentially offer future customers smarter wireless-supported services, which could be proposed as an additional feature in combination with the TIAG platform to cover wireless applications.

3.4 Robust real-time networking for AI applications

TIAG is leading the subtask in WP3 related to the development of the SW architecture that supports the integration of AI methods and allows for the collection and storage of machine data as well as for managing and deploying SW remotely and locally. In particular, TIAG designs and implements a next-generation Management System enabling the AI models' deployment and execution at the edge. This system is part of TIAG's Nerve platform,

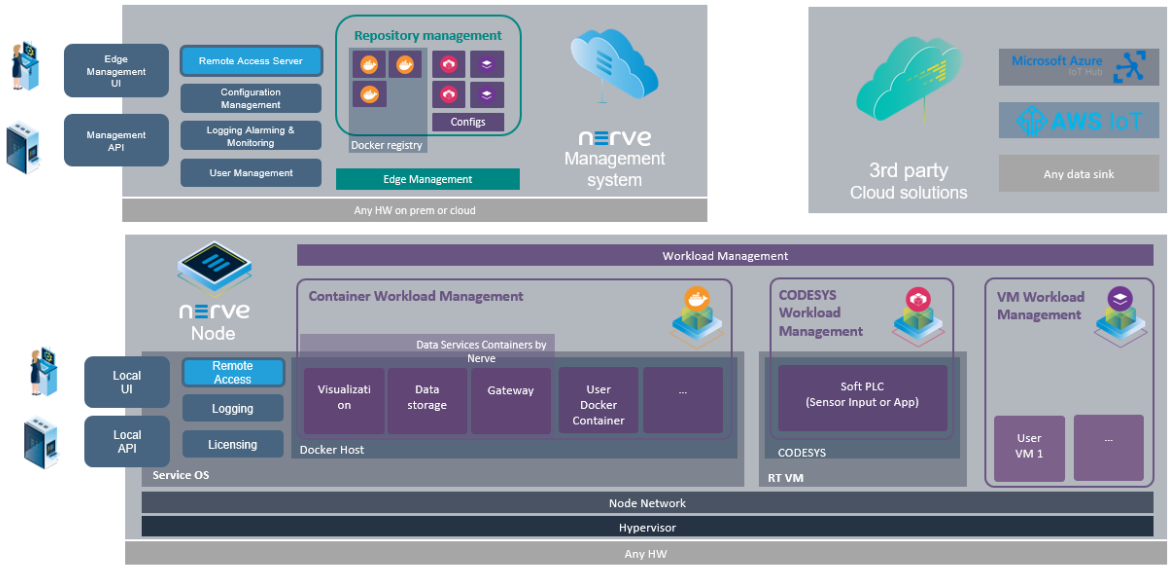


Figure 7: TTTech Industrial Nerve Architecture

combining technologies – from hardware to protocols to operating systems – that can be used to implement digitalization and allow for the collection, processing, and analysis of machine data in real-time to increase efficiency and production optimization. The users of this edge computing technology are, in the first place, machine builders but also innovative companies such as automated indoor plant prodution system owners. In a nutshell, Nerve provides a solid foundation for the proposed IIoT use case and consists of node software that runs on devices at the edge, as well as a management system that runs in the cloud or on-premises. The entire system can run offline as well. With Nerve, the user has two interfaces: the Local UI and the Management System. It is possible to use the TIAG's Nerve hardware devices (also called Nerve nodes) such as, e.g., MFN100 or another legacy HW from, e.g., Kontron, Siemens, Vecow, etc. The Nerve Management System (Figure 8) is a web-based service that permits management of registered Nerve nodes in the system. The system can be hosted by

TTTech Industrial or installed on-premise, which provides flexibility for the deployment of devices, setting up network configuration, and management of the workloads (CODESYS, Virtual Machine, Docker). The machine data can be acquired using the integrated data gateway via communication protocols such as OPC UA, MQTT, Kafka, Modbus, and Siemens S7 and visualized using tools such as GRAFANA. NERVE also supports connection to industrial remote IO systems via EtherCAT and PROFINET for reading sensors and machine data using integrated Soft PLC. The Nerve architecture is shown in Figure 7.

The development focus of this project in the reporting period lay on docker workloads and the composition of those to support software developers in building, sharing, executing, and verifying applications anywhere. With *Docker Compose workloads*, users have the possibility to manage multiple services in the Nerve system and establish a link between them. One can also use a Docker container in combination with a compose file to use the full extent of Docker Compose. In contrast, a Docker workload in the Nerve system is designed to cover the basic functionalities and configuration options of a Docker container through the UI of the Management System. We are concentrating on this approach, in particular, due to the requirements of UC11 where several partners' applications will co-exist and run on the same computing platform. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. In this regard, the Nerve Management System will enable the following services:

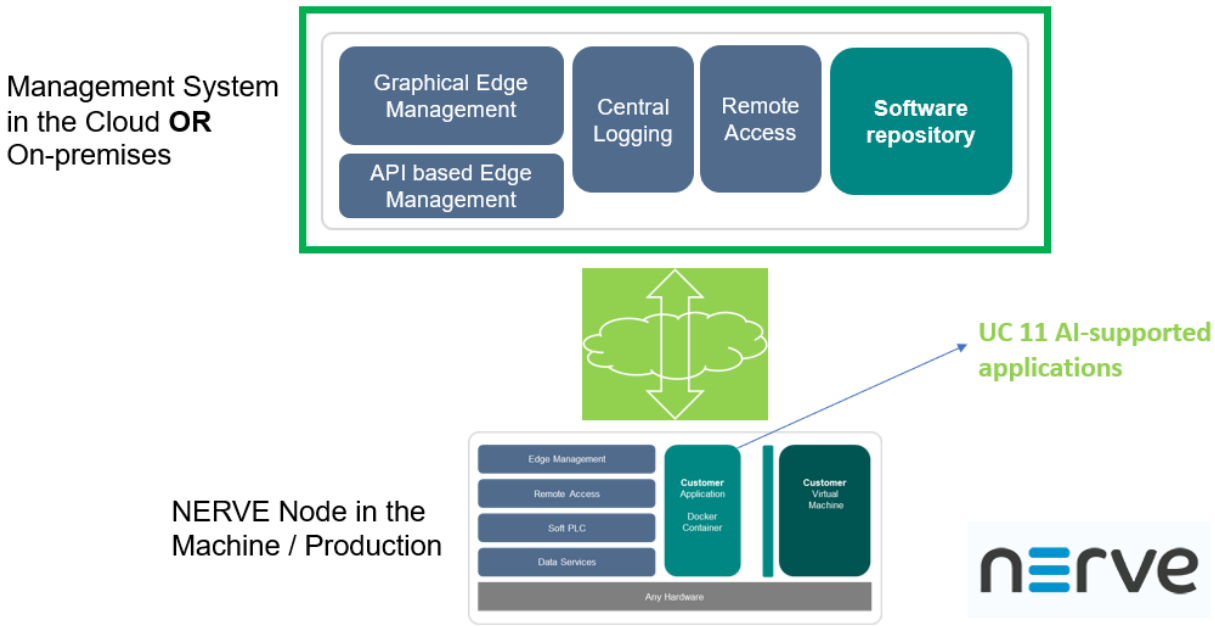


Figure 8: TTTech Industrial Management System

- Use Nerve Data Services to collect live and training data
- Create and train AI models using the preferred machine learning toolkit
- Deploy models and configurations and integrate them into a DevOps flow using the API
- Connect the trained models to real-time data using applications running on Nerve
- Visualize results in local or cloud-based dashboards

As mentioned above, however, a specific type of workload management underlying technology must be selected and tailored to an application or a combination of them. For this reason, the architectural concepts with Docker Compose workloads are a promising approach to be developed and evaluated in close alignment with UC 11 partners.

3.5 Model based AI Gym concept

Since AI usage in industrial production is already very diverse, this approach to an AI gym provides a model-based infrastructure to build SOA/microservice solution architectures that have AI-based microsystems. Enabling efficient changes of AI algorithm, updates of trained AI models, updates of training data sets, federated learning, etc.

For this purpose, the Eclipse Arrowhead Domain Specific Language (DSL) is used, enabling the design of solution architectures. The first level is black box abstract architecture, followed by a white box implementation architecture. Using the Eclipse Arrowhead microsystem model library, the solution engineering can be significantly simplified. Thus, most often, only the AI-based microsystems and the data set storage need to be modeled in more detail.

From a solution modeled in such a way, the low code machinery of Eclipse Arrowhead can be used to produce an almost executable-ready implementation of the intended solution.

Once completed to a working implementation, first-time updates of AI algorithm, data sets, training procedures, etc., an AI-Gym for the solution use case is established.

Using this solution, tailored AI-Gym changes will be made at the modeling level, where the next working test can be generated almost automatically (my disclaimer for expected bugs to be discovered).

The approach has been tested with students in our AppleID AI Master program. Here, 12 students successfully used the modeling approach to develop black-box and partly white-box architecture models for 12 different use cases. Based on the successful results, the next step is the integration of automated code generation and containerizing.

An example of architecture was intended to be depicted in a figure. Bugs in Eclipse IDE connected with the new Apple M3Max chip prevented such inclusion.

4. Use case integration of the common cyber architecture

This section briefly summarizes the planned individual and collaborative contributions of partners and groups focusing on different use cases.

4.1 AI-supported Industrial IoT utilizing the common cyber architecture

The common cyber architecture utilizing Service Oriented Architecture was evaluated in a use-case about industrial production of plants (Use-case 11). The targets are increasing yield, lowering energy and water use, and detecting the health and growth of plants using AI-based tools. In our case, this system represents indoor plant production facility without soil in an enclosed environment without sunlight. Hence, all energy, normally provided by the sun, must come from artificial lighting, e.g., LEDs. There are a variety of parameters to be considered in this complex system.

The system we are using is called a hydroponic system. Its characteristics are that it is soilless, and the plants are grown in a nutrient-rich solution. In this case, it is a running system that uses NFT (Nutrient Film Technic) channels. The roots of the plants (salads) are constantly submerged in the flowing nutrient solution. Normally, a substrate, like rock wool or perlite, is used. However, PYT (owner of the indoor plant production facilities in Austria) does not use any substrate to hold the roots in place. A pump circulates the nutrient solution permanently through the system, and air conditioning and ventilation are used to control the climate in the production room. This system allows for a highly controlled and resource-friendly growth setting.

The overall use case deployment is divided into three sub-demonstrators, as shown below in Figure 9:

1. NLP-based and AH-compliant sandbox prototype for controlling production parameters to increase the productivity of an automated vertical plant production system. The overall AI-based and IoT-based production simulation solution will be deployed on the TTTech industrial cloud-based edge computing platform. The edge computing platform will host several services, including the AH core system, an AI/NLP-based module that analyses the data coming from the sensors, the autonomous control loop based on MAPE-K, and an AH-compliant client based on the Generic Autonomic Management as a Service (GAMS) module. (TTTech, RSA, PYT; ongoing integration in St. Pölten, Austria).

2. ML-based mechanisms for visual plant health control, detection of anomalies using camera data, training data, and developed algorithms. The security analysis framework is also a part of this work to ensure that UC11 and the AH concepts being developed are in alignment with the Cyber Resilience Act. A Security Control architecture passed on an NLP-derived compliance mechanism is shown for the use case based on the analysis. AIT utilizes NLP to derive security compliance mechanisms that are out of standard to show compliance with the overall architecture. (AIT, PYT; ongoing integration in Oberwart, Austria)

3. AI-powered cross-domain communication gateway for reliable and efficient wireless connectivity demonstrator. The edge computing platform modules from TTTech will be integrated with the HW by GUT to enable secure wireless communication and test in

addition to the proposed PoC for other domain applications (GUT, TTTech; ongoing integration in Gdansk, Poland).

For all the above-mentioned demonstrators of UC11, the functional and non-functional requirements, as well as preliminary specifications, were defined in M1-M12. RSA, GUT, AIT, and TTTech are extending and testing the Arrowhead framework with Arrowhead-compliant AI modules and relevant standards, a cloud-based management system, and tools and methods for secure communication between IoT devices. MAPE-K framework (Measure, Analyze, Plan, Execute, Knowledge) was selected and will be extended and adapted to the needs of the use case. See use case partner contributions described in this deliverable for more details. Below, we present a simplified system architecture of the AI-enhanced hydroponics vertical farm system. The MAPE-K-based approach will operate on both edge and cloud, between sensors and actuators, to monitor the indoor plant production environment and optimize the plant's growth.

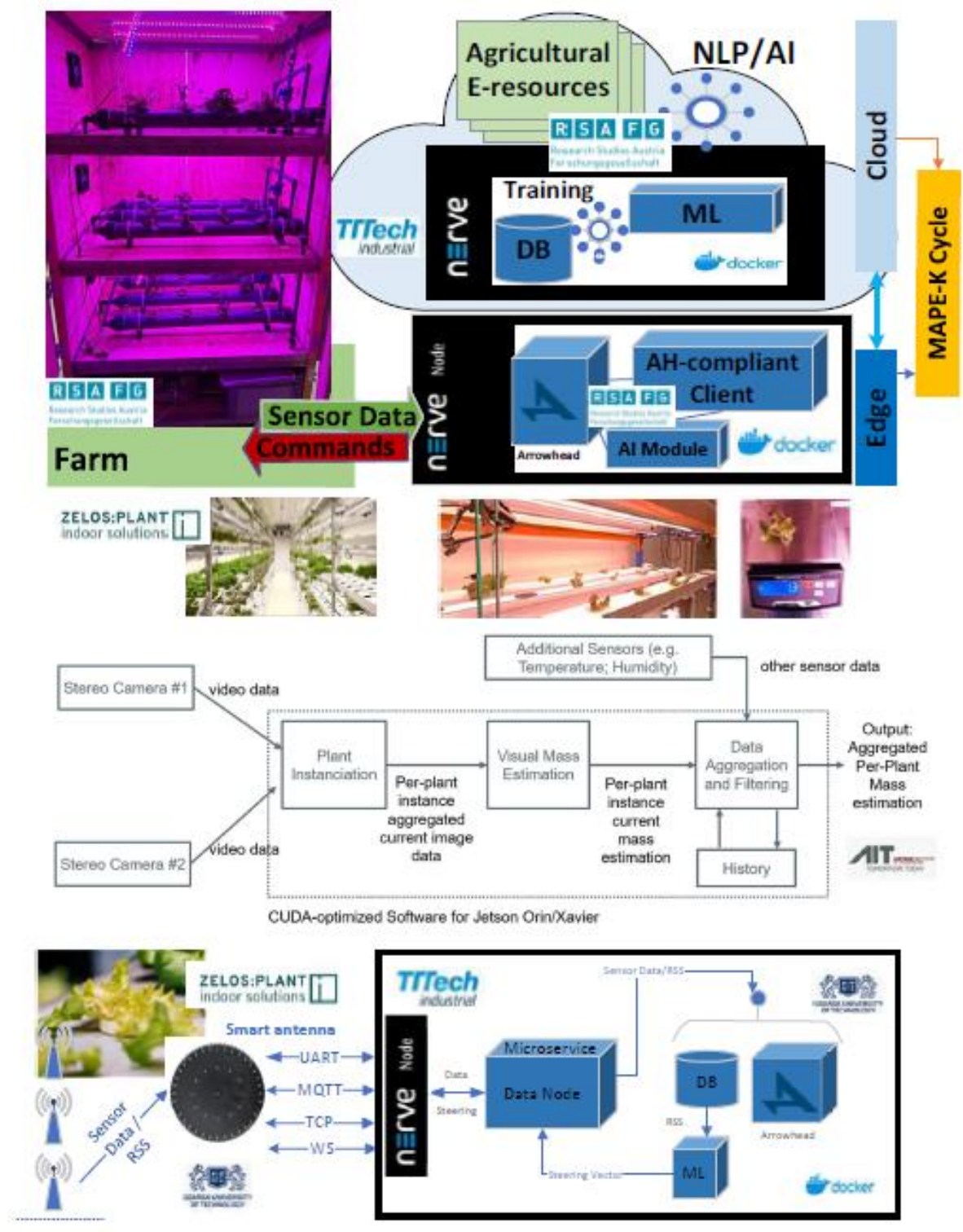


Figure 9: UC 11 AI-supported Industrial IoT for Indoor Food Production: System Architecture (TIAG, PYT, AIT, RSA, GUT)

4.1.1 A prototype for evaluating investigated aspects of the common architecture

RSA built a prototype in a laboratory environment using hydroponic technology to implement the desired autonomous control loop system. This prototype constitutes three parts, i.e., cloud, edge, and the plantation facility. The cloud part hosts the NLP pipeline, which we use to create the knowledge base. The edge part hosts several services, including an AI module that analyses the data coming from the sensors, the autonomous control loop based on MAPE-K, and the microservices provided by the deployment framework.

In practice, several types of sensors and actuators are installed in the plantation facility to collect the data the autonomous control loop requires. The simplified plantation facility is a container for vertical plant production that employs hydroponic technology to host and grow the plants. Sensors, actuators, and microcontrollers are installed in this part to monitor and control the environment in the plantation facility. Obviously, the proper regulation of the parameters of the environment is decisive when it comes to growing healthy plants. Figure 10 depicts the prototype, which integrates several tools and technologies, such as NLP and AI, MAPE-K to automate policy creation, monitoring, and execution, and Eclipse Arrowhead framework to execute service deployment.

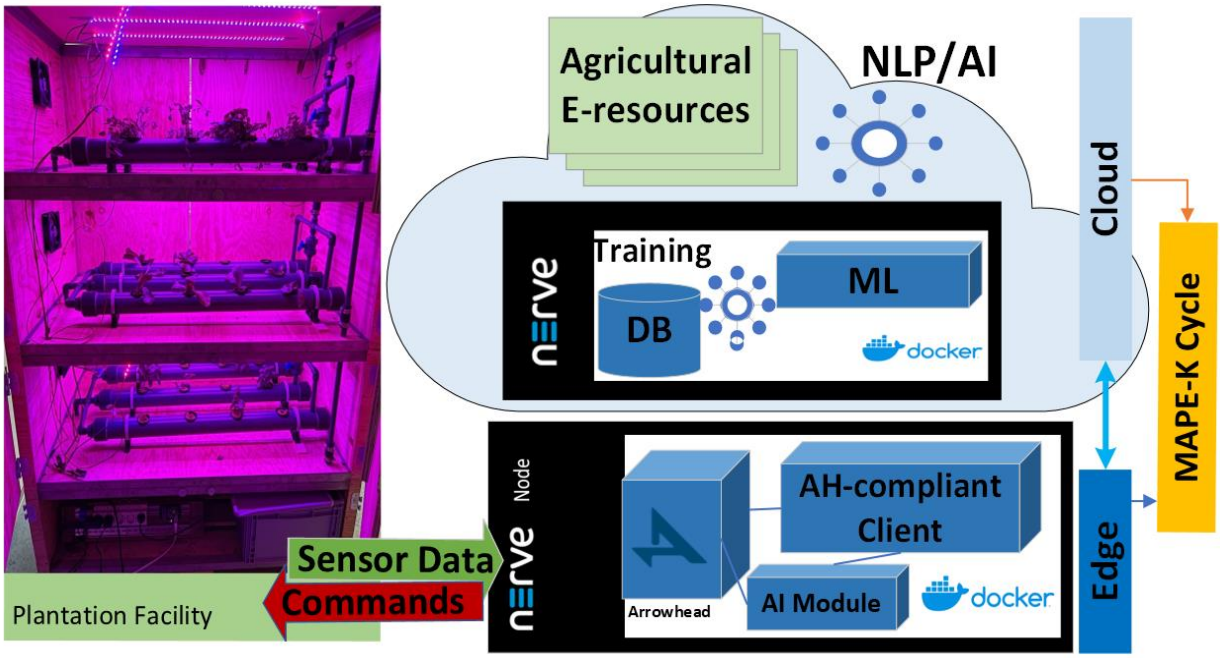


Figure 10: The Hydroponic Prototype

4.1.2 Architecture for integrating Micro services at the edge level

In industrial indoor plant production, integrating microservices at the edge level involves designing an architecture that can efficiently manage and coordinate the various services required for monitoring and controlling the production environment.

Figure 11 depicts the architecture for integrating microservices at the edge level in industrial indoor food production, showcasing the different components and their interactions within the system.

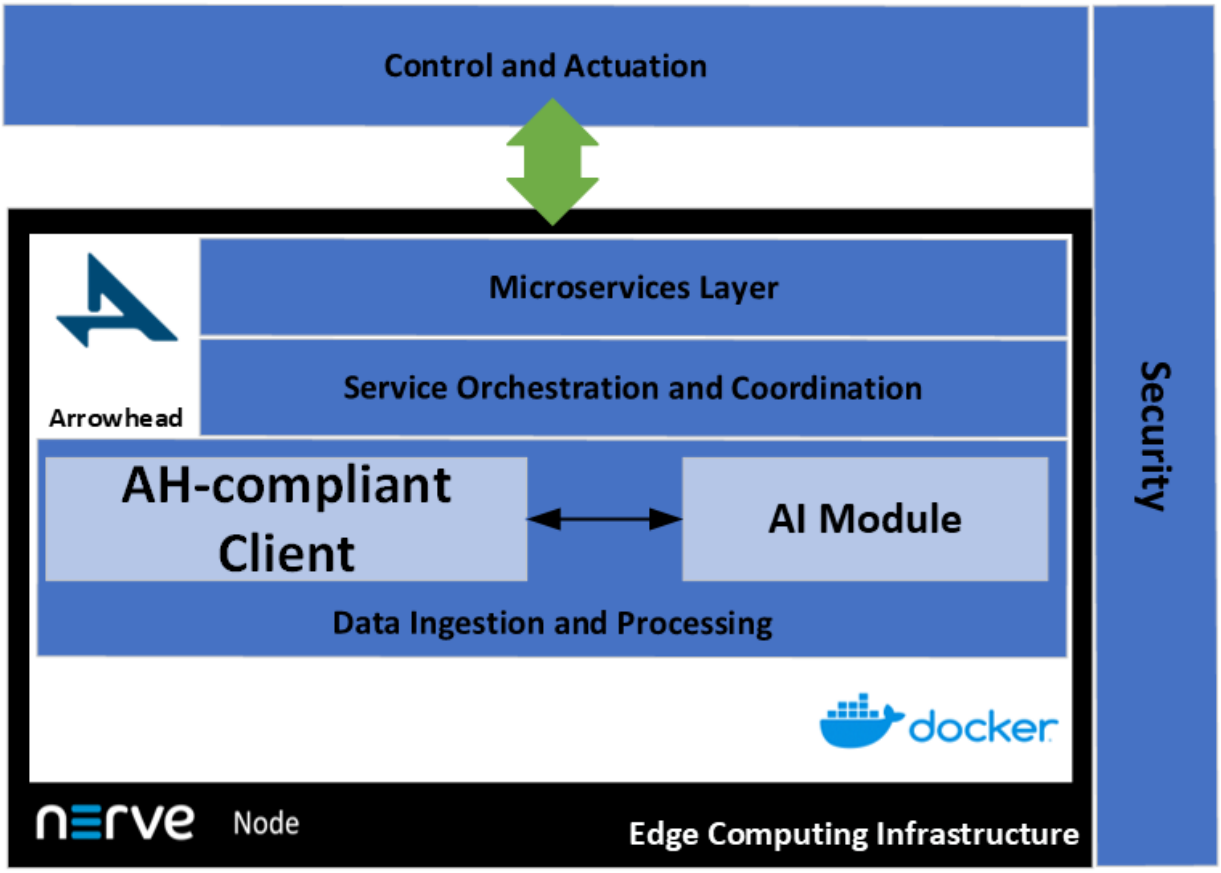


Figure 11: Architecture for integrating Micro services at the edge level

Explanation of the components:

1. **Edge Computing Infrastructure:** This represents the physical hardware at the edge, i.e., the Nerve node, which is responsible for running the microservices.

2. **Microservices Layer**: This layer consists of individual microservices responsible for specific functions, such as data ingestion, processing, control, and communication.

3. **Service Orchestration and Coordination**: This component manages the deployment and coordination of microservices using Eclipse Arrowhead.

4. **Data Ingestion and Processing**: These microservices ingest data from sensors, process it using stream processing frameworks, and analyze it to derive actionable insights.

5. **Control and Actuation**: These microservices implement control logic to adjust environmental parameters and interact with actuators to regulate equipment and machinery.

6. **Security**: This component ensures the security and reliability of the system through encryption, authentication, redundancy, and compliance with industry standards.

By leveraging microservices architecture at the edge level, industrial indoor food production facilities can achieve greater flexibility, scalability, and efficiency in managing complex production processes while minimizing latency and bandwidth constraints associated with cloud-based solutions.

4.2 AI-Enhanced Value Chain

Different modeling (system, processes, ML, planning) questions related to the SOA princeples for building complex systems were investigated and evaluated in use-case 2. The main goal of here (Enhanced Value Chain), as a contribution to WP3 by PCL, TUE, and RUG is to create a framework where customized production is possible within a mass production and design environment.

For this goal, the system architecture, shown in Figure 12, is created. On the system design side (left side), system modeling is considered the key for user profile creation and tracing user needs & requirements to the system functional and physical decomposition and to the system part attributes. System modeling is also considered the knowledge management technique between system design and production.

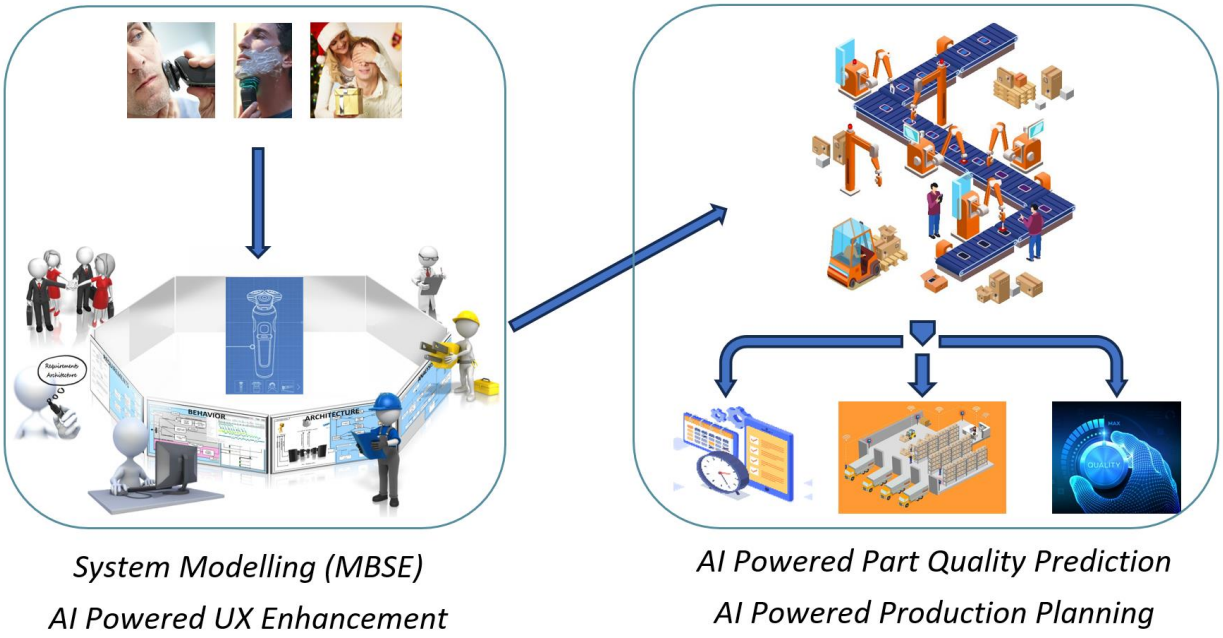


Figure 12: Use Case 2 System Architecture

AI-Powered UX Enhancement algorithms are also created and deployed to individual products to learn user behaviors better and guide users during use.

On the production side (right side), production process modeling, AI-powered part quality prediction, and AI-powered production planning are considered key tasks.

With system modeling, user preferences shall be modeled and traced to customized part properties and knowing the production process flow and process transfer function between process parameters and part quality, process parameters shall be controlled to produce the required parts always with high quality. AI-powered production planning shall enable continuous and efficient operations in such a dynamic production environment.

The following tasks are created within Use Case 2:

- System Modelling [run by PCL];
- Production Process Modelling [run by RUG];
- ML Modelling between process parameters and part quality estimations [run by TUE];
- Production Planning / LLM supported production planning [run by TUE].

Four partners work together in Use Case 2: Philips Consumer Lifestyle (PCL), Eindhoven University of Technology (TUE), University of Groningen (RUG) and SEMAKU.

4.2.1 System Modelling

The main purpose of system modeling is to capture the different user profiles, to trace user needs & requirements towards system decomposition and system part attributes, and to share the knowledge between design and production. Therefore, system modeling mainly takes part in the system design phase.

System models shall be linked to process models to achieve complete traceability from user attributes toward process steps, and parameters will be achieved. This will be a key to finding which process parameter to change to achieve user-driven functions and attributes.

System models, as shown in Figure 13, consist of the following steps: User Profile creation, user value study, system feature creation, system functional decomposition, feature-to-function allocation, function behavior definition, system requirements creation, mathematical modeling, system physical decomposition, risk assessment and knowledge transfer between design and production. In Use Case 2, we use Cameo System Modeler tool and SysML language for our system models.

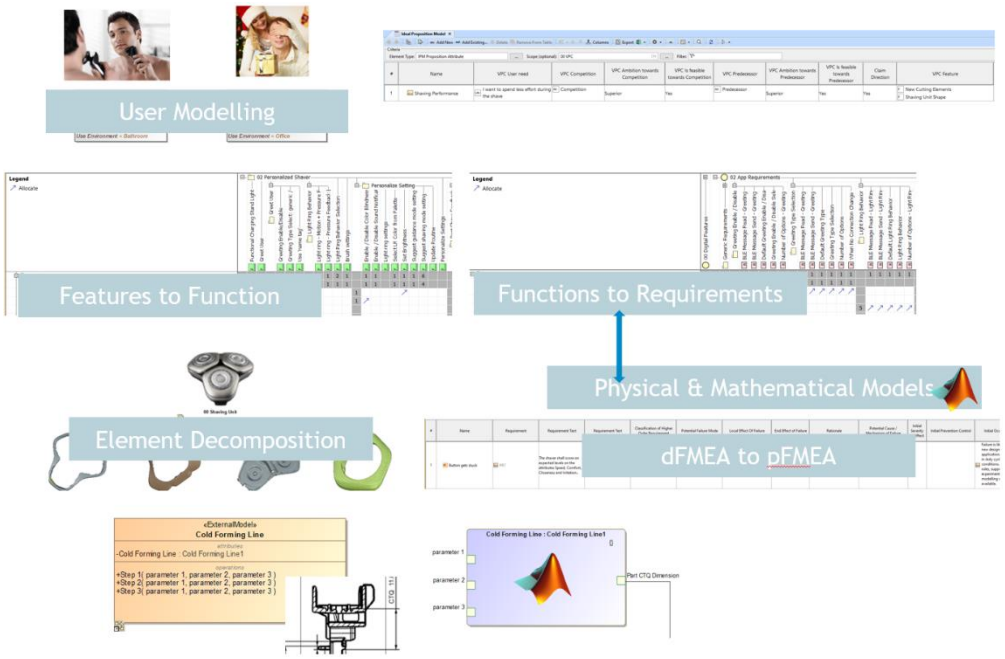


Figure 13: System Modelling Steps

System modeling has a limited contribution to WP3; its main contribution is to WP5.

4.2.2 Modelling Production Processes as Business Processes

Modern manufacturing enterprises realise large-scale automation and increased production efficiency by using Internet of Things technology to foster collaboration between digital systems — such as cyber-physical systems and information systems — and humans [Xu2021], [Lee2015]. This integration is a complex undertaking as it needs to provide the flexibility to allow it to adapt to changing business needs and has to deal with the heterogeneity of the different technologies in terms of their interfaces, communication protocols, and data formats. Service-oriented architecture approaches, such as those used by the Eclipse Arrowhead framework [Pali2017], promote interoperability between these types of systems by considering services as the basic unit for information exchange. Each service fulfills a basic business function, and business value is realised through the coordinated execution of sets of services. The Eclipse Arrowhead framework provides the Choreographer core service for defining and executing workflows on top of these basic services by leveraging orchestration and service consumption. However, the types of workflows that the Choreographer core service can execute do not provide enough expressiveness for modeling the complex processes that are typical for the manufacturing industry.

The three main limitations are the fact that it is only capable of executing simple, sequential workflows, the lack of support for the integration of user tasks, and the inability to incorporate machines without digital interfaces or manual tasks performed by people without the direct support of information systems. In order to provide a solution that can model the complex processes found in modern manufacturing enterprises and offer the full benefits of automating tasks across different services, the Eclipse Arrowhead framework must fulfill a number of additional requirements:

- Improved service registration process;
- Flexible service orchestration, for example, based on service descriptions or required functionality;
- Service orchestration for AI-based services;
- Support for the execution of complex workflows;
- The possibility to incorporate human tasks into workflows;
- Support for machines without digital interfaces or manual tasks performed by people without the direct support of information systems.

The last three requirements can be met by modelling the production processes and their supporting processes as business processes and incorporating a business process management system into the Eclipse Arrowhead framework to execute these processes [Weske2012].

Business processes are capable of expressing complex control flows, including error handling and the reaction to external events, and offer support for both human activities and manual activities. They are further capable of modelling processes that involve multiple stakeholders, either within a single organisation or across different organisations. Business process management systems provide the means to model, execute, monitor, adapt, and simulate these business processes. This integration enables the automation of processes across machines and facilitates the integration of AI algorithms, such as those developed in Work Package 2, into the process. Moreover, the process adaptation techniques allow for the reconfiguration of the machines involved in a production process while the process is running, making it possible to optimise the production process with respect to various metrics.

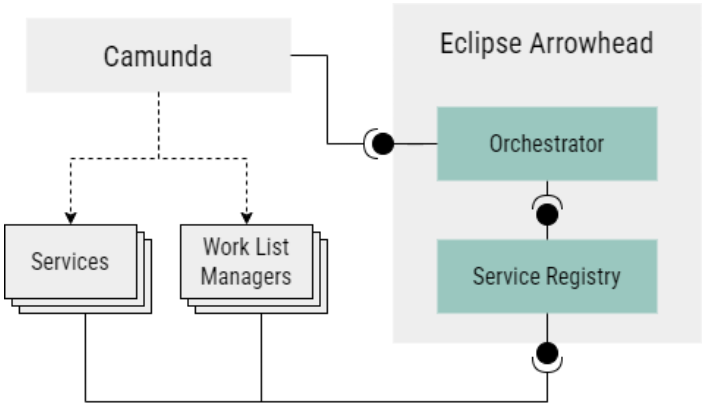


Figure 14: Integrating Camunda into the Eclipse Arrowhead framework allows Camunda to use the Orchestrator core service to look up service endpoints at runtime.

A preliminary investigation has been initiated to study the feasibility of integrating Camunda [Camunda] — an open-source business process management system that uses the widely adopted BPMN standard [OMG2011] to model and execute business processes — into the Arrowhead framework, providing the users of the framework with direct support for the modelling and execution of business processes. For executing service-based tasks, Camunda can use the Orchestration core service of Arrowhead to realise dynamic binding to service endpoints during runtime (as depicted in Figure 14).

With this dynamic binding, the process can be optimised by selecting the optimal service endpoint for a particular task. In order to support optimisation with respect to sustainability, additional sustainability-related metrics will be introduced that capture the various dimensions of sustainability, including the environmental, social, and economic dimensions. Explicit support for batch activities [Pufahl2018] is also planned, as batch processing is a common working practice, particularly in the manufacturing domain. It is, for example, not uncommon for certain resources (e.g., manufacturing machines) to be able to handle multiple items (e.g., parts) simultaneously: an oven may be able to accommodate several parts that require the same treatment. Logistics operations such as transporting parts between production lines or transporting parts to a storage area are other common occurrences of batch activities in the manufacturing industry.

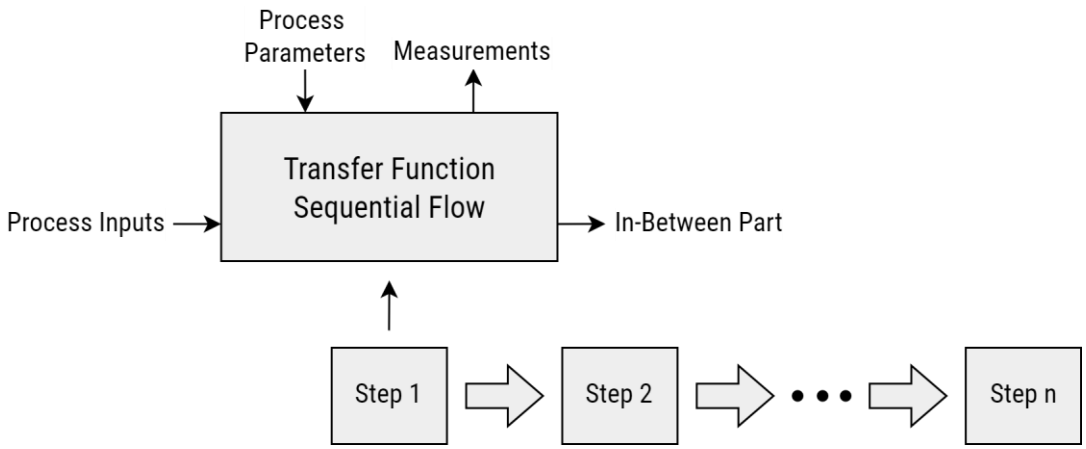


Figure 15: Abstract representation of a production process, where each step is defined by a transfer function that maps the input and process parameters to an output and a part quality prediction.

The capabilities provided by the integration of a business process management system into the Eclipse Arrowhead framework are showcased based on the scenarios provided by Philips Consumer Lifestyle in Use Case 2. One of the objectives of Philips Consumer Lifestyle is the incorporation of AI algorithms into a number of their production processes in order to predict the quality of parts after they have passed through one or more production steps. Based on these predictions, the configuration of the machine can be adjusted so as to optimise the predicted part quality (an abstract production process is shown in Figure 15).

The plan is to model the flow of these production processes, their input and output data, and the relevant metrics — such as those related to part quality — using business processes and execute them with Camunda. The monitoring capabilities of Camunda and the data collected throughout the production process make it possible to provide information about the status of a part within the production process. Process mining techniques can additionally be used on the process data to determine the most common execution paths of a process and to analyse the movement of the parts through production lines, which is especially useful for tracing parts through production lines that consist of multiple parallel lines. Business processes can also support the escalation process that Philips Consumer Lifestyle uses whenever it is detected that the produced parts are of insufficient quality. This process involves a large number of manual steps that process engineers have to perform in order to collect the relevant data from various systems and ingest this data into an analysis tool that they use to identify the root cause of the quality issue.

Several steps in this process can be automated by modelling it as a business process and realising the required services, with the option to support some steps by AI algorithms, thereby providing better support to the process engineers in finding the root cause and reducing the amount of time they spend on this analysis process. The automation of this process across the machines also allows for more real-time quality monitoring to be performed, promoting the detection of problems with the quality earlier on, which is expected to result in a reduction in the amount of raw material and resources such as energy that are wasted and improved economic value for the business.

4.2.3 ML Modelling between process parameters and part quality prediction

As a part of Use Case 2, we study ML Modelling techniques to create part quality estimations based on the process parameters. In a running production process, as the process parameters vary, the part quality varies too. By using these ML models, a transfer function between process parameters and part quality shall be created and be used run-time to understand the impact of changing process parameters. Within this task, we have 3 sub-tasks. Investigation of concept drift problem in such ML models, exploring the potential model types and creating an ML model for one of our processes.

Integrating Large Language Models (LLMs) into production processes offers a solution for quality prediction, allowing operators to use natural language for process analysis and decision-making. LLMs facilitate root-cause analysis and adapt to concept drift, seamlessly integrating with cyber architectures through fine-tuning. However, the complexity of verifying LLM outputs in industrial settings is a challenge, that can be addressed by the Statement Verification Framework. This framework enhances LLM accuracy, using NLP, machine learning, and symbolic reasoning, thereby enhancing reliability in dynamic manufacturing environments.

The statement verification framework's primary function is to ensure that outputs from Large Language Models remain accurate and relevant over time, despite the evolution of language and context (concept drift). To achieve this, the framework could incorporate continuous learning mechanisms that adapt to new data without forgetting previous knowledge. One

potential design could involve a dynamic updating system that identifies drift by monitoring the performance of LLMs on validation sets that are periodically refreshed with recent data. For root-cause analysis in manufacturing, the framework could use causal inference models to map the logical flow of statements, helping to identify potential sources of errors or inefficiencies within the processes.

The architecture of the framework could be modular, with separate components for different aspects of the verification process. For handling concept drift, an online learning module could be implemented, which would continuously integrate new data into the model's training process. To assist with root-cause analysis, a symbolic reasoning module might be integrated, converting natural language inputs into symbolic forms that are easier to analyse for causal relationships. Moreover, the framework could include a feedback loop that not only identifies discrepancies in LLM outputs but also suggests possible corrections, thereby creating a self-improving system.

Building such a framework would likely require tools that support advanced machine learning techniques. Tools like TensorFlow or PyTorch could provide the necessary computational backbone for designing neural networks, while open-source LLMs will supplied by Hugging Face and deployed via LangChain.

4.2.4 Production Planning / LLM Supported Production Planning

As a part of Use Case 2 aligned with WP3 objectives, we study automated production techniques and/or LLM-supported production planning concepts. We study advanced planning techniques within a production site based on real-time inventory levels, logistics capabilities, material substitutions, production machinery health status, etc. The goal is to create a planning framework and to study the data-gathering techniques for such ML-based planning tools. In addition, operational efficiency is crucial in such a dynamic production environment. We study an LLM tool to support resolving the production issues by understanding the issues from the prompts of operators, suggesting solutions, and altering the production plans accordingly.

4.2.4.1 Production Planning using LLMs

Incorporating the Statement Verification Framework into production planning, specifically in the context of LLM-supported production planning, offers a structured approach to enhancing operational efficiency within manufacturing environments. This integration can address the dynamic nature of production sites, where a number of different components are crucial for effective planning. The framework could be architected to function as an intermediary layer that validates the accuracy and relevance of LLM-generated suggestions, ensuring that production plans are optimized and reliable. By interfacing with real-time data streams, the framework can continuously update and verify the LLM outputs against the current production context, ensuring that the planning remains adaptive and accurate.

For the solution space and design details, the framework can be designed to handle the complexities of production planning by utilizing advanced NLP and machine learning algorithms to interpret and analyse operator prompts. This enables the LLM tool to not only understand production issues but also to propose viable solutions and adjust production plans dynamically. The verification framework ensures that these suggestions are accurate and feasible, aligning with the current production parameters and constraints. Doing so supports a more responsive and efficient production planning process, where decision-making is data-driven and aligned with the operational goals and realities of the manufacturing environment.

4.2.4.2 Automated Planning Framework

Production planning is an activity that relies on a large amount of data to fulfill its purpose. These data points also usually come from several different systems. Systems that usually don't have the same measures, product names, and product sizes, among other things. Apart from that, some of the inputs of the planning process are forecasts that have randomness and errors embedded in them. Adding to that, the planning process is usually very manual, as little to no data extraction is done centrally/automated. Those reasons corroborate the fact that the planning process needs to be divided and well understood to see where the partner can benefit the most by adding an AI model, framework, or data-gathering tool.

One of the problems that can be investigated is related to the forecasted variables. After understanding the models, considerations that are made, and forecast errors, new techniques can be researched, formulated, and tested to improve such results. Human behavior needs to be taken into consideration because when forecasted values have large errors, humans can insert slack variables to take those errors into account in the middle of the planning process, which can hinder the benefits from the improvement in accuracy.

To address operational efficiency, we investigate the integration of an LLM tool. This involves understanding operational challenges through operator prompts, generating suggested solutions based on historical data and machine learning algorithms, and then adjusting production plans accordingly. This can be extended for communication of customers/suppliers and the system.

Another approach that might become relevant is to improve data gathering/availability. There is a probability that the same data is used in different parts of the planning process but might be taken in different time frames. This can mean that this data collection, if done automatically, will improve the speed of the planning process but also will assure that the same value will be used throughout the process. In that regard, a framework can be researched and developed that will increase the efficiency and effectiveness of the data collection for the planning process.

A third stream of study can be on robust models that can take into consideration the forecast errors to ensure we can be prepared for the most probable scenarios.

Another possibility can be the planning framework. After careful consideration of the current literature and with the prospect that we can advance humanity's collective knowledge, a framework for a data-driven planning process can be created with the help of field specialists, addressing the most concerning points of each part, with the potential use of a set of technologies and methodologies, such as implementing advanced optimization techniques for dynamic production planning.

Within the potential architecture, this use case would involve the integration of various components in line with WP3. For any part of the production planning endeavor, curated data is of utmost importance. Data collection, storage, and a single source of truth are a necessity. For data collection and storage, several tools are needed, such as data lakes, API tools, IIoT capabilities for the machine's data, data streaming from on-site machines, and digital twins, among others. For a single source of truth, there is a need for a master data system that can bridge information from different systems, such as ERP and MES systems, but that can also help interpret data coming from machines.

For forecasted data, tools should exist that would inform the error of each of those forecasts once we have the real data to compare. With this, the performance of such forecasts can be verified on a cyclic basis.

For machine learning models to run, we need several MLops (Machine Learning Operation) tools; they combine elements from DevOps, data engineering, and machine learning to maintain the entire ML lifecycle, from data preparation and model development to deployment and monitoring.

Real-time data acquisition systems that monitor real-time inventory levels, logistics infrastructure, and machinery health may also be included. In addition, the key to both planning and resolution of issues would be machine learning algorithms for predictive analysis and decision-making.

4.3 Servitization and service delivery enabled by Zero administration

Servitization is an outcome-based business model where customers pay for services that require equipment constantly or intermittently on-site rather than buying the equipment themselves.

As a part of evaluation of common cyber architecture, the objective of this use case as a contribution to WP3 by LTU, HQ, SEB, and SA is to connect market actors involved in service delivery through data/event sharing and legally binding smart contracts. In each customer case, a smart contract will serve as a hub for data sharing and transparency, monitoring events, supporting equipment service activities, and initiating financial transactions in the context of outcome-based business models. Typical parties in a use case include at least one equipment manufacturer, several deployment and service partners, financial partners, and many end customers. We have outlined four key capabilities (Figure 16): 1) manage financial value & remaining lifetime, 2) improve uptime through predictive maintenance, 3) SLA monitoring through smart contracts, and 4) smart evaluation of error codes.

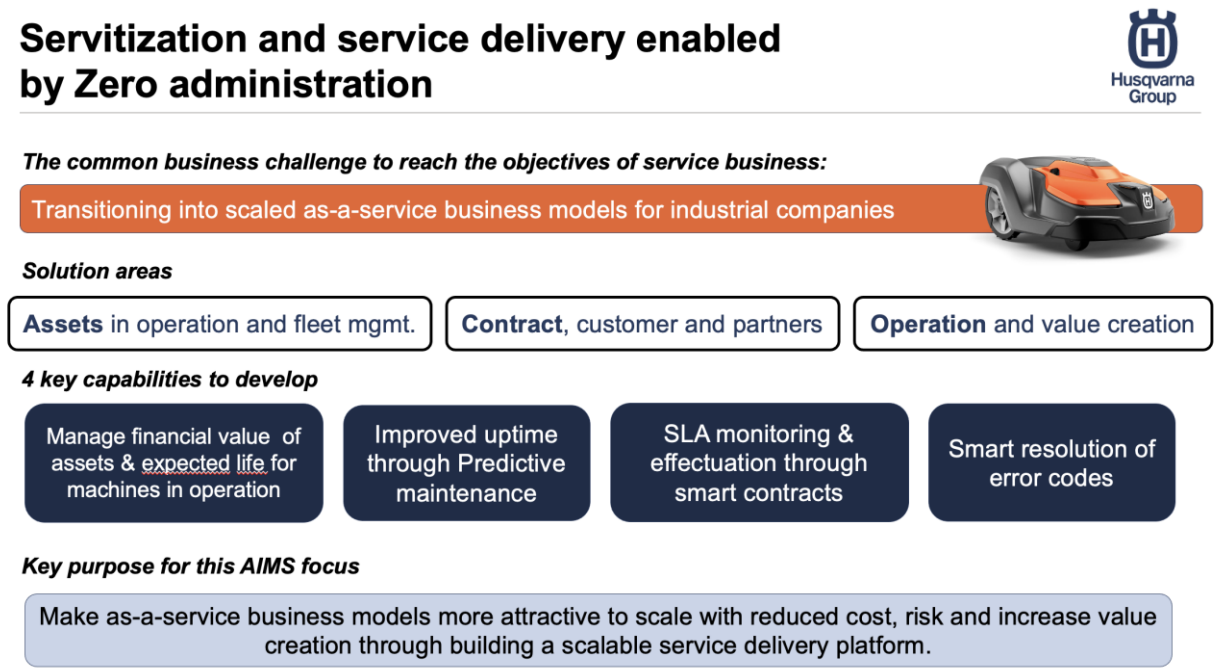


Figure 16: Capabilities outline

A target of our servitization platform is to support rapid cross-party workflows where parties interact back and forth, supported by automation to reduce administrative work. This is

enabled by working with streaming data throughout the full work process. The platform supports a range of use cases. Models include data-sharing portals, streaming analytics, mathematical-statistical functions, and Machine Learning models on-device and in the cloud. Different model categories can also be combined for more advanced use cases. The interaction between the involved parties has been outlined (Figure 17).

The prime use case (UC3) targets lawn mowing services carried out by autonomous robotic mowers. Initially, the target is large-size customers such as sports fields and commercial fields of various kinds, but it can later be extended to residential customers.

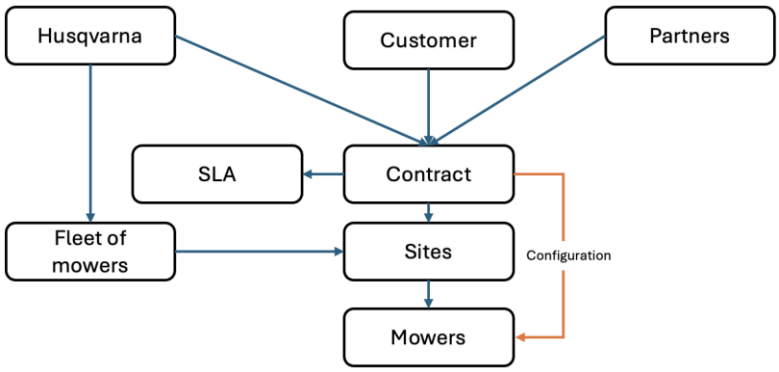


Figure 17: Interaction between servitization parties

A distributed platform for data storage, sharing, and event generation framework between parties will be developed. This also stores and executes the required smart contracts. To provide user interfaces for proof of concept, a standard front-end solution is integrated with the platform. The platform is designed to be easy to use for data scientists, engineers, domain experts, and other groups interested in developing edge AI solutions. The conceptual architecture can be seen in Figure 18.

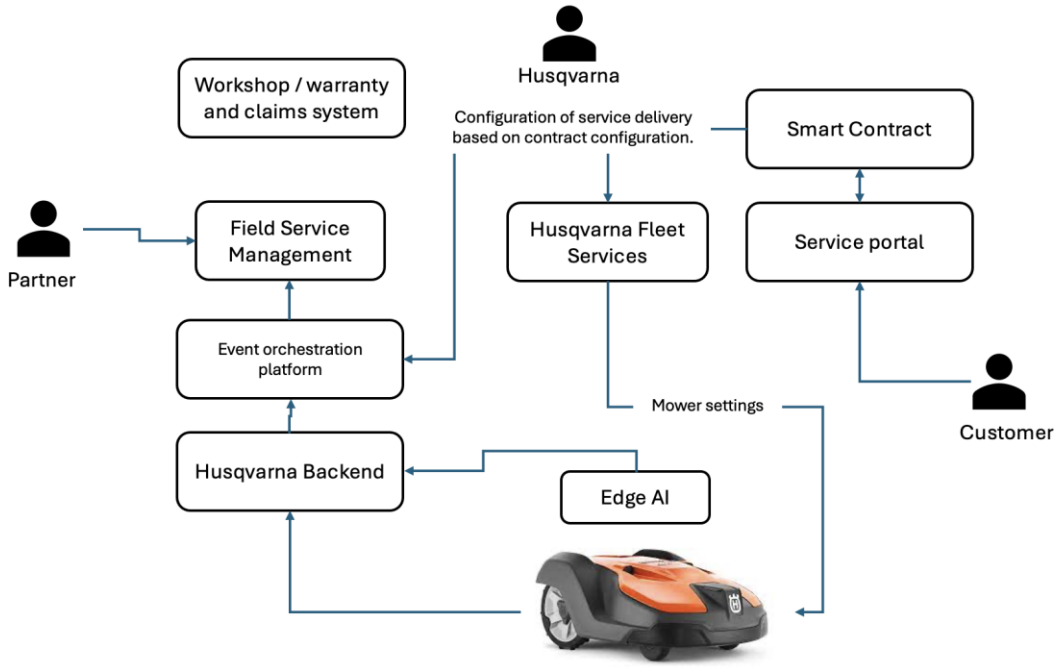


Figure 18: Conceptual architecture (technology agnostic)

The PoC implementation will adhere to the SOA and microservices paradigms using Arrowhead Eclipse arrowhead core systems and other best-of-breed open-source solutions for stream analysis, event handling, and smart contracts. There is a preliminary technology integration diagram (Figure 19)

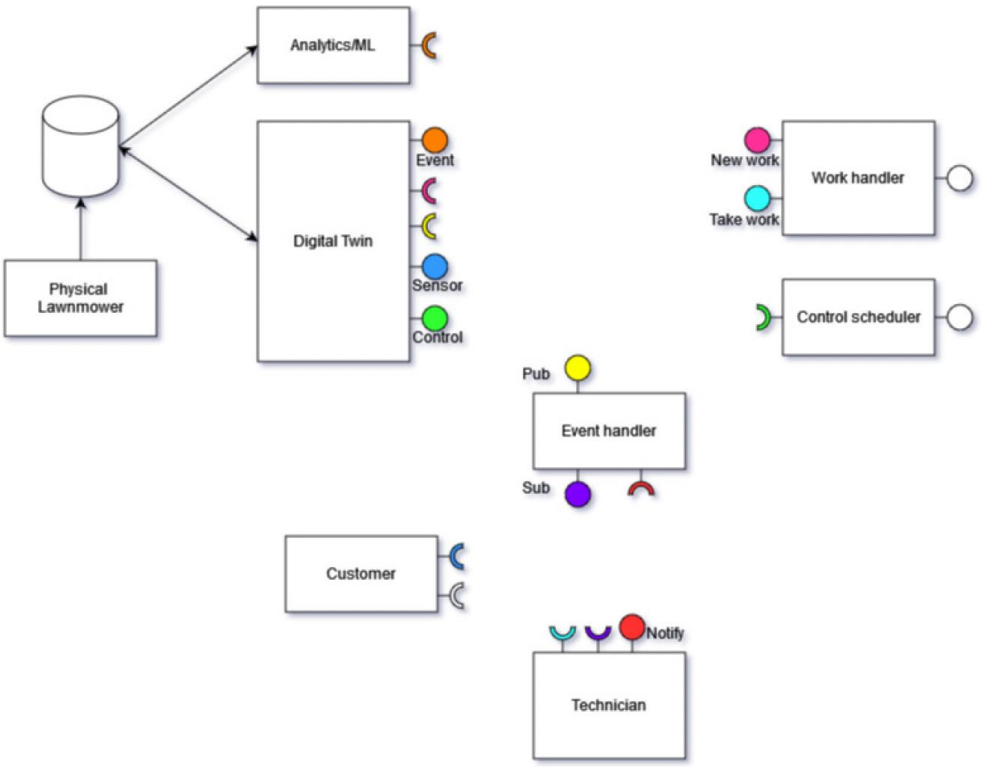


Figure 19: Preliminary Arrowhead-based architecture diagram

Partner responsibilities are as follows:

- HQ leads the use case with knowledge of state-of-the-art servitization and provides basic data sharing from experimental lawnmowers (physical and simulated).
- SEB provides financial expertise on the valuation of assets (mowers) as well as analyzing the revenue streams between HQ, the Service Providers, and customers based on lawn mower data in connection with smart contracts. The insights will contribute to supporting the as-a-service business model with regard to, e.g., financing and payments.
- LTU provides knowledge and solutions for the distributed framework for event sharing and smart contract execution and is also responsible for the proof-of-concept user interfaces.
- SA is responsible for the deployment of AI model framework(s) suitable for real-time on-device advanced analytics (streaming analytics) execution, as well as the development of on-device ML models.

Across the parties, we will provide an implementation of techniques for real-time data streams and modeling of results and visualization.

UC3 targets objectives in WP3 and WP5. The WP3 objectives are targeted as follows:

Scope (outcomes) and means of achievement

- To considerably reduce the administrative overhead in servitization by automatic event and data sharing supported by AI.
- To define a servitization architecture based on microservices systems of systems, using Eclipse arrowhead core systems and other best-of-breed open-source solutions for stream analysis, event handling, and smart contracts.
- To define a Proof-of-concept of the SoS-based architectures where the implementation can contribute to Eclipse Arrowhead

Evaluation and Success metrics

- Validation and verification of administrative overhead.
- Provision of written contributions and publications (both Master's theses and peer-reviewed papers).
- Documentation using the Eclipse Arrowhead documentation model.
- Reference implementation including, SW libraries and training material.

4.4 AI techniques at different layers in Machine Tool Domain

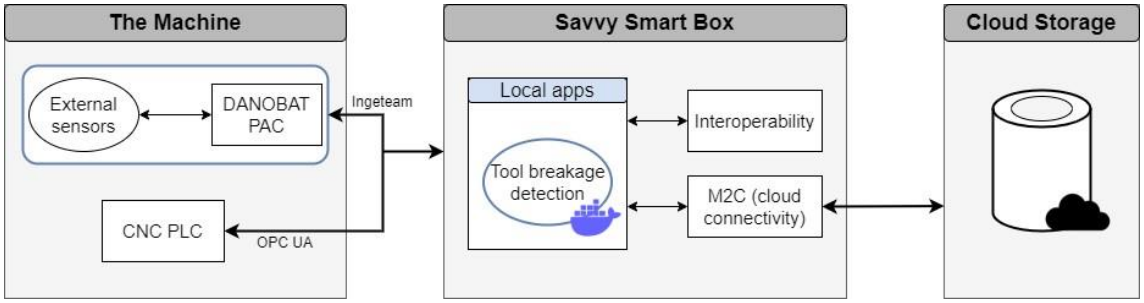


Figure 20: General architecture

IDEKO, SORALUCE, GOIMEK, and SAVVY are major contributors in task T3.2, which includes use-case UC5 as a particular implementation of a microservice-based platform enabling AI integration at the edge level. This serves as particular evaluation case for the research objectives and results in T3.2.

Objectives

1. Research in digital architectures, e.g Docker and DevOps of industrial software.
2. Acquisition, reading, processing, and extraction of indicators on the edge (from the sensors installed on the machine).
3. Creation of models for the machines, and their processes, oriented to the quality of the part, the state of health of the machine and its critical components, tool life and energy efficiency.
4. Develop AI models for automatic tool breakage detection.
5. To have an architecture for the deployment of AI models on the edge.

Architecture

- External sensors: Data collection from external sensors and virtual sensors.
- Danobat PAC: This is an Ingeteam controller that enables high-frequency data collection. It allows to run algorithms in Matlab, thus avoiding sending the raw data to the cloud if it is not necessary. It can also generate CSV with the raw data and send it to an FTP server.
- CNC PLC: OPC UA is used to connect to the CNC if available.
- Savvy Smart box: It is an IoT gateway capable of connecting to different sources such as CNC/PLC machines and controllers. It is the device that is responsible for data collection and transmission in a secure way. This data can be stored locally on the device itself or in the cloud. It contains a set of connectivity libraries that allow data collection using different source protocols. Within the device, we have the following services enabled for this use case:
    - Connectivity for automation: these are the libraries that connect to the different sources of data.
    - Dockers: space to deploy applications via containers.
    - Cloud connectivity: Internal protocol to connect the device to the cloud and enable secure information exchange.
- Cloud: Own cloud that allows data storage for access from the outside.

The proposed architecture is configured as an integral system to address the different challenges of the industrial environment, prioritizing research in digital technologies such as Autonomous Cyber-Physical Systems, Docker, and DevOps in the field of industrial software.

One of the main objectives of this architecture is data management at the edge, with a clear focus on acquiring, processing, and extracting indicators from both physical and virtual sensors. This approach provides critical information related to part quality, the state of health of the machine and its critical components, tool life, and energy efficiency. Another main objective is to develop and deploy models on the edge for automatic tool breakage detection.

The Danobat PAC, an Ingeteam controller, is responsible for high-frequency data collection, as shown in Figure 20. This component offers the flexibility to run algorithms in Matlab, allowing local processing to be performed before sending the data to the cloud. It also has the ability to send data directly to the cloud and can generate CSV files with raw data for subsequent transmission to an FTP server.

Regarding the connection with the machines, the use of standards such as OPC UA is prioritized, while the interoperability module guarantees the connection to legacy machines, ensuring effective integration with CNC-s from any manufacturer, allowing the acquisition of essential data for monitoring and control of operations.

Savvy Smart Box is a physical device that contains a system for sensor data collection and analysis, in order to capture and process all the information generated by industrial processes, that is, work with the information generated by the machines and automated devices. It is designed to facilitate the connection of machines to the cloud storage infrastructure. This IoT gateway establishes a secure and scalable possibility to connect machines and other equipment to the monitoring platform (it provides connectivity of machines and devices). It sends the information securely to the private segment assigned to each customer. In addition, customized applications can be deployed on it. This device is where the data lifecycle starts. It is possible to access the machine data through the CNC

libraries. It is prepared to work with high-frequency data, as well as data with less demanding frequencies. This device has the versatility of storing data locally or in the cloud, providing flexibility in information management.

Cloud connectivity is established by means of an internal protocol, guaranteeing security in the exchange of information between the device and the cloud. The cloud provides a storage environment designed to facilitate access to data from external locations, closing the information management cycle in this industrial architecture.

These components together provide a complete and robust framework to address the specific challenges of the industrial environment, from data acquisition, and analysis at the edge to its storage, processing and availability in the cloud.

Link with the use case

Automatic tool breakage detection in milling operations is a problem of interest in the manufacturing industry, as tool breakage can significantly affect product quality and increase production costs. Milling processes can be long-term processes in which the machine works unattended and can run for hours without human intervention. However, there is the possibility of errors occurring during this milling process. These errors can cause failures in the final part, which can lead to re-machining of the part or, in the worst case, rejection of the machined part.

Therefore, to solve this problem, an artificial intelligence system capable of detecting early tool breakage would be very beneficial in reducing production costs and avoiding defective parts and wasted time. This tool breakage detection model would be deployed in the Savvy Smart Box edge device (local apps) via containers.

To solve this problem and to find out the behavior of the data, some tests were carried out under experimental conditions on a milling machine in the IDEKO laboratory (with known cutting conditions and with a known tool). The AI models fitted to this data were able to detect the tests carried out with broken tools.

In real production, the vibration data generated by the accelerometers installed in the machine are collected, the vibration signal is processed, the characteristics are extracted, and these indicators are sent to the cloud. However, when the developed models are taken to real production, the results obtained are not so good. The models are not able to detect tool breakages.

4.5 Automated Architecture Compliance Assessment

AUSTRIAN INSTITUTE OF TECHNOLOGY (AIT) plans to assess the architectures and developments in WP3 with a focus on standards and regulations and consider the new Cyber Resilience Act (CRA). We utilize our advanced model-based security analysis tool, ThreatGet (https://www.threatget.com/), which identifies potential security issues based on a database that is currently manually filled with known threats and attack path synergies. The tool and its current workings are shown in Figure 21.

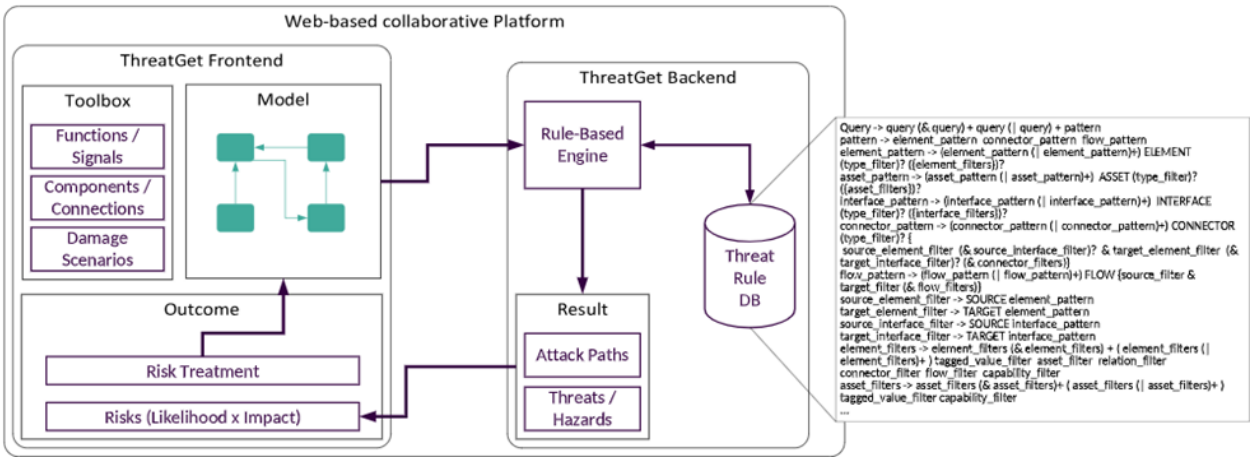


Figure 21: ThreatGet Architecture

By enhancing this manual process with a Natural Language Processing (NLP)-enhanced algorithm, we aim to process public threat intelligence available in unstructured text formats, such as those found in CVE (Common Vulnerabilities and Exposures) and NVD (National Vulnerability Database) listings. This process involves extracting structured rules in a Domain Specific Language (DSL), thereby enabling the automated evaluation of system vulnerabilities, assessing their relevance, and determining their potential impacts on the targeted industrial and production systems. The envisioned process and AI enhancements are shown in Figure 22.

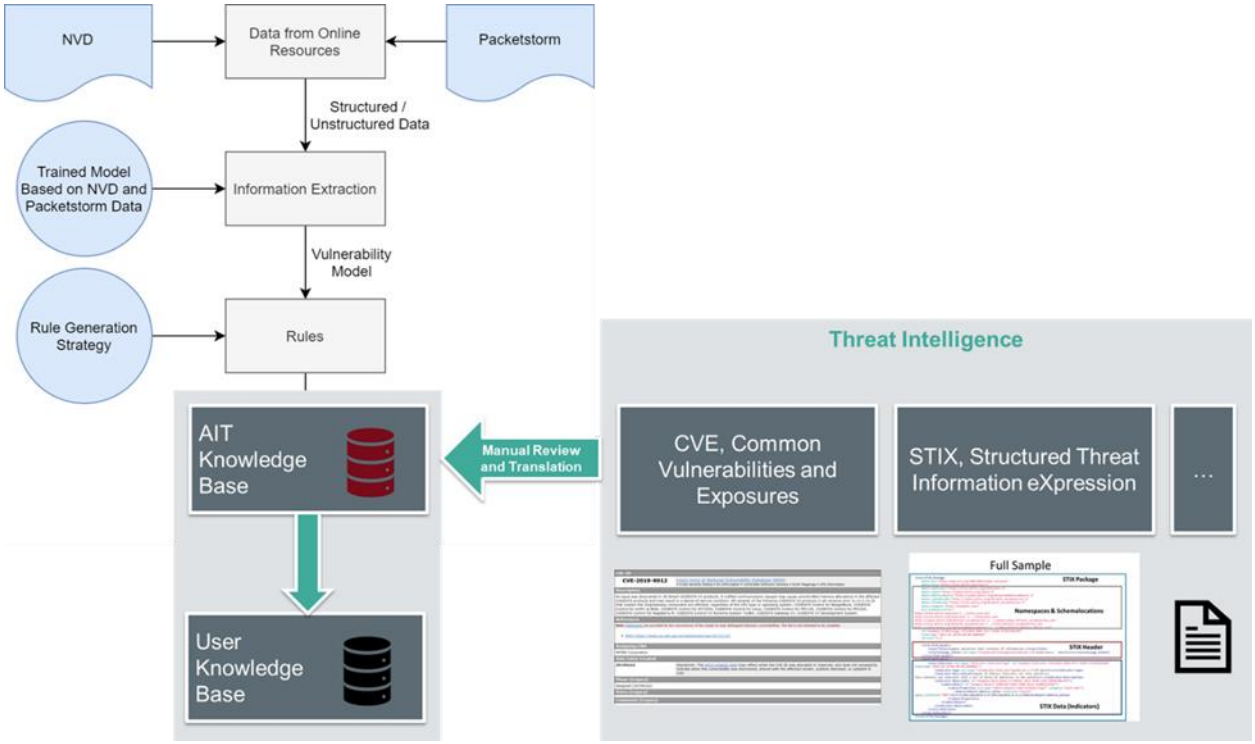


Figure 22: ThreatGet AI-enhanced process

To align our approach with the CRA, our immediate focus was on identifying the key CRA requirements that our methodology must satisfy. This involves a detailed analysis to ensure our approach meets the regulatory expectations for security in digital and connected ecosystems.

In addition, we prepared an initial threat database, which will act as a repository of known vulnerabilities and associated threat vectors relevant to the domains on which UC11 “AI-supported Industrial IoT for Indoor Food Production” is built. Concurrently, we are dedicated to the development of a security model for the use case. This model will serve as the foundation for our efforts.

By leveraging AI to augment our threat intelligence capabilities, we ensure that our database remains current and comprehensive, reflecting the ever-evolving landscape of cybersecurity threats. With this, we can re-evaluate the security at a later time and address the risk assessment and vulnerability monitoring aspect of the CRA.

This aspect also represents a first preparatory step towards the assessment of architectures, which is planned in T3.4.

4.6 Architecture for MLOps solutions

AI integration and MLOps are central research question related to the common cyber architecture for AI-enabled CPSoS built on SoA principles. DAC works focused on preparing the architecture of the MLOps solution, encompassing tools for both the development and production of ML algorithms. Also, in the first year of the project, the first implementation of Machine Unlearning was done, which was taken into account as the next step for integration in the prepared MLOps solution.

The components of the envisioned MLOps platform are shown in Figure 23. The components or entire toolchains are envisioned to be accessible through an API, allowing for further integration with other services in, for instance, the local cloud.

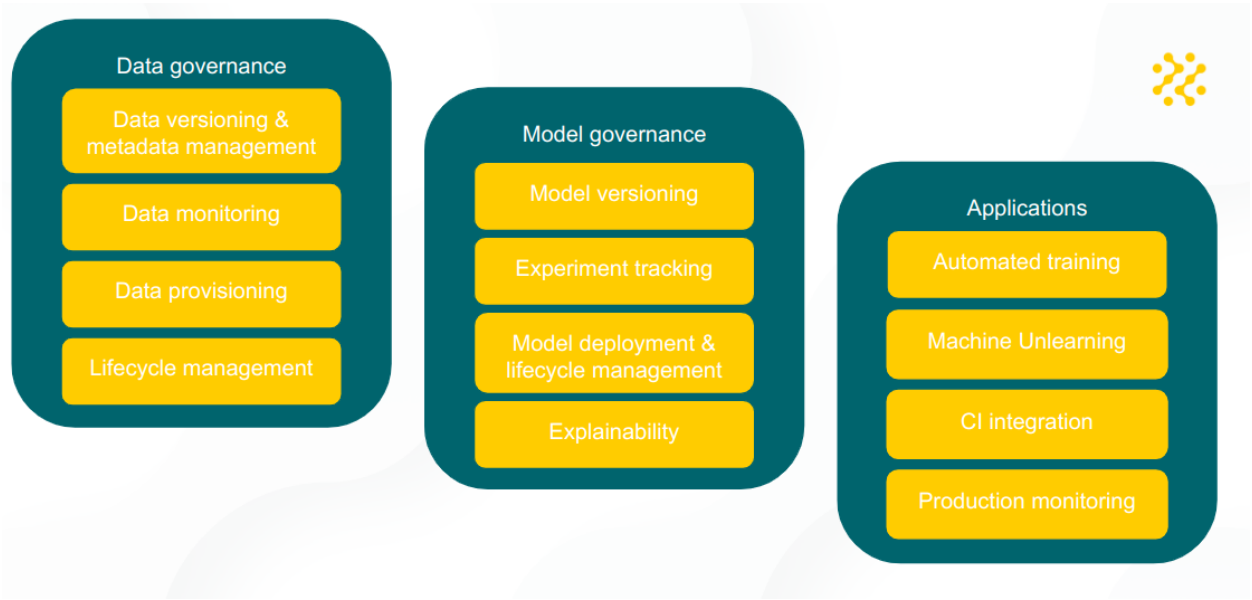


Figure 23: MLOps platform components

There are three main blocks, which interact with each other as a part of the MLOps framework. These blocks and their components can be arranged to have either basic functionalities, such as tracking experiments at the development stage or versioning the subsequent versions of the datasets, or more sophisticated, such as automated model training and deployment or data shift detection and model redeployment at the production stage. Below, these two pipelines are described.

Pipeline 1: Automated Model Development with Managed Workflows and Retraining Using New Data

In this scenario, the MLOps framework orchestrates an automated model development process with managed workflows and retraining capabilities using new data. Data governance components, such as data versioning and metadata management, ensure the availability and quality of datasets for training. Experiment tracking within model governance captures the details of model training experiments, including the datasets used and the resulting performance metrics. Automated training, facilitated by applications, leverages these experiment records to iteratively train models with various datasets and hyperparameters, ensuring reproducibility and scalability. As new data becomes available, data monitoring components detect its presence and notify the system. This triggers the provisioning of the new data, managed through data governance mechanisms, for model retraining. The automated training then utilizes the updated datasets to retrain models, ensuring they remain up-to-date and accurate.

Pipeline 2: Production Monitoring for Detecting Data Shift and Acting

In this use case, the MLOps framework focuses on production monitoring to detect data shifts and maintain model performance. Data monitoring, a key component of data governance, continuously observes production data for any deviations from expected patterns or anomalies. Model governance oversees the deployment and lifecycle management of models in production environments. Applications, specifically production monitoring tools, leverage data monitoring alerts to trigger the monitoring of deployed models for performance degradation or drift. Explainability, provided by model governance, provides a layer to the user that helps in understanding model behavior and identifying potential issues related to data shifts. Integration with CI is an enabler for automated testing of deployed models against production/reference data to detect drift or performance issues. Data provisioning and lifecycle management ensure the availability and quality of production data for monitoring purposes, completing the loop in detecting and mitigating data shifts in the production environment. On the other hand, data monitoring can be used to continuously collect new data coming directly from the target environments.

In the next part of the project, subsequent blocks will be further implemented, and integration with the execution environments and external frameworks will take place.

4.7 Deployment of new AI approaches/services on the edge

As a part of evaluating of the research results in WP3, CISC will work on the deployment of new AI approaches/services on the edge and cloud components to provide pre-processing of sensitive sensor data.

In manufacturing processes to track products reliably over the RF (radio frequency) channel, the concurrent use of the radio spectrum by multiple users without detrimental mutual interference must be guaranteed. CISC addresses interference and throughput issues with an emphasis on NFC (Near-field communication) and RFID (**R**adio **F**requency **ID**entification) technology. CISC is an established provider of measurement and verification tools for the evaluation of emerging communication protocols and RF channels. This comprises simulation/emulation of the communication between devices and the development of test scenarios and test cases. In the scope of this project, AI-enabled techniques are analyzed and developed to predict potential interoperability issues with physical devices (products).

UHF RFID devices operate in an unlicensed ISM band. They have to follow regulatory requirements regarding frequency, power, and timing. The number of ETSI channels in the lower band is fifteen, but it is effectively limited to four by the four-channel plan. The FCC band of 902 MHz – 928 MHz is divided into 52 channels. With the increasing use of RFID globally, the number of devices accessing wireless media increases significantly as 28 billion RAIN tag chips have been sold in 2021, exceeding 100 billion pcs sold globally since the technology exists (source: RAIN Alliance Market Report 2023).

RFID-enabled operations in industry, healthcare, retail, and other applications rely on efficient utilization of the frequency channels to guarantee reliability and speed. In road-tolling applications, for instance, vehicles need to be identified within milliseconds to allow for smooth traffic. The application suffers a lot under tags from different applications. Any delay in the execution of data transactions between the reader and tags would require introducing speed limits at the tolling plaza, causing traffic jams and a consequent reduction in driving safety.

Improper operation of a few readers or interfering signals from other sources jams frequency channels and causes retransmissions of data over the RF link, slowing down the data exchange and read rates. The RAIN RFID protocols offer mechanisms to adjust the parameters of transactions to optimize performance. Applications with isolated fast-moving objects operate readers with different settings than applications with large volumes of static tagged items. The communication protocol allows for adaptively adjusting the parameters of the data exchange. Decisions for the adjustment are based on the number of collisions of packets that are sent during the inventory.

The ability to predict disturbing patterns would increase the efficiency of applying proper techniques for adjustment of the communication protocol parameters, extracting signals from the noise, and increasing the throughput of the system.

The CISC reader tester is based on a fully configurable Tag Emulator that allows measurement of reader transmit power and the reader receiver sensitivity through variable backscatter strength. Furthermore, the Tag Emulator allows testing the reader’s behavior in dependence on timing variations like BLF (Backscatter frequency) tolerance. CISC Sniffer is a tool that analyzes RFID readers and tag signals. Triggering on various events was implemented to record and log the communication between the reader and tags. Sniffer provides up to 100 MB RF sampled data per second that will be used in the training phase of the AI system. Furthermore, devices will be challenged using the emulation platform in real time. Communication protocols and specific algorithms for real-time handshaking with the DUT (device under test) are to be developed, and measurements and tests will be carried out. The results will be provided as an input for standardization. In particular, the goal is to identify other spectrum users and in order to use AI to define how to best configure readers including timing for best coexistence.

Furthermore, CISC is focusing on applying AI techniques to improve the reliability of the production of RFID systems by detecting anomalies in production. This covers early detection of production issues by analyzing parameters from both the measured performance of UUT and the parameters of production lines. The goal is to use RAIN/NFC Reader to monitor and predict potential production issues and provide early warning before failing tags get produced.

Methods:

- Collect information about devices produced in the line, in particular performance parameters.

- Collect information from available sensors in the production line (cameras for die attachment monitoring, temperature sensors, etc.)
- Apply AI methods on the collected data to evaluate probability of product quality-check failure (federate learning).

The outcome of this approach is to reduce waste (IC/silicone; antenna/metal; label/plastics) by

1. improvement of the production process setup and
2. reduction of number of devices not passing the QC.

4.8 Micro service based platform enabling AI integration at edge level

4.8.1 Detailed description of the work plan

The digital and innovative methods of Industry 5.0 are creating new digital opportunities to increase the productivity, flexibility, quality and safety of process chains and products. In order to utilise this new potential, for example in the areas of digital planning processes or machine learning and smart data analytics, the BMW Group is pursuing the goal of generating a full digital twin. This enables the virtual planning and simulation of all processes required for the production system even before the first production step. This makes it possible, for example, to minimise downtimes during commissioning and reconfiguration and to check compliance with quality parameters or the planning and adherence to production sequences. In the long term, this leads to an optimisation of the process chains, saves material costs, reduces maintenance costs and increases sustainability. In the long term, a full digital twin therefore offers an indispensable tool for the development and utilisation of cyber-physical systems and the creation of unprecedented transparency in complex process chains.

4.8.2 Research outline

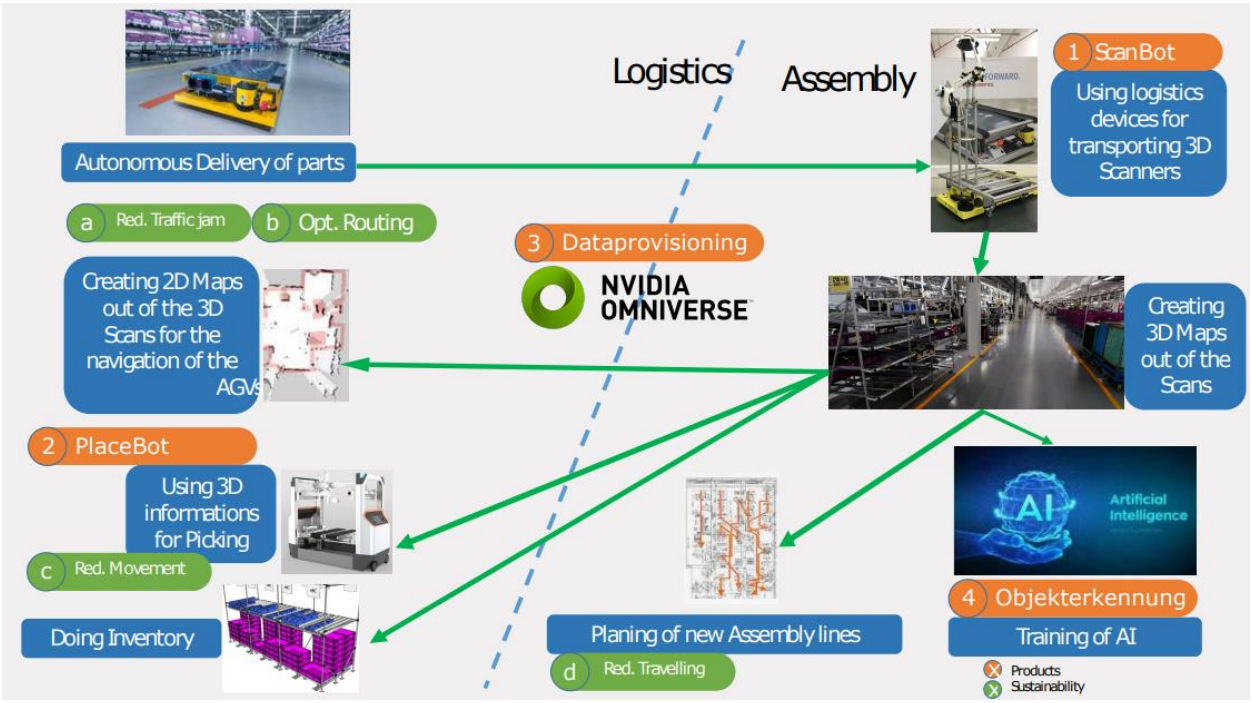


Figure 24: Research topics

Access to continuously updated 3D data, which is to be generated using an autonomous scanning robot (1), creates the basis for the rapid commissioning of new transport and material handling systems, such as the PlaceBot (2), as well as a continuous inventory in logistics. On the assembly side, planning processes can be organized more efficiently. A decisive advantage on both sides is the use of scan data to create a digital representation of our production facilities in Omniverse (3), which provides the absolutely necessary data basis for full digital twin models and countless other use cases in the context of Industry 5.0, such as the possibility of using scan data to simulate all physical properties of the real world and thus create an optimal training environment for artificial intelligence (4) based on BMW's requirements.
The findings from the research work within WP5 are also to be incorporated into work packages 3, 7 and 8 and thus contribute to enabling efficient (as defined in the individual use cases) AI-supported and sustainable production automation on the one hand, and to the development of personas and roles in coordination with the use cases on the other. In addition, support with market-oriented measures or the introduction of solutions to optimize business processes (e.g. digitalization of factories, AI-supported production).

4.8.3 Objectives

The main goal of this task is to enable efficient (as defined in the individual use cases) AI-supported and sustainable production automation by extending and maturing the existing edge-level SOA and event-driven interaction models implemented with the Arrowhead Framework. In particular, the extensions will address the requirements for the properties of AI-based functionalities as well as the requirements for the properties of flexibility, scalability, security, and safety associated with such functionalities.
BMW Group AG is participating in this work package by cooperating with the WP lead and applying the results obtained in the BMW production environment. It also conducted an analysis of BMW AI hardware in order to analyse the market with regard to AI-based applications in the context of edge hardware.

4.8.4 Initial results

As our work in this area is still ongoing, we currently do not have sufficient results to document. However, we are continuing our efforts and are still working on it.

4.9 Collecting requirements and relation to the common architecture

FMX collects and processes the requirements within WP3 harmonizing with the research objectives of the work package. The currently identified requirements are presented in the following two sections, regarding the Use Cases 13 and 14.

4.9.1 Software and hardware realted requirements

In Use Case 13, FMX is advancing the frontier of intralogistics automation systems through the integration of AI-enabled software and hardware. The following figure (Figure 25) provides a comprehensive overview of the key concepts and strategies driving this initiative.

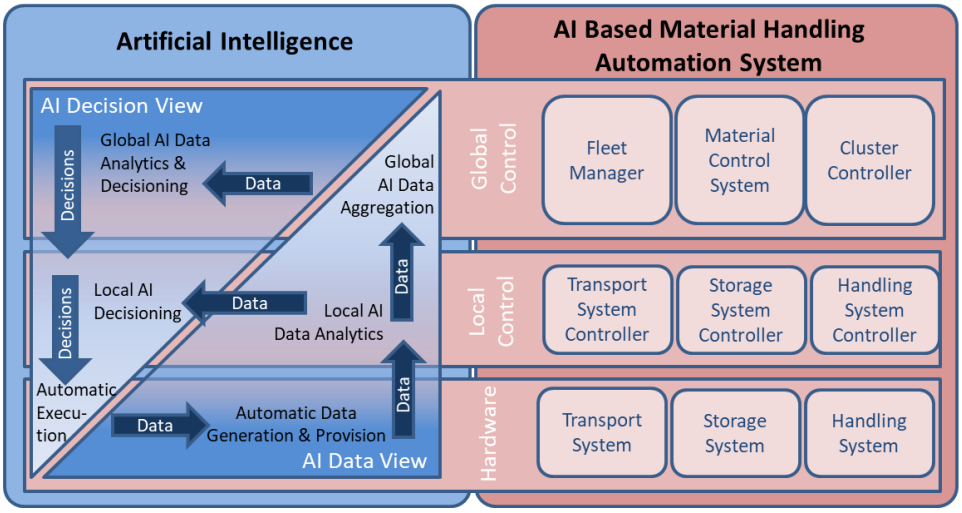


Figure 25: Basic architecture.

It illustrates that the new generation of material handling automation systems, which are developed in this project, consists of three levels:

- hardware (lowest level in the diagram above), which basically creates and provides the necessary data for AI based control
- local control software (medium level above), which analyzes and locally aggregates data coming from hardware, potentially makes local decisions and provides it to the high-level software
- global control software (highest level), which further aggregates data provided by the local control level and does global data analytics and decisioning using AI

In other words, the lower two levels describe AI-enabled material handling automation systems which in the highest level can be controlled by AI based global control software like Cluster Controller, Material Control System and/or Fleet Management System.

In pursuit of this objective to develop AI-enabled and controlled automation systems, FMX has outlined a set of critical requirements for AI-enabled intralogistics automation systems within WP 3:

- Real-time Data Acquisition
- Online and Offline Data Storage
- Data Accessibility
- Decision Making Capability

Real-time Data Acquisition entails the seamless collection of data in real-time, including for example sensor information. This data acquisition process involves both local controllers within specific subsystems and global controllers that oversee the entire factory operation.

Online storage of data ensures immediate accessibility for controllers to make real-time decisions, while ***offline storage*** facilitates deeper analyses and long-term planning. All controllers must have unhindered access to the stored data.

This ***data accessibility*** ensures that decision-making processes across the intralogistics system are informed by the most up-to-date and comprehensive information available.

Furthermore, controllers of intralogistics automation systems must possess the ***capability to make informed decisions*** based on the data they have access to. This involves leveraging AI

algorithms and other decision-making frameworks to optimize intralogistics operations in real-time.

The integrity and availability of consistent and valid data serve as the bedrock for the efficiency and effectiveness of all subsequent processes within the intralogistics automation system. The envisioned architecture allows running controllers to tap into online data storage at any given moment, empowering them to execute real-time decisions with precision and efficiency. Furthermore, the inclusion of offline stored data enables in-depth analyses and strategic planning that can shape the trajectory of the intralogistics system over the long term.

Currently, these requirements are undergoing a meticulous translation into a specific course of action tailored for implementation within Use Case 13. This course of action will be seamlessly integrated into all material storage and handling systems developed under the project, ensuring the realization of advanced AI-enabled intralogistics automation solutions.

4.9.2 Requirements for Digital Twin systems

Use Case 14 revolves around expediting the advantages of automation via virtual commissioning leveraging digital twins. The focus lies on pinpointing commissioning tasks for (partial) virtualization. Rather than rely to the time-sensitive nature of on-site commissioning, the idea is that these tasks undergo a preliminary virtual commissioning phase earlier in the automation project and are, therefore, effectively removed from the time-critical path. To qualify for virtualization, a task must meet four essential criteria:

- Virtualizability
- Applicability
- Adjustability
- Effectiveness

Virtualizability identifies the task's potential to be transitioned into a virtual environment in general, rendering on-site execution unnecessary.

Applicability ensures that outcomes from the virtual commissioning phase seamlessly integrate into the real-world automation project.

Adjustability entails the flexibility of virtual commissioning results to accommodate slight deviations in real-world conditions, offering a tolerance for digital twins to slightly differ from real-world scenarios.

Lastly, **Effectiveness** measures the benefits of virtualizing a commissioning task, quantifying the cost or time savings.

Regarding these general requirements, the partners of Use Case 14 are currently breaking them down into smaller and task-specific requirements. Furthermore, based on the described requirements, the partners are currently identifying specific tasks that align with these requirements, ensuring their viability for integration into virtual commissioning workflows. For instance, tasks such as roadmapping, scan field configuration, or handling robot trajectory planning are candidates for virtualization due to their inherently digital nature and minimal dependence on physical presence.

Details of the requirements and specifications of the envisioned solution are described in Deliverable 6.1.

4.10 ML algorithms for workflow optimization

Since AI-enabled CPSoS are major drivers in WP3 composability researches, Signify is developing, with its partners in WP5 (UC7) and WP2, ML algorithms for workflow optimization of luminaire manufacturing. This includes, e.g., the development of smart and intelligent schedulers/planner algorithms for our 3D printing factories (together with partner TU/e), Digital Twins of assembly flows/assembly layouts (generic, multiple factories), and the use of Computer Vision for luminaire assembly execution tracking and Operator Guidance (Augmented Worker/Future of work).

Algorithms are co-developed and shared with the WP2 partners and Signify makes use of the ML library that is provided by WP2.

The algorithms will be validated (in year 2/3) in Signify's MES/ERP platforms in dedicated selected factories in Europe. That is a prerequisite for an industrial company like ours. A first selection of potentially receiving factories has been made, and the first experiments at these locations are already ongoing prior to later transfer and launch.

As ML algorithms need to be reliably operationalized, validated, maintained, and widely deployed, there is the option to apply ML Ops. This method can be used to bridge our work in WP2, and Eclipse/Arrowhead might be an intermediate isolated test platform for us. This might, in particular, hold for alternative (own) algorithms developed by our academic partner TU/e that might require such an isolated environment (to be confirmed later). Also, it might allow the use/re-use of the developed generic ML methods by the other AIMS5.0 partners.

4.11 Gate **Boarding of Sensors**

An evaluation task for the common cyber architeture defined in WP3 is Task 6.3.4 (Use Case 09, 10, 13, 17, 18, 19, 20) out of WP6, of which main scope is using sensor systems for advanced semiconductor manufacturing – Smart Sensor Platform by UzL, IFD, IFAT, IFAG, HSZG, TUD, SYSTEMA is to develop smart sensor platforms for advanced semiconductor manufacturing using the MES, machine learning, and AI.

For most of the use cases within WP6, sensors are used, which deliver the data for manufacturing, which is the core for machine learning and AI applications. A main task is to find devices via Arrowhead, e.g., IO-link tool, in case of unknown IP addresses on the web. Partner like Infineon within WP6 are searching tools which enables to find a new IP address (via set IP with DCP). The challenge is to get an industrial standard to set up an IP address. The problem is that each semiconductor equipment vendor has its own interface, and many vendors are supporting DCP (e.g., Pepper & Fuchs or IFM), which often comes from profit (this is a low-level protocol). Therefore, the goal is to define architectures in each equipment that gives a signal automatically that the device is on (on a secure level, using certificates). Furthermore, we will avoid to build own tools (sensor gateways could be built via Arrowhead). One idea for a simple solution will be the change of IP addresses via RES interfaces (but the device must be found among many other devices on the net). A simple tool will be created for RES, which can be used as a new Arrowhead device.

The partners out of the mentioned use cases work together with the WP3 team and mainly LTU on an architecture for onboarding the sensors. This architecture will be established and presented within year 3 of the project.

4.12 Improved security and reliability measures in communication systems

As part of its objectives, GUT (Gdańsk University of Technology) is committed to researching parts of the common cyber architecture - and as a part of its evaluating in industrial environment - refining various mechanisms essential for the effective management and control of jamming and interference mitigation techniques. In addition, GUT strives to improve security and reliability measures in communication systems. The key focus is the development of rapid and efficient methods for alerting and responding to emerging threats to maintain secure communication. GUT aims to improve the resilience and efficiency of communication networks, ensuring robustness in the face of evolving challenges.

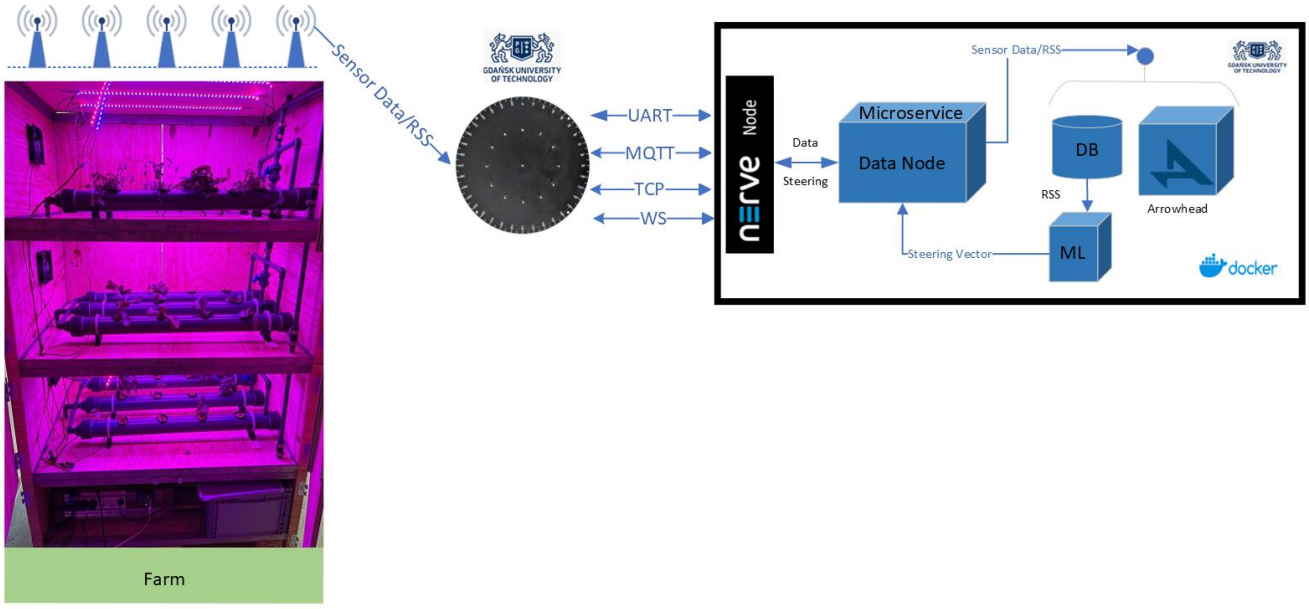


Figure 26: Conceptual architecture with relation to other system

As part of Work Package 1 (WP1), GUT intends to collect sensor data provided by collaborators, together with Received Signal Strength (RSS). Communication will be maintained using the beamforming capabilities of an intelligent antenna. GUT will explore the full range of 4096 switching configurations to determine the optimal setup, taking into account environmental factors and sensor placement. The integrated custom processing unit, coupled with the ESPAR antenna, enables the propagation of collected sensor data and RSS through various interfaces such as UART, MQTT, WS and TCP. In addition, a dedicated edge device called "Nerve" will receive this data and distribute measurements to specific microservices as required.

The ML microservice will take on the task of analysing communication characteristics using Received Signal Strength (RSS) data specific to the configuration of the ESPAR antenna. Through internal algorithmic calculations, it will determine the optimal steering vector for the antenna, thereby improving connectivity. In addition, the incoming data will be both locally stored and shared via the Arrowhead platform in accordance with the objectives outlined in Work Package 2 (WP2).

The outlined solutions will hold the concept of microservices, which will be containerized making it reusable and scalable.

4.13 Autonomous, reactive and teachable surveillance network (system of systems)

Since AITIA leads tasks T3.1 and T3.2, moreover subtasks ST3.2.1, AITIA has an extensive organizational role while also being a technical contributor to the work package. Moreover, AITIA is the leading responsible partner for Deliverable D3.1, but also delivers technical contribtuion as it presented below.

The industrial system Sentinel, developed by AITIA in collaboration with BME, is about to automatically process surveillance camera images and autonomously trigger customized actions. This use case presents how the execution of this complex task can be achieved by harnessing Service Oriented Architecture (SOA) capabilities.

4.13.1 Objectives

The main goal is to develop a surveillance solution that does not require active human supervision, enables automated interventions and targeting a wide range of application areas, such as:

- **Automated product quality assurance**
  Comparing a product image to a reference image or recognize typical damages and deformations.

- **Reactive security supervision**
  Processing images about an area where no significant difference is expected compared to a reference image. (E.g.: Warehouse outside of business hours).

- **Reactive restricted area monitoring**
  Detecting unexpected objects (human, forklift, etc…) within an area.

- **Automated quantitative reporting**
  Counting target objects (pests on plantations, trucks on roads, etc…) within an area.

4.13.2 Architecture

Eclipse Arrowhead Framework has been chosen to meet and take advantage of the design patterns of Service Oriented Architecture. The components (nodes) of the Sentinel Surveillance Network are represented as service providers and service consumers belonging to the same Arrowhead Local Cloud. There are six types of application systems:

- **Sentinel Nodes**
  Responsible for taking photos at the given locations.

- **Image Nodes**
  Responsible for evaluating the photos (recognizing objects and/or comparing photos).

- **Coordinator Nodes**
  Responsible for controlling the photo processing and action triggering.

- **Action Nodes**
  Responsible for executing additional business logic (interventions).

- **Storage Nodes**
  Responsible for storing photos, metadata and processing results.

- **Monitor Nodes**
  Responsible for ensuring the human GUI (editing configurations, receiving alerts).

Each node type can be present in several instances within the sentinel network, but a typical setup is described in Figure 27.

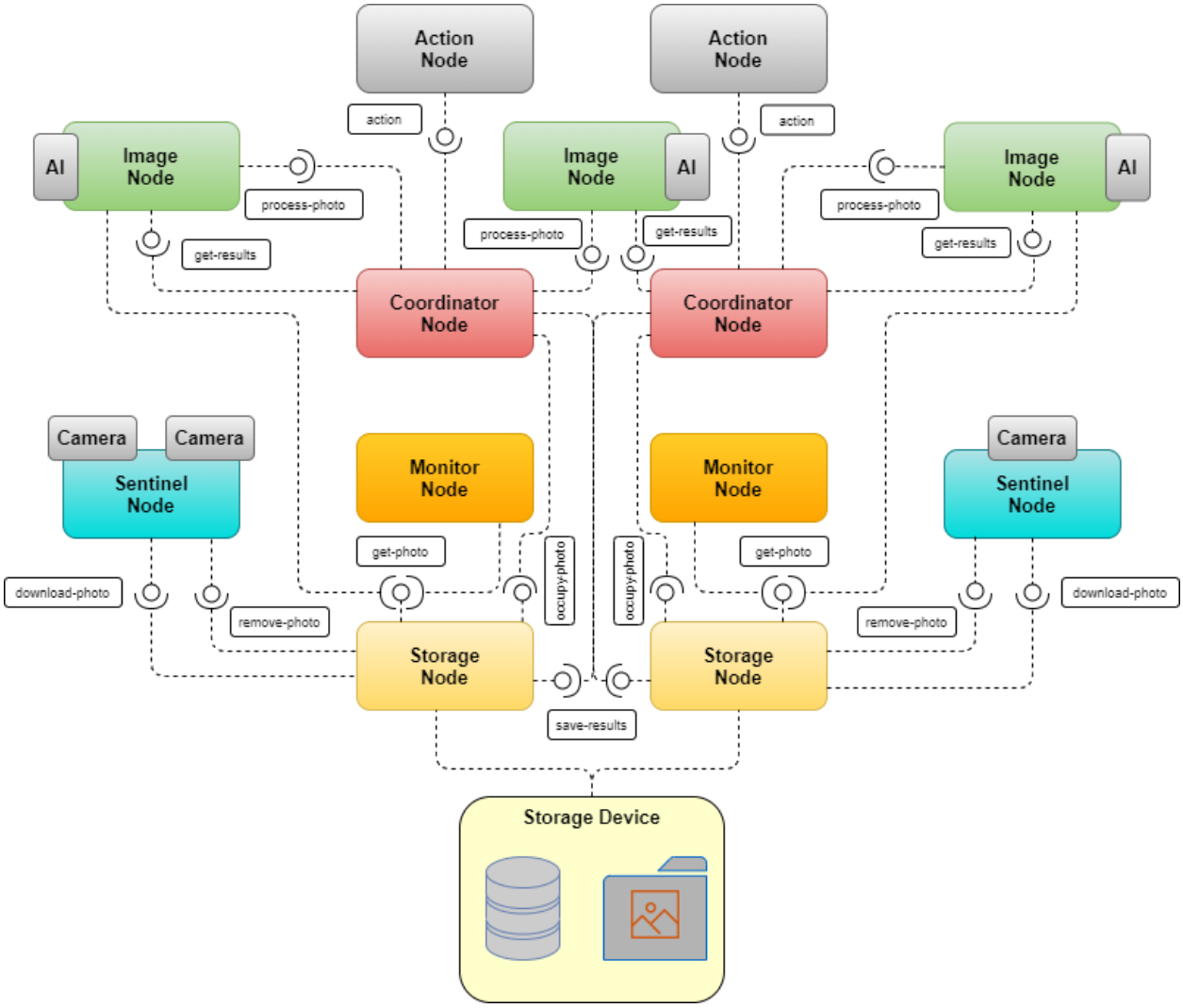


Figure 27: Architecture of Sentinel system

- **Sentinel Node**
  Its most important task is to integrate the surveillance cameras into the network. The various devices are using vendor-specific communication solutions. Therefore, a common interface has been introduced to make device integration possible. At the moment, only the ISAPI protocol is implemented.

- **Image Node**
  There are two types of tasks an image node can perform: object recognition and image comparison. Object recognition is implemented by using the YOLO machine learning method. Currently, two models are being used, one for humans and one for forklift recognition. Image comparison is implemented by using two algorithms based on Structural Similarity Index (SSI) and Mean Square Error (MSE).

- **Coordinator Node**
  Its most important tasks are to assign processing jobs to image nodes and trigger action node services based on the analysis of the processing results. Both the image node and action node services are discovered by using the dynamic orchestration strategy of the Eclipse Arrowhead Framework. Actions are triggered based on dynamically configurable policies.

- **Action Node**
  Any provider service from an Arrowhead Local cloud can be triggered as an action, which can accept an image processing result JSON as an input.

- **Storage Node**
  A file system and a database are used to store the actual images (in the file system), the belonged metadata, and processing results (in the database). In case multiple storage nodes are present in a local cloud, they must use the same database and file system.

- **Monitor Node**
  Desktop application for interacting with the surveillance network.

4.13.3 Microservice Environment

As it was mentioned before, the Eclipse Arrowhead Framework (4thgeneration) was used to establish a proper microservice environment for the Sentinel Network. Several features were exploited for the faster and more effective development process:

- **Event-driven workflow**
  The nodes in a Sentinel network are using events to communicate with each other (besides the actual service consumptions). For this purpose, the Event Handler Supporting System is also a fundamental part of the Arrowhead Local Cloud. The different types of events and their publishers and subscribers are described in Figure 28.

- **Dynamic configuration**
  It was required to be able to change and apply configuration values without the need of restarting the various components. This requirement was fulfilled by using the Configuration Supporting System in the local cloud.

- **Scalability**
  Since every node type can be present in multiple instances within the network, it was crucial to discover each other dynamically. The dynamic strategy of the Orchestrator Core System makes this easily possible. This feature also helps resolve scalability problems.

- **Authorization & Authentication**
  The usage of the various services and event subscriptions must be restricted only to the sentinel components because other application systems can be present in the same local cloud. Authorization Core System fulfills this need. Furthermore, the Arrowhead X.509 certificates make every component identifiable.

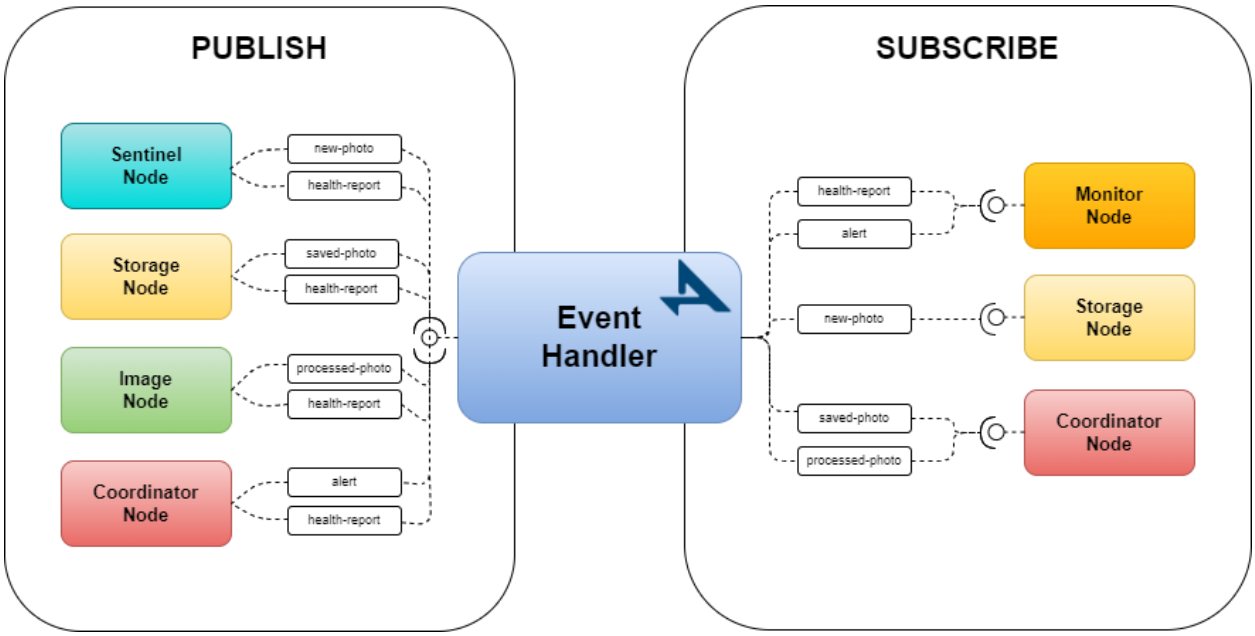


Figure 28: Publish-subscribe approach within Sentinel

Requirements
During the development phase of the prototype, various requirements has been risen against to 4th generation of Eclipse Arrowhead FW that should be addressed in the coming 5th generation:

- Policy based authorization
  It is cumbersome to settle all the peer-to-peer authorization rules for the sentinel components; a policy-based authorization would be much more effective.

- Event level authorization
  At the moment, the components need to have at least one service level permission to be allowed to receive events from a specific provider. However, there are such components that are not consuming any service but need to receive events.

- System group level configuration
  Currently, all system instances need to have their own stored configuration values, but in some cases, multiple systems can share the same configuration (as an example, in a Sentinel Network, all the Coordinator Nodes have to have the same configuration, like action map, thresholds, etc…).

- User account management
  Many consumers system could require user account management, in the Sentinel project the Monitor Node is such a system, so it would reasonable to provide a general solution for this purpose within the Eclipse Arrowhead FW.

Currently, the main focus of research and development is enhancing the monitoring system and integrating MLOps features (image model management and creation support).

5. Conclusions

D3.1 provides requirements which forms a road map for the further development of the AIMS5.0 cyber architecture. Which will be implemented as updates and extensions to the Eclipse Arrowhead framework. These updates and extension will be including in the major release of Eclipse Arrowhead v5.0 which is currently being defined as a cooperation between AIMS5.0, Arrowhead fPVN and a number other smaller EU and national projects.

These updates will provide an extensive support for using AI in automation solutions based on the Eclipse Arrowhead cyber architecture which will be validated and verified in a set of concrete industrial use cases targeting TRL 6-7.

6. Publications

Numerous technical achievement and work performed in this work package were published as scientific papers. They were submitted to international, peer-reviewed conferences to disseminate the accumulated results. A list of related publications can be found below:

[BAN1] Belal Abu Naim et al., „A Generic Framework for Resource-Limited Microcontrollers Deployment in I-IoT Systems“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.

[SW1] "Real-time Container Orchestration based on Time Utility Functions", S. Walser, J. Ruh, S. Craciunas, 20th IEEE Int. Conference on Factory Communication Systems, IEEE, 2024

[GD1] G. Dimirakopoulos et al., "On the Way to Realize the 5th Industrial Revolution: Achievements, Challenges and Research Areas," NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 2023, pp. 1-6, doi: 10.1109/NOMS56928.2023.10154446.

[SVS1] S. Vattaparambil Sudarsan, O. Schelén and U. Bodin, "Multilevel Subgranting by Power of Attorney and OAuth Authorization Server in Cyber–Physical Systems," in IEEE Internet of Things Journal, vol. 10, no. 17, pp. 15266-15282, 1 Sept.1, 2023, doi: 10.1109/JIOT.2023.3265407.

[HG1] Gergely L. Hollósi et al., „AIMS5.0 AI Toolbox: Enabling Efficient Knowledge Sharing for Industrial AI“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.

[PV1] Pal Varga et al., „MLOps in CPS - a use-case for image recognition in changing industrial settings“, NOMS 2024-2024 IEEE/IFIP Network Operations. In press.

[HCS1] Cs. Hegedus et al., „Tailoring MLOps Techniques for Industry 5.0 Needs“, CNSM 2023-2023 IEEE/AnServApp Network Operations. In press.

7. References

[Salman2019] S. M. Salman, V. Struhár et al., “Fogification of industrial robotic systems: Research challenges,” in Proc. Fog-IoT, 2019.

[Shaik2020] M. S. Shaik, V. Struhár et al., “Enabling fog-based industrial robotics systems,” in Proc. ETFA, 2020.

[Morabito2017] R. Morabito, I. Farris et al., “Evaluating performance of containerized iot services for clustered devices at the network edge,” IEEE Internet Things J., vol. 4, no. 4, 2017.

[Xavier2016] B. Xavier, T. Ferreto et al., “Time provisioning evaluation of kvm, docker and unikernels in a cloud platform,” in Proc. CCGrid, 2016.

[Struhár2023] V. Struhár, S. S. Craciunas et al., “Hierarchical resource orchestration framework for real-time containers,” ACM TECS, 2023.

[Vaucher2018] S. Vaucher, R. Pires et al., “SGX-aware container orchestration for heterogeneous clusters,” in Proc. ICDCS, 2018.

[Tosatto2015] A. Tosatto, P. Ruiu et al., “Container-based orchestration in cloud: state of the art and challenges,” in Proc. CISIS, 2015.

[Gkamas2022] T. Gkamas, V. Karaiskos et al., “Performance evaluation of distributed database strategies using docker as a service for industrial iot data: Application to industry 4.0,” Information, vol. 13, no. 4, 2022.

[Lasi2014] H. Lasi, P. Fettke et al., “Industry 4.0,” Business & information systems engineering, vol. 6, no. 4, 2014.

[Abeni2019] L. Abeni, A. Balsini et al., “Container-based real-time scheduling in the linux kernel,” ACM SIGBED Review, vol. 16, no. 3, 2019.

[S-Julián2023] R. S-Julián, I. Lacalle et al., “Self-* capabilities of cloud-edge nodes: A research review,” Sensors, vol. 23, no. 6, 2023.

[Struhár2021] V. Struhár, S. S. Craciunas et al., “React: Enabling real-time container orchestration,” in Proc. ETFA, 2021.

[Lipari2003] G. Lipari and E. Bini, "Resource partitioning among real-time applications," 15th Euromicro Conference on Real-Time Systems, 2003. Proceedings., Porto, Portugal, 2003, pp. 151-158, doi: 10.1109/EMRTS.2003.1212738.

[Xu2021] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, ‘Industry 4.0 and Industry 5.0—Inception, conception and perception’, *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, Oct. 2021, doi: 10.1016/j.jmsy.2021.10.006.

[Lee2015] J. Lee, ‘Smart Factory Systems’, *Informatik Spektrum*, vol. 38, no. 3, pp. 230–235, Jun. 2015, doi: 10.1007/s00287-015-0891-z.

[Pali2017] P. Varga *et al.*, 'Making system of systems interoperable – The core components of the arrowhead framework', *Journal of Network and Computer Applications*, vol. 81, pp. 85–95, Mar. 2017, doi: 10.1016/j.jnca.2016.08.028.

[Weske2012] M. Weske, *Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-28616-2.

[Camunda] Camunda, 'Camunda Platform 8'. Camunda. [Online]. Available: https://camunda.com

[OMG2011] OMG, 'Business Process Model and Notation (BPMN)'. Jan. 2011.

[Pufahl2018] L. Pufahl and D. Karastoyanova, 'Enhancing Business Process Flexibility by Flexible Batch Processing', in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, H. Panetto, C. Debruyne, H. A. Proper, C. A. Ardagna, D. Roman, and R. Meersman, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 426–444. doi: 10.1007/978-3-030-02610-3_24.