Ref. Ares(2022)3263954 - 27/04/2022

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

Intelligent Motion Control under Industry 4.E

D2.3 Overall requirements on IMOCO4.E reference framework

Due Date: M08 – 2022-04-30

Abstract:

The deliverable D2.3 describes the overall requirements of the IMOCO4.E reference framework. First, D2.3 defines an initial version of the IMOCO4.E reference framework with the help of an architecture viewpoint, an AI dataflow viewpoint, a digital twin functional viewpoint, and a digital twin lifecycle viewpoint. Second, D2.3 proposes an abstraction of the building blocks based on component and interface modelling. Finally, D2.3 enumerates system-level, architecture layer-specific and overall building block specific requirements based on the combined expertise of all IMOCO4.E beneficiaries. The deliverable provides requirements not only on the hardware and software components but also on digital twinning, the application of AI, and ultimately, the interfacing between all the IMOCO4.E framework components. The proposed initial version of the IMOCO4.E reference framework and the overall requirements go beyond the current state-of-the-art intelligent motion control systems. The final version of the IMOCO4.E reference framework shall be reported in the deliverable D2.4.

The project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking, under Grant Agreement n°101007311. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation programme and Netherlands, Czech Republic, Latvia, Spain, Greece, Portugal, Belgium, Italy, France, Ireland

Project Information

Grant Agreement Number	**101007311**
Project Acronym	IMOCO4.E
Project Full Title	Intelligent Motion Control under Industry 4.E
Starting Date	1st September 2021
Duration	36 months
Call Identifier	H2020-ECSEL-2020-2-RIA-two-stage
Topic	ECSEL-2020-2-RIA
Project Website	https://www.imoco4e.eu/
Project Coordinator	Arend-Jan Beltman
Organisation	SIOUX TECHNOLOGIES BV (SIOUX)
Email	Arend-Jan.Beltman@sioux.eu

Document Information

Work Package	WP2 Business requirements and reference system architecture							
Lead Beneficiary	ITEC BV.							
Deliverable Title	Overall requirements on IMOCO4.E reference framework							
Version	3.0							
Date of Submission	27/04/2022							
Author(s)	Sajid Mohamed (ITEC), sajid.mohamed@nexperia.com							
Contributor(s)	Gijs van der Veen (ITEC), gijs.van.der.veen@nexperia.com Hans Kuppens (SIOUX), hans.kuppens@sioux.com All BB owners (OROLIA, TNO, INL, UNIMORE, TEK, UNIBS, UWB, EDI, GNT, IMST), EMC, UGR, NURO, Many partners (ADI, BUT, CRIT, ECS, EDI, EMD, FAG, GDM, GNT, HS, IMST, INL, INTRA, MADARA, NORMET, OROLIA, PCL, PMS, TEK, TNO, TUE, UCC, UNIBS, UWB, VTT)							
Internal Reviewer(s)	Hans Kuppens (SIOUX), hans.kuppens@sioux.com Martin Čech (UWB), mcech@kky.zcu.cz							
Document Classification	**Draft**				**Final**	X		
Deliverable Type	**R**	X	**DEM**		**DEC**		**OTHER**	
Dissemination Level	**PU**	X	**CO**		**CI**			

History					
Version	**Issue Date**	**Status**	**Distribution**	**Author**	**Comments**
1.0	03/04/2022	Draft	PU	ITEC	Internal distribution to all partners for feedback
1.1	07/04/2022	Draft	PU	ITEC	Addressing partner feedbacks - Textual changes for clarity, additions in Section 2.4, updated Figure 3, Figure 6, Table 11

2.0	08/04/2022	For internal review	PU	ITEC	Proposal for internal review to UWB and SIOUX
2.1	15/04/2022	Draft	PU	ITEC	Added more abbreviations, incorporating initial feedback from the reviewer (SIOUX) – updated Figure 2, Figure 3, Figure 7, re-organised Section 2.5, added Figure 11
2.2	19/04/2022	Draft	PU	SIOUX	Internal review - Minor textual changes, comments for improvement
2.3	25/04/2022	Draft	PU	ITEC	Addressing reviewer (SIOUX, UWB) comments and feedback – textual changes, added abbreviations, added references, updated Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 10, Figure 11, all tables (1-12) – added one, updated some, and removed eight requirements
2.4	25/04/2022	Draft	PU	ITEC	Updating requirements' numbering, ID, and tasks; removed a comment from Table 11; Formatting changes
3.0	27/04/2022	Final	PU	ITEC	Final version for public dissemination

Type of Contribution	
Partner	**Description of Contribution to Contents**
ITEC	Ideation, reference framework definition, requirements gathering survey (drafting, coordination, results processing), requirements classification, preparation and editing of the document
SIOUX	Ideation, reference framework definition, requirements gathering survey, internal review
EMC, UGR, NURO	Ideation - requirements gathering survey and requirements classification
Many partners, ITEC, SIOUX, EMC	Contributed to the requirements gathering survey
All BB owners	Coordination of requirements gathering within BB, leading the BB workshop for consistency checking and finalisation of requirements per BB.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Explanation
AC	Alternating Current
AI	Artificial Intelligence
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
BB	Building Block
BBx (e.g. BB1)	Building Block x (e.g. Building Block 1)
BLDC	Brushless Direct Current
CAN	Controller Area Network
COTS	Commercial off-the-shelf
CPU	Central Processing Unit
DC	Direct Current
DoA	Description of Action
DSP	Digital Signal Processor
DT	Digital Twin
DTA	Digital Twin Aggregation
DTI	Digital Twin Instance
Dx.x (e.g. D2.3)	Deliverable x.x (e.g. Deliverable 2.3)
ECSEL	Electronic Components and Systems for European Leadership
EM	Electromagnetic
EMC	Electromagnetic Compatibility
EMF	Electromagnetic Field
EMI	Electromagnetic Interference
ERP	Enterprise Resource Planning
EU	European Union
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
FPGA	Field-Programmable Gate Array
FOC	Field-Oriented Control
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMI	Human Machine Interface
HIL	Hardware-In-the-Loop
IADT	Inspection, Analysis, Demonstration, and Test
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Commission
I-MECH	Intelligent motion control platform for smart Mechatronics systems
IMOCO4.E	Intelligent Motion Control under Industry 4.E
IoT	Internet-of-Things
ISO	International Standardization Organization
IO or I/O	Input Output
IT	Information Technology
JU	Joint Undertaking

MBSE	Model-Based Systems Engineering
MES	Manufacturing Execution System
MIL	Model-In-the-Loop
MIMO	Multi Input Multi Output
ML	Machine Learning
MPSoC	Multiprocessor System-on-Chip
LIDAR	Laser Imaging Detection and Ranging
MPU	MicroProcessor Unit
OPC-UA	Open Platform Communications Unified Architecture
OT	Operation Technology
PC	Personal Computer
PIL	Processor-In-the-Loop
PIPL	Personal Information Protection Law
PLC	Programmable Logic Controller
QoS	Quality-of-Service
RoHS	Restriction of Hazardous Substances
SAP	Systems, Applications and Products in data processing
SCADA	Supervisory Control And Data Acquisition
SE	System Exploitation
SI	System Integration
SIL	Software-In-the-Loop
SLAM	Simultaneous Localization And Mapping
SO	System Operational
SoC	System-on-Chip
ST	Scientific and Technological
TPU	Tensor Processing Unit
TRL	Technology Readiness Level
TSN	Time-Sensitive Networking
Tx.x (e.g. T2.3)	Task x.x (e.g. Task 2.3)
UART	Universal Asynchronous Receiver-Transmitter
WEEE	Waste Electrical and Electronic Equipment
WP	Work Package
WPT	Wireless Power Transfer
WPx (e.g. WP2)	Work Package x (e.g. Work Package 2)
XIL	X-in-the-Loop

Executive Summary

The deliverable D2.3 addresses the overall requirements on the IMOCO4.E reference framework. The IMOCO4.E reference framework envisions a generic architecture platform for designing, developing, and implementing novice and complex motion-controlled industrial systems. In this deliverable, an IMOCO4.E reference framework is defined over several discussions after gathering the brownfield reference architectures from the pilots, use-cases, and demonstrators. The proposed initial version of the IMOCO4.E reference framework and the overall requirements go beyond the current state-of-the-art intelligent motion control systems.

The requirements definition on the IMOCO4.E reference framework covers the system-level requirements, architecture layer requirements (as explained in the DoA), the BB requirements, and interfaces and connectivity requirements. The requirements definition is the first step in the W-model iterative approach of IMOCO4.E framework development (explained in the DoA) and drives the IMOCO4.E development activities. The requirements were gathered through a survey sent to all beneficiaries in the IMOCO4.E consortium.

This deliverable classifies the requirements based on defined classification criteria, assigns priorities, enlists the validation method and identifies the applicable IMOCO4.E tasks. Further, this deliverable considers the scientific and technological (ST) development objectives, system integration and interoperability (SI) objectives, system operational (SO) objectives, and system exploitation (SE) objectives of the IMOCO4.E project. The requirements specified in this deliverable form the basis for tasks 3.1, 4.1, 5.1, 7.1 and 7.2, which will gather detailed requirements and specifications for the IMOCO4.E development activities. The requirements and the reference framework will be refined further in D2.4. The deliverable D2.4 would explain the general specification and design of the final IMOCO4.E reference framework.

1. Introduction

1.1 Purpose of this Deliverable

This deliverable contains the overall (system-level) requirements on the IMOCO4.E reference framework. This deliverable also proposes an initial version of the IMOCO4.E reference framework and the expected interfaces between the 10 building blocks to give direction for IMOCO4.E tasks that will gather more detailed requirements about the implementation of functions and features. Further, this deliverable considers the ST, SI, SO and SE objectives of the IMOCO4.E project [1]. This deliverable is the first outcome of Task 2.3. Task 2.3 is devoted to the definition of the IMOCO4.E reference system architecture. Deliverable D2.4 will gather the refined requirements and present the detailed specification of the final IMOCO4.E reference framework.

This document does not include all IMOCO4.E requirements because the focus is on system-level requirements. Detailed requirements and specifications at the building block level can be more extensive, and these are left to the respective deliverables in tasks 3.1, 4.1 and 5.1. The detailed requirements for the pilots and demonstrators would be disseminated through the deliverables in tasks 7.1 and 7.2.

1.2 Structure of this deliverable

Chapter 1 contains sections that help the reader understand the context of this deliverable with respect to the overall project. Chapter 2 explains the first version of the IMOCO4.E reference framework. The different viewpoints of the reference framework in relation to the BBs of this project are also presented. Chapter 3 details the requirements gathering process, requirements classification and the requirements coding scheme used throughout this deliverable. Chapter 4 enumerates the overall system-level requirements of the IMOCO4.E reference framework. Chapter 5 enumerates the architecture layer-specific requirements of the IMOCO4.E reference framework. Chapter 6 briefly summarises the BBs and lists the overall BB specific requirements of the IMOCO4.E reference framework. Finally, Chapter 7 concludes the deliverable.

1.3 Intended readership

Deliverable D2.3 is disseminated as a public document. This means that the IMOCO4.E reference framework and requirements identified in this deliverable are available to everyone.

1.4 Relation to I-MECH project

The successfully completed I-MECH (intelligent motion control platform for smart mechatronic systems) project, funded by the ECSEL JU programme, is the predecessor to the IMOCO4.E project. The main differences between the I-MECH and IMOCO4.E projects are the introduction of the factory (supervisory) layer (Layer 4) and the emphasis on AI methods and digital twin integration. In I-MECH, the focus was on a single machine, whereas in IMOCO4.E, Layer 4 enables communication and interfaces between multiple machines. The AI methods are explored in detail through BB8 on AI-based components. The individual BBs inherently consider the digital twin integration during their design and development. BB9 focuses on the data storage, cyber security, and trustworthy data management required for integrating AI and digital twins.

Figure 1 gives an overview of what is new in the IMOCO4.E project. Figure 1 also shows the commonalities between the I-MECH and the IMOCO4.E projects. As you can see, there is an overlap of three architecture layers – Layer 1, Layer 2 and Layer 3. This overlap means there may be requirements from the I-MECH project carried over to the IMOCO4.E project. In addition, the IMOCO4.E project may re-use or extend some of the concepts and requirements from the I-MECH project. The main differences are related to the AI components and Digital twins. The requirements related to these are covered mainly in the Overall system-level requirements in Chapter 4. In addition, BB8 on AI components cover the specific BB-specific requirements on AI BB9 on cyber-security tools, and trustworthy data management addresses how data is secured and managed for integration with AI and digital twin toolchains.

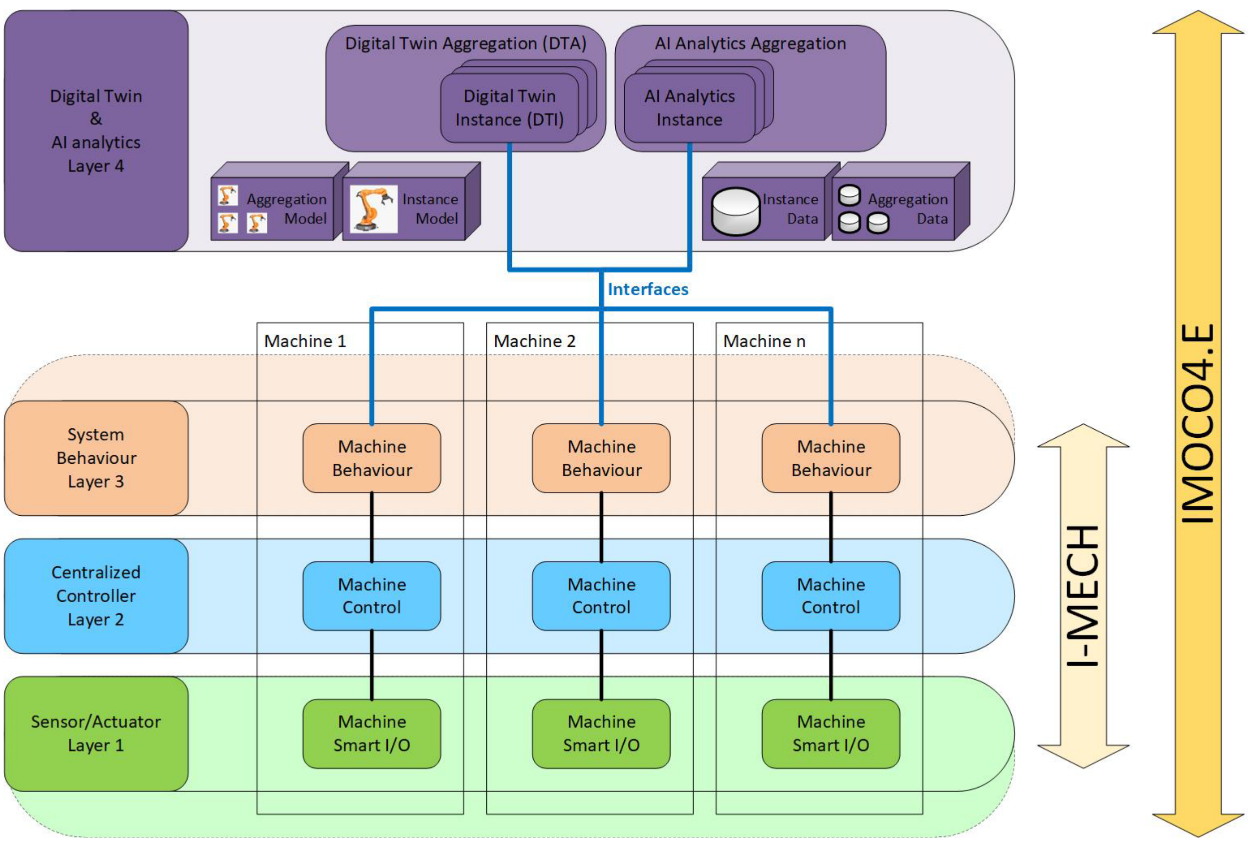

Figure 1. Envisioned IMOCO4.E reference framework (derived from the DoA [1]). The difference between the IMOCO4.E reference framework and the I-MECH project is also illustrated.

1.5 Relation to other IMOCO4.E deliverables and recommendations for D2.4

The initial definition of the IMOCO4.E reference framework is based on the state-of-the-art, emerging trends and identified shortcomings from the deliverable D2.1 and the identified needs from the deliverable D2.2. The deliverable D2.2 was done in parallel with this deliverable. The definitions of relevant terminology, e.g., an architecture and a framework, are also provided in the deliverable D2.1. The enumerated requirements in this deliverable try to address the identified shortcomings and need wherever possible. The goal of this deliverable is to keep the requirements generic and at the system level. Detailed requirements and specifications may be derived from the overall requirements and reported in future deliverables D3.2, D4.2 or D5.1.

The following are the recommendations for the deliverable D2.4:

- The detailed specification and definition of the final IMOCO4.E reference framework shall be reported in D2.4.
- Application and domain-specific instantiations and implementations of the IMOCO4.E reference framework shall be reported in D2.4.
- The final version of the IMOCO4.E reference framework viewpoints shall be reported in D2.4.
- Based on the AI and digital twin viewpoints, AI and digital twin toolchains and frameworks shall be reported in D2.4.
- The detailed specification of the architecture interfaces shall be reported in D2.4.
- The BB abstraction model shall be finalised in D2.4.
- The finalised BB interactions and interfaces of the IMOCO4.E development shall be reported in D2.4.

2. IMOCO4.E reference framework – the initial version

In this chapter, the initial version of the IMOCO4.E reference framework is defined after considering the identified shortcomings from the state-of-the-art, needs for the future, and the brownfield architectures from the relevant industrial partners. The steps leading to the definition of the reference framework are the following:

1. Identifying shortcomings from the state-of-the-art reference architectures. Understanding the current state-of-the-art and identifying the gaps were essential for defining the IMOCO4.E reference framework. The state-of-the-art methods for motion-driven high-tech applications and shortcomings were reported in the deliverable D2.1.
2. Identifying the needs for future smart production from the mechatronics and robotic point of view. The IMOCO4.E reference framework should address these needs through relevant requirements. The needs summary gathered per BBs was used to define the requirements for (some of) the relevant topics. The needs would be part of the deliverable D2.2.
3. Gathering and characterising the brownfield architectures of the pilots, demonstrators and use cases from the industrial partners in the IMOCO4.E consortium. The brownfield architecture (in the context of the IMOCO4.E project) characterises the existing legacy systems (software, hardware, or platform) that would be part of the corresponding pilots, demonstrators or use cases during the IMOCO4.E framework development and integration. Understanding the constraints and including interfacing options with the brownfield architecture in the generic IMOCO4.E reference framework definition is necessary.

Further, this deliverable considers the scientific and technological (ST) development objectives, system integration and interoperability (SI) objectives, system operational (SO) objectives, and system exploitation (SE) objectives of the IMOCO4.E project [1]. IMOCO4.E strives to deliver a reference framework consisting of AI and digital twin toolchains and a set of mating building blocks for resilient manufacturing applications.

2.1 Overview

The IMOCO4.E reference framework will be open, flexible, scalable, future-proof, and fully functional. It can be exploited in the industry in high-performance motion control applications with several overlaps to health, mobility, manufacturing, and supply chain management domains. A flexible framework is configurable from the lowest layer (Layer 1) to the human interfaces and supervisory layer (Layer 4). IMOCO4.E will also bring an MBSE (Model-Based System Engineering) approach to all the architecture layers. Furthermore, these layers will demonstrate how AI supports the optimisation of processes using the digital twin instances.

The first version of the IMOCO4.E reference framework definition comprises the following viewpoints.

1. Architecture viewpoint
2. AI viewpoint
3. Digital twin viewpoint

These viewpoints are explained in detail in the following sections. Additionally, the BBs are abstracted as components, and the interfaces between BBs are modelled using interface modelling.

2.2 Architecture viewpoint

The architecture viewpoint of the IMOCO4.E reference framework is illustrated in Figure 2. The components in the architecture viewpoint are the following.

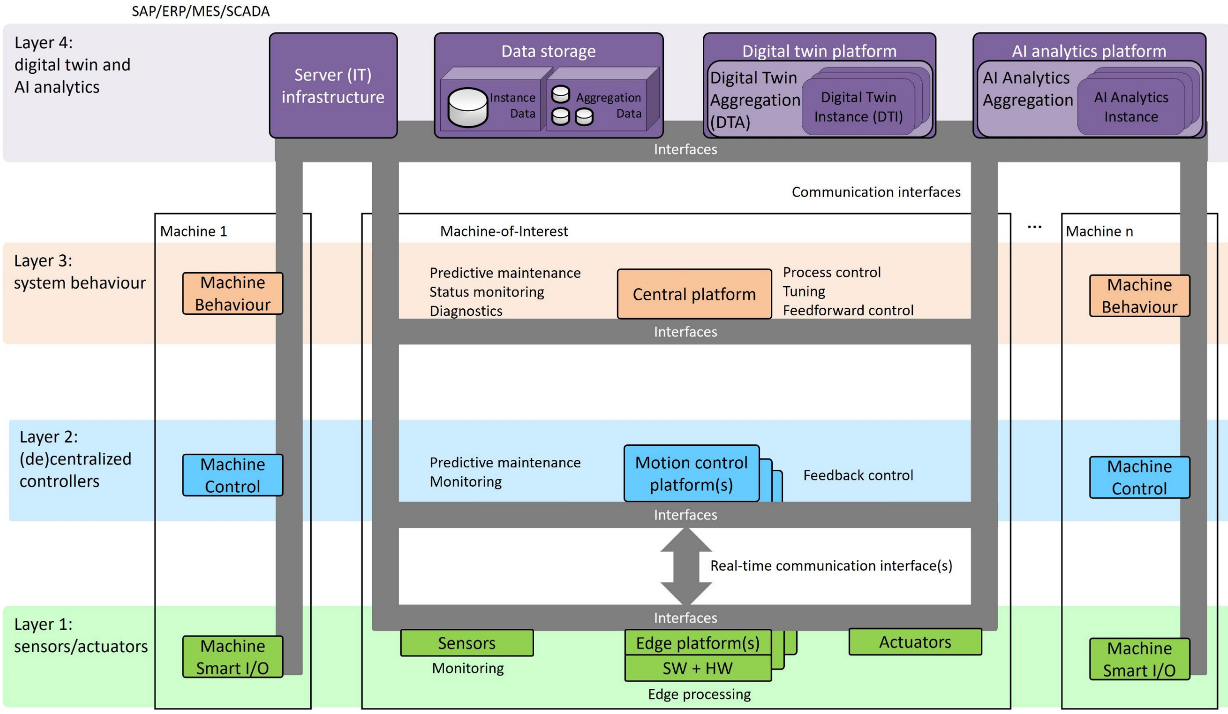

Figure 2. IMOCO4.E reference framework architecture viewpoint – initial version

- *Sensors*: Component for detection or measurement of physical properties or parameters. E.g., temperature sensor, pressure sensor, etc. Smart sensors may include some local (or edge) processing.
- *Actuators*: Component responsible for moving and controlling a mechanism or system, for example, opening a valve. Smart actuators may include some local intelligence (and processing).
- *Platforms*: A platform refers to the combination of software (tasks, messages, mapping, scheduling) and hardware (computation, communication, memory). The software performance relies heavily on the predictability and reliability of the deployed hardware. The software can overcome errors to a certain degree when a few hardware functions fail. Still, the overall performance will degrade when input signals, i.e., data, are corrupted in hardware before these are in the 'digital' domain. Therefore, it is essential that the hardware used within the IMOCO4.E project provides reliable signals and data. The platform components considered in the IMOCO4.E reference architecture are the following.
    - Edge platforms (Layer 1): When data needs to be processed at the edge, the IMOCO4.E framework will rely on edge platforms. E.g. for high-speed vision processing, an edge platform is essential since sending image streams over the fieldbus is not ideal due to limited fieldbus bandwidth.
    - Motion control platforms (Layer 2) are mainly required for accurate and predictable feedback control at a high sampling rate (in the kHz range). Such control can be

centralized or decentralized. The state of the components controlled by the platforms can also be monitored, and necessary predictive maintenance actions can be taken.
    - A central platform (Layer 3), e.g. a PC, is required for coordinating the machine operation. Process control, feedforward control, parameter setting/tuning, machine status monitoring, predictive maintenance and diagnostics are some of the tasks/applications that run on the central platform
    - *AI platform* (Layer 4) refers to the AI analytics and training infrastructure. The AI platform consists of AI analytics instances and AI analytics aggregation (explained in Section 2.3). The detailed specification of the AI platform will be reported in the deliverable D2.4.
    - The *digital twin (DT) platform* (Layer 4) refers to the software and hardware necessary for the digital twin instances (DTI) and digital twin aggregation's (DTA) modelling, operation, monitoring, maintenance, and update. DTI and DTA are explained in Section 2.4.
- *Server (IT) infrastructure* is the backbone in a factory or production line for interactions with human users and factory operations, e.g. integrating the machine operation with SAP, ERP, MES or SCADA solutions.
- *Data storage* for the IMOCO4.E reference framework refers to the data necessary for the AI and digital twin platforms. Instance data refers to the data for a machine instance, and the aggregation data refers to the data for DTA and AI analytics aggregation.
- *Interfaces* are the most necessary components in the IMOCO4.E reference framework. Interfaces can be fieldbuses, real-time communication protocols, wireless communication protocols, internet communication and so on. As shown in Figure 2, interfaces can be present between any two architecture layers on the same machine or between architecture layers on two different machines through Layer 4. The interface between Layer 1 and Layer 2 is typically a real-time communication interface (e.g. EtherCAT). The detailed specification of the interfaces will be reported in the deliverable D2.4.

2.3 AI viewpoint

In the IMOCO4.E project, "the focus is on the following distinct aspects of AI for motion control: environmental awareness, motion planning, deployment, image pre-processing, signal analysis for drive diagnostics, long-term predictive maintenance of the machine" (from the DoA [1]). The AI viewpoint provides the data flow, interfaces and BB interactions required to achieve the project goal.

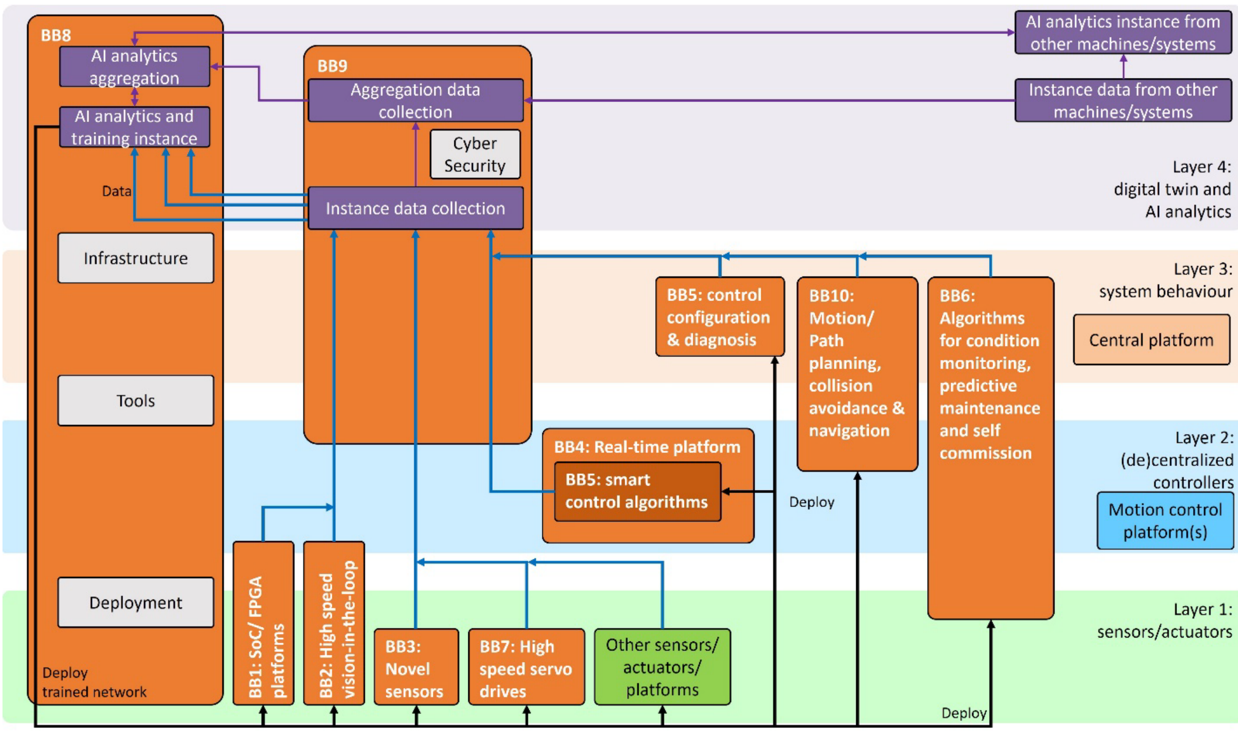

Figure 3. AI viewpoint with BB interactions

The AI viewpoint with BB interactions of the IMOCO4.E reference framework is illustrated in Figure 3. The general principle is that data is collected from Layer 1 (sensors, edge platforms and actuators) and used by the AI framework for modelling, training, optimisation, analytics and/or services. Additionally, data can be collected from Layer 2 (e.g. from BB5 internal signals). Furthermore, it is convenient to have configuration data (e.g. from Layer 3) available in the data collection, such that the dataset is complete and consistent at all times. The data flows from layers Layer 1, Layer 2 and Layer 3 to the AI framework and back to the corresponding BBs are illustrated in Figure 3. BB8 deals with AI-based components and forms the core BB for integrating the AI framework in the IMOCO4.E methodology. BB8 will specify in detail the AI infrastructure, tools and deployment methods in future deliverables. The data necessary for the AI framework is collected, secured and stored based on the methodology developed as part of BB9 (cybersecurity and trustworthy data management). Some definitions are as follows:

- The *AI instance* refers to the AI framework for a machine instance (or some machine components).
- The *AI aggregation* refers to the aggregation of all AI instances.

The functionality of an AI instance and AI aggregation varies based on the stage in the machine lifecycle and is illustrated in Figure 4. During the design phase of the machine lifecycle, the main functionality of an AI instance is modelling and training. The AI model that is suited for the design objective and satisfies the requirements is identified. Typically, the machine prototype data is used to train the AI model. The AI modelling and training could also start with machine simulation (before the machine prototype is available). Then, the trained AI model is deployed in the machine prototype for testing and validation. The AI instance is optimised for inference performance using the assembled machine data and characteristics during the manufacturing phase. The optimised inference AI model is then deployed in the assembled machine for testing and validation. Finally, during the services phase of the machine lifecycle, the AI instance is used for data analytics and offering other services, e.g. process optimisation. The data monitored from the machine in operation is the input for the AI analytics algorithm, and the AI instance offers optimal services. The AI platform coordinates the AI instance. If required, the AI platform can be independent (with its own hardware and software).

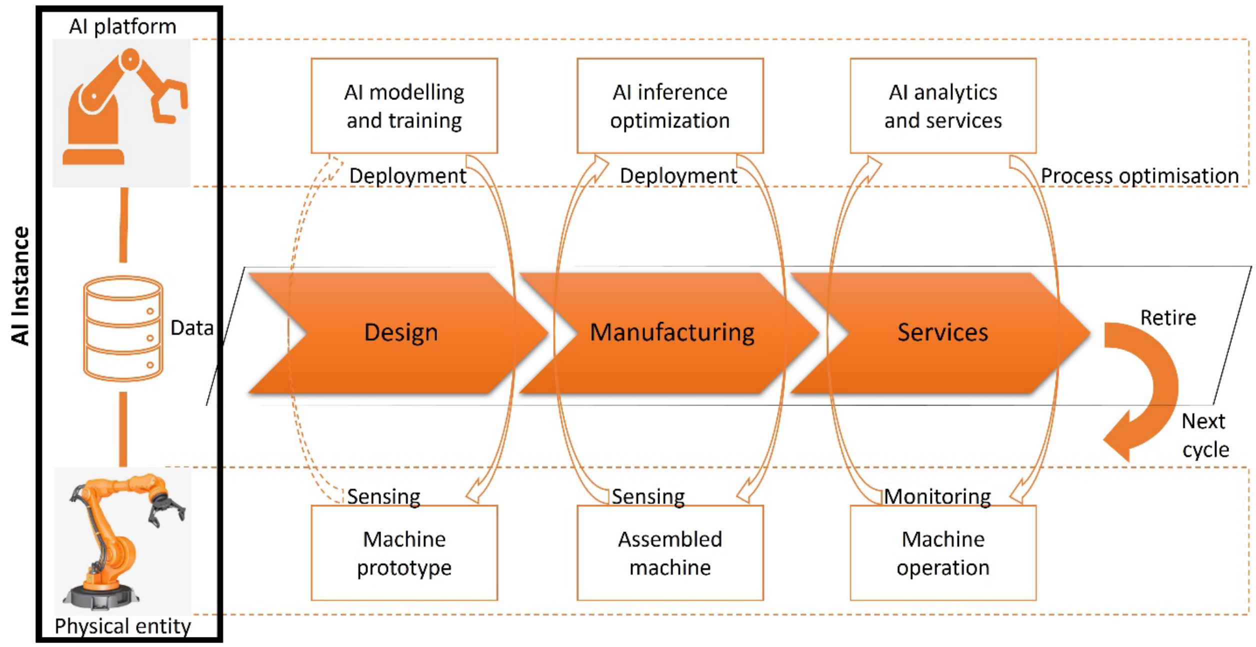

Figure 4. AI viewpoint during a machine lifecycle

2.4 Digital twin viewpoint

A variety of definitions have been employed for the concept of a digital twin. The original concept of a digital twin envisages a system that couples physical entities with virtual models, leveraging the benefits of both the virtual and physical environments to benefit the entire system [2]. Another widely accepted representation of a digital twin is the five-dimensional digital twin model illustrated in Figure 5 and derived from [3]. In the IMOCO4.E framework, a digital twin comprises five dimensions – the physical entity, virtual model, data, service, and connection or interaction. The physical entity is the foundation of the digital twin. Virtual models model the physical entity and reproduce the physical geometries, properties, behaviours and rules [3]. A virtual model is an abstraction of (a part of) the physical entity. Data is the core component and key driver of the digital twin. Data can be obtained from the physical entities, generated from the virtual models, obtained from services or provided by domain experts and users. Services are an essential and highly relevant component of a digital twin due to the modern emphasis on the Everything-as-a-Service. Connections between physical entities, virtual models, services, and data enable information and data exchange. A model is a part of the digital twin and can be independent of the digital twin. However, a digital twin has the models as one of the five dimensions.

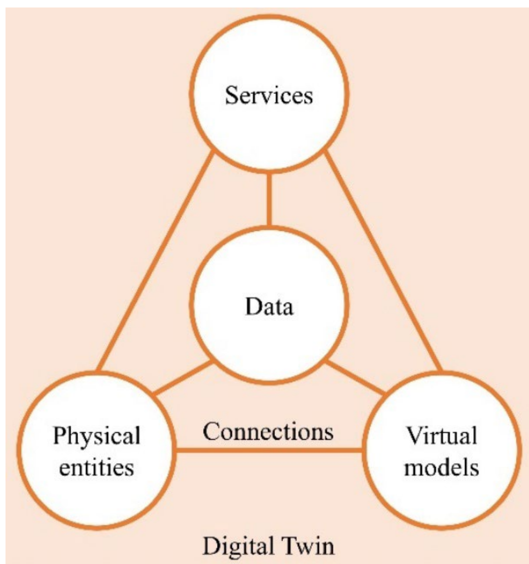

Figure 5. Five-dimensional digital twin (derived from [3]). The five dimensions are physical entities, virtual models, data, services, and their connections.

The digital twin viewpoint during a machine's lifecycle is illustrated in Figure 6. The lifecycle viewpoint for a digital twin is extremely important. "Digital twins are of most use when an object is changing over time, thus making the initial model of the object invalid, and when measurement data that can be correlated with this change can be captured. These changes could be undesirable, for instance, wear in bearings or fatigue in metal components, or they could be neutral but important, for instance, variations in supplied material properties. If an object does not change much over time, or if data associated with that change cannot be captured, then a digital twin is not likely to be useful" [4]. An object here can be (a part of) the machine of interest.

The IMOCO4.E framework concepts related to digital twin (derived from [2]) are the following:

- a *digital twin prototype* is a virtual description of a prototype machine containing all the information to create the physical twin prototype. The digital twin prototype can vary from component level to system level.
- a *digital twin instance (DTI)* refers to a specific instance of a physical machine that remains linked to the specific machine throughout its lifecycle.
- a *digital twin aggregation (DTA)* combines all the digital twin instances.

A digital twin is helpful throughout the machine's lifecycle. During the design phase, virtual design models form the basis of the machine prototype development. Machine prototype specifications are required by the virtual models for designing an efficient system through iterative optimisation and virtual verification. A digital twin can be used during the design phase - to design and test a new algorithm, explore use cases, etc., before deploying it to the actual physical system. The physical system may not yet be available at this point. A digital twin also expedites the test time (and hence faster time-to-market) since the physical system has limited test capacity. Testing on the physical system can be expensive if hardware fails due to testing. The digital twin prototype is used for real-time sensing and control and process optimisation during the manufacturing phase. A digital twin instance during the service phase enables predictive maintenance, fault detection and diagnosis, state monitoring, process optimisation and so on.

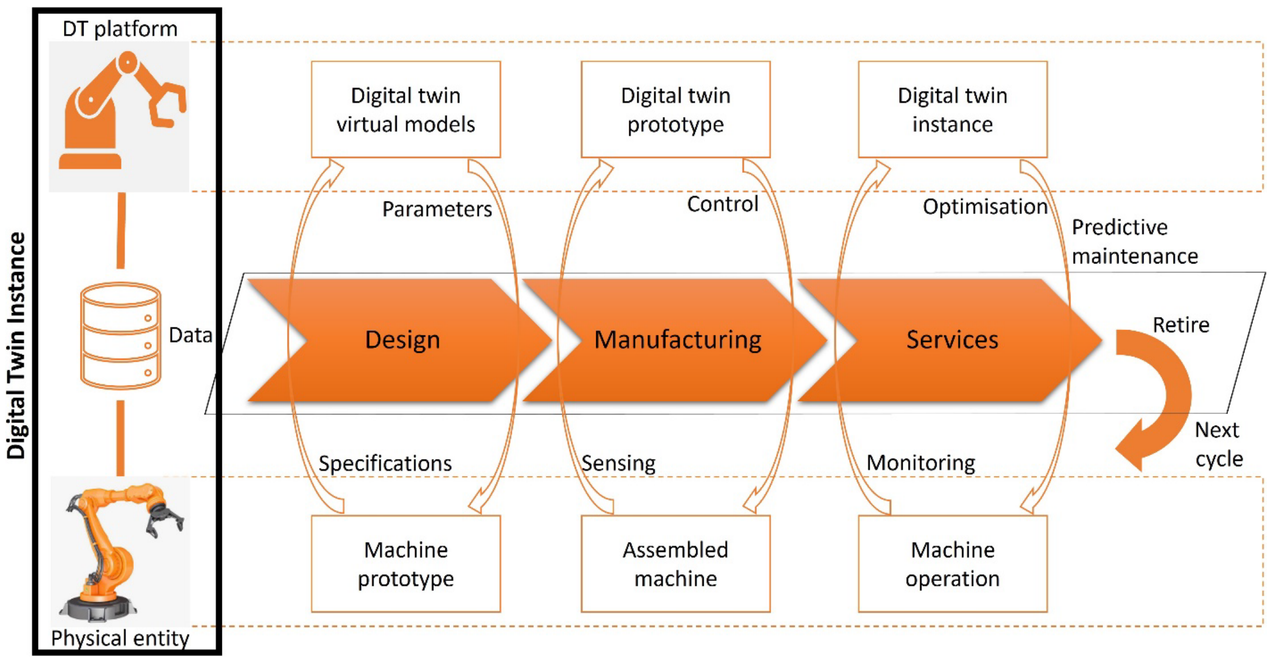

Figure 6. Digital twin viewpoint during a machine lifecycle

The digital twin viewpoint with BB interactions is illustrated in Figure 7. The general principle here is that the physical entity comprises the machine (the sensors, platforms, actuators and interfaces represented through the various BBs, and other components, e.g. COTS). The virtual entity of the digital twin is represented by the digital twin (DT) platform (explained in Section 2.2). The digital twin virtual models are part of the DT platform. The services and analytics are performed through the AI framework (BB8). The BB9 handles the data collection, storage and cyber-security. The DT platform uses the data from the physical twin, services, and models. Finally, the DT platform sends the parameter changes for optimal machine performance to the relevant physical components or provides warnings or predictive maintenance schedules to the human users.

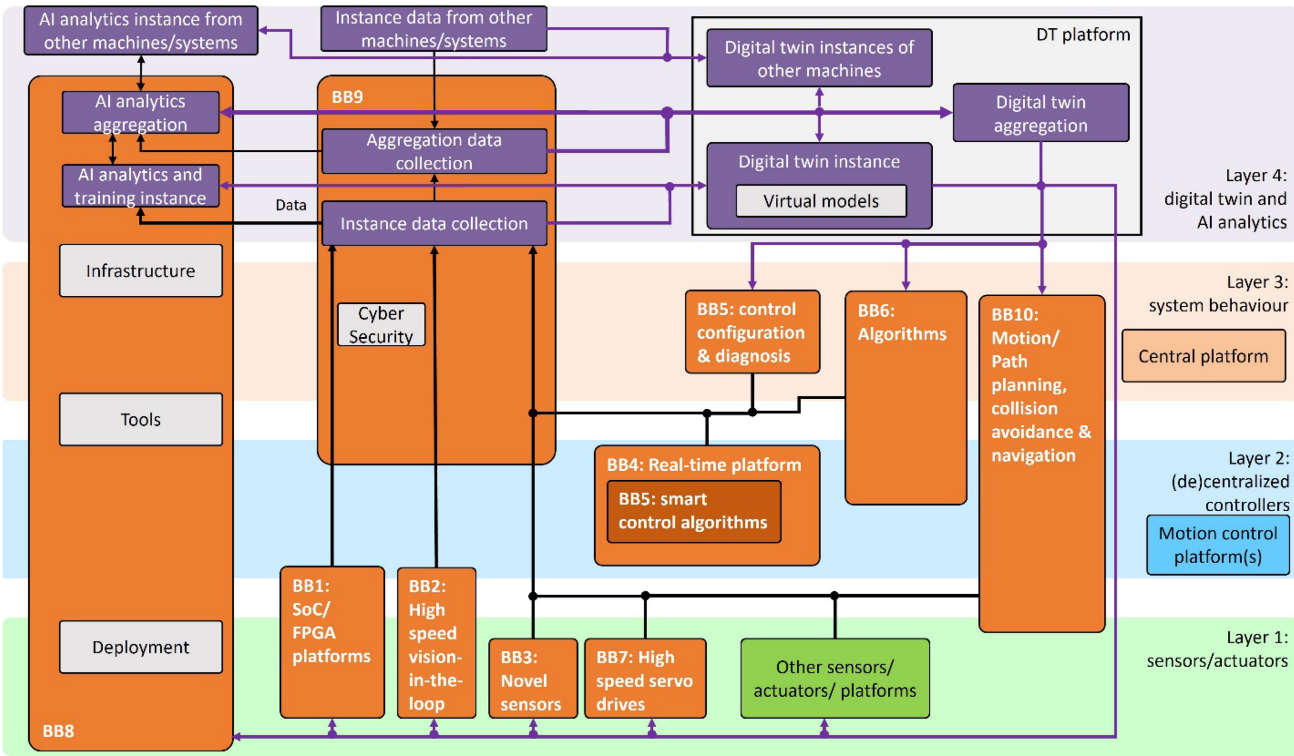

Figure 7. Digital twin viewpoint with BB interactions

2.5 Component abstraction and interface modelling

A component-based model of computation [5] is used to abstract the IMOCO4.E framework components, e.g., the BBs, system or machine. A component-based model-of-computation can be classified into a component model and an interface model. A component model (illustrated in Figure 8 (a)) specifies the behaviour of the component. An interface model specifies how a component can be used. A component can have one or more configurations, and each configuration is specified using configuration parameters. A configuration has an input, an output, a required resource, a provided resource and a quality. The input and the output model the functionality of a particular component configuration. The hardware resources associated with the configuration of a component are modelled using the provided and required resources. The quality specifies the (non-functional) properties of the component, e.g. battery storage capacity, camera frame rate, and quality of control. These properties can typically be optimised for performance according to the requirements.

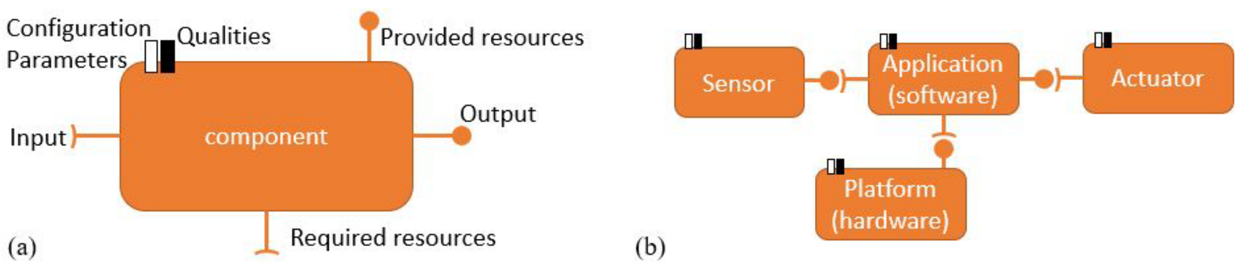

Figure 8. (a) Component model - abstraction of a component in the IMOCO4.E framework (derived from [5]). (b) An interface model example using some IMOCO4.E framework architecture components.

An interface model using the IMOCO4.E architecture components – sensor, application (software), platform (hardware), and actuator – is illustrated in Figure 8 (b). An application (software) require the input from the sensor and provides the output to an actuator. The application also needs a hardware platform to execute. Each of these components can have a configuration and a quality associated with it. E.g. a camera sensor provides an image stream or video as output. The quality of the camera sensor can be the frame rate of the image stream and the configuration parameters can be the video parameters [5].

2.5.1 BB abstraction and BB interface modelling

The BBs are abstracted based on the component model, as illustrated in Figure 9 (a). The BB1 and BB4 are platforms, and they provide resources. BB3 is a sensor and outputs the functional data. BB7 is an actuator and receives the control input to actuate. Other BBs (BB2, BB5, BB6, BB8, BB9, BB10) have a generic structure, as shown in Figure 9 (a), with optional input, output or required resources as determined by the application or function. An example interface model considering the IMOCO4.E BBs is illustrated in Figure 9 (b). The interface shows a sensor (BB3), e.g. a camera sensor, feeding functional information or data to the high-speed vision-in-the-loop system (BB2). BB2 outputs the relevant functional input to the control algorithm in BB5, which in turn sends a control input to BB7 (the servo drive actuator). BB2 and BB5 require platform resources (BB1 and BB4).

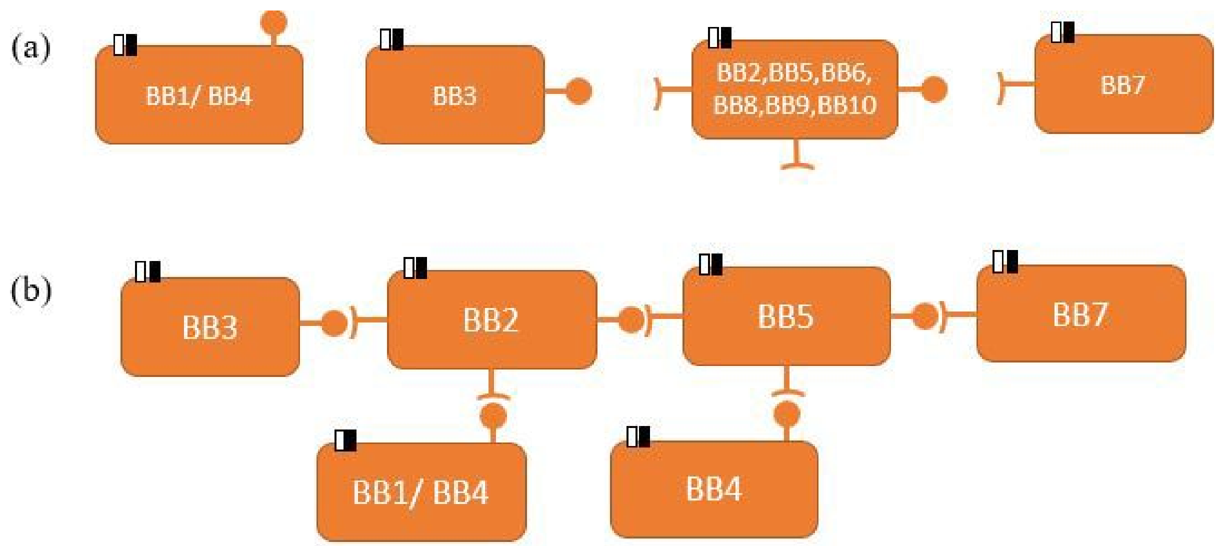

Figure 9. (a) Abstraction of the IMOCO4.E BBs based on the component model. (b) An example interface model of the IMOCO4.E BBs.

Representing BBs as components help to quantify the interfaces necessary for testing and validation. For example, a DTI or DTA can be customised precisely for validating a BB based on this model (see Figure 10 for how this interface looks for the different BBs). E.g. for testing and validating BB1 or BB4, only the component model and the interface model need to be specified as illustrated in Figure 10 (a). Figure 10 (b) illustrates the interfaces for BB2, BB5, BB6, BB8, BB9 and BB10, Figure 10 (c) illustrates the interface for BB3, and Figure 10 (d) illustrates the interface for BB7. The digital twin instance can then provide the required interface. XIL simulation can be performed with the digital twin for testing and validation.

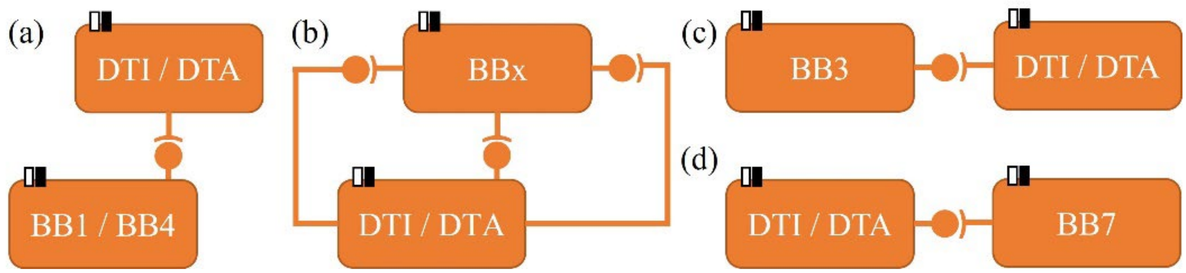

Figure 10. Component specification of a BB helps to define interfaces for digital twin testing and validation. Interfaces between DTI or DTA and (a) BB1/BB4, (b) BBx, (c) BB3, and (d) BB7.

2.5.2 System and environment in the interface model

An intelligent motion control system interacts with the human user, environment and the system hardware. The interactions can be captured using the component model, as illustrated in Figure 11. The (part of the) system-of-interest can be modelled as illustrated in Figure 8 (b) using the sensor, application, actuator and platform components. The system-of-interest can be a component, BB, machine, or multiple machines (e.g. production line). Each of the components in the system-of-interest can be hierarchical, as explained in [5]. For brevity, the concept of hierarchy in the component model is not included in this deliverable, and the reader may refer to [5] for more details.

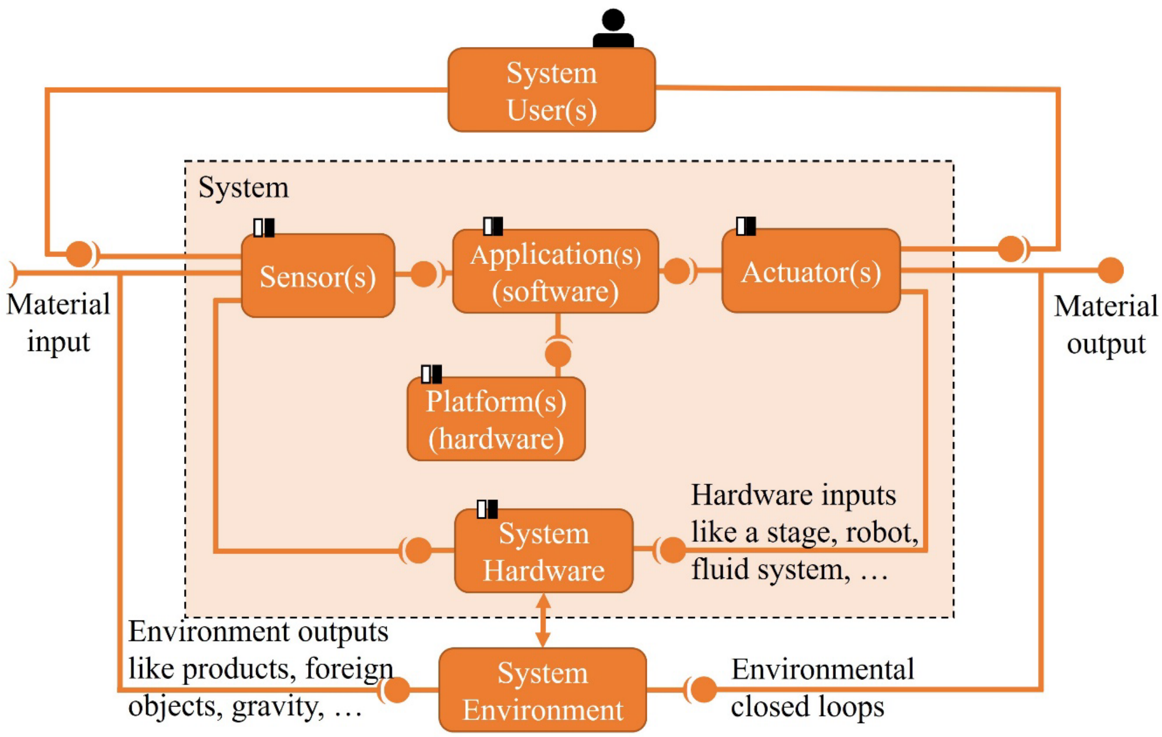

Figure 11. System and environment in the interface model

The system consists of the system-of-interest with the corresponding system hardware like a stage, robot or fluid system. The dynamics of the system hardware can be sensed by the sensors and affected by the actuator's actions. The system, e.g. a production line, can also have material input and material output. In a semiconductor back-end manufacturing production line, the wafer and lead frame are material inputs, and the semiconductor chips are the material output. The system environment determines factors like products, foreign objects, gravity or climate. The system environment is affected and changed by the system actuators, and the system input can be somehow affected by the system output history. This results in a closed loop of the system that goes "through the environment", i.e. outside the system itself (represented in Figure 11 as environmental closed loops). The system user operates or interacts with the system. The user can be a remote operator or can have human-machine interactions.

A digital twin virtual model of the system and environment would have the environment simulated in an environment (physics) simulator, the system hardware in a hardware (dynamics) simulator, the software running on a virtualised platform, system user by the augmented or virtual reality simulator environments, and simulator interfaces between the system-of-interest and the other components (system environment, system hardware, material input/output, and system user).

3. Requirements specification for IMOCO4.E

This chapter explains the requirements gathering process, requirements classification, and the requirements coding scheme. After several discussions with task and deliverable owners of T2.3, T3.1, T4.1 and T5.1 (the tasks related to requirements and specifications), these were finalised. Further refinements were done internally with the WP2 core team. The requirements gathered, priority classification, verification method and identifying the related tasks help meet the ST, SI, SO and SE objectives of the IMOCO4.E project [1] for the overall requirements specification. In this deliverable, a one-to-one mapping of these objectives to the requirements is not done.

3.1 Requirements gathering process

First, an initial requirements classification and requirements coding scheme were finalised (after several discussions with relevant partners). Then, an initial requirement gathering survey template was formulated. ITEC as the lead partner of task T2.3, filled in this survey to understand the challenges and difficulties. The survey was refined based on ITEC's feedback. The requirements gathering process started with a T2.3 kick-off workshop. The details of the requirements gathering survey and the process were explained to all beneficiaries/partners in the consortium. The steps involved for a partner to fill in a requirement were explained through a flow chart (illustrated in Figure 11).

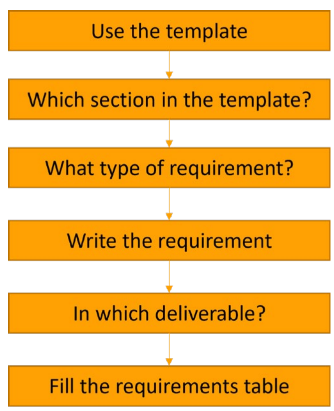

Figure 12. Steps followed by a beneficiary while filling in the requirement

Every partner was asked to use the provided template document in MS Word and upload it to the IMOCO4.E SharePoint. The choice of MS Word over MS Excel for the template was intentional due to better formatting options and the possibility to include math equations available in MS Word. The template document contained the following sections: (1) requirements specification for IMOCO4.E – this section explained the requirements classification and coding scheme followed; (2) overall (system-level) requirements; (3) architecture layer requirements; (4) BB requirements; (5) Pilot requirements[1]; (6) Demonstrator requirements[1]; and (7) Use case requirements[1]. The type of requirement refers to the requirements classification (as detailed in Section 3.2).

After identifying the section in the template and the type of the requirement, the partner can write the requirement. The partners were asked to adhere to KISS (Keep It Simple Short) and SMART (Smart, Measurable, Achievable, Realistic, Time-bound) [6] objectives while writing their requirements. The

[1] These requirements are specific to the application, and likely confidential. Hence not included here in D2.3, since D2.3 is generic and public.

partners were also asked to provide their suggestions on which deliverable (they believe) is the requirement suited for. Partners were also asked to provide the priorities, verification method, comments and the related IMOCO4.E tasks for the written requirement.

The partners who completed the survey would upload their templates to a common SharePoint folder. The decision to gather templates from the partners separately was made so that tasks 3.1, 4.1 and 5.1 can track the requirements per partner for their deliverables (as not all partners are involved in all these tasks). ITEC then collated the requirements from each partner based on the defined sections in D2.3. BB owners and partners in the IMOCO4.E consortium were invited to another WP2 workshop to work on finalising the requirements. BB owners were asked to check the overall BB-specific requirements for consistency, redundancy, and finalisation. The reviewed requirements are now part of this deliverable.

3.2 Requirements classification

We classify the requirements using the following characteristics (derived and extended from the ISO 25010 standard on software and data quality [7]):

1. *Interfaces and connectivity* – represent the interfaces and connectivity explained in the IMOCO4.E reference framework. Interfaces and connectivity exist between the architecture layers, building blocks and platforms.
2. *Maintainability* - represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in the environment and requirements. This characteristic is composed of the following sub-characteristics:
    a. *Modularity* - A system is modular when it can be decomposed into several components that may be mixed and matched in various configurations. The components can connect, interact, or exchange resources by adhering to a standardised interface.
    b. *Analysability* - Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, diagnose a product for deficiencies or causes of failures, or identify parts to be modified.
    c. *Testability* - Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.
3. *Performance* - This characteristic represents the performance relative to the number of resources used under stated conditions.
4. *Compatibility* - Degree to which a product, system or component can exchange information with other products, systems or components and perform its required functions while sharing the same hardware or software environment.
    a. *Interoperability* - Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.
    b. *Co-existence* - The degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products without detrimental impact on any other product.
5. *Usability* - Degree to which specified users can use a product or system to achieve defined goals with effectiveness, efficiency and satisfaction in a specified context of use.
    a. *Operability* - The degree to which a product or system has attributes that make it easy to operate and control.

        The systems have to operate in various environments, e.g. semiconductor, physical and chemical (cleanroom) laboratory environments, and automotive production areas with welding equipment. There will not be a one size fits all boundary constraint. The main differences will be in:

- Measurement ranges of the physical quantities and their tolerances w.r.t. to their electrical representation
- Temperature, pressure, humidity range
- Pollution degree
- Power quality
- EM environment, including EM-fields from nearby wireless connectivity, motion control and wireless power transfer (WPT)

6. *Reliability* - Degree to which a system, product or component performs specified functions under specified conditions for a specified period.
7. *Security* - The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorisation.
8. *Portability* - IMOCO4.E methodology will enable each machine to maintain excellent performance under slight variations in machine conditions, using ML and advanced learning control. This allows the portability of production processes across multiple machines since processes will run almost identically on these machines.
   a. *Adaptability* - The degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.
9. *Cost* – The cost of material goods, development or hardware for the system-in-consideration
10. *Scalability* – The ease with which a system or component can be modified to accommodate increased usage (resources, platform – hardware and software), increased datasets, and remains maintainable.
11. *Tools/toolchains* – Software tooling and toolchains that facilitate the design, development, deployment, operation and maintenance of the system under consideration (e.g. machine, factory).
12. *Safety* - The term safety, in particular, applies in the IMOCO4.E methodology to the human environment w.r.t. generated noise, pollution, radiation, and dangerous motion from autonomous robots and production machinery.
13. *Functional suitability* - represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

3.3 Requirement coding scheme

Each requirement ID is prefixed with Rxxx (xxx – the requirement number), the deliverable ID (D2.3 for this deliverable), the applicable IMOCO4.E relation(s),

- *Lx*: layer x
- *Bx*: BB x
- *Px*: pilot x
- *Dx*: demonstrator x
- *Ux*: use-case x

the optional reference framework-specific relation,

- *hw*: hardware
- *sw*: software
- *fw*: firmware

- *com*: communication

and the optional requirement classifier.

- *SAF*: safety
- *SEC*: security
- *DAT*: data

E.g., R003-D2.3-L2-L3-L4-B5-B6-B8-B10, R100-D2.3-P2-hw-SAF, R010-D2.3-P2

We ensure that the requirement IDs are unique so that the other deliverables can reference the defined requirement IDs within the IMOCO4.E project. For brevity during reference, we have unique numbering to the requirements. Throughout this document, any reference, e.g., to R003, refers to the corresponding requirement, i.e., R003-D2.3-L2-L3-L4-B5-B6-B8-B10.

The requirements are prioritised through the MoSCoW method [8].

- *M*: must have (necessary requirements for the IMOCO4.E project)
- *S*: should have (additional desired requirements with high priority)
- *C*: could have (additional requirements with low priority)
- *W*: would have (future requirements, ideally after the completion of the IMOCO4.E project)

We consider the IADT requirement verification method [9] as follows:

- *I*: inspection (observation using basic senses)
- *D*: demonstration (use the system as it is intended)
- *T*: test (more precise and controlled demonstration using scientific principles and procedures)
- *A*: analysis (validation of the system by scientific methods)

The expected technical maturity will be quantified using the technology readiness level (TRL) criteria [10].

	TRL	Description
Research	1	Basic principles observed
	2	Technology concept formulated
	3	Experimental proof of concept
Development	4	Technology validated in lab
	5	Technology validated in (industrially) relevant environment
	6	Technology demonstrated in (industrially) relevant environment
Deployment	7	System prototype demonstration in operational environment
	8	System complete and qualified
	9	Actual system proven in operational environment

4. Overall (system-level) requirements

In this chapter, the overall system-level requirements are specified. IMOCO4.E will combine a systematic, iterative requirement analysis activity exploiting knowledge from the partners gained in the motion control field with the iterative development and refinement of an architecture, method, and platform for accurate motion control in mechatronic systems. Overall (system-level) requirements refer to the generic requirements that span multiple architecture layers and/or BBs.

Table 1. Overall (system-level) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R001-D2.3	IMOCO4.E reference framework shall only implement interfaces and protocols with an open standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2
R002-D2.3	All software interfaces in the IMOCO4.E reference framework shall comply with a documented standard	M	I		T3.1, T4.1, T5.1, T7.1, T7.2
R003-D2.3-L2-L3-L4-B5-B6-B8-B10	AI-components, control algorithms, and digital twin models may use additional data or sensory input interfaces to train the model. However, after completion, it shall only make use of existing data and interfaces of the brown field system.	S	I	Detailed specifications would be part of D4.2 and D5.2	T4.1 T5.1
R004-D2.3-D4	Internet connection for the possibility of remotely inspecting the behaviour of perception and control modules	S	I		T3.1, T7.2
R005-D2.3-D4	Screen and input hardware for inspection and correction of perception and control modules	S	I		T3.1, T7.2
R006-D2.3-L1-hw-sw	The IMOCO4.E reference framework shall support EtherCAT connectivity	S	I		T3.1, T4.1, T5.1
Maintainability (modularity, analysability, testability)					
R007-D2.3-L2-L3-L4-B5-B6-B8-B10	AI-components, (control) algorithms, and digital twin models shall be documented (input, output, parameter interface and user guidance).	M	I	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1
R008-D2.3-L2-L3-L4-B5-B6-B8-B10	AI-components, (control) algorithms, and digital twin models shall be testable in simulation (e.g. by means of digital twins) and deployable on the physical target.	M	T	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1

R009-D2.3	The developed system is easily extendable	S	D	E.g. EtherCAT device can be added	T3.1, T4.1
Performance					
R010-D2.3-L1	Predictive maintenance of inverter components provides the reliable message that some components prone to wear out deteriorate in performance and require maintenance.	S	D		T5.1, T5.3
Compatibility (interoperability, co-existence)					
R011-D2.3-L1	Storage, transportability, installation and disposal (of machines and components) need to satisfy formal regulations: Restriction of hazardous substances (RoHS) directive/ Waste Electrical and Electronic Equipment (WEEE) Compliance	M	I		T6.5, T6.6
R012-D2.3-L2-L3-L4-B5-B6-B8-B10	All (smart) control algorithms, AI-components and digital twin models shall be compatible with different data / sample rates.	S	D	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1
R013-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	All smart control algorithms, AI-components and digital twin models shall be compatible with or can be integrated into Matlab / Simulink environment.	S	D	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1
R014-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	All smart control algorithms, AI-components and digital twin models shall be compatible and/or configurable/tunable for different variations of similar system / robot	S	T	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1
R015-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	All smart control algorithms, AI-components and digital twin models that are intended for real-time operation shall be compatible with code generation from Simulink.	S	I	Detailed specifications would be part of D4.2 and D5.2	T3.1, T4.1, T5.1
R016-D2.3-U3-hw-sw	Co-existence of Information Technology (IT) and Operation Technology (OT) on the same network infrastructure will be supported.	M	I		T3.3, T3.4
R017-D2.3-L1-L2	Co-existence, interoperability with I-MECH BBs	S	I		T6.1
Usability (operability)					
R018-D2.3	The production machines shall operate autonomously with minimum intervention for maintenance. Most production machines shall be operational 24/7. The designs need to be	S	D		T5.1

	forecastable maintainable with as short as possible intervention.				
R019-D2.3-P5-hw	HMI, wherever applicable, must be intuitive and clear.	M	I		T4.5, T6.4, T7.1
R020-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	Algorithms, AI-components and digital twin models that are intended for real-time deployment shall not adversely affect the responsiveness of the system to user requests.	M	T		T3.1, T4.1, T5.1
R021-D2.3-L2-L3-L4-B5-B6-B8-B10-P4	Algorithms, AI-components and digital twin models that are intended for relieving the operator from complex tasks shall be safe and predictable.	M	T		T4.1, T5.1
Reliability (fault tolerance, availability)					
R022-D2.3	Availability control – the semiconductor components intended for use within the IMOCO4.E project are available on the market	S	I	Situation in the semiconductor market	T6.5, T6.6
R023-D2.3-P5-hw	Components used (LIDARs, cameras, sensors, actuators, etc.) must meet the application environmental conditions. E.g. components in the manipulator for Pilot 5 must meet the requirements for mining use - class IP65 minimum classified enclosures. Operating temperature: -20 °C - +65 °C.	M	T		T4.1, T4.5, T7.1
R024-D2.3	Recovery of the system to a previous point in time (e.g. restore from backup) must leave the overall system in a consistent state. It is allowed for the system to be eventually consistent.	S	T	Especially in distributed architectures, this becomes an issue when, e.g. state changes cannot be replayed between microservices.	T6.1, T6.4
Security (cyber-security, integrity, confidentiality, authenticity)					
R025-D2.3-P5-hw	The control system algorithms must be ported as binaries/executables. Access level authentication must exist.	S	I	A must-have for Pilot 5.	T4.5, T6.4, T7.1
R026-D2.3	Compliance with data privacy regulations (e.g. GDPR, PIPL) and related cybersecurity regulations or frameworks (e.g. https://www.iotsecurityfoundation.org/iotsf-issues-update-to-popular-iot-security-compliance-framework/)	M	I		T6.2, T6.4
R027-D2.3	The compromise of a single asset/component of the system should not compromise the security	M	T		T6.4

	of other components. For example, every device must have unique security credentials (such as client certificates), and a layered approach must be taken to protect cloud infrastructure.				
R028-D2.3	External API (via ethernet) should be secure	S	T		T6.4
Portability (adaptability, replaceability)					
R029-D2.3-L2-L3-L4-B5-B6-B8-B10	If a (digital twin) model or algorithm is applicable to multiple layers (e.g. for real-time deployment and for condition monitoring), it will allow for easy adaptability/re-use across, for instance, by the selection of variants of different abstraction levels.	C	I	Detailed specifications would be part of D4.2 and D5.2	T4.1 T5.1
R030-P5-hw	Components used must be easily replaceable in its application environment (e.g. in an underground environment or factory production line).	M	I		T4.5
Tools/toolchains					
R031-D2.3	IMOCO4.E reference framework shall provide support for model-based design. A set of tools and toolchain(s) are selected and configured to support a model-driven engineering approach.	M	D		T2.3, T3.1, T4.1, T5.1, T6.1
R032-D2.3	IMOCO4.E reference framework shall integrate MIL/SIL/PIL/HIL simulation and validation toolchains	M	D		T2.3, T6.1
R033-D2.3	IMOCO4.E reference framework tools/toolchains shall use open standard(s) for i.data storage ii.communication with other tooling	M	D		T2.3, T6.1
R034-D2.3	IMOCO4.E reference framework, tools and toolchains shall have a shallow learning-curve	S	D		T6.1
R035-D2.3	IMOCO4.E reference framework should interface with the domain expert user following the domain concepts, terminology and workflow	S	I		T2.3, T6.1
R036-D2.3	IMOCO4.E framework shall be reliable	M	D		T6.1
R037-D2.3-L2-L3-L4-B5-B6-B8-B10	All smart control algorithms, AI-components and digital twin models that are intended for real-time deployment shall be compatible with code generation from Matlab / Simulink.	M	I	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1, T6.1
R038-D2.3-L2-L3-L4-B5-B6-B8-B10	All smart control algorithms, AI-components and digital twin models shall be testable in a simulation / virtual environment.	M	I	Detailed specifications would be part of D4.2 and D5.2	T4.1, T5.1, T6.1
R039-D2.3	Should support FMI/FMU approach	S	I		T6.1
Safety					

R040-D2.3	Safety software needs to be fail-safe	M	T		T6.4
R041-D2.3	**Electromagnetic compatibility: emission and immunity requirements** All electric and electronic equipment must satisfy the EMC directives, as applicable to the products considered: 2014/30/E.U. (EMC directive) IEC/EN 55011/61326	S	I		T6.4
R042-D2.3	**Radio equipment, Radio frequency emissions:** All products that incorporate wireless and/or radio-related functions must satisfy the Radio Equipment Directive (2014/53/E.U.). The EMC requirements are superseded, i.e., extended by the ETS 301-489-1. Other applicable standards: IEC 61000-6-2 EMF immunity (industrial) IEC 61000-6-4 EMF emission (industrial)	S	I		T6.4
R043-D2.3	**Electrical safety** All electric and electronic autonomous robots and production machinery need to be electrical safe according to the international requirements (and their national deviations): IEC 60601 (medical equipment, depending on application) IEC 60204-1 (electrical safety of machines) 2006/42/E.C. (Machine directive) 2014/35/E.U. (low-voltage directive)	S	I		T6.4
R044-D2.3	Compliance with safety standards. People should be safe to operate in the same room the machines (Pilots/Use-cases/Demonstrators) are operating.	M	I		T6.4
R045-D2.3-D4	Compliance with good manufacturing practice. Parts must not disconnect and contaminate product or packaging.	M	I	Particularly relevant for cosmetics	T6.4
R046-P5-hw	Motion control functionalities must cover mobile machine safety directives	M	T	For mobile machines/robots	T6.4
R047-D2.3-L2-L3-L4-B5-B6-B8-B10	Any smart control algorithms, AI-components and digital twin models shall not adversely affect the safety of the system.	M	T		T3.1, T4.1, T5.1, T6.1
R048-D2.3-P3-P5-hw	Collision avoidance methods must meet application-specific requirements (e.g., mining safety operations or medical applications).	M	I		T4.5, T7.1

5. Architecture layer requirements

In this Chapter, we summarise the overall architecture layer-specific requirements. The four architecture layers we have defined in the IMOCO4.E reference framework are:

i) Layer 1 – sensors and actuators
ii) Layer 2 – centralised controller
iii) Layer 3 – system behaviour
iv) Layer 4 – digital twin and AI analytics

The mission of the IMOCO4.E project is to bring adequate edge intelligence into the Instrumentation and Control Layers, analyse and process machine data at the appropriate levels of the feedback control loops, and synchronise the digital twins with either the simulated or the real-time physical world. At all layers, AI techniques are employable.

Table 2. Architecture layer-specific requirements

ID	Requirement	Priority	Verify	Tasks
R049-D2.3-L1	The interface between Layer 1 and Layer 2 shall be time-critical and support sample rates of at least 20 kHz	S	T	T3.1, T4.1, T6.1
R050-D2.3-L1-com	The interface between Layer 1 and Layer 2 shall be an industry standard (e.g. EtherCAT)	M	I	T3.5, T6.1
R051-D2.3-L1	Wireless connectivity of sensors for mechanical manifestations acquisition.	S	D	T3.2
R052-D2.3-L1-hw-DAT	Sensors monitoring the condition of an asset should be able to push measurements to the IMOCO4.E big-data infrastructure.	M	T	T5.1, T6.1
R053-D2.3-L1-hw	Layer 1 should be able to connect additional sensors for optimal motion control	S	I	T3.5
R054-D2.3-L4	Positions/rotations of moving parts are required on interface to provide visual feedback.	S	I	T5.1, T6.4
R055-D2.3-L1	Sensors provide sufficient diagnostic information to evaluate their weaknesses or damages.	C	D	T3.2
R056-D2.3-L2	Automatic testing should be applied to all control functionalities/algorithms	S	T	T4.1
R057-D2.3-L3-D4	Systems perception models should be retrainable by a non Machine Learning expert.	S	D	T3.1
R058-D2.3-L2	The platform should support real-time sensor data processing for localisation and navigation calculation	M	D	T4.5
R059-D2.3-L1-hw-sw-fw-com	Feed-forwarded data w.r.t. motion excitation (VAJ) needs to be available (in time) to enable (EMI-noise) compensation (applicable to actuators)	S	I	T3.5
R060-D2.3-L1-hw-DAT	Sensors monitoring the condition of an asset should provide real-time measurements without disruptions	S	T	T3.1
R061-D2.3-L4	Cloud-hosted company data should be accessible to the company only.	M	I	T6.4

R062-D2.3-L1	Hardware components must be interchangeable; boundary constraints must be set: volume, temperature, power supply, etc.	S	I	T3.1
R063-D2.3-L2	The optimised Neural Networks must be able to run on the available (brownfield) hardware platforms.	M	D	T3.3, T3.4
R064-D2.3-L4	System must be (cloud-)vendor neutral. Specific dependency on vendor-specific solutions (such as Azure IoT) must be prevented unless other providers provide similar functionality (such as hosted Kubernetes).	M	I	T5.1, T6.4
R065-D2.3-L1-hw	The hardware needed to compensate for EMI noise needs to be more compact, lighter, flexible, cheaper than conventional shielding and filtering	S	I	T3.1, T6.5
R066-D2.3-L1	Implemented sensors communication interface/protocol allows simultaneous connection of multiple devices.	C	D	T3.2
R067-D2.3-L1-hw-sw	Needs to be available from low to high power applications, low voltage DC to high(er) voltage AC.	S	I	T3.5
R068-D2.3-L4-sw-DAT	Big data infrastructure must be able to accept and store streams of data from all sensors of Layer 1 enabling both historical and real-time data analysis.	M	T	T5.1, T6.4
R069-D2.3-L1-hw-DAT	A sufficient set of sensors for providing a complete view of the condition of all assets, for which predictive maintenance methods must be applied, should be guaranteed. Note: The sensors under development in the IMOCO4.E project may not be exhaustive	C/W	I	T3.2

6. Building block requirements

In this chapter, the overall BB-specific requirements are specified. Requirements that are generic for the BB and required for interfacing the BB component with other IMOCO4.E framework components, e.g., BBs, are specified in this deliverable. Detailed specifications or application-specific requirements would be part of future deliverables.

6.1 BB1 – SoC/FPGA platforms for smart control and signal processing

BB1 improves the control profile of high-performance applications by offloading application-specific functionalities to the FPGA and co-processors present in the edge network. The direct interface to digital signals yields lower latency, improved system operations, and better local control guidance for digital twins. The Building Block will show data gathering directly from drives and machine sensors, data pre-processing, local control loops and bounded-latency communications with the upper layer. The BB will consider high sampling/signal processing rates as well as the implementation of different electrical and functional interfaces in a modular way, aligned with the pilot, demonstrator and use case requirements. Interoperability is a key feature achieved through standard communication modules, such as TSN. Signals and events will be exposed to the user through detailed diagnostics/monitoring interfaces.

Table 3. BB1 (SoC/FPGA platforms for smart control and signal processing) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R070-D2.3	The interfaces to BB1 shall be an industry standard	M	I		T3.1
R071-D2.3	BB1 shall support standard and vendor-neutral Wired 1G Ethernet	M	I	1000-Base-T (Copper twisted pair), 1000-Base-X (Optical fibre). Expected TRL: 7	T3.3, T3.4
R072-D2.3	BB1 shall support standard API between application controller and distributed edge instance, exposing local configuration and telemetry.	C	T	The MPSoC FPGA platform will provide standard and vendor-neutral API (such as OPC-UA), exposing control and monitoring to the application controller. Expected TRL: 5	T3.4
R073-D2.3	IEEE 802.1Q Mixed-critical communication support	M	T	Time-constrained, rate-constrained and best-effort QoS concurrently delivered. IEEE 802.1Qbv time-aware traffic shaper will be instantiated in logic.	T3.3, T3.4

				Expected TRL: 5-6	
R074-D2.3	The BB1 should have interface with camera sensors	M	T		T3.4
R075-D2.3	The BB1 should have enough memory to allow for buffering more than 6 image from the camera sensors	M	T		T3.4
Maintainability (modularity, analysability, testability)					
R076-D2.3	Time Sensitive Network data plane will expose telemetry monitoring and configuration to a centralised TSN data management	M	I	Netconf and Yang data structures will be used to provide these capabilities to a Centralised Controller. Expected TRL: 3	T3.4
Performance					
R077-D2.3	The interface to/from BB1 shall support update rates of at least 20 kHz to Layer 2 and/or BBs	M	D		T3.1
R078-D2.3	Sub-µs time synchronisation based on IEEE 1588 and IEEE 802.1AS	M	T	Network time exposed to distributed instances implemented on software of programmable logic. Expected TRL: 6	T3.3
R079-D2.3	End to End deterministic latency for time-constrained TSN data streams.	M	T	Safety-critical data streams will be delivered with low latency and minimum latency deviation (jitter) in the presence of lower priority data streams. Expected TRL: 6	T3.1
R080-D2.3	Delivery guarantee for rate constrained TSN data streams.	M	T	Bandwidth guarantee for prioritised data streams in the presence of lower priority data streams.	T3.1
Compatibility (interoperability, co-existence)					
R081-D2.3	Software and Programmable Logic based applications can be instantiated either on Xilinx or Intel Altera MPSoC FPGA platforms	S	I		T3.3, T3.4
Usability (operability)					
R082-D2.3	BB1 shall have a configuration interface to modify all (pre-defined) configuration parameters without requiring firmware changes	M	D		T3.1
Portability (adaptability, replaceability)					
R083-D2.3	BB1 shall offer ways of working and toolchains that extend across multiple (generations of) platforms	M	D		T3.1
Cost					

R084-D2.3	A basic version of BB1 shall have a target cost of goods in the order of €200	M	I		T3.1
Scalability					
R085-D2.3	BB1 shall offer a scalable amount of computational resources, e.g. by means of the firmware implementation or by offering a family of processing units with different capacities	M	D		T3.1
Tools/toolchains					
R086-D2.3	BB1 shall use a toolchain that is open-source or industry-standard	S	I		T3.1
Safety					
R087-D2.3	Exchanging control between low level and high-level controllers shall not affect human or machine safety of the total solution	M	T		T3.1

6.2 BB2 – high-speed vision-in-the-loop

BB2 will optimise high-speed vision architectures and AI and digital twin algorithms for the deployment on embedded "edge" devices (Embedded GPUs, FPGAs, MPUs, ASICs), with applications in perception, localisation, planning, maintenance. This building block will be deployable as a smart sensor for higher control layers.
Furthermore, BB2 will generate input data via imaging and high-performance computing for digital twin implementations using novel architectures under strict time-sensitive constraints.

Table 4. BB2 (high-speed vision-in-the-loop) requirements

ID	Requirement	Priority	Verify	Tasks
R088-D2.3-B2-P2	BB2 shall use standardised interfaces to the various involved layers (Layer 1 and Layer 2 specifically)	M	I	T3.1, T4.1
R089-D2.3-B2-P2	Must have physical mounting plate and volume claim	M	I	T3.1
R090-D2.3-B2-P2	BB2 must offer interfacing to industrial cameras via an industry-standard interface	M	D	T3.1
R091-D2.3-B2-P2	BB2 can be connected to Matlab/Simulink	C	A	T3.1, T4.1
R092-D2.3-B2-P2	BB2 shall be modular with testable interfaces	S	I	T3.1
R093-D2.3-B2-P2	The low-cost BB2 implementation shall offer a vision position update rate of at least 75 Hz	M	D	T3.1
R094-D2.3-B2-D4	Object detection and pose estimation model processes > 5 frames per second	S	D	T3.1
R095-D2.3-B2-P2	The high performance BB2 implementation must have frame rate of 5 kHz	M	I	T3.1
R096-D2.3-B2-P2	Must have Latency <300 µsec	M	I	T3.1
R097-D2.3-B2-P2	Must have Image frame size > 0.25 Mpixel	M	I	T3.1
R098-D2.3-B2-P2	BB2 shall have a configuration interface to modify all (pre-defined) configuration parameters without requiring firmware changes	M	D	T3.1
R099-D2.3-B2-P2	BB2 must be able to operate disconnected from connections to the internet	M	I	T3.1
R100-D2.3-B2-P2	BB2 shall offer customizability such that non-standard algorithms can be implemented	M	D	T3.1
R101-D2.3-B2-P2	BB2 shall have a cost target of <1000€, excluding camera, optics and motion drives (for lower frame rates)	M	I	T3.1, T7.1
R102-D2.3-B2-P2	BB2 shall have a cost target of <25000€, including camera; excl optics and motion drives. (for higher frame rates)	C	I	T3.1, T7.1
R103-D2.3-B2-P2	BB2 shall offer a scalable processing platform, to enable higher update rates or more complicated image processing pipelines if required	M	I	T3.1, T7.1
R104-D2.3-B2-P2	BB2 would preferably be based on open software tooling	M	I	T3.1
R105-D2.3-B2-P2	Must be safe for humans, products, machine/system and environment	M	D	T3.1, T4.1, T5.1

6.3 BB3 - Novel sensors (new type of sensors, wireless communications, self-powered, low-powered)

BB3 concerns the sensors to be developed that will be incorporated in Layer 1. Several sensors targeting different parameters are pursued, including physical/mechanical parameters such as vibration, temperature, and pressure; radar sensor; 3D depth sensor; vision sensor; and 2D/3D camera. These sensors must comply with the requirements defined by the respective Pilots/Use-cases/Demos in which they will be deployed as well as those originated by the needs of the upper layers/BBs that will use the data generated.

Table 5. BB3 (novel sensors) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R106-D2.3-B3	Wireless communication interface for sensors not located in the closed vicinity of centralised controller.	S	D		T3.2
R107-D2.3-B3	Low energy wireless communication interface for sensors with at least 500 kbit/s burst data rate and operating range with at least a few tens of meters. Network star topology is preferred.	C	T		T3.2
R108-D2.3-D2	Controller must provide communication and power supply, both wireless.	M	D	Specifications would be part of D3.1/D3.2	T3.3
Performance					
R109-D2.3-D2	Sensors must be able to read temperatures within the range of -40ºC to 85ºC, with at least 0.5ºC accuracy and in the range of 0ºC to 45ºC with at least 0.3ºC accuracy.	M	T	Specifications would be part of D3.1/D3.2	T3.2
R110-D2.3-D2	Sensors must be able to read variations of pressure and temperature at at least 10 Hz.	M	T	Specifications would be part of D3.1/D3.2	T3.2
R111-D2.3-D2	Pressure and temperature measurement data must be communicated at at least 1 Hz.	M	T		T3.2
Usability (operability)					
R112-D2.3-B3	Power supply for expected lifetime operation integrated with the sensor, e.g. battery power, wireless power or energy harvesting, depending on the device's power requirements.	C	D		T3.2
R113-D2.3-B3	Sensors analyse their internal performance parameters to evaluate the reliability of output data and report this information.	C	D		T3.2

6.4 BB4 - Real-Time Smart-Control Platform

BB4 provides a common computational platform where advanced sensors or TSN networking devices are interfaced. Systems following the recommended setup enable fast and real-time execution of compute-intensive AI workloads at the edge and reduce the latency to transmit the data to the digital twin. In the proposed IMOCO architecture, BB4 covers Layer 2 and is connected to Layers 1 and 3, from which input data are gathered by Layer 1 and processed using algorithms from Layer 3.

Table 6. BB4 (real-time smart control platform) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R114-D2.3	BB4 shall offer industry standard interfaces to: - Encoders (e.g. Biss-C, EnDAT) - Drives (e.g. EtherCAT) - Layers 1 and 2 (e.g. EtherCAT)	M	I	COTS components are needed to interoperate with TSN	T3.1, T4.1, T5.1
R115-D2.3	BB4 must be able to run on an “ARM” based platform.	M	I		T4.1
R116-D2.3	BB4 can be connected to Matlab/Simulink	C	A		T4.1
Maintainability (modularity, analysability, testability)					
R117-D2.3	BB4 shall be ready for the vertical distribution of smart control algorithms	M	D	Interfacing with BB5	T4.1
Performance					
R118-D2.3	BB4 shall support control loop update rates of at least 20 kHz	M	D		T3.1 T4.1
Compatibility (interoperability, co-existence)					
R119-D2.3-B5	BB4 should be compatible with code generated from Simulink	M	D	Interfacing with BB5	T4.1
R120-D2.3	BB4 shall be compatible and portable with x86-based platforms	S	D		T4.1
Portability (adaptability, replaceability)					
R121-D2.3	BB4 shall offer customizability to run any combination of custom control loops in parallel, including MIMO control loops	M	D		T4.1
R122-D2.3	BB4 shall offer customizability such that non-standard tasks (i.e., tasks which are typically performed in research) can be performed. Examples include flexibility in allowed controller structures and reference/feedforward signals.	S	T		T4.1
Cost					
R123-D2.3	BB4 shall have a target cost of goods of €1000 for a basic version	M	I		T4.1

6.5 BB5 - Smart control algorithms library

BB5 constitutes a framework for smart control algorithms. The framework covers key solutions to improve the dynamic and accuracy performance of mechatronic systems, covering both centralised and decentralised approaches. These algorithms can be both model-based and data-driven and build upon reliable knowledge of the dynamics of the system, obtained via physical modelling, data-driven learning (system identification), or both. BB5 covers Layer 2 and Layer 3 in the proposed IMOCO4.E reference architecture. While the core of the algorithms is the main software part in Layer 2, all the configuration and diagnosis are done through Layer 3.

Table 7. BB5 (smart control algorithms library) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R124-D2.3	AI-based algorithms should be compatible with commercially available TPU's	C	I	Allows for application in embedded solutions	T4.1
R125-D2.3-L2	BB5 can be connected to Matlab/Simulink (Layer 2 to Layer 3) for configuration, etc.	C	A		T4.1
Maintainability (modularity, analysability, testability)					
R126-D2.3-L2	All smart control algorithms shall have a clear documentation that explains input, outputs, description, and parameter settings	M	I		T4.1
R127-D2.3-L2	Smart control algorithms and models shall be tested in simulation	M	T	Validated in WP6	T4.1, T4.2
R128-D2.3-L2	Control functionalities should be able to be tested by automatic means and accordingly documented (requirement traceability)	C	T		T4.1
Compatibility (interoperability, co-existence)					
R129-D2.3-L2-L3	The library in BB5 shall be compatible with BB4	M	D		T4.1, T4.3, T4.4, T4.6
R130-D2.3-L2-L3	All models should be compatible with Matlab/Simulink	S	D		T4.2, T4.3, T4.4
R131-D2.3-L2-L4	Modelling should be compatible with code generation tools	M	D		T4.1, T4.2, T4.3, T4.4
R132-D2.3	BB5 shall be compatible with smart control blocks developed in I-MECH	S	D		T4.2, T4.3, T4.4
Usability (operability)					

R133-D2.3	BB5 shall be able to be executed in real-time on provided execution platforms (e.g. via BB1, BB4)	M	T		T4.1
Reliability (fault tolerance, availability)					
R134-D2.3-L2-L4	Control algorithms will have self-diagnosis functions	S	I		T4.1
Portability (adaptability, replaceability)					
R135-D2.3	BB5 shall offer customizability such that non-standard tasks (i.e., tasks which are typically performed in research) can be performed. Examples include flexibility in allowed controller structures and reference/feedforward signals.	S	T		T4.1
Safety					
R136-D2.3-L2-L4	Smart control algorithms of collaborative robots (Cobots) need to be compliant with safety standards	M	A		T4.1, T4.2, T4.3, T4.4
Digital twin					
R137-D2.3-L2-L4	Data-driven models shall be compared to analytical models and/or validated real robots	S	A		T4.1, T4.2, T4.3, T4.4

6.6 BB6 - Algorithms for condition monitoring, predictive maintenance and self-commissioning of industrial motion control systems

BB6 will provide a framework for meaningful data selection, condition indicators computation and models and digital twins training for classification and prediction. The BB6 will provide software components for AI-augmented condition monitoring, self-commissioning and predictive maintenance. BB6 components will fulfil the general system requirements of the IMOCO4.E framework with respect to interoperability functionalities. The algorithms will be implementable on different hardware platforms.

Table 8. BB6 (algorithms for condition monitoring, predictive maintenance and self-commissioning of industrial motion control systems) requirements

ID	Requirement	Priority	Verify	Comments	Tasks
Interfaces and connectivity					
R138-D2.3-D4	Internet connection for the possibility to remotely inspect behaviour of hardware modules relevant for condition monitoring, predictive maintenance and self-commissioning of industrial motion control systems	S	I		T5.1
Performance					

R139-D2.3-B6	Condition monitoring and predictive maintenance are capable of running on different layers depending on computational complexity and memory requirements.	S	T		T5.1, T5.3, T5.7
R140-D2.3	High-frequent data-collection and logging of sensors / actuators signals upon failures in the system should be present.	M	I	Allow for creating a "database" of failures to learn predictive maintenance mod	T5.1
R141-D2.3-L4	Models for Virtual Commissioning purposes should include real-time communication limitations	M	I		
Compatibility (interoperability, co-existence)					
R142-D2.3-L4	Virtual Commissioning functionality should be compatible with major PLC manufacturers	M	I		T5.1
R143-D2.3	Self-commissioning shall be modular w.r.t. the system architectures and sizes	M	I		T5.1
Usability (operability)					
R144-D2.3-B6-sw	Appropriate visualisations should be able to provide insights into the operational health of monitored assets	S	D		T5.1
Portability (adaptability, replaceability)					
R145-D2.3	BB6 shall offer customizability such that non-standard tasks (i.e., tasks which are typically performed in research) can be performed. Examples include flexibility in allowed controller structures and reference/feedforward signals.	S	T		T5.1
R146-D2.3	Standardised interfaces shall be defined to guarantee replaceability of the components, if available.	M	I		T5.1
Scalability					
R147-D2.3	Model for Virtual Commissioning should determine standard interfaces so that system modelling can be adapted to real system complexity	M	A		T5.1
Functional suitability					
R148-D2.3-B6-sw	Predictive maintenance software components should create real-time notifications about anticipated malfunctions of monitored assets.	M	A		T5.1
Digital twin					
R149-D2.3	Self-Commission algorithms should be tested and provided with a XIL system	M	A		T5.1

6.7 BB7 - High performance servo-drives

This BB will deliver high performance, highly configurable current amplifier for servo control applications. This provides a flexible low-level actuator control in Layer 1, which can be used in high fidelity motion control platforms with stringent performance requirements.

Table 9. BB7 (high-performance servo-drives) requirements

ID	Requirement	Priority	Verify	Tasks
Interfaces and connectivity				
R150-D2.3	EtherCAT connectivity	M	D	T3.5
R151-D2.3	Biss-C protocol over RS422 (abs. encoder)	M	D	T3.5
R152-D2.3	RS422 conectivity (abs. encoder)	C	D	T3.5
R153-D2.3	Incremental encoder connectivity	M	D	T3.5
R154-D2.3	UART connectivity	C	D	T3.5
R155-D2.3	CAN connectivity	C	D	T3.5
R156-D2.3	Digital IO connectivity	M	D	T3.5
R157-D2.3	Motor brake connectivity	M	D	T3.5
R158-D2.3	Safe torque off IO	M	D	T3.5
R159-D2.3	DC motor connectivity	M	D	T3.5
R160-D2.3	Servo motor connectivity (FOC)	M	D	T3.5
R161-D2.3	BLDC motor connectivity	W	-	T3.5
Maintainability (modularity, analysability, testability)				
R162-D2.3-fw	Drive firmware parameters/variables monitoring (analysability, testability)	M	T	T3.5
R163-D2.3-fw	Drive firmware internal process monitoring (analysability, testability)	M	T	T3.5
R164-D2.3-hw	Drive hardware modular design (modularity, testability)	M	I	T3.5
Performance				
R165-D2.3	Provide fast data transfer (oversampled variables) of selected parameters/variables over EtherCAT from/to drive to develop and test smart control algorithms, e.g. repetitive control	M	T	T3.5
R166-D2.3	EtherCAT transfer rate with the servo drive shall be up to 1kHz	M	T	T3.5
R167-D2.3	Current loop rate up to 24 kHz	M	T	T3.5
R168-D2.3	Application loop rate up to 8 kHz	M	T	T3.5
Compatibility (interoperability, co-existence)				
R169-D2.3	Co-existence with I-MECH and IMOCO4.E BBs	C	D	T3.5
R170-D2.3	Compatibility with IEC 61800, ISO 15745, IEC 61784	S	D	T3.5
Usability (operability)				
R171-D2.3	The servo drive (BB7) shall be used for internal projects within the IMOCO4.E consortium (internally developed and used manipulators, test stands, etc., expected TRL 5)	M	T	T3.5

R172-D2.3	The servo drive (BB7) shall be used for external purposes (potential prototype applications outside the consortium, expected TRL 5)	W	-	T3.5
Reliability (fault tolerance, availability)				
R173-D2.3	HW design/testing for high reliability	S	T	T3.5
R174-D2.3	SW design/testing for high reliability	W	-	T3.5
Portability (adaptability, replaceability)				
R175-D2.3	Ability to adapt the drive hardware to different requirements (hardware, firmware)	S	I	T3.5
Scalability				
R176-D2.3	BB7 shall support multiple drives on an EtherCAT bus, the scalability will be achieved by adding multiple EtherCAT devices in the system	S	D	T3.5
Tools/toolchains				
R177-D2.3	Tools for drive firmware parameters/variables monitoring	M	T	T3.5
R178-D2.3	Tools for drive firmware internal process monitoring	M	T	T3.5
R179-D2.3	Tools to research, develop, monitor, tune smart control algorithms	M	T	T3.5
Safety				
R180-D2.3	Requirements for electrical safety IEC/EN 61800-5-1: Adjustable speed electrical power drive systems - Safety requirements - Electrical, thermal and energy	M	I	T3.5
Digital twin				
R181-D2.3	Real-time communication of signals/parameters/variables of drive subsystem functionality (e.g., FOC, velocity calculation) to be verified against the independent model (digital twin)	S	T	T3.5

6.8 BB8 - AI-based components

BB8 unites partners involved in AI-related activities into a single team, thus creating an opportunity for innovation and new research ideas. BB8 activities include AI-based vision-in-the-loop algorithm realisation, deep learning model optimisation for speed and efficiency, AI-based control algorithms, and other deep learning and AI-based algorithms development and deployment.

Table 10. BB8 (AI-based components) requirements

ID	Requirement	Priority	Verify	Tasks
R182-D2.3	To support integration across all layers, BB8 shall offer industry-standard interfaces to each of the IMOCO4.E layers to exchange data	M	I	T3.1, T4.1, T5.1
R183-D2.3	Interfaces to deploy learned networks are present Note: The main targets are BB1, BB2, BB5, and BB6	M	I	T3.1, T4.1, T5.1
R184-D2.3	On-site update in-the-field	S	I	T5.1
R185-D2.3-B8-D4	Sim2Real transfer provides synthetically trained object detection algorithms that detect objects of interest in 80% of images with said objects	S	D	T5.1
R186-D2.3	BB8 shall support real-time inference (limited and deterministic)	S	D	T3.1, T4.1, T5.1
R187-D2.3	Support and be operational in multiple Pilots/Demos/Use-cases	S	I	T3.1, T4.1, T5.1
R188-D2.3	Any user could operate (without expert knowledge)	S	D	T5.1
R189-D2.3	Minimize downtime	S	D	T5.1
R190-D2.3	BB8 shall offer AI components including one or more forms of verifiability, for example: - Providing a human-interpretable view of the algorithm - Providing a framework to assess reliability in a simulation/digital twin environment	M	D	T3.1, T4.1, T5.1
R191-D2.3	Only authorised users have access to systems and data	M	I	T5.1
R192-D2.3	BB8 shall offer customizability such that non-standard tasks (i.e., tasks which are typically performed in research) can be performed. Examples include flexibility in allowed controller structures and reference/feedforward signals.	S	T	T3.1, T4.1, T5.1
R193-D2.3	BB8 shall support a computing continuum in the sense that BB8 can operate in all layers, i.e. from the instrumentation layer up to the cloud layer.	M	D	T3.1, T4.1, T5.1
R194-D2.3	EU-first tools	S	I	T3.1, T4.1, T5.1
R195-D2.3	Digital twin supports the decision making of sensors in the case of errors	S	D	T3.1
R196-D2.3	Latency requirements will be adhered to by the digital twin	S	D	T3.1, T4.1
R197-D2.3	All (generated) software for hardware targets should also run on digital twin (unmodified).	S	D	T3.1, T4.1

6.9 BB9 - Cyber-security tools and trustworthy data management

BB9 will facilitate trustworthy and secure data exchange between the IMOCO4.E connected components and permanent storage. BB9 will be based on a distributed messaging system with a publish/subscribe model that will be able to handle data streams of multiple types exchanged between multiple endpoints in parallel and in real-time. BB9 will be able to aggregate, transform and fuse incoming data that will be persistently stored and become accessible as historical data. BB9 will only allow authorised users and components to access appropriate data and services, while data will be secured at rest and in flight. Data reliability will be ensured through data replication over secure channels. BB9 will be able to handle time-sensitive data streams in real-time while conforming to the bandwidth and latency requirements of connected IMOCO components.

BB9 will support the automated detection of cyber-security threats and vulnerabilities that can be inferred from applying anomaly detection techniques to the BB9 data streams. BB9 will alert the user in real-time if any supported cyber-security threat and vulnerability is detected and present an assessment. BB9 will provide a GUI for configuring operational parameters and for visualising information related to BB9 functions (e.g., system health status, data traffic, performance metrics, alerts) in the form of a dashboard.

The BB9 architecture will be modular, extendable and scalable, comprising a collection of containerised components that allow easy deployment and configuration as needed. Due to its interoperable design and lateral functionality, BB9 will be used to connect other BBs and will be applied in multiple Pilots / Demonstrators / Use Cases. Specifically, BB9 will facilitate the communication and data exchange between BBs by providing a framework based on a publish/subscribe messaging model, where components from all BBs will be able to release and consume information across multiple endpoints.

Table 11. BB9 (cyber-security tools and trustworthy data management) requirements

ID	Requirement	Priority	Verify	Tasks
R198-D2.3-B9-com-DAT	Support real-time information exchange with a protocol based on message set abstraction (publish/subscribe model) that is able to handle parallel data streams between multiple endpoints	M	D	T5.2
R199-D2.3-B9-com-DAT	BB9 will be able to aggregate, transform and fuse incoming text-based data from multiple sources and of multiple data types (e.g., time-series and cross-sectional data, real and simulated data, raw sensor data, inference result data from AI components).	M	D	T5.2
R200-D2.3-B9-com-DAT	BB9 will provide persistent storage for the aggregated and fused data (see R199-D2.3-B9-com-DAT) in the cloud infrastructure (historical data).	M	D	T5.2
R201-D2.3-B9-com-DAT	BB9 will allow all authorised components to access incoming data streams collected from multiple sources (see R199-D2.3-B9-com-DAT) in real-time via a dedicated API.	M	D	T5.2
R202-D2.3-B9-com-DAT	BB9 will allow all authorised components to access historical data stored in the cloud infrastructure (see R200-D2.3-B9-com-DAT) via a dedicated API.	M	D	T5.2
R203-D2.3-B9-sw	BB9 architecture to be based on microservices to be delivered in containerised form and deployed on the edge/cloud (e.g. using Docker/Kubernetes cluster)	S	D	T5.2

R204-D2.3-B9-com-DAT	BB9 will be able to handle time-sensitive data streams between multiple endpoints in real-time while conforming to the bandwidth and latency requirements of connected IMOCO4.E components.	S	T	T5.2 T3.4
R205-D2.3-B9-SEC	BB9 must be able to generate alerts in real-time (e.g., related to supported cyber-security threat detection, see R215-D2.3-B9-SEC).	M	D	T5.2
R206-D2.3-B9	BB9 will be designed to support and be operational in multiple Pilots/Demonstrators/Use Cases	S	D	T5.2
R207-D2.3-B9	A GUI will be provided for configuration purposes of BB9.	C	D	T5.2
R208-D2.3-B9	BB9 will provide an appropriate dashboard for visualising data and providing insight related to the operation of BB9 (e.g. system health status, data traffic, performance metrics, alerts)	C	D	T5.2
R209-D2.3-B9-DAT	Data safety will be ensured through Data Replication support over secure channels between the infrastructure cluster nodes.	S	D	T5.2
R210-D2.3-B9	BB9 will be able to continue operating despite receiving and processing invalid or wrong data.	S	D	T5.2
R211-D2.3-B9	BB9 will provide high computing availability, having a continuous, uninterrupted, fault-tolerant operation.	S	D	T5.2
R212-D2.3-B9-SEC	Only authorised users will be allowed to access the system.	S	D	T5.2
R213-D2.3-B9-SEC	Access to the system's data and services will be granted only to authenticated users and components that have been granted the necessary privileges.	S	D	T5.2
R214-D2.3-B9-SEC	Data security will be ensured at rest and in flight.	S	D	T5.2
R215-D2.3-B9-SEC	BB9 will support the automated detection of cyber-security threats and vulnerabilities that can be inferred from applying anomaly detection techniques to the BB9 data streams.	S	D	T5.2
R216-D2.3-B9-SEC	The system will alert the user if any supported cyber-security threat and vulnerability is detected and present an assessment (see R215-D2.3-B9-SEC).	S	D	T5.2
R217-D2.3-B9	BB9 will be fully scalable so that it can easily be adapted to new integration needs or changes in performance, reliability and data volume requirements.	S	D	T5.2
R218-D2.3-B9	All used libraries/frameworks/components must not have known security vulnerabilities nor infringement of (open source) license conditions.	S	D	T5.2

6.10 BB10 - Motion / path planning, collision avoidance and navigation algorithms

BB10 provides a framework of smart algorithms for path and motion planning, collision detection and avoidance, and autonomous navigation. The framework includes important solutions for the operation of mobile and stationary (manipulatory) robotic systems, ranging from autonomous warehouse transport vehicles to mining and tunnelling robots.

Table 12. BB10 (motion/path planning, collision avoidance and navigation algorithms) requirements

ID	Requirement	Priority	Verify	Tasks
R219-D2.3-B10-sw	The control system algorithms can be integrated in a real HIL testing environment.	M	T	T4.5
R220-D2.3	The LIDAR sensor must be suitable for the usage of SLAM.	M	D	T4.5
R221-D2.3	Enough processing power is needed to work with real-time sensor data (for localisation and navigation calculation).	M	D	T4.5
R222-D2.3-B10-sw	Visual servoing for motion control is based on real-time camera systems integrated into the control system.	M	T	T4.5
R223-D2.3-B10-sw	Path planning algorithm must be possible to be done in near-real-time in the control system.	M	T	T4.5
R224-D2.3-B10-sw	Collision avoidance must be possible to execute in real time	M	T	T4.1, T4.5
R225-D2.3-B10-sw	The machine vision algorithms used in visual servoing comprise of ML open-source libraries, i.e. compatible with BB8.	S	T	T4.5
R226-D2.3-B10-sw	The libraries and algorithms used must be open source or industrial standard.	M	I	T4.5
R227-D2.3-B10-sw	Possible to port different programming languages and operating systems/embedded control systems	M	I	T4.1 T4.5
R228-D2.3-B10-sw	The calibration and parametrisation of the algorithms and sensors related must be able to be configured on-site.	M	T	T4.5
R229-D2.3-B10-sw	The motion control algorithms are intuitive to operate from a usability perspective.	M	T	T4.5
R230-D2.3-B10-sw	Path generation should work automatically with minimal input from the operator	M	T	T4.1, T4.5
R231-D2.3-B10-sw	User can intervene automatic path execution safely	M	T	T4.1, T4.5
R232-D2.3	The short-term future path of the robot should be predictable for human traffic participants.	S	D	T4.1 T5.1
R233-D2.3	Path planning should take into account the presence and movement of human traffic participants and generate cooperative movement behaviour.	C	D	T4.1 T5.1
R234-D2.3-B10-sw-SAF	The automatic movements are tolerant to failures of the control system (servo drives, sensors, actuators, software singularities)	M	T	T4.1 T4.5

R235-D2.3-B10-sw	Measurement outliers and incomplete data (LIDAR data) should not lead to dangerous or unexpected behaviour.	M	T	T4.1, T4.5
R236-D2.3	The optimised Neural Networks must be able to run on the existing hardware (I.MX), e.g. Nvidia Jetson Or the mini-pc	M	D	T3.3, T3.4
R237-D2.3	The FPGA hardware shall not cost more than 1500€ and the embedded hardware not more than 500€.	M	D	T3.3, T3.4
R238-D2.3-P5-hw	The target cost of the goods in the visual servoing application for the camera system (excluding servomotors and drives) < 1000€.	M	I	T4.1 T7.1
R239-D2.3-B10-sw	Visual servoing and motion control algorithms related are modular so that these can be ported to other manipulators	W	T	T4.1, T4.5
R240-D2.3-B10-sw	Algorithms can be adapted/extended to different sensor types and boom types.	W	T	T4.1, T4.5
R241-D2.3-B10-sw	The implemented motion control algorithms are directly interoperable with a HIL toolchain and dynamic digital twin counterpart of the boom.	M	I	T4.1, T4.5
R242-D2.3-B10-sw	Algorithms can be tested and verified in a simulation environment with digital twins.	M	I	T4.1, T4.5
R243-D2.3-B10-sw	The motion control algorithms are fail-safe	M	T	T4.5
R244-D2.3-B10-sw-SAF	The algorithms must comply with mobile machinery directives	M	T	T4.1, T4.5

7. Conclusion

This deliverable proposes an initial version of the IMOCO4.E reference framework and specifies the overall requirements for the IMOCO4.E reference framework. The proposed IMOCO4.E reference framework and the overall requirements go beyond the current state-of-the-art, and meet the ST, SI, SO and SE objectives. The overall requirements would be refined further in the deliverable D2.4. In addition, this deliverable will act as a reference for the other requirements related tasks T3.1, T4.1, T5.1, T7.1 and T7.2. Ideally, the requirements coding scheme used in this deliverable would be used in other requirements related deliverables D3.1, D3.2, D4.1, D4.2, D5.1, D5.2, D7.1 and D7.2. The final version of the novel IMOCO4.E reference framework would be part of D2.4.

Based on the requirements gathered and reported in this document, the IMOCO4.E reference framework shall provide:

- *Configurability* (and *reconfigurability*) - in terms of the degree of parallelism, by choosing the most suitable computing engine on the heterogeneous SoC, e.g., CPU cores, Adaptable Engines (FPGA fabric), Ascend cores and Intelligent Engines for AI/ML/DL and/or DSP operations.
- *Flexibility* – The IMOCO4.E reference framework and methods are flexible for applying to different pilots, demonstrators, and use-cases.
- *Portability* - IMOCO4.E methodology will enable each machine to maintain excellent performance under slight variations in machine conditions using ML and advanced learning control. This enables the portability of production processes across multiple machines since processes will run almost identically on these machines.
- *Modularity* – The IMOCO4.E reference framework can be decomposed into several components that may be mixed and matched in various configurations. The components can connect, interact, or exchange resources by adhering to a standardised interface.
- *Maintainability* – The IMOCO4.E reference framework can be effectively and efficiently modified for improvements, corrections, or adaptation to changes in the environment and requirements.
- *Interoperability* – The IMOCO4.E framework systems, products or components can exchange information and use the information that has been exchanged.
- *Uniformity* – The IMOCO4.E framework components to be developed, modelled and used shall become (more) uniform in their description to allow faster integration in the design flow and as such, more modular to serve exchangeability during the lifecycle of the systems created.
- *Virtual commissioning* – The IMOCO4.E framework shall support the usage of digital twins for the programming of control code and its testing.
- Additionally, the IMOCO4.E reference framework shall support *Openness*, *Extensibility*, *Traceability*, *Dependability*, and *Scalability*.

References

[1] IMOCO4.E Description-of-Action (Annex I Part B).
Public link (with limited details): https://www.imoco4e.eu/overview;
SharePoint link for IMOCO4.E consortium (confidential).

[2] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. “Characterising the Digital Twin: A systematic literature review.” *CIRP Journal of Manufacturing Science and Technology* 29 (2020): 36-52.

[3] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and A. Y. C. Nee. “Enabling technologies and tools for digital twin.” *Journal of Manufacturing Systems* 58: 3-21, 2021.

[4] Louise Wright, and Stuart Davidson. “How to tell the difference between a model and a digital twin.” *Advanced Modeling and Simulation in Engineering Sciences* 7:13, 2020.

[5] Martijn Hendriks, Marc Geilen, Kees Goossens, Rob de Jong, and Twan Basten. "Interface Modeling for Quality and Resource Management." *Logical Methods in Computer Science* 17, 2021.

[6] Mike Mannion, and Barry Keepence. “SMART requirements.” *ACM SIGSOFT Software Engineering Notes* 20.2: 42-47, 1995.

[7] ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, *ISO/IEC JTC1/SC7/WG6*, 2011.

[8] Jennifer Stapleton. “DSDM, dynamic systems development method: the method in practice.” *Cambridge University Press*, 1997.

[9] Charles S Wasson. “System engineering analysis, design, and development: Concepts, principles, and practices.” *John Wiley & Sons*, 2015.

[10] Héder, Mihály. "From NASA to EU: the evolution of the TRL scale in Public Sector Innovation." *The Innovation Journal 22.2: 1-23*, 2017.