

Workpackage	Deliverable ID
<b>WP2 Business requirements &amp; Reference system architecture</b>	<b>D2.4 General specification and design of I-MECH reference platform</b>
<b>Summary</b>	
<p>This document describes the specification of the I-MECH reference platform that will be realized during the I-MECH project. The I-MECH reference platform will enable the demonstration of all key I-MECH principles and the validation of building blocks.</p>	
<b>Author</b>	Gijs van der Veen (Nexperia)
<b>Keywords</b>	Motion control platform, Interoperability, Open standards compliant, Instrumentation, Sensors, Actuators, Drives, Mechatronics, Vision-in-the-loop, Servo control, Predictive diagnostics, Multi-Many-Core, OPC UA, RAMI 4.0, I-MECH reference platform, Self-reflection

Coordinator                      Sioux CCM  
 Tel.                                    0031 (0)40.263.5000  
 E-Mail                                [info@i-mech.eu](mailto:info@i-mech.eu)  
 Internet                               [www.i-mech.eu](http://www.i-mech.eu)

## Table of contents

1	Introduction .....	5
1.1	Scope.....	5
2	Towards a minimal specification of the I-MECH reference platform .....	6
2.1	I-MECH principles to be demonstrated on the I-MECH reference platform .....	7
3	General specification of the I-MECH reference platform.....	9
3.1	MATLAB/Simulink as a model-based development environment .....	9
3.2	EtherCAT as field bus between layers 1 and 2 .....	10
3.3	OPC UA as non-time-critical machine interface .....	10
3.4	Centralized platform architecture.....	12
3.5	Interoperability .....	14
3.6	Interface between layer 2 and layer 3 .....	14
4	Architecture-related specifications at building-block level.....	15
4.1	BB1: Smart sensor platform, BB2: wireless sensor platform and BB5: High-speed servo amplifier. ....	15
4.2	BB4: High-speed vision .....	15
4.3	BB3: Robust condition monitoring and predictive diagnostics .....	15
4.4	BB6/7/8/9: Control and commissioning building blocks.....	16
4.5	BB10/11: Multi-many core control platform and RTOS .....	17
5	Concluding remarks.....	19

## List of Figures

Figure 1	The scopes of the I-MECH project and of the I-MECH center .....	6
Figure 2	I-MECH reference platform building blocks as a minimal intersection of pilot, use-case and demonstrator requirements on functionality (left) and interfaces (right) (courtesy: Technolution, WP6).....	6
Figure 3	Model-based design workflow as a gradual transition from simulation hardware to real hardware (adapted from WP6 minutes of meeting). .....	10
Figure 4	Functional mapping of I-MECH intelligence functions to the I-MECH layers (D6.2). .....	12
Figure 5	Decentralized (left) and centralized (right) I-MECH platform architectures. The centralized architecture (right) will be reflected in the I-MECH reference platform. ....	13
Figure 6	Envisioned layout of the I-MECH reference platform.....	13
Figure 7	A demonstration of interoperability in connecting proprietary instrumentation layer EtherCAT modules to the I-MECH system, preserving the I-MECH standard interfaces in the control layer. ....	14
Figure 8	I-MECH reference platform showing the mapping of BB3 (in red) onto the system architecture. ....	16
Figure 9	An architectural overview of BB10/BB11 as part of the I-MECH reference platform .....	18
Figure 10	The role of BB10/BB11 in processor-in-the-loop(PIL) testing of the I-MECH reference platform .....	18
Figure 10	The role of BB10/BB11 in hardware-in-the-loop (HIL) testing of the I-MECH reference platform .....	19
Figure 11	Schematic draft of the I-MECH reference platform architecture. Note that this image only indicates the principal locations of building blocks, whereas actual building blocks have functionalities distributed throughout the system. ....	20

## List of tables

Table 1	I-MECH principles and features to be demonstrated using the I-MECH reference platform .....	7
Table 2	Information flows and possible interfaces.....	11

## (Open) Issues & Actions

Open Issues (and related actions) that need central attention shall be part of a file called "[IAL - Issues & Action List – Partners](#)" which can be found in the [Google Drive Partner Zone](#).

ID	Description	Due date	Owner	IAL ID
OI-01	<del>Certain architecture aspects require expert input</del>	10-2018	Van der Veen, Beltman?	
OI-02	Define BB1/2 local hardware interfaces in D3.2	10-2018	Van der Veen	
OI-03	EtherCAT clock distribution and synchronization should receive attention in D3.2 as well as feasible and compatible sample rates for slave devices (what happens if rates are not integer multiples of each other?). Is PWM in BB5 also in sync with the EtherCAT clock?	10-2018	Van der Veen	
OI-04	There is little defined yet regarding the interface/protocol between layers 2 and 3	10-2018	Van der Veen	
OI-05	What will constitute the HAL contributed by BB11? I.e. what services are offered and who will take care of this?	10-2018	Van der Veen	
OI-06	What is needed in layer 2 to use layer 1 EtherCAT devices.	10-2018	Van der Veen	

## Document Revision History

Revision	Status	Date	Author	Description of changes	IAL ID / Review ID
R01	Draft	24-JUL-18	Van der Veen	Initiate	
R02	Draft	20-AUG-18	Van der Veen	Intermediate feedback from subsidiaries	
R03	Draft	23-AUG-18	Van der Veen	Draft R03 to be distributed to task leaders	
R04	Draft	12-SEP-18	Van der Veen	Draft after collecting TL feedback, ready for formal review process	
R05	Draft	1-OCT-18	Van der Veen	Ready for submission to ECSEL	

## Contributors

Revision	Affiliation	Contributor	Description of work
R01	Nexperia	Van der Veen	First draft
R02, R03, R04	Nexperia	Van der Veen	Subsequent drafts
R03	CCM	Beltman, Kuppens	Feedback on draft
R03	Technolution	Van Eert	Feedback on draft
R03	EDI, UNIBS, FAGOR, CCM, BUT, TUE, UNIMORE	Ozols, Visioli, de Yurre, Kuppens, Blaha, Goswami, Solieri	Feedback on draft, contributions

## Document control

		Status							
		Revision							
Reviewer Name	Role	Selection							
Arend-Jan Beltman	Coordinator	X							

--	--	--	--	--	--	--	--	--	--

## File Locations

Via URL with a name that is equal to the document ID, you shall introduce a link to the location (either in [Partner Zone](#) or [CIRCABC](#))

URL	Filename	Date
		dd-MMM-yyyy

## Literature

Ref	Name	Publisher	Year
[1]	<a href="#">Implementation of a low-latency EtherCAT slave controller with support for gigabit networks</a>	TU/e, Guerra Marin, R.	2017

## Abbreviations & Definitions

Abbreviation	Description
MIL	Model-In-the-Loop
SIL	Software-In-the-Loop
PIL	Processor-In-the-Loop
HIL	Hardware-In-the-Loop
OPC UA	Open Platform Communications Unified Architecture ( <a href="https://opcfoundation.org/about/opc-technologies/opc-ua/">https://opcfoundation.org/about/opc-technologies/opc-ua/</a> )
EoE	Ethernet over EtherCAT
FMI	Functional Mock-up Interface ( <a href="https://fmi-standard.org/">https://fmi-standard.org/</a> )
DoA	Description of Action: Annex 1B of the I-MECH grant agreement

Definition	Description

# 1 Introduction

Work package 2 in the I-MECH project considers the specification of the I-MECH reference platform and high-level architecture. Previous deliverables in this work package are:

- D2.1 State-of-the-art of motion control solutions
- D2.2 Needs for smart production from a mechatronics point of view
- D2.3 Overall requirements on I-MECH reference platform

Together, these documents describe the state of the art regarding motion control platforms and provide an inventory of requirements for the I-MECH reference platform. These requirements have been collected from all the I-MECH consortium partners. In addition, D2.3 provides a coarse sketch of the I-MECH platform architecture. This sketch is intended to derive the key requirements to facilitate *interoperability, openness, extensibility, modularity and maintainability* based on combining a custom set of building blocks. A definition of these terms is provided in D2.3.

In order to create an achievable target, this deliverable defines an I-MECH reference architecture which meets the minimal set of requirements in order to demonstrate the central I-MECH principles and which can be validated in the pilot, use-case and demonstrator applications.

## 1.1 Scope

Previous deliverables in WP2 contained many detailed functional requirements (for instance at building-block level). These requirements have been interpreted and refined in the specification documents (D3.1, D4.1 and D5.1 regarding the layers and building blocks, D7.1 regarding pilot-specific requirements). This document will only consider architectural requirements and the functional requirements relating to the overall I-MECH reference platform.

## 2 Towards a minimal specification of the I-MECH reference platform

The set of requirements as well as the flexibility in the architecture described in D2.3 are substantial. Within the scope of the I-MECH project one cannot attempt to develop and demonstrate a platform which meets all these requirements and fits all architectures that are foreseen (e.g. centralized vs decentralized). Rather, the I-MECH reference platform should focus on:

- Demonstrating the essential I-MECH principles as described in the DoA (see also, e.g. the I-MECH methodology in D6.2)
- Delivering a minimal set of requirements that fits the pilots, use-cases and demonstrators and allows the I-MECH platform to be validated in these applications. *Note that this specific item is not addressed here, since this document is at architecture level. This item is the responsibility of building block owners and the WP3, WP4, WP5 and WP7 task leaders.*
- Being an open platform in the sense that future improvements can be readily implemented

These ideas are schematically shown in Figure 1 below where we distinguish between the *I-MECH reference platform* developed within the duration of the project and future I-MECH platforms which will be improved and enriched versions of the I-MECH platform.

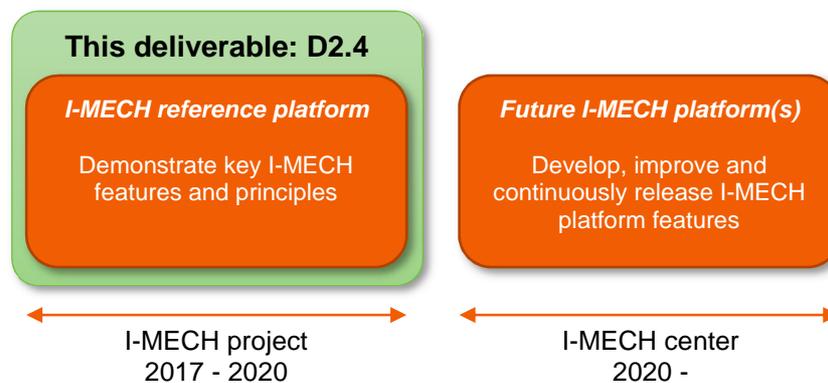


Figure 1 The scopes of the I-MECH project and of the I-MECH center

Similar ideas were discussed during the I-MECH face-to-face meeting (June 12-13, 2018, The Netherlands) in a workshop about work package 6 (Implementation and Integration of the I-MECH platform). Regarding building blocks (see Figure 2) it was agreed that the building blocks that are to be delivered within the scope of the I-MECH project will meet the least common denominator in terms of individual pilot and demonstrator requirements, to validate their functionality and interoperability within the I-MECH reference platform. As indicated in Figure 2, this also holds for the interfaces.

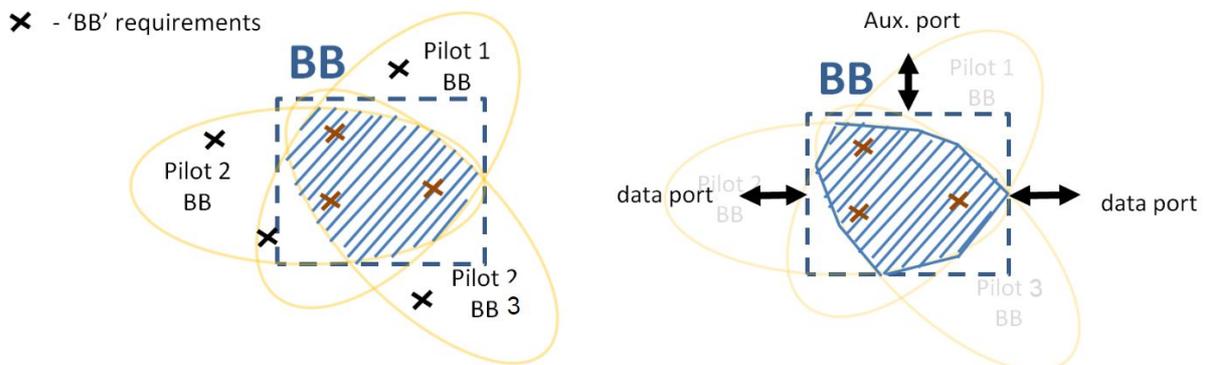


Figure 2 I-MECH reference platform building blocks as a minimal intersection of pilot, use-case and demonstrator requirements on functionality (left) and interfaces (right) (courtesy: Technolution, WP6).

Limiting the scope of the I-MECH reference platform as discussed leads to increased feasibility and lower risk of not meeting the project goals (e.g. an increased success factor). Even though not all requirements gathered in D2.1-D2.3 will be satisfied, a sufficient set of requirements will be satisfied to demonstrate the I-MECH objectives and principles. Furthermore, the I-MECH reference platform will be open, interoperable and extensible, so that future improvements and expansions are facilitated. For instance, the I-MECH reference platform may adopt a centralized architecture with an EtherCAT bus, whereas future I-MECH platforms may adopt multiple other field buses.

*Note: Referring to Figure 2, this deliverable will not discuss the detailed specifications of each building block in relation to individual pilot, use-case and demonstrator needs and the technological objectives specified in the DoA (chapter 1). This topic is reserved for deliverables D3.2, D4.2, D5.2 and D7.1 (the finalized requirements deliverables per work package). The essence of D2.4 (this document) is to sketch a minimal architecture (at higher abstraction level) which will suit all pilots, use-cases and demonstrators.*

## 2.1 I-MECH principles to be demonstrated on the I-MECH reference platform

Sections 1.1.2, 1.1.3 in the description of action (DoA) describe the scientific and technological development objectives of the I-MECH project. Some of these objectives can be viewed as *principles* that should be verifiable on the I-MECH reference platform. The following table lists I-MECH principles that will be essential features of the I-MECH reference platform and will be demonstrated within the scope of the I-MECH project:

Table 1 I-MECH principles and features to be demonstrated using the I-MECH reference platform

Principle and/or feature	Description	Verification method
1	Model-based design and testing (see D4.7, D6.2).	The I-MECH platform will demonstrate a model-based design, integration and testing workflow.
1	Model-based design and testing (see D4.7, D6.2).	<p>the model-based design and testing workflow will be demonstrated in (one or more) pilot, use case or demonstrator applications:</p> <ul style="list-style-type: none"> <li>A model of the system (including controller and plant) can be set-up in Simulink. The plant model can include interfaces if considered relevant. MIL and SIL simulations can be performed on the developer PC.</li> <li>Code and an executable can be generated from Simulink and sent to the I-MECH reference platform. Currently, the central hardware is expected to be x86-compatible or FPGA-based (TU/e CompSoC) and the executable will be targeted to such platforms.</li> <li>The I-MECH reference platform will perform its motion control task as expected under PIL and HIL conditions.</li> </ul>
2	Modularity	<p>Building blocks can be combined to create many variations of the I-MECH reference platform tailored to the application</p> <p>At instrumentation layer level:</p> <ul style="list-style-type: none"> <li>Zero or more layer 1 building blocks can be attached to the I-MECH platform and these different configurations can be addressed without manual software changes to interfaces. I.e., the building blocks are recognized and all I/O and auxiliary data can be accessed and the only software changes necessary are used to configure the BB.</li> </ul> <p>At control layer level:</p> <ul style="list-style-type: none"> <li>Control building blocks can be assembled from a library of building blocks with a standardized set of interfaces., which may only need context-specific configuration. Layer 2 offers a standardized interface to which control BBs can connect.</li> </ul>

3	Self-reflection, self-diagnostics and a service-oriented architecture	<p>Layer 1 building blocks will support a service-oriented architecture and self-reflection, implying that interfacing such devices will not require manual programming of device drivers in layer 2 when new devices are added (e.g. by means of the EtherCAT standard). The only effort necessary will be to perform design-time and runtime configuration. These devices will be able to diagnose themselves and report the status to upper layers. Layer 2 will provide access to services running in layer 3 related to commissioning, scripting, diagnostics and supervisory control.</p>	<p>See (2); attached devices will be identified automatically in layer 2. Configuration and communication with layer 1 will be according to a unified interface shared by all layer 1 BBs (e.g. EtherCAT).</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>I-MECH drives/amplifiers (BB5) with different specifications can be interchanged with minimal software configuration changes, since any I-MECH drive can receive a current (A), velocity (m/s) or position (m) setpoint in SI units, depending on the operating mode.</li> <li>A connected I-MECH smart sensor or drive will communicate data about its units, resolution and sample rate (and integer/floating point data representation) and can be configured remotely from layer 2/3. The protocol to achieve this is yet to be defined (e.g. the SERCOS protocol already provides such a feature, perhaps EtherCAT does too).</li> <li>All attached devices can be queried for their status and diagnostics</li> <li>The smart sensor platform BB1 will, among others, be able to interface to analog drives, analog sensors and (quadrature) encoders. Configuration of BB1 will enable layer 2 to access these devices while providing the “electronic datasheets”, unit conversions, diagnostics, etc...</li> </ul> <p>A future ambition, perhaps out of the scope of the I-MECH reference platform, is that mapping of input and output signals is completely automatic, as well as automatic configuration based on electronic data sheets. This could be implemented by adding OPC UA parallel to the EtherCAT protocol for sharing of non-time-critical data.</p>
4	Openness, interoperability and extensibility	<p>Interface definitions are open, so that non-I-MECH layer 1 and layer 2 building blocks based on existing (and proprietary) software and hardware can be added to the system. Furthermore, the system can be readily extended with new features which build upon the uniform interfaces.</p>	<p>At instrumentation layer level:</p> <ul style="list-style-type: none"> <li>Proprietary layer 1 building blocks can be added to the I-MECH reference platform and made available via the standard layer 2 hardware abstraction interface by coding an adapter and implementing a model for MIL if necessary (e.g., allowing addition of a commercial EtherCAT drive or I/O module)</li> </ul> <p>At control layer level<sup>1</sup>:</p> <ul style="list-style-type: none"> <li>Simulations can be added to the I-MECH model-based simulation environment via S-Functions and or FMI (Functional Mock-up Interface)</li> <li>Existing proprietary or open-source controller code can be linked to the I-MECH platform by creating an adapter to the available I-MECH control layer interface, so that it runs in the I-MECH MBSE environment and on the I-MECH platform (BB10/BB11).</li> </ul>
5	Maintainability, predictive maintenance	<p>Related to (2, 3); the I-MECH modularity and self-reflection abilities should support easy detection of modules requiring</p>	<ul style="list-style-type: none"> <li>Faults can be simulated at layer 1 building block level. The upper layers will report the fault in a precise and localised manner (e.g. by interpreting low-level data). The module can be swapped with minimal reconfiguration effort.</li> </ul>

<sup>1</sup> Note: The DoA (cf. SI 2) specifies that software BBs should be deployed to and tested on three commercial platforms. This is not considered a topic for this deliverable, as we restrict attention to (deployment onto) the I-MECH reference platform itself.

maintenance and easy  
exchange of modules

Interfaces are present to deliver diagnostic  
information from layers 1 and 2 to layer 3.

### 3 General specification of the I-MECH reference platform

The reference architecture described in D2.4 is updated with respect to the architecture proposed in D2.3. The update involves several modifications:

- In the reference platform architecture in D2.3, building blocks are portrayed as belonging to a single layer. The update below shows more explicitly the distributed nature of certain building blocks
- The reference platform architecture is chosen such that it can be realized within the scope (time and effort) of the I-MECH project

#### 3.1 MATLAB/Simulink as a model-based development environment

Simulink is one of the *de facto* standards for model-based design of control systems and is adopted by many of the I-MECH partners. The I-MECH reference platform will use Simulink as an environment for implementation of the entire control layer and for parts of the instrumentation layer (mainly for simulating, interfacing and configuring the instrumentation layer). A Simulink diagram will form the start of the design of a control system, progressing iteratively from pure simulation (MIL) via SIL and PIL towards hardware-in-the-loop (HIL). The complete MIL→SIL→PIL→HIL workflow involving all stages is currently only foreseen to be mandatory for the I-MECH reference platform itself, but not yet for all attached layer-1 building blocks. In this way, the principle can be demonstrated and verified without having to extend the principle to all building blocks yet.

Currently, the executable created from Simulink is foreseen to run on x86-compatible hardware (with or without a dedicated RTOS) or on a SoC (CPU/FPGA combination in various forms), i.e. the platforms offered by BB10. Note that Simulink is the initial choice and alternative environments can be considered in the future.

The Simulink environment for the I-MECH platform will make use of the FMI (Functional Mock-up Interface) to provide interoperability with other simulation models and environments. (Functional Mock-up Units; FMUs). This will allow the MIL stage of development to rely on models with several levels of detail in dedicated modeling environments such as Amesim and OOFELIE:Multiphysics. The SIL/PIL/HIL stages of development can also be supported if the FMU also provides source code or an executable for the target platform.

The code generation capability of Simulink also facilitates the use of control building blocks (BB6-BB9) outside the I-MECH reference platform. E.g. an ILC controller from the I-MECH Simulink block set can be implemented as C++ code in a proprietary control platform that supports C++. Conversely, an existing control implementation can be imported (e.g. as an S-Function) using an adaptor to couple it into the I-MECH Simulink framework. This supports the openness and interoperability of I-MECH modules.

Deployment specifications:

- MATLAB/Simulink will be used as a model-based development environment
- A Functional Mock-up Interface (FMI) will be developed and demonstrated in a MIL context, enabling a coupling to other simulation environments (Functional mock-up units (FMUs) implemented in e.g., Amesim and OOFELIE:Multiphysics). Demonstration of the FMI in SIL/PIL/HIL situations are of interest but not mandatory.
- Layer 2 will run on BB10/BB11 (either a COTS x86 platform or a SoC/FPGA)
- MATLAB/Simulink will enable successively:
  1. full simulation (MIL) of the hardware implementation within the Simulink environment
  2. code generation and deployment for SIL/PIL simulation
  3. code generation and deployment for HIL operation
 These phases will (initially) only be supported for the platforms offered by BB10/BB11. Furthermore, BB10/BB11 will dictate the requirements on compatible x86 platforms.

The model-based design approach is shown in Figure 3 in terms of MIL/PIL/SIL/HIL stages. Note that there may be some differences in interpretation of MIL/PIL/SIL/HIL. For an accurate definition refer to D6.2.

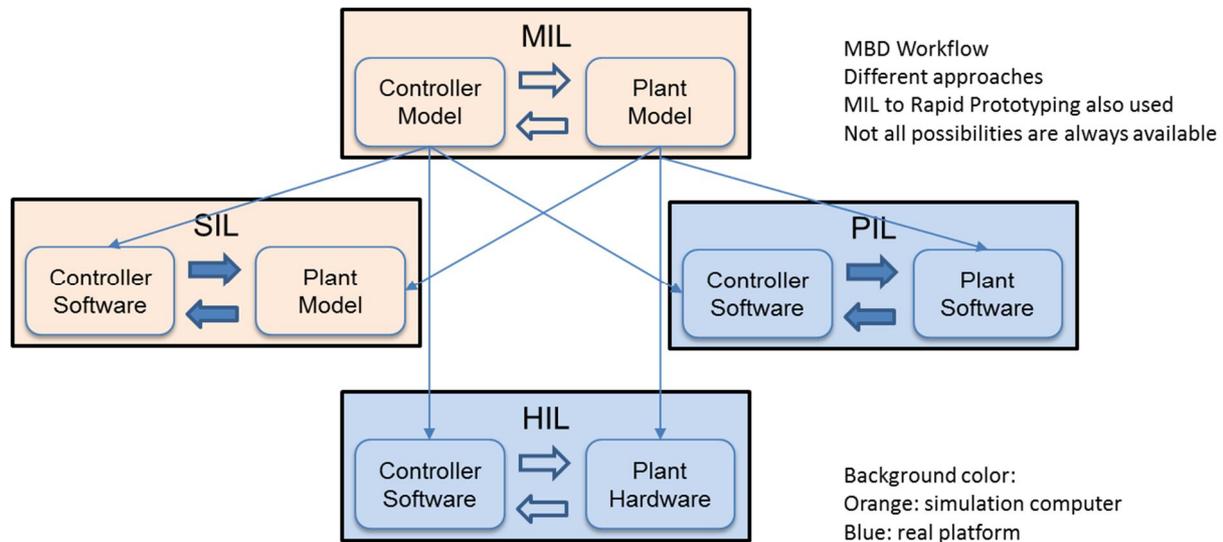


Figure 3 Model-based design workflow as a gradual transition from simulation hardware to real hardware (adapted from WP6 minutes of meeting).

### 3.2 EtherCAT as field bus between layers 1 and 2

The EtherCAT interface is adopted by many I-MECH partners and by many motion control companies outside the consortium. A possibility to achieve higher overall data rates with multiple layer 1 modules is to consider a real-time ethernet port switch/multiplier with a 1 Gbit uplink with several shorter 100 Mbit EtherCat chains.

#### Interface specifications

- The EtherCAT protocol will be used for communication between the control and instrumentation layers.
- The control layer (BB10/BB11) will implement an EtherCAT master
- Building blocks 1, 2 and 5 (instrumentation layer components) will be implemented as EtherCAT slaves. BB2 can either be an independent EtherCAT slave, or an add-on module on BB1, which appears to make more sense.
- A centralized control architecture will be demonstrated (see 3.4). An exception may be that current control loops are closed locally on the drives in view of the high sample rates required (BB5).
- The interface (EtherCAT) between layers 1 and 2 will demonstrate a service-oriented architecture and self-reflection. This implies that there will be a software layer at the end of the EtherCAT master (in layer 2), which provides a uniform interface for all I-MECH layer 2 building blocks. Hence, layer 2 building blocks need not be aware of the layer 1 implementation and fieldbus.
- The standardized EtherCAT I-MECH software interface will accommodate interface adapters, which allow non-I-MECH EtherCAT devices to be represented as I-MECH devices to the software running in layers 2 and 3 (see WP4, task 4.7).
- An ultimate ambition would be for layer 1 devices to be able to communicate their capabilities (e.g. electronic datasheet exchange), possibly via OPC UA; see 3.3 below. In view of the project duration, this ambition may not be achievable as an integrated solution within the I-MECH reference platform, however, but could be tested as a separate principle.
- BB10/BB11 shall not prohibit the implementation of other fieldbus standards in the future.

### 3.3 OPC UA as non-time-critical machine interface

OPC Unified Architecture will serve as the protocol for non-time-critical communication interfaces. The time-critical interface in the I-MECH platform is the interface between layers 1 and 2 which will make use of the EtherCAT protocol. As indicated in Figure 4, components in each layer have several “intelligence” functions next to their intrinsic functions, such as configuration, status monitoring and predictive maintenance. These functions require data that is not time-sensitive and could be transferred using OPC UA. This is summarized in Table 2 below.

Since OPC UA only defines *how* data is exchanged and not the data model (i.e. *which* data is exchanged), an OPC UA companion specification will need to be defined for I-MECH<sup>2</sup>.

In summary:

- Ethernet is used for communication between the control layer and the system behavior layer (layers 2-3 and 3-4, depending on the definition of a layer).
- The behaviour layer (layer 3) will implement the OPC UA protocol for communication with the factory automation layer (layer 4)
- An OPC UA companion specification will be chosen (or drafted/extended) to accommodate the data exchange required by the I-MECH platform. This companion specification will likely align with existing robotics-related and/or machine-tool-related companion specifications. This is a topic to be investigated.
- (Optional) The instrumentation layer could also implement OPC UA for non-time-critical information related to configuration, monitoring and predictive maintenance (possibly using Ethernet over EtherCAT, EoE). This would facilitate automatic device enumeration and configuration.

Table 2 Information flows and possible interfaces

Information flow	Present in layers	Time-critical?	EtherCAT	OPC-UA <small>(e.g. using Ethernet over EtherCAT)</small>
Process flow control - Realtime control loop data	1,2	yes	ü	
Configuration data - Auto-calibrate - Auto-configure - Datasheet exchange	1,2,3	no	ü	ü
Process monitoring data - KPIs - Analytics	1,2,3	no	ü	ü
Predictive maintenance data - Anomaly and failure prediction - Machine operation data	1,2,3	no	ü	ü

<sup>2</sup> <https://opcfoundation.org/markets-collaboration/>

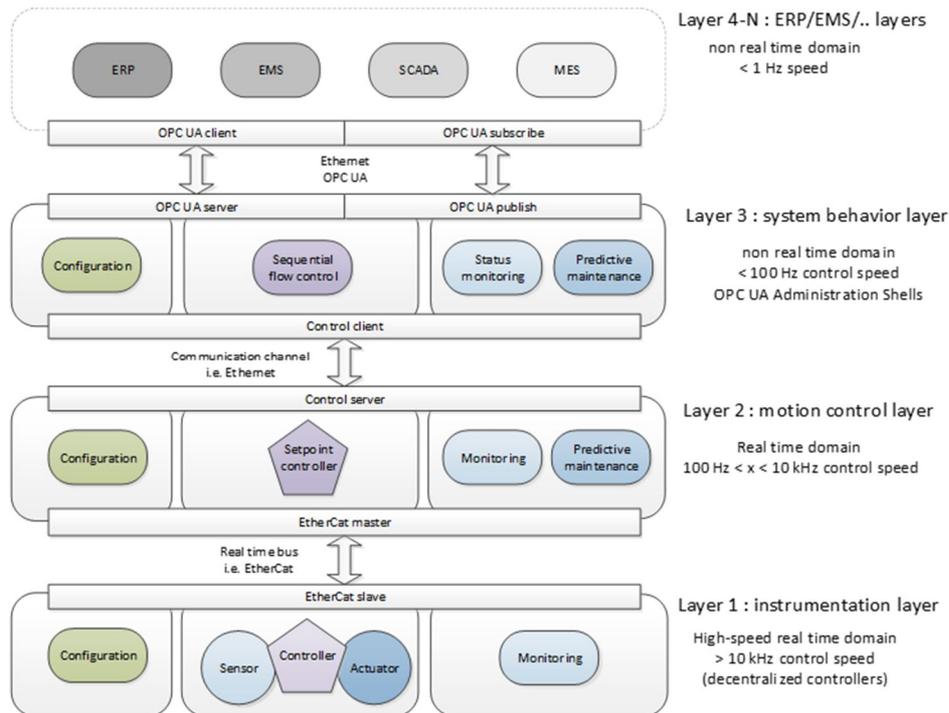


Figure 4 Functional mapping of I-MECH intelligence functions to the I-MECH layers (D6.2).

### 3.4 Centralized platform architecture

For reference, Figure 5 shows centralized and decentralized control architectures. The I-MECH platform is envisioned to serve both architectures. For demonstration of the I-MECH principles, however, it is not necessary to have a *decentralized* architecture. A centralized architecture also simplifies matters regarding model-based development, since the number of interfaces between executables is reduced.

For these reasons, the I-MECH reference platform will initially serve a centralized control architecture. An exception is high-speed servo amplifiers (BB5), which often have local high-rate current feedback loops and sometimes local position/velocity feedback loops.

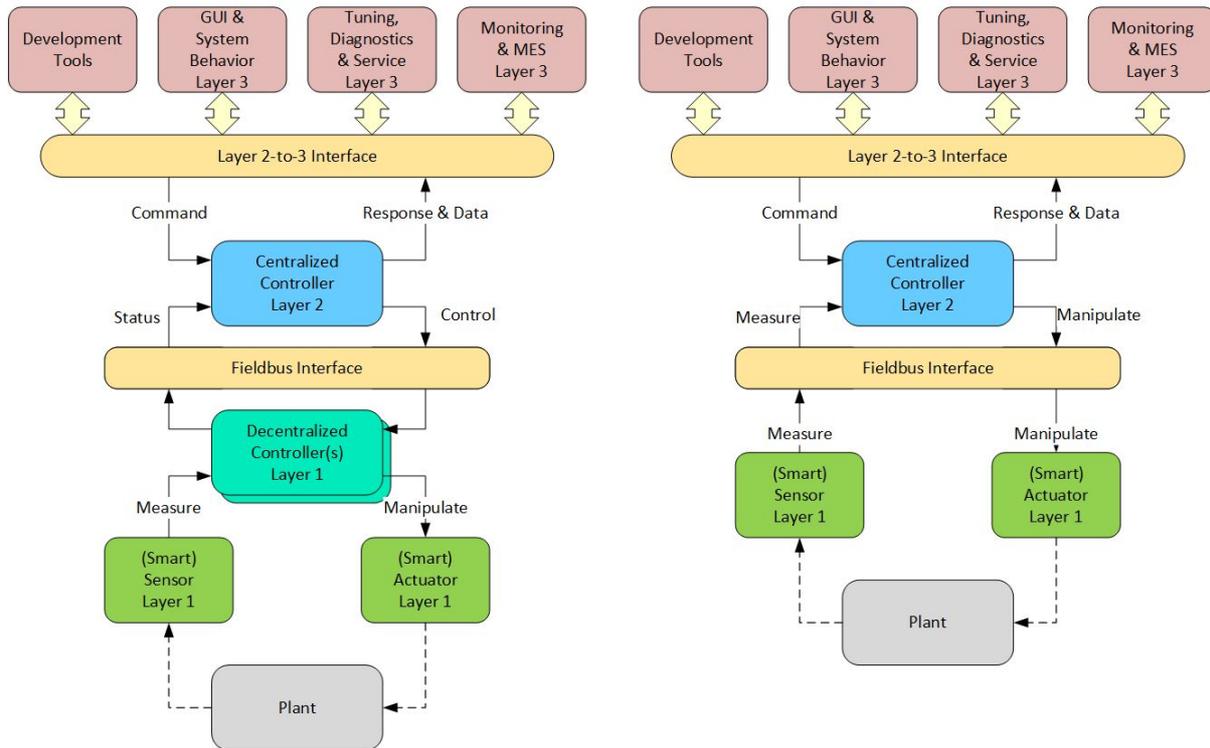


Figure 5 Decentralized (left) and centralized (right) I-MECH platform architectures. The centralized architecture (right) will be reflected in the I-MECH reference platform.

The architecture of the I-MECH reference platform is sketched in Figure 6, indicating the interfaces and data flows between the layers.

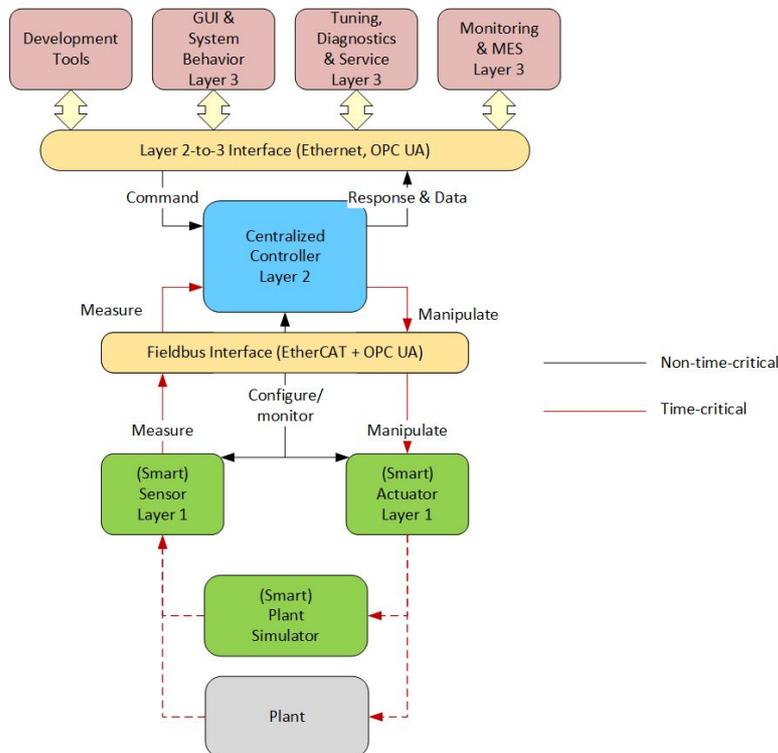


Figure 6 Envisioned layout of the I-MECH reference platform

### 3.5 Interoperability

Figure 7 shows an example of interoperability. Since the I-MECH platform will provide a generic EtherCAT master, any commercially available EtherCAT slave can be connected to it. In the EtherCAT interface of layer 2 a custom software adapter will be added, which exposes the commercial EtherCAT device via the I-MECH standard interface, so that software running in layer 2 will not need to be aware of the distinction between I-MECH and non-I-MECH layer 1 building blocks. In other words: I-MECH EtherCAT slaves will have a uniform data model which is directly compatible with the I-MECH platform and non-I-MECH EtherCAT devices will need a data translation layer in between. The uniform data model (to be defined) will involve aspects like standard naming conventions, support for different configurations and adaptiveness and extensibility of the data model. Obviously, there may be limitations in supported features, as the commercial device may not have all the I-MECH smart building block features.

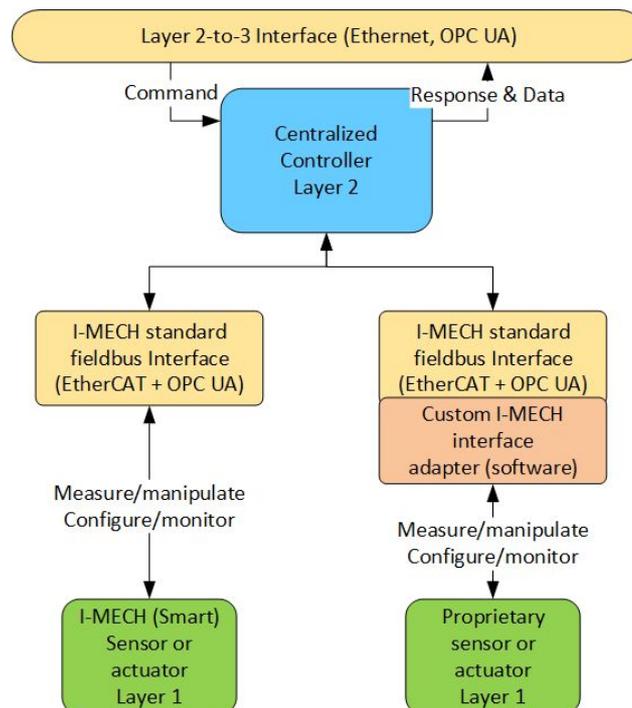


Figure 7 A demonstration of interoperability in connecting proprietary instrumentation layer EtherCAT modules to the I-MECH system, preserving the I-MECH standard interfaces in the control layer.

Similar principles will hold for control building blocks in layer 2: proprietary or open-source non-IMECH control building blocks can be added to the I-MECH platform. This can be done either by creating the necessary interface in the model-based development environment (Simulink), or by preparing an interface translator for this code, which makes the code behave like an I-MECH building block at interface level.

### 3.6 Interface between layer 2 and layer 3

The interface between the control layer (layer 2) and the behaviour layer (layer 3) will make use of an Ethernet interface. It is not yet clear which protocol will be used for communication, although the services to be offered by layer 2 are clear. The protocol may may be OPC UA, but can also be another protocol. In that case, an OPC UA server may run a level higher (level 3), providing communication with the factory automation layer (ERP, MES, ...).

The behaviour layer must be able to connect several client applications (simultaneously) to a server running on the motion control layer. Examples of such clients may be:

- A graphical user interface (GUI) which enables user interaction with the devices connected to the I-MECH platform,
- The engineering programming environment, which allows scripting of motion platform tasks, data acquisition, signal processing, calibration and tuning,

- An OPC UA server, performing translation of OPC UA instructions to layer 2 instructions and vice versa,
- Access to non-realtime functions of control building blocks (BB6 - BB9), such as tuning, calibration, configuration, monitoring, diagnostics and signal tracing,
- Access to non-realtime functions of layer 1 building blocks (BB1, BB2, BB5) such as configuration, tuning, firmware update, monitoring, diagnostics and signal tracing

## 4 Architecture-related specifications at building-block level

Detailed building block specifications have been described in D2.3 from the project point of view, in D7.1 from the pilot, use-case and demonstrator point of view, and in D3.2, D4.2 and D5.2 from the layer points of view. These requirements will not be duplicated in this deliverable. Rather, the scope of each building block will be tailored to the I-MECH reference platform *architecture* as outlined previously.

### 4.1 BB1: Smart sensor platform, BB2: wireless sensor platform and BB5: High-speed servo amplifier

BB1, BB2 and BB5 are similar in the sense that these are instrumentation layer building blocks. From that point of view, they should satisfy the same architecture requirements regarding the interface with layer 2.

These building blocks

- will implement an EtherCAT slave for communication with layer 2
- will have a simulation model available to be able to simulate behaviour for MIL/PIL/SIL testing, implemented as a Simulink model, as well as Simulink units which configure and interface to these building blocks
- will support configuration and monitoring from layers 2 and 3
- will implement self-diagnostics and self-reflection

BB1 and BB2 will standardize on a set of interfaces between the BB1/2 processor and the sensors/ADC/DAC (e.g SPI, I2C). These are to be specified in D3.2.

With regards to the wireless sensor platform, BB2 can either be an independent EtherCAT slave, or an add-on module on BB1, which appears to make more sense since it provides an integrated solution. In the latter case, BB1 will take care of the interface to layer 1.

Depending on the implementation, BB5 may be implemented as an EtherCAT slave or interfaced directly to BB1 via a dedicated interface.

Figure 4, borrowed from D6.2, shows that the building blocks in each layer shall have configuration, monitoring and predictive maintenance functionalities next to their intrinsic functionality.

### 4.2 BB4: High-speed vision

BB4 is expected to rely on a GPU (possibly assisted by an FPGA). with a focus on rapid acquisition and pre-processing of vision data. With this focus in mind, BB4 is not expected to rely on the model-based design techniques of I-MECH. BB4 will likely interface with layer 2 or layer 3 via a dedicated high-speed interface and require tailored interface code. An alternative implementation of BB4 is as an EtherCAT slave, where BB4 sends processed data (such as coordinates) to the EtherCAT bus.

### 4.3 BB3: Robust condition monitoring and predictive diagnostics

BB3 operates across layers 1 to 3. Figure 8 shows that elements of BB3 may be present at any level within the I-MECH reference architecture. Essentially, all interfaces should be prepared for predictive maintenance and condition monitoring by means of BB3.

Within layer 1, BB3 is expected to be present in two forms:

- As a dedicated sensor with or without integrated intelligence to perform (pre-processing)
- As a capability added to each existing input-output module (e.g., smart sensors, amplifiers) to provide diagnostics data upon request

In its most basic form, BB3 will be demonstrated by providing diagnostic messages or performance data (i.e., interpretations of measured quantities) at the level of layer 3, and being able to configure diagnostics (i.e. select signals, alarm levels, algorithms) from that layer. If BB3 features are implemented in layer 1 building blocks, the presence of signals (alarms, parameters, key performance indicators) will be automatically known to other layers because layer 1 devices communicate their data model (e.g. over EtherCAT or via an electronic data sheet). In this way these diagnostic parameters become available in layers 2 and 3 without manual intervention.

From the point of view of layer 2, each controller building block as well as the underlying motion control software framework will also provide diagnostic data in the same standardized format shared by layer 2 and layer 3 diagnostics features. Layer 2 or layer 1 devices can also perform pre-processing and data reduction on rapidly sampled signals, for instance by down-sampling or extracting spectral properties.

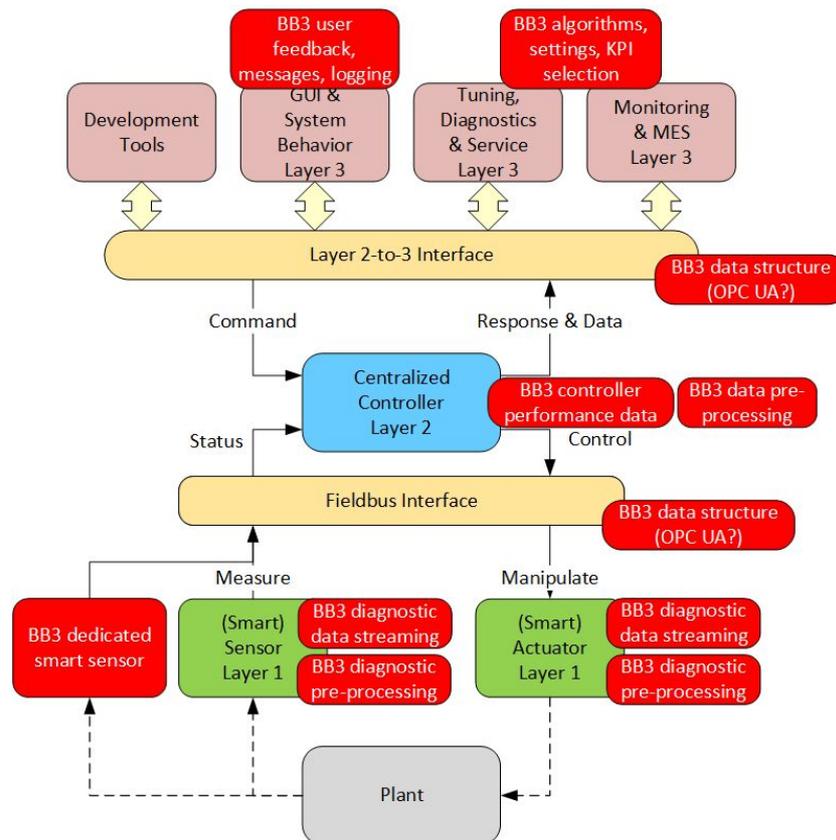


Figure 8 I-MECH reference platform showing the mapping of BB3 (in red) onto the system architecture.

#### 4.4 BB6/7/8/9: Control and commissioning building blocks

Control and commissioning blocks predominantly operate (i.e., perform their intrinsic task) in layer 2. However, configuration and monitoring take place in layer 3, whereas closed-loop feedback may also take place in layer 1 (e.g. inside a servo amplifier).

In the I-MECH reference platform, the following architecture will be demonstrated:

- Control building blocks (BB7, BB8, BB9) run on layer 2 in discrete real time with a given sample rate
- Control building blocks (BB7, BB8, BB9) designated to run in layer 2 will be developed in a Simulink environment and support model-based development (i.e. MIL/SIL/PIL/HIL and code generation for layer 2, in which MIL/SIL imply software running on the development PC and PIL/HIL imply code executing on the layer 2 platform)

- The commissioning tools in BB6 will run in layer 3/4 to commission and configure the control building blocks running in layer 2, possibly via the I-MECH standard OPC UA<sup>3</sup> interface.
- Layer 2 control building blocks shall have a standardized interface
  - Towards the common layer 2 control framework, which provides access to time-sensitive layer 1 I/O signals and layer 1 status and events and which controls timing, tasking and scheduling
  - Towards layer 3 to support commissioning, configuration, monitoring and to receive commands from the behaviour layer

## 4.5 BB10/11: Multi-many core control platform and RTOS

BB10 and BB11 jointly provide the hardware and software which sustains layer 2. BB10 will be a hardware platform with a real-time OS possibly including a hypervisor (BB11) on which the executable (generated from the model-based simulation environment) can run.

### Architecture

BB10 will have two implementations: a COTS based hardware platform, on x86 architecture, and an FPGA based one, on ARMv8.

BB11 will be composed by three independently reusable and composable sub-blocks:

- BB11.OS: operating system, based on Erika (EVI);
- BB11.HV: hypervisor, based on Xen (UNIMORE);
- BB11.PL: FPGA platform, based on CompSOC (TUE).

These three BB11 sub-blocks are not strictly interdependent, but they rather represent three independent pieces whose composition and integration will be explored. Two options will be surely implemented and applied to I-MECH pilots:

- BB11.PL on BB10.FPGA, applicable to Pilot 1 (SIOUX);
- BB11.OS on BB11.HV on BB10.COTS, applicable to Pilot 3 (IMA).

An overview of BB10 and BB11 system architecture is depicted below.

---

<sup>3</sup> CCM and Technolution are looking into the possibility of organizing a training to familiarize consortium partners with the OPC UA technology and how to implement it in building blocks.

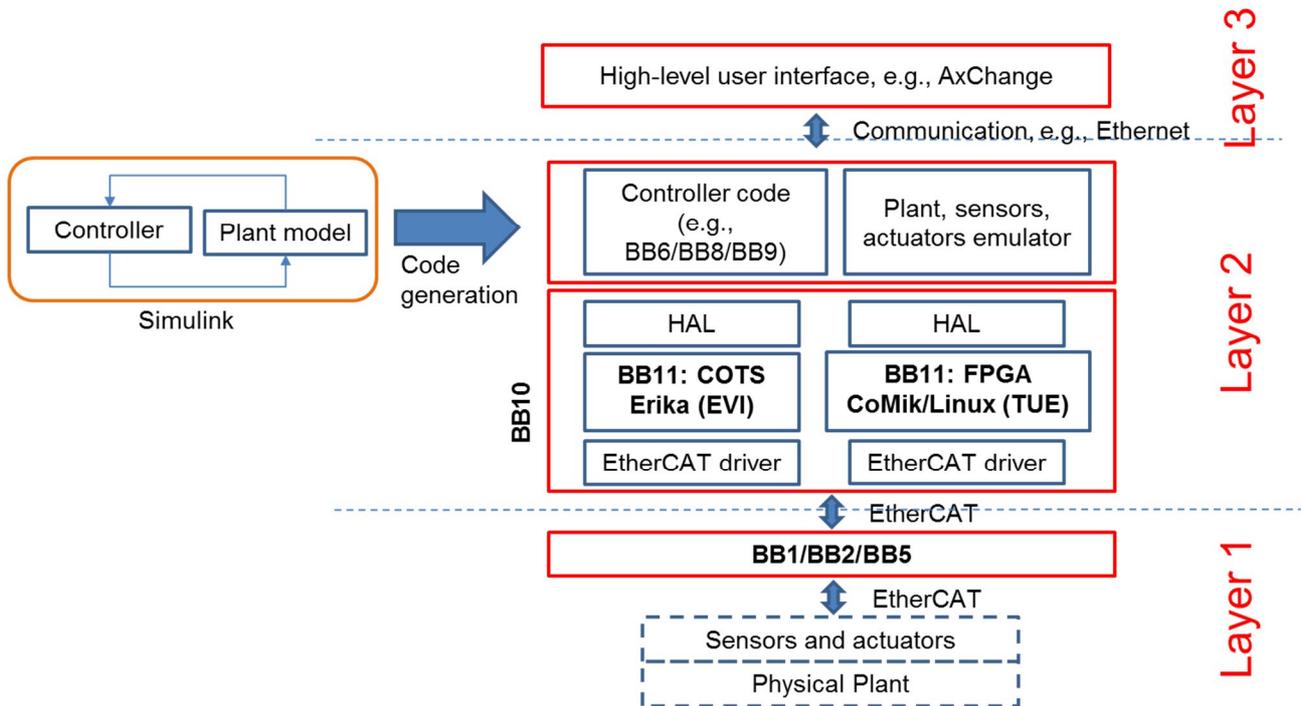


Figure 9 An architectural overview of BB10/BB11 as part of the I-MECH reference platform

**Interface**

- Codes of plant and controller will be generated from Simulink targeting either COTS or FPGA.
- The BB10 will be interfaced with BB1, BB2 and /or BB5 over EtherCAT.
- BB11 on COTS and FPGA will offer EtherCAT interface to the Layer 1.

**Testing and model-based development**

BB10/B11 will be used for PIL and HIL simulation and testing. How processor-in-the-loop (PIL) and hardware-in-the-loop (HIL) testing are foreseen is schematically shown in Figure 10 and Figure 11 below.

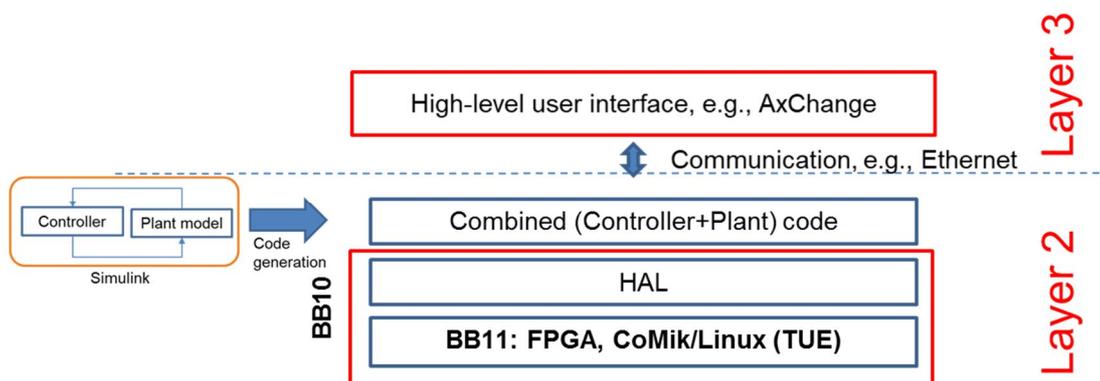


Figure 10 The role of BB10/BB11 in processor-in-the-loop (PIL) testing of the I-MECH reference platform

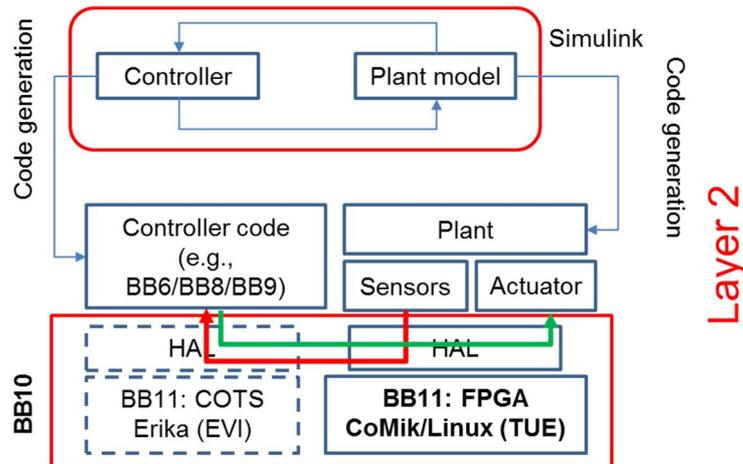


Figure 11 The role of BB10/BB11 in hardware-in-the-loop (HIL) testing of the I-MECH reference platform

## 5 Concluding remarks

This deliverable presents the decisions and choice made to arrive at a description of the I-MECH reference platform at architecture level. This architecture is based on a list of principles (I-MECH principles) which the I-MECH consortium intends to demonstrate using the platform. Although these principles can be interpreted and implemented in a very broad sense, a minimal set of architecture requirements is derived from them. This is done to arrive at a reference architecture which can be realized within the duration of the I-MECH project.

A complete overview of the I-MECH reference platform architecture is shown in Figure 12 below. The figure does not encompass all details but focuses on the interfaces and interactions. Specifically, this figure does not convey a complete mapping of building blocks onto the I-MECH reference platform. In reality, building blocks comprise a hardware and software component, each of which may reside at different locations in the reference platform. For instance, BB1 is a layer 1 device (hardware and software), but will require software in later 2 to use its data and in layer 3 to configure and monitor the device.

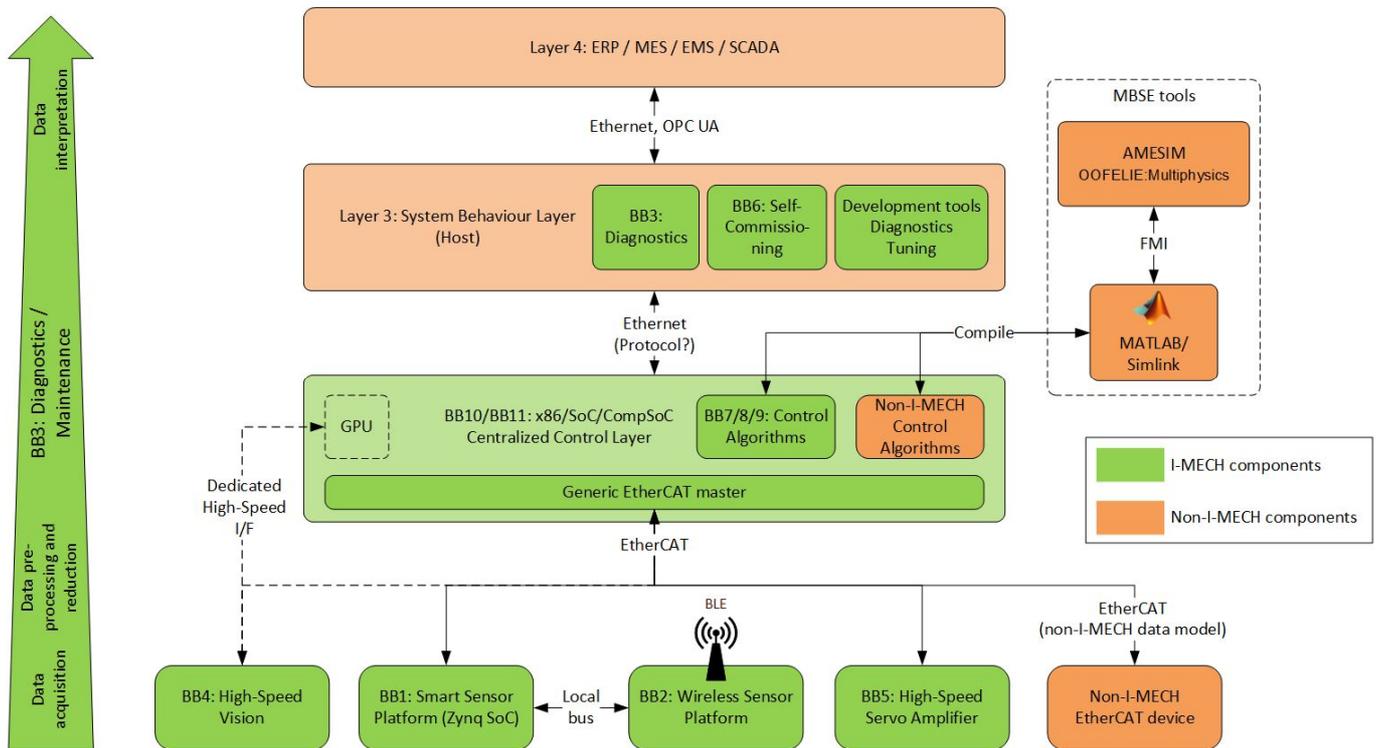


Figure 12 Schematic draft of the I-MECH reference platform architecture. Note that this image only indicates the principal locations of building blocks, whereas actual building blocks have functionalities distributed throughout the system.



**Acknowledgement**

This project has received funding from the Electronic Component Systems for European Leadership Joint

Undertaking under grant agreement No 737453. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program and Netherlands, Czech

Republic, Latvia, Spain, Greece, Portugal, Belgium, Italy, France, Ireland