A4BLUE

Adaptive Automation in Assembly For BLUE collar workers satisfaction in Evolvable context

GA number: 723828

Deliverable D2.6

Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system – Final Version

Partner responsible	ENGINEERING - INGEGNERIA INFORMATICA SPA
Contributors	TEK, RWTH, ILL, KOM
Type	Report (R)
Dissemination level	Public (PU)
Date:	30/04/2018
Version:	1.0
Due Date	M19
Status of the document	Final
Total number of pages:	175
Suggested readers:	All
Keywords:	Reference Architecture, FBB Decomposition, Business Processes

Document History			
Ver.	**Date**	**Changes**	**Author**
0.1	29/08/2017	First review from AIRBUS after D2.1	ENG
0.2	19/03/2018	Updated document structure to reflect changes from D2.1 to D2.6	ENG
0.3	10/04/2018	Updated common sections and FBBs (partially9	ENG
0.6	25/04/2018	Integrated contributions from ILL, RWTH and TEK	ENG
0.7	26/04/2018	Integrated reviews from AIRBUS	ENG
0.8	27/04/2018	Integrated reviews from CRAN, CESA, ILL	ENG
1.0	30/04/2018	Below is a summary list with all the main changes made for each A4BLUE RA module compared with D2.1. **MAIN CHANGES FROM D2.1:** • Addressed user level requirements and challenges in section 5.2 revisited to update the last version identified in D1.4 and D1.5 • Description of FRT.04 updated in section 5.2.4 to reflect the final approach • Description of FRT.05 updated in section 5.2.5 to reflect the final approach • Description of FRT.08 updated in section 5.2.8 to reflect the final approach and include notification management • Description of FRT.09 updated in section 5.2.9 to reflect the final approach • Traceability tables included in section 5.3 to comply with the Verification and Validation strategy described in D2.5 • Updated MOD.SH,MS description in section 6.1 to reflect the final approach • Updated MOD.SH,AM description in section 6.1 to reflect the final approach • Updated MOD.SH,DM description in section 6.1 to reflect the final approach • Updated MOD.SH,MHMI description in section 6.1 to reflect the final approach • Updated MOD.SH.AS description in section 6.1 to reflect the final approach • Updated MOD.EN.CAM description in order to take into the newly reasoning capabilities (Sections 6.2.1 and 8.2.1) • Update MOD.EN.EM description in order to	ALL

		reflect the change in the CEP engine integrated (Sections 6.2.2 and 8.2.2) • Updated MOD.BU.KM description in order to take into the account emerging needs from CESA and AIRBUS pilots (Sections 6.3.1 and 8.3.1) • Added MOD.BU.ACE component (Sections 6.3.4, 7.3.4, and 0) • Improved MOD.BU.MON descriptions (Sections 6.3.3, 7.3.3, and 8.3.3)	

List of authors

Company	Author	Contribution
ENG	Angelo Marguglio	Master Document Management and contributions on components owned by ENG
TEK	Patricia Casla	Contributions on general sections and on components owned by TEK
ILL	Giuseppe Donvito	Contributions on components owned by ILL and overall document review
KOM	Javier Salazar	Contributions on components owned by KOM and overall document review
RWTH	Tobias Adlon	Contributions on components owned by RWTH and overall document review

List of reviewers

Company	Reviewer	Contribution
AIRBUS	Laure Parigot	Review and improvements from a business perspective
TEK	Francisco Javier Diez	Review from a technical perspective

Executive Summary

The current deliverable (i.e. D2.6: Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system - Final Version) aims to report on the initial results of tasks T2.1 (Reference model for evolving assembly systems) and T2.2 (A4BLUE functional and modular architecture). This represents the updated and final version of D2.1, providing the grounding information for all the other technical work-packages (i.e. WP3, WP4, and WP5). Tasks T2.1 and T2.2 follow the incremental-iterative approach (alpha-beta loop) identified for the overall project, therefore this version of the deliverable provides a comprehensive view on the design of the A4BLUE Reference Architecture, incorporating the main feedbacks received during the development of the alpha prototypes deliverd by M18.

The purpose of both tasks T2.1 and T2.2 is to identify key challenges to be addressed in A4BLUE, as well as to refine the initially proposed reference implementation and define the functional architecture. To this end the main inputs of such activities are:

- DoA: Annex 1 (part A) of AMENDMENT Reference No AMD-723828-4 to the “Grant Agreement number: 723828 — Adaptive Automation in Assembly For BLUE collar workers satisfaction in Evolvable context (A4BLUE)”
- D1.4 - Requirements book and human factors best practice guidance - Final Version: an exploratory investigation of ‘multidimensional requirements’ for the A4BLUE solution and its results, capturing both user level requirements and high level requirements (i.e. the formal requirements to which organisations must comply that typically come from legal and prescriptive sources).
- D1.5 - Scenarios definition - Final Version: D1.5 includes the description of the AS-IS/TO-BE scenario of the four business cases driving the project (i.e. the two industrial scenario in AIRBUS and CESA, and the two laboratory scenario in TEKNIKER and RWTH).

The methodology followed to complete deliverable D2.6 included:

- Table of Contents (ToC) and document scope: once the ToC was agreed among the partners, a detailed template including a description of the required information along with examples was circulated to guide the section coordinators in the collection of the required information from all the contributing partners.
- EU initiatives analysis and alignment to the project objectives: a general overview of the most important EU initiatives dealing with the definition of a common architecture for the Smart Industry domain has been presented in order to ground the A4BUE approach into well-known approach.
- Specifications, reference model definition, business processes analysis, and FBB specification: starting from the requirements books and from the use case descriptions, the A4BLUE approach has been designed using UML diagrams.
- Follow up activities to monitor the work progress through different types of meeting: (1) General project follow up web meetings (every three weeks) involving representatives from all the partners; (2) WP2 specific conference meetings; (3) General project F2F meeting.

Furthermore, several revision iterations of D2.6 were put in place along the duration of the alpha phase (M4-M18) involving representatives of both the technical partners (i.e. ENG, ILL, TEK, RWTH, KOM) and the business partner (i.e. AIRBUS, CESA, TEK, WQTH) to check that the provided information was as clear as possible to be used in the scope of the other technical work-packages (i.e. WP3, WP4, and WP5).

Table of Contents

List of Figures

List of Tables

Abbreviations

AIOTI	Alliance of Internet of Things Innovation
API	Application Programming Interface
BD	Big Data
CAM	Collaborative Asset Management
CDM	Collaborative Decision Making
CEP	Complex Event Processing
CPS	Cyber-Physical Systems
CRUD	Create, Read, Update And Delete
CWE	Collaborative Working Environment
DoA	Description of Actions
DSS	Decision Support System
DyCEP	Dynamic Complex Event Processing
ECA	Event-Condition-Action
EFFRA	European Factories of the Future Research Association
EPA	Event Processing Agent
EPL	Event Processing Language
F2F	Face-to-Face
FBB	Functional Building Block
FCM	Firebase Cloud Messaging
FI-PPP	Future Internet Public-Private Partnership
FITMAN	Future Internet Technologies for MANufacturing
FW4I	FIWARE for Industry
GE	Generic Enabler
H2M	Human-to-Machine
HLA	High-Level Architecture

HQ-SE	FI-STAR Health Questionnaire
IA	Intangible Asset
IdM	Identity Management
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IIOT-RA	Industrial IoT Reference Architecture
IIS	Industrial Internet Systems
IoT	Internet of Things
IoT DM	IoT Device Management
KETs	Key Enabling Technologies
KPI	Key Performance Indicator
LDS	Local Discovery Service
LoA	Level of Automation
LSTM	Long Short-Term Memory
M2H	Machine-to-Human
M2M	Machine-to-Machine
NGSI	Next-Generation Service Interface
OCB	Orion Context Broker
P2P	Peer-to-Peer
PSCB	Publish / Subscribe Context Broker
RA	Reference Architecture
RAMI 4.0	Reference Architectural Model Industrie 4.0
RM	Reference Model
RNN	Recurrent Neural Network
SDO	Standards Development Organization
SE	Specific Enablers

TA	Tangible Asset
ToC	Table of Contents
UGC	User Generated Content
UI	User Interface
VA	Virtualized Asset
VAR	Virtual Asset Representation
VARM	Virtual Asset Representation Model
VBA	Visual Basic for Applications

Glossary of terms

Cyber-physical systems	Cyber-physical systems (CPS) are the basic technology platforms for IoT and IIoT and therefore the main enabler to connect physical machines that were previously disconnected. CPS is a system which links real (physical) objects and processes with information-processing (virtual) objects and processes via open, in some cases global, and constantly interconnected information networks. A CPS optionally uses services available locally or remotely, has human-machine interfaces, and offers the possibility of dynamic adaptation of the system at runtime.
FITMAN	FITMAN (Future Internet Technologies for MANufacturing) was a large-scale use case project, successfully completed by September 2015. Its mission was to assess the FIWARE platform in the context of ten industrial trials of various sizes and belonging to several manufacturing sectors. FITMAN also developed its own specialized Open Source components – Specific Enablers (SE) – filling some of the gaps existing between FITMAN's use case requirements and FIWARE platform's capabilities.
FIWARE	FIWARE is an open initiative in the scope of the Future Internet PPP (FI PPP) program, aiming at the creation of a sustainable ecosystem of Cloud-ready generic components – aka Generic Enablers (GE).
Generic Enablers	Generic Enablers (GE) are cloud-ready generic components that may be used as the foundational building blocks of Future Internet solutions in any area, effectively supporting the new wave of digitalization of EU industry and society. They are essentially software tools offered by FI-WARE, they are for public use and are royalty free.
Industrial IoT Reference Architecture	The Industrial IoT Reference Platform (IoT for Manufacturing - IoT4 Platform) is the synthesis of the three Smart / Digital / Virtual domains into one unified Reference Architecture that brings together FIWARE GEs, FITMAN SEs and lessons learned during the development of FITMAN's ten Trial Platforms. Its broader scope addresses complex real-world use cases involving multiple facets of the Manufacturing Enterprise.
Industry 4.0	Industry 4.0 is the next phase in the digitization of the manufacturing sector, driven by four disruptions: the astonishing rise in data volumes, computational power, and connectivity, especially new low-power wide-area networks; the emergence of analytics and business-intelligence capabilities; new forms of human-machine interaction such as touch interfaces and augmented-reality systems; and improvements in transferring digital instructions to the physical world, such as advanced robotics and 3-D printing. It includes cyber-physical systems, the Internet of things and cloud computing.
Internet of Things	The term Internet of Things (IoT) represents a paradigm that covers a wide range of topics in the information-based era. The goal of IoT is to enable any object to be connected anytime and anywhere with anything and anyone.

RAMI 4.0	RAMI 4.0 is the Reference Architectural Model for Industrie 4.0, it is s a Service-Oriented Architecture that combines all elements and IT components of Industrie 4.0 in a three-dimensional layer model for the first time. Based on this framework, Industrie 4.0 technologies can be classified and further developed.
Specific Enablers	A Specific Enabler (SE) is a component similar to a GE which offers functions relevant to domains specific, for example manufacturing, media, eHealth, energy and agrifood.

1 INTRODUCTION

1.1 CONTEXT AND SCOPE OF THIS DELIVERABLE

The current deliverable (i.e. D2.6: Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system – Final Version) aims to report on the initial results of tasks T2.1 (Reference model for evolving assembly systems) and T2.2 (A4BLUE functional and modular architecture).

This represents the updated and final version of D2.1, providing the grounding information for all the other technical work-packages (i.e. WP3, WP4, and WP5). Tasks T2.1 and T2.2 follow the incremental-iterative approach (alpha-beta loop) identified for the overall project, therefore this version of the deliverable provides a comprehensive view on the design of the A4BLUE Reference Architecture, incorporating the main feedbacks received during the development of the alpha prototypes deliverd by M18.

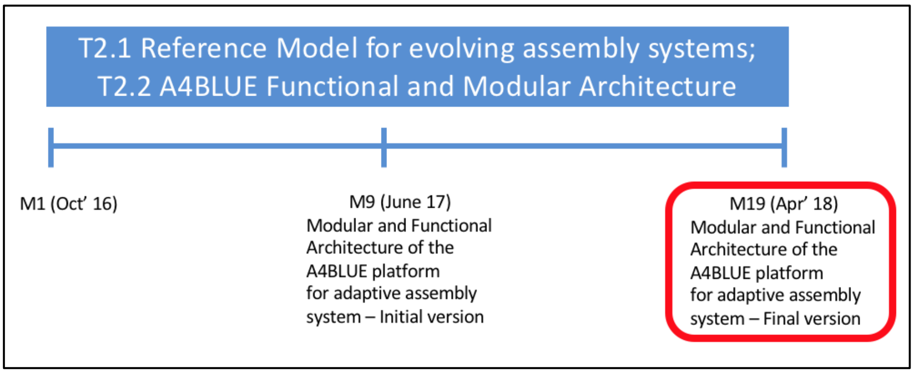


Figure 1 Tasks T2.1 and T2.2 timeframe and related deliverables

The purpose of both tasks T2.1 and T2.2 is to identify key challenges to be overcome by the A4BLUE project as well as common and application specific requirements to refine the initially proposed reference implementation and define the functional architecture.

WP2 (and especially T2.1 and T2.2) is the starting point of the technical work to be performed in the scope of A4BLUE. The following Figure 2 shows the relationship of tasks T2.1 and T2.2 with the rest of task involved in WP2.

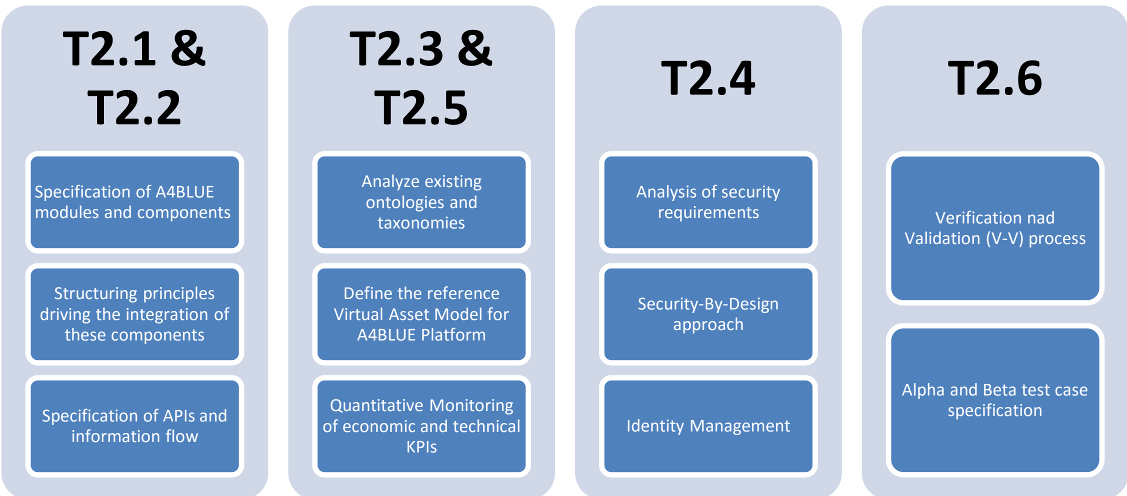


Figure 2 Relation of the tasks T2.1 and T2.2 with the rest of tasks in WP2

1.2 RELATIONSHIP WITH OTHER TASKS

D2.6 is divided in five main parts involving:

- **Introduction**: This section Identifies the tasks of the project related to the deliverable including information on objectives as well as a short description of the relationship of the current deliverable with the results of other tasks and work-packages.
- **Methodology**: This section describes the approach followed in tasks T2.1 and T2.2 to complete the deliverable D2.6.
- **EU Context**: An Analysis of other EU initiatives relevant to the A4BLUE architecture design, complemented by a description on how A4BLUE can be aligned or supported by them.
- **Specifications**: This is the core part of the document including relevant information from each module composing the A4BLUE overall solution. This description includes (1) the specifications of the modules representing an external view of the system derived from the requirements defined in WP1; (2) the identification of the A4BLUE Reference Model providing the solution to be adopted in A4BLUE in terms of Functional Building Blocks; (3) the analysis of main business processes to be put in place in the final solution to realize a coherent system from the individual modules; (4) the detailed specification of FBB in terms of system interfaces; (5) the overall Functional and Modular architecture of the A4BLUE project.
- **Conclusions:** This section provides summarised information on the A4BLUE Reference Architecture to pave the way to the technical developments in WP3, WP4 and WP5.

An overall view of the document structure can be seen in the Figure 3 below.

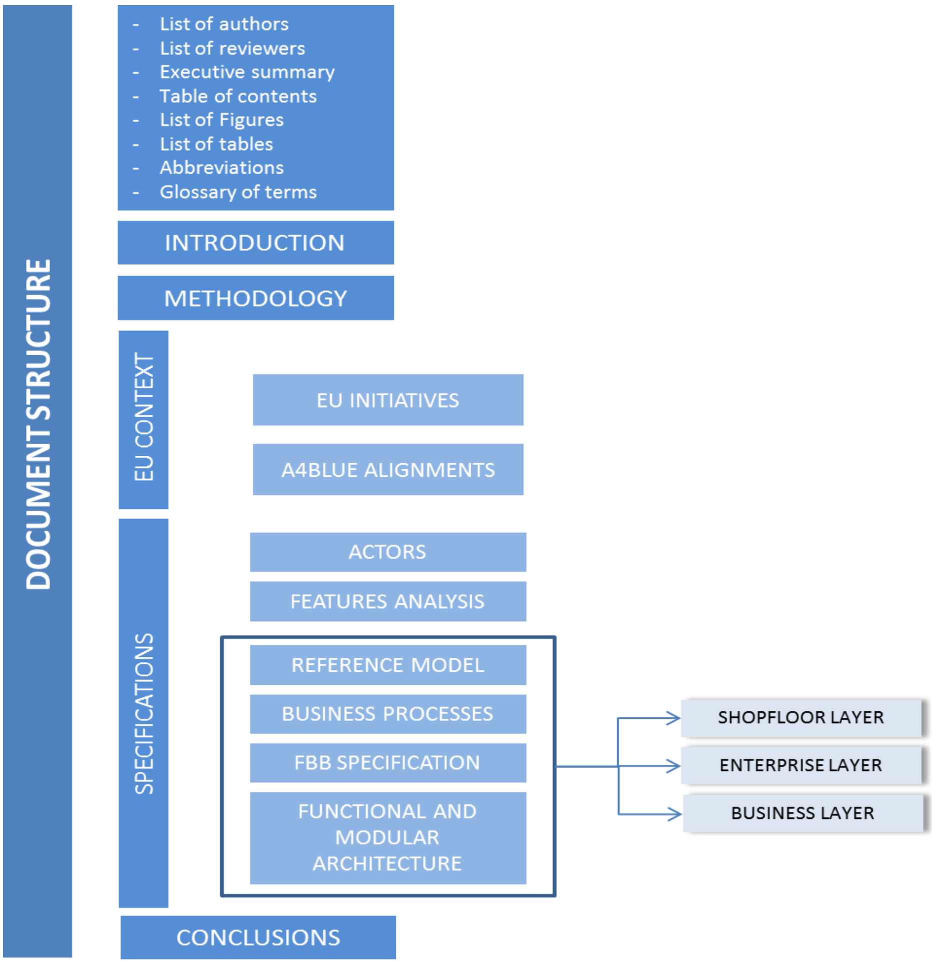


Figure 3 Diagram Document Structure

2 METHODOLOGY

2.1 REFERENCE IMPLEMENTATION

The A4BLUE Adaptive Framework will be designed based upon the following pillars: virtualisation, integration, adaptation management, worker assistance support and monitoring as shown in Figure 4.

In general terms, modularity and adaptability will be supported by exploiting the Service-Oriented Architecture (SOA) and the Event Driven Architecture (EDA) patterns and scalability will be implemented by a cloud approach.

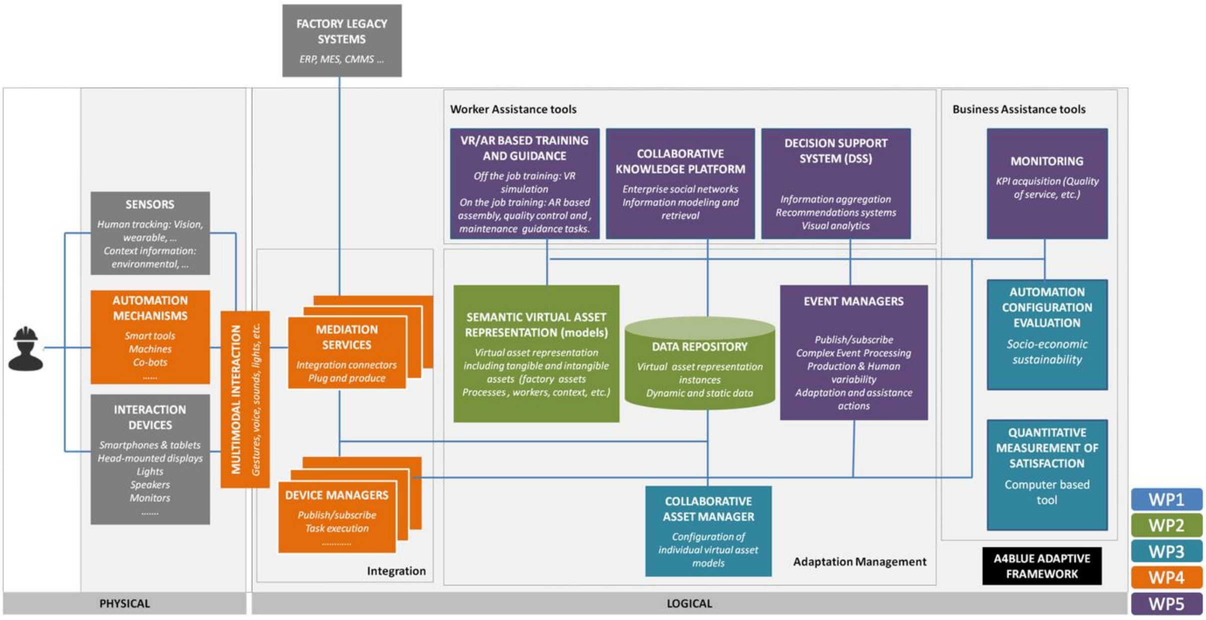


Figure 4 A4BLUE Reference Implementation (from DoA)

The main components of such a vision are further detailed in the following Table 1.

Pillars	Reference Modules	Description
Virtualization	Collaborative Asset Manager	The CAM will be in charge of providing virtualisation capabilities by managing the virtual asset model, containing representations of both Tangible Assets (i.e. physical objects with economic value like machinery, robots, tools, materials, workers, etc.) and Intangible Assets (i.e. key business drivers such as process definitions, technical or scientific knowledge, worker skills and their level of adaptability, business relationships, technological solutions, etc.).
Integration and interoperability	Mediation Services	These services will enable the integration of heterogeneous hardware and software components, while complying with the relevant standards (e.g. OPC UA, etc.).

Pillars	Reference Modules	Description
	Device Managers	The Device Managers will represent the logical part of the automating mechanisms (CPS) and will include management capabilities to support the decentralisation of the decision making.
	Multi-modal Interaction	Multimodal interaction involves both “Multichannel Human-Automation/robot interaction” and “Active safety”. “Multichannel Human-Automation/robot interaction” implements the multimodal, multichannel input/output mechanisms for the shop floor operator interaction with the automation mechanisms (e.g. robots) through gesture, voice, lights, sounds, AR devices (if applicable), etc. “Active safety” implements active safety mechanisms to adapt the behaviour of the automation/robot considering the safety mode in place (i.e. safety rated stop mode or speed & separation monitoring mode) and the operator’s profile.
Adaptation management	Event Manager	The Event Manager will be in charge of continuously capturing and analysing relevant events and reacting to them, on publish/subscribe basis, by triggering appropriate adaptive actions.
Personalised worker’s assistance tools	VR/AR based training and guidance	A4BLUE will sustain both off the job training (i.e. at a site away from the actual work place) and on the job guidance (i.e. taking place in at the workplace in the real working situation) by using appropriate VR/AR hardware and software components.
	Collaborative Knowledge Management	Facilitating the transfer of knowledge, especially implicit and informal ones, from skilled workers to young or less experienced workers to help them make the right decision at the right time.
	Decision Support System (DSS)	The Decision Support System is aimed to support workers on relevant decisions for the assembly, maintenance, inspection operations. It aggregates relevant information produced in the domain of the A4BLUE system and provides visual analytics capabilities to support workers in the decision-making process.
Business assistance tools	Monitoring	The monitoring module is aimed to support the evaluation process to be performed by supporting the collection and visualisation of key performance indicators (KPIs) to assess the impact, from an economic and social perspective. It is aimed only to support the collection of the relevant performance metrics produced in the domain of the A4BLUE solution other kind of KPIs will be out of the scope for this component.

Pillars	Reference Modules	Description
	Automation Configuration Evaluation	An evaluation tool able to assess the levels of automation (assembly system configurations/assembly process) from a socio-technical as well as from an economical perspective. The socioeconomic evaluation results will be used as a basis for further improvements of the assembly system structure as well as the selection and elaboration of automation solutions.
	Quantitative Measurement of Satisfaction	A psychometric instrument for quantitative measurement of satisfaction which can be used to assess levels of worker satisfaction in relation to human-automation systems and wider work environment characteristics. It will include a computer-based tool that enables shop floor operators to easily complete recurrent worker satisfaction questionnaires and evaluate the results.

Table 1 A4BLUE Reference Modules (from DoA)

The above-mentioned components will be further discussed later in this deliverable, since they have paved the way for defining the REFERENCE MODEL described in Section 6.

2.2 THE METHODOLOGICAL APPROACH

This document presents the architecture of A4BLUE-based systems, based on the use of multiple, concurrent views. Multiple views allow to address separately the concerns of the various stakeholders of the A4BLUE project, mainly technical partners and business partners, and to handle separately the functional and non-functional requirements separately. The **A4BLUE Reference Architecture** (RA) will be designed using an architecture-centered, scenario- driven, iterative development process.

The A4BLUE RA deals with the design and implementation of the high-level structure of new adaptive systems based on project outcomes. It is the result of assembling a certain number of architectural components in some well-chosen forms to satisfy the major functionality and non-functional requirements of the system.

Following the “4+1 view model” approach (as defined by Kruchten in [1]) this document deals with abstraction, decomposition and composition of several viewpoints. Not all the views prescribed in this approach will be part of D2.1, since they fall in other project outcomes.

The proposed approach uses a model composed of five main views (or perspectives):

- The *Logical view*, which is the object model of the design, is concerned with the functionality that the system provides to end-users. See Section 6 – “REFERENCE MODEL” for further details.
- The *Process view*, which captures the concurrency and synchronization aspects of the design, dealing with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behaviour of the system. See Section 7 – “BUSINESS PROCESSES” for further details.
- The *Physical view*, which describes the mapping(s) of the components onto the concrete instances and reflects its distributed aspect, presenting the inter-connections between these components. See Section 8 – “FBB SPECIFICATION” for further details.

- The *Development view*, which describes the static organization of the solutions to be implemented in its development environment. See outcomes from WP3, WP4, and WP5.
- The *Scenarios*, which become a fifth view, used to derive the architecture that is in fact partially evolved from these scenarios as we will see later. See the outcomes from WP1 and Section 5 – "SPECIFICATIONS"

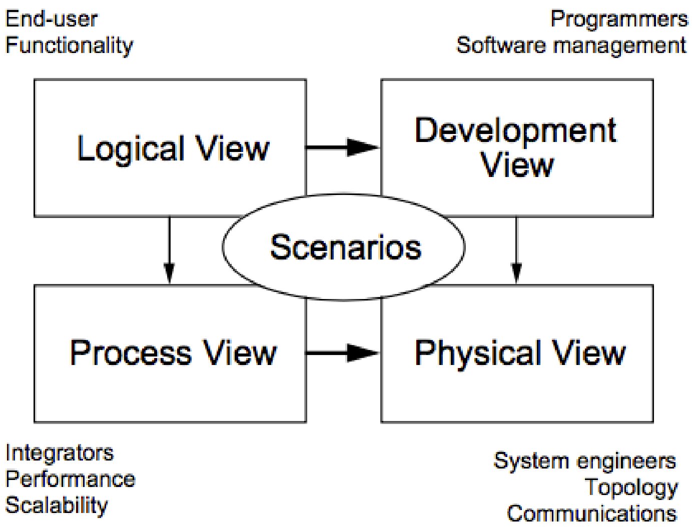


Figure 5 The "4+1" view model

Apart from adherence to the 4+1 view approach, the present deliverable has taken account of prior WP1 developments (notably the requirements described in D1.4 and the use cases described in D1.5) in order to develop the A4BLUE architecture. In particular, the architecture of the project satisfies several of the functional and non-functional requirements of the project. Some of key requirements driving the development of the architecture are the ones relating to compliance to standards (e.g. RAMI4.0 and IIRA, as described in Section "3 - THE CONTEXT"), implementation of adaptive automation taking into account process/product/worker variability. Note however that the A4BLUE architecture has considered high-level requirements of D1.4, rather than the low-level technical ones that will be further analysed in the scope of WP3, WP4 and WP5. This is because the presented A4BLUE architecture focuses on high-level decisions with system wide impact on A4BLUE based systems, rather than on low-level technical details that will be elaborated as part of detailed design and implementation.

In addition, the objectives of the A4BLUE RA can be decomposed as in the following:

- To develop a **reference architecture** for the A4BLUE platform for the implementation of its solution by using an iterative approach
- To define the **logical structure** of the infrastructure components in the A4BLUE stack
- To define the **functional components** implementing each infrastructure component in order to support the evolving adaptive assembly system concept

In order to achieve the above-mentioned objectives, it is worth noting how different levels of an architecture description could include both design and implementation aspects, as summarized in the following Figure 6.

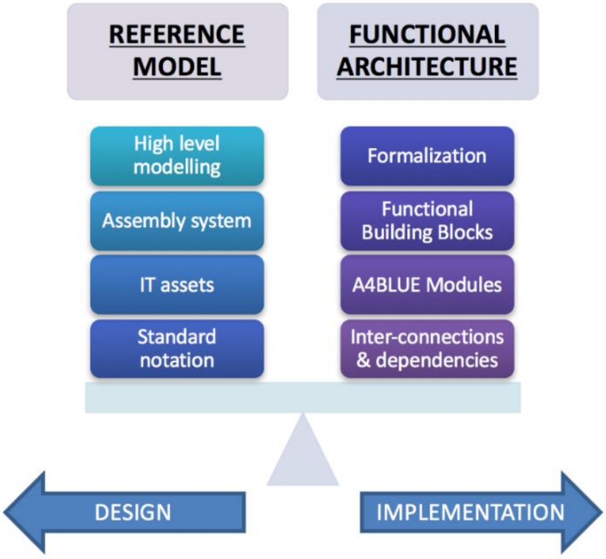


Figure 6 Architecture description: Design vs. Implementation phases

The Reference Model will ideally connect the domain of the problem (as described in WP1 outcomes, namely D1.4 and D1.5) and the domain of the solution (further described in WP2, WP3, WP4, and WP5 outcomes). At design stage, a high level mapping and a common notation are needed to pave the way to further analysis and detailed design to be formalized in Functional Building Blocks (FBB) and their mutual inter-dependencies.

To further structure the envisioned process to derive from the Reference Model a FBB Specification, the following Figure 7 shows the suggested iterative and incremental approach.

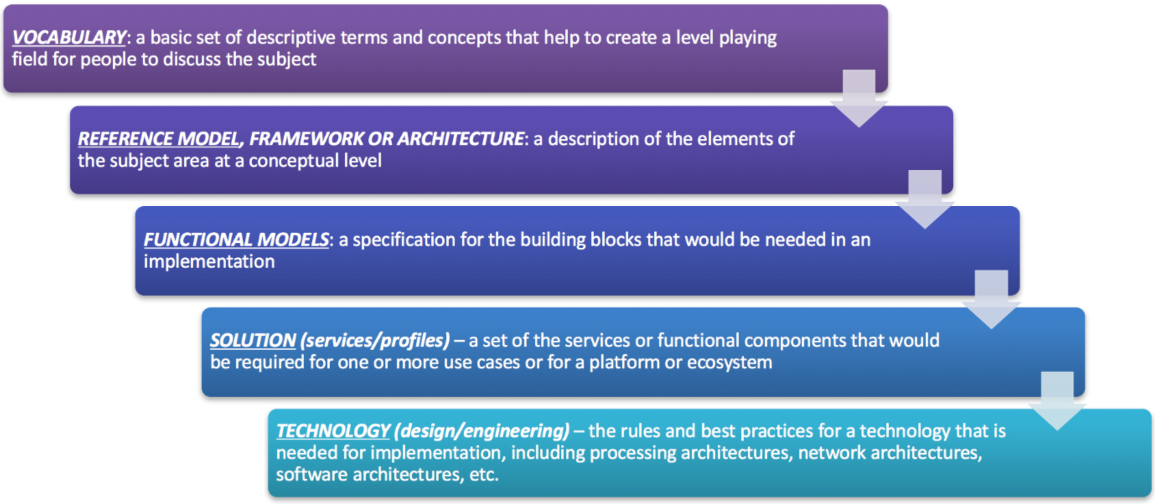


Figure 7 Architecture description: iterative and incremental approach

Starting from defining a foundational glossary, common concepts and principles (mainly in Section 5 – "SPECIFICATIONS") will set the scene to start describing the elements of the architecture at a conceptual level (i.e. the REFERENCE MODEL described in Section 6); then a more functional analysis will conduct to the identification of the main building blocks needed for the implementation stage (namely in Section 8 – "FBB SPECIFICATION"). Here we pass to the domain of the solution. Here we start with the definition of a set of services that would be required by one or more application scenario identified in WP1; then these functions will be deatiled in terms of background assets and technologies to be used during development; this technogical view will be enriched also with a decomposition in sub-components presenting the business processes among them (in Section 7 – "BUSINESS PROCESSES").

3 THE CONTEXT

The Internet of Things (IoT) is an intensively discussed topic in industry, government and academia. The industrial application of IoT are named more specifically **Industrial Internet of Things (IIoT)** and are a fundamental part of Industry 4.0 solutions, offering a significant innovation potential for entire industries. In such a context, one of the main problems is the technologically focused discussion. Proposals exist for several technical applications within the IoT (e.g. Smart Grid, Smart Home, Smart Factory); however, the core of the so-called Cyber-Physical Systems (CPS) — the technological basis of IoT — has not been defined clearly. In addition, the implementation of CPS in the industrial ecosystem has not yet been discussed properly.

In the context of digitization, the term Internet of Things (IoT) represents a paradigm that covers a wide range of topics in the information-based era. Several terms correspond to IoT in different applications or industries (e.g., Smart Grid, Smart Home, or Smart Factory). In all of these applications, physical objects and digital information are converging into smart systems, based generally on cyber-physical systems (CPS). The technical basis of these cyber physical systems consists of many different and fast-developing technologies. The goal of IoT is to enable any object to be connected anytime and anywhere with anything and anyone.

Cyber-physical systems are the basic technology platforms for IoT and IIoT and therefore the main enabler to connect physical machines that were previously disconnected (see Figure 8).

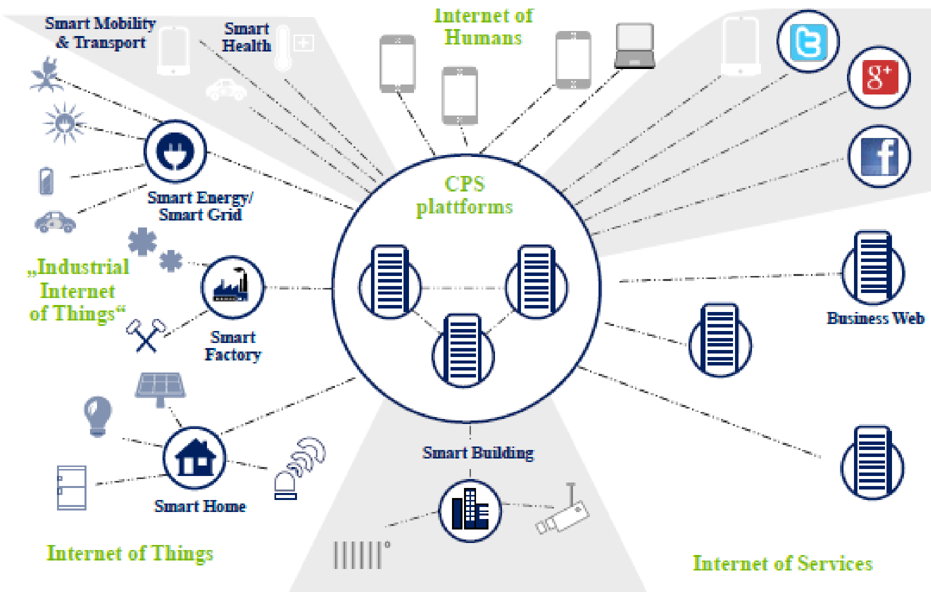


Figure 8 Cyber-physical systems: the core of IoT and IIoT

There are many expectations with regard to the ways in which the current era will change the management of manufacturing, supply chains and innovations; some of these are listed below:

- Interconnected physical systems
- Mobile Information
- Minimized reaction times

Firstly, nearly every piece of hardware contains a system that can communicate to other systems. Therefore, as physical systems become cyberized, they are able to communicate with other physical systems or, via an interface, with human beings. Within an information network, physical systems will increasingly be able to control themselves automatically, being able to behave as autonomous systems in a changing and dynamic environment.

Secondly, the information is becoming truly mobile, as opposed to merely decentralized or cloud-based. Information will be exchanged between CPSs, whose decision making processes are self-controlled and not dependent on centrally stored data.

Finally, technical reaction times are minimized. CPSs will make self-controlled decisions within a globally distributed information system. Because information is now ubiquitous, the dynamics of these processes are speeded up [1].

These are among the main reasons why it is time to pass to a new architectural approach.

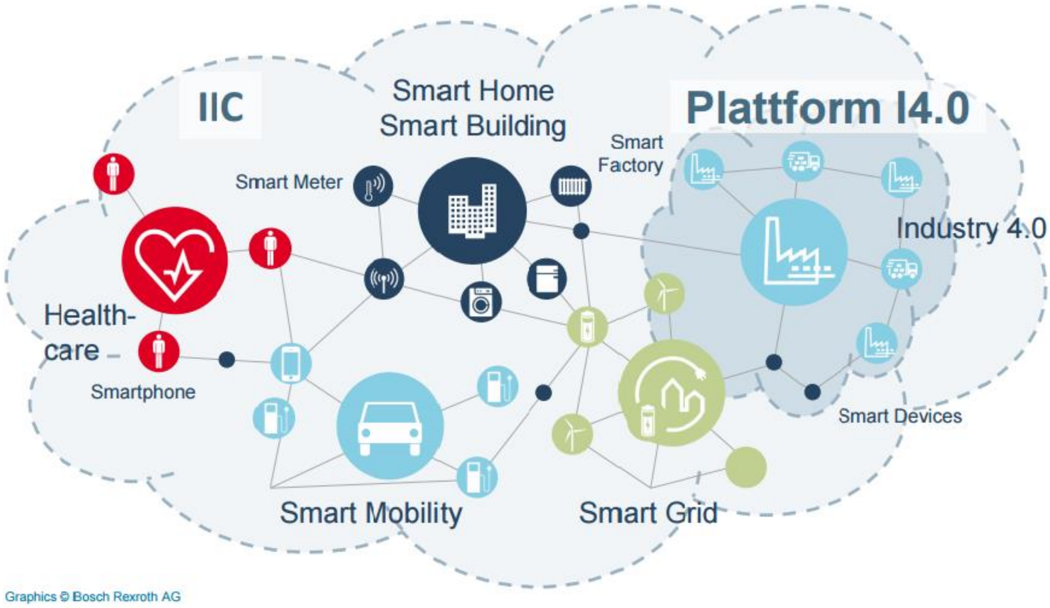


Figure 9 Industrie 4.0 in the Internet of Things and Services

3.1 RAMI 4.0

One of the more interesting architectural approaches in the field of Industrie 4.0 is **The Reference Architectural Model Industrie 4.0** (abbreviated RAMI 4.0) [1], since it combines the crucial elements of Industrie 4.0 in a three-dimensional layer model for the first time. As its name clearly states, it is the outcome of Platform Industrie 4.0[1], the German public-private initiative addressing the fourth industrial revolution – i.e., merging the digital, the physical and the biological worlds into *cyber-physical production systems*. This specification was firstly published in July 2015 and provides a first draft of the reference architecture for the Industrie 4.0 initiative trying to group different aspects in a common model and to assure the end-to-end consistency of *"… technical, administrative and commercial data created in the ambit of a means of production or of the workpiece*" across the entire value stream and their accessibility at all times.

Even if the RAMI is essentially focused on the manufacturing process and production facilities, it tries to focus all essential aspects of Industrie 4.0. The participants (a field device, a machine, a system, or a whole factory) can be logically classified in this model and relevant Industrie 4.0 concepts described and implemented. Based on this framework, Industrie 4.0 technologies can be classified and further developed. The RAMI consists of a three-dimensional coordinate system that describes all crucial aspects of Industrie 4.0. In this way, complex interrelations can be broken down into smaller and simpler clusters (see Figure 10).

[1] http://www.plattform-i40.de/I40/Navigation/EN/Home/home.html

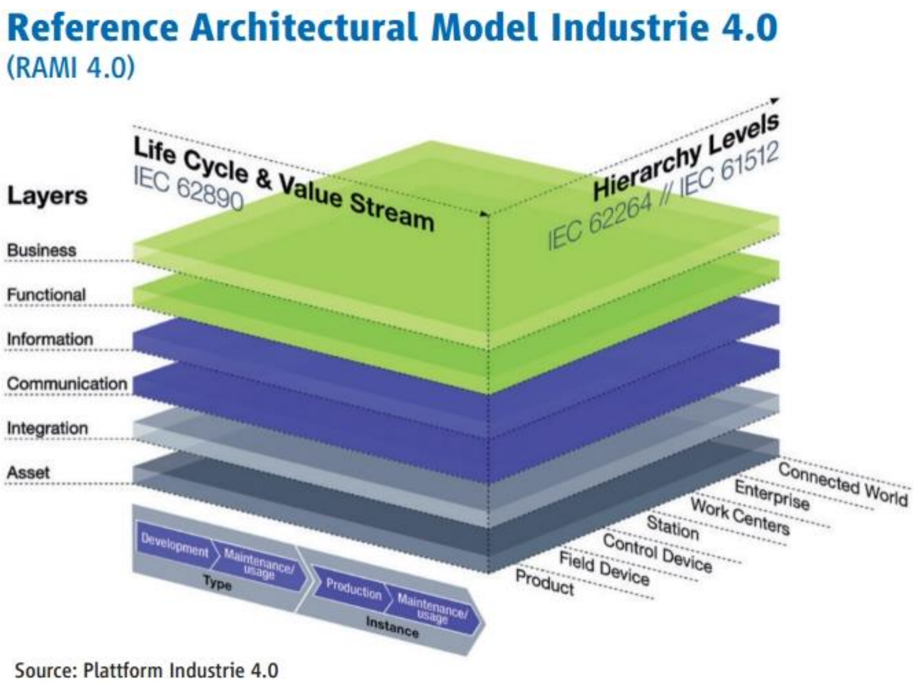


Figure 10 RAMI 4.0 Reference Architecture

The “Hierarchy Levels” axis indicated on the right horizontal axis are hierarchy levels from IEC 62264, the international standards series for enterprise IT and control systems. These hierarchy levels represent the different functionalities within factories or facilities.

The “Life Cycle & Value Stream” axis represents the life cycle of facilities and products, based on IEC 62890 for life-cycle management. Furthermore, a distinction is made between “types” and “instances”. A “type” becomes an “instance” when design and prototyping have been completed and the actual product is being manufactured. In order to represent the Industrie 4.0 environment, these functionalities have been expanded to include workpieces, labelled “Product”, and the connection to the Internet of Things and Services, labelled “Connected World”.

The six layers on the vertical axis serve to describe the decomposition of a machine into its properties structured layer by layer, i.e. the virtual mapping of a machine. Such representations originate from information and communication technology, where properties of complex systems are commonly broken down into layers (see Figure 11).

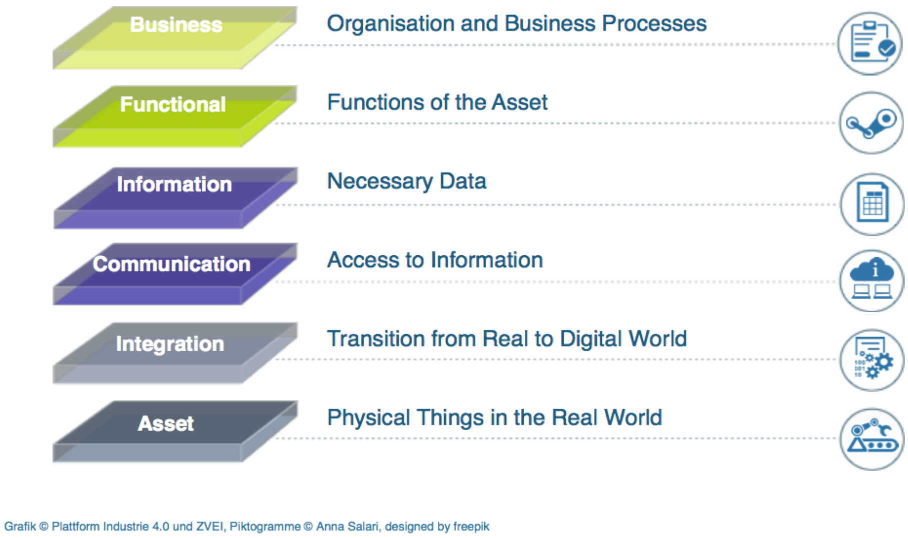


Figure 11 RAMI 4.0 Layers

Within these three axes, all crucial aspects of Industrie 4.0 can be mapped, allowing objects such as machines to be classified according to the model. Highly flexible Industrie 4.0 concepts can thus be described and implemented using RAMI. The reference architectural model allows for step-by-step migration from the present into the world of Industrie 4.0 [6].

One of the main objectives of RAMI is to provide an end-to-end (i.e., since the inception of the product's idea to its dismantling or recycling) framework that is able to connect and consistently correlate all technical, administrative and commercial data to create value streams providing added value to the manufacturer. Elements "active" within the RAMI layers are called *Industrie 4.0 component* (**I4.0 component**). In summary, an *I4.0 component* can be characterized as having the following features:

- an *I4.0 component* provides data and functions within an information system about an, even complex, object;
- an *I4.0 component* exposes one or more end-points through which its data and functions can be accessed;
- *I4.0 components* have to follow a common semantic model.

Therefore, the RAMI framework aims to define how *I4.0 components* can communicate and can be coordinated to achieve the manufacturing objectives. A distinguishing element of *I4.0 components* is the **Administration Shell**: i.e. the "smart element" that transforms an object into an *I4.0 component*. Figure 12 shows the logical relationship between an *I4.0 component* and the physcal *object(s)* it represents.

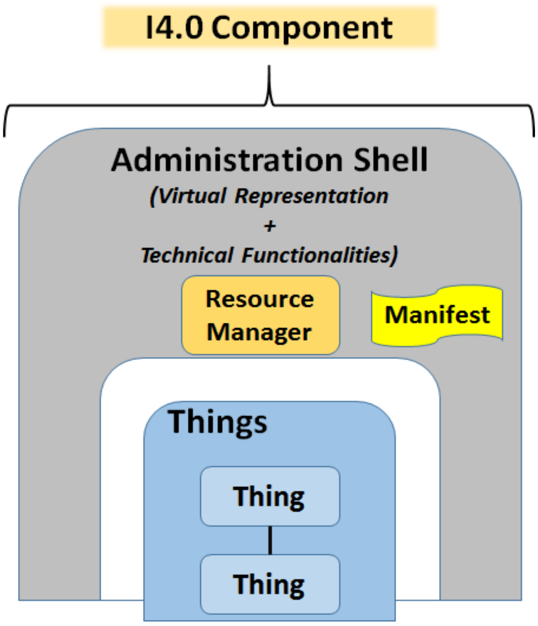


Figure 12 I4.0 Component

The *Administration Shell* is the element in charge of exposing the *I4.0 component* end-point(s) and, therefore, able to interact with other external elements and act as "resource manager" for the represented *object(s)*. The *Manifest* contains the meta-information of the *I4.0 component* and, therefore, constitutes the basis for the virtual representation within a *RAMI4.0* context of the *object(s)*. Among the other data, the *Manifest* contains mandatory *I4.0 component* data necessary to identify it or to communicate with it.

Currently RAMI does not provide detailed, strict indications for standards related to the communication or information models, even if some references are provided in the current architecture document. In particular for:

- the *Communication Layer* an element to be taken into account is the *OPC UA* (Basis IEC 62541) specifications;
- the *Information Layer* the current, initial indications point to the *IEC Common Data Dictionary* (IEC 61360), the *Electronic Device Description*[2] (EDD), and the *Field Device Tool* (FDT) specifications [2];
- the *Functional and Information Layer* the *Field Device Integration*[3] (FDI) specification as integration technology.

The RAMI specification currently indicates, for end-to-end engineering, the *AutomationML* [3] and the ProSTEP iViP[4] specifications. Anyway, the RAMI reference model will adhere to relevant standards in the field and will try to highlight missing features and stimulate the standardization bodies to fill the gaps.

With respect to the latter point, OPC UA is central to the RAMI strategy. It's the successor of the much popular (in Microsoft-based shopfloors) OPC machine-to-machine communication protocol for industrial automation. As opposed to OPC, OPC UA is open, royalty-free, cross-platform and supports very complex information models. I4.0 Components will be required to adopt OPC UA as their interfacing mechanism, while also relying on several IEC standards (e.g., 62832, 61804, etc.) for information sharing.

3.2 IIRA

The Industrial Internet Reference Architecture (IIRA)[5] has been developed and is actively maintained by the Industrial Internet Consortium (IIC), a global community of organizations (>250 members, including IBM, Intel, Cisco, Samsung, Huawei, Microsoft, Oracle, SAP, Boeing, Siemens, Bosch and General Electric) committed to the wider and better adoption of the Internet of Things by the industry at large. The IIRA, first published in 2015 and since evolved into version 1.8 (Jan 2017), is a standards-based architectural template and methodology for the design of Industrial Internet Systems (IIS). Being a RA, it provides an ontology of IIS and some architectural patterns, encouraging the re-use of common building blocks and promoting interoperability. It is worth noting that a collaboration between the IIC and Platform Industrie 4.0, with the purpose of harmonizing RAMI 4.0 and IIRA, has been announced[6] and the kick-off events of a world tour are now taking place[7].

IIRA has four separate but interrelated *viewpoints*, defined by identifying the relevant stakeholders of IIoT use cases and determining the proper framing of concerns. These viewpoints are: business, usage, functional and implementation.

- The *business viewpoint* attends to the concerns of the identification of stakeholders and their business vision, values and objectives. These concerns are of particular interest to decision-makers, product managers and system engineers.

[2] http://www.eddl.org

[3] FDI Cooperation, "FDI - Field Device Integration Technology", January 2012 (http://www.fdi-cooperation.com/tl_files/images/content/Publications/FDI-White_Paper.pdf)

[4] http://www.prostep.org/en/medialibrary/publications.html

[5] http://www.iiconsortium.org/IIRA.htm

[6] http://www.iiconsortium.org/iic-and-i40.htm - to date, no concrete outcomes of such collaboration have been yet published.

[7] http://www.iiconsortium.org/iiot-world-tour/index.htm

- The *usage viewpoint* addresses the concerns of expected system usage. It is typically represented as sequences of activities involving human or logical users that deliver its intended functionality in ultimately achieving its fundamental system capabilities.
- The *functional viewpoint* focuses on the functional components in a system, their interrelation and structure, the interfaces and interactions between them, and the relation and interactions of the system with external elements in the environment.
- The *implementation viewpoint* deals with the technologies needed to implement functional components, their communication schemes and their lifecycle procedures.

Overall, the functional viewpoint tells us that control, management and data flow in IIS are three separate concerns having very different non-functional requirements, so that implementation choices may also differ substantially. On the other hand, the implementation viewpoint describes some well-established architectural patterns for IIS: the Three-tier, the Gateway-mediated Edge Connectivity and Management and the Layered Databus. The *Three-tier architectural pattern* distributes concerns to separate but connected tiers: Edge, Platform and Enterprise. Each of them play a specific role with respect to control and data flows, as depicted in the following picture Figure 13Figure 13.

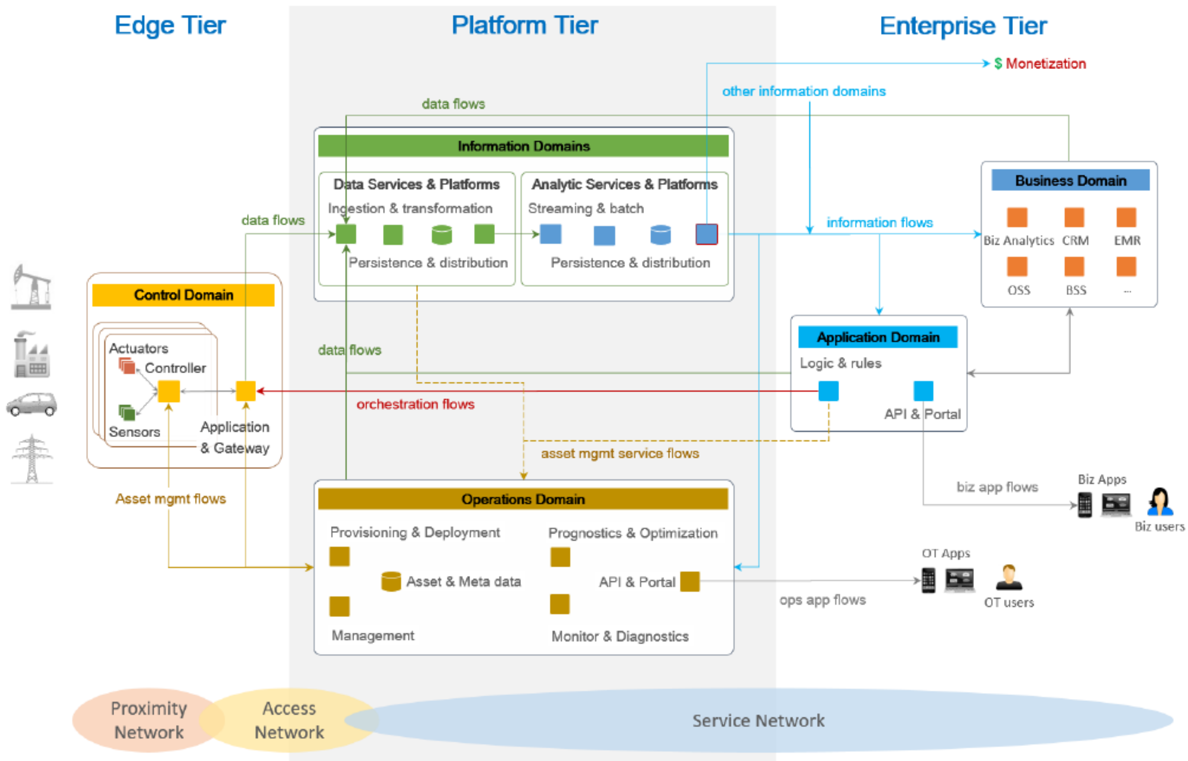


Figure 13 IIRA Three-Tier Architecture Pattern (implementation + functional viewpoint)

The implementation viewpoint indeed provides some very relevant building blocks for the A4BLUE platform. Another relevant contribution given by IIRA RA is in the definition of the following architectural pattern: the *Layered Databus*. According to this design, a IIS can be partitioned into multiple horizontal layers that together define a hierarchy of scopes: machine, system, system of systems and internet. Within each layer, components communicate with each other in a peer-to-peer (P2P) fashion, supported by a layer-specific databus. A databus is a logical connected space that implements a common data model, allowing interoperable communications between endpoints at that layer. For instance, a databus can be deployed within a smart machine to connect its internal sensors, actuators, controls and analytics. At the system level, another databus can be used for communications between different machines. At the system of systems level, still another databus can connect together a series of systems for coordinated control, monitoring and analysis.

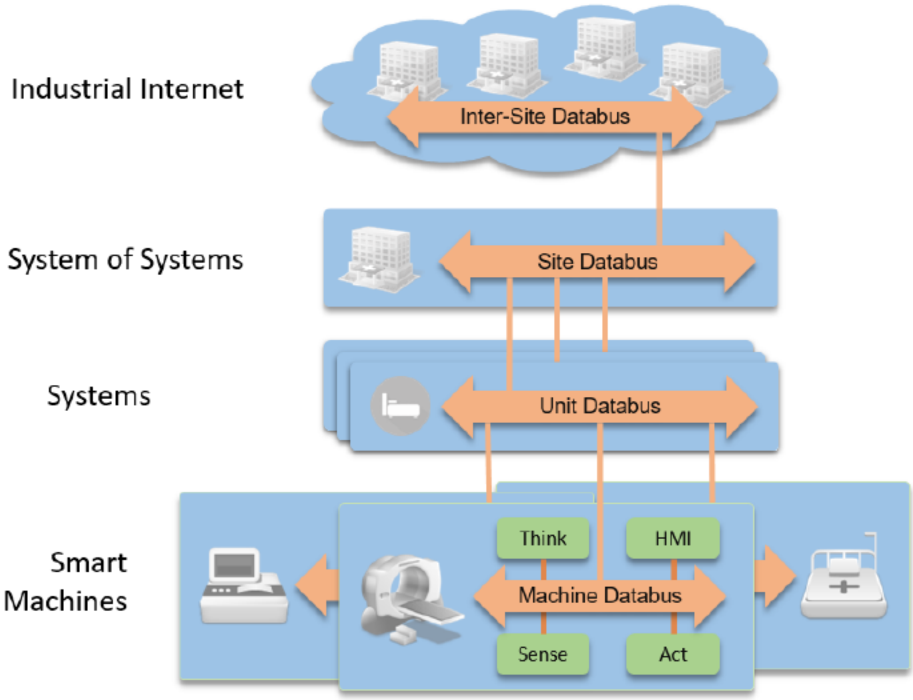


Figure 14 IIRA Layered Databus Architecture Pattern (implementation viewpoint)

3.3 FIWARE FOR INDUSTRY

FIWARE is an open initiative in the scope of the Future Internet PPP (FI PPP) program, aiming at the creation of a sustainable ecosystem of Cloud-ready generic components – aka Generic Enablers (GE) – that may be used as the foundational building blocks of Future Internet solutions in any area, effectively supporting the new wave of digitalization of EU industry and society. The FIWARE Community members are committed to materialise the FIWARE mission, that is: "to build an open sustainable ecosystem around public, royalty-free and implementation-driven software platform standards that will ease the development of new Smart Applications in multiple sectors". From the very beginning, FIWARE was built thanks to the joint efforts of different actors; and now, FIWARE goes a step further in the creation of a community to gather web entrepreneurs, mentors, investors, students, academia, industry and public sector innovators to keep progressing. In FIWARE technologies, developers can gather context information at large scale from many different sources. FIWARE also helps to easily process, analyse and visualize managed context information, easing the implementation of the smart behaviour and the enhanced user experience required by next-generation Smart Applications. Using FIWARE, organizations can capture the opportunities that are emerging with the new wave of digitalisation brought by combining the Internet of Things with Context Information Management and Big Data services on the Cloud. FIWARE Community is not only formed by contributors to the technology (the Open Source Community working on the FIWARE platform), but also those who contribute in building the FIWARE ecosystem and making it sustainable over time.

In the same FI PPP scope presented for FIWARE, **FITMAN** (Future Internet Technologies for MANufacturing) was a large-scale use case project, successfully completed by September 2015. Its mission was to assess the FIWARE platform in the context of ten industrial trials of various sizes and belonging to several manufacturing sectors. FITMAN also developed its own specialized Open Source components – Specific Enablers (SE) – filling some of the gaps existing between FITMAN's use case requirements and FIWARE platform's capabilities. Moreover, three reference architectures were designed by assembling the available building blocks (GEs + SEs) into baseline platforms, each one targeted at a specific EFFRA domain – i.e., Smart Factory, Digital Factory and Virtual Factory. Each baseline platform was aimed at fulfilling a series of common requirements that are intrinsic to its domain of reference.

The **FIWARE for Industry (F4I)** initiative is the main exploitation vehicle for the results of the FITMAN project. FW4I was created by the FITMAN consortium but also involves a larger community of end users and software developers. FIWARE for INDUSTRY is, therefore, an open initiative aiming at collecting and coordinating EU-National-Regional projects interested in the evolution of Generic and Specific Enablers developed in several research programmes. The F4I ecosystem is in constant evolution as many new projects (e.g. those belonging to the Connected Factories cluster) are joining and expressing their interest to develop new FIWARE-based components or to evolve existing GEs or SEs implementations.

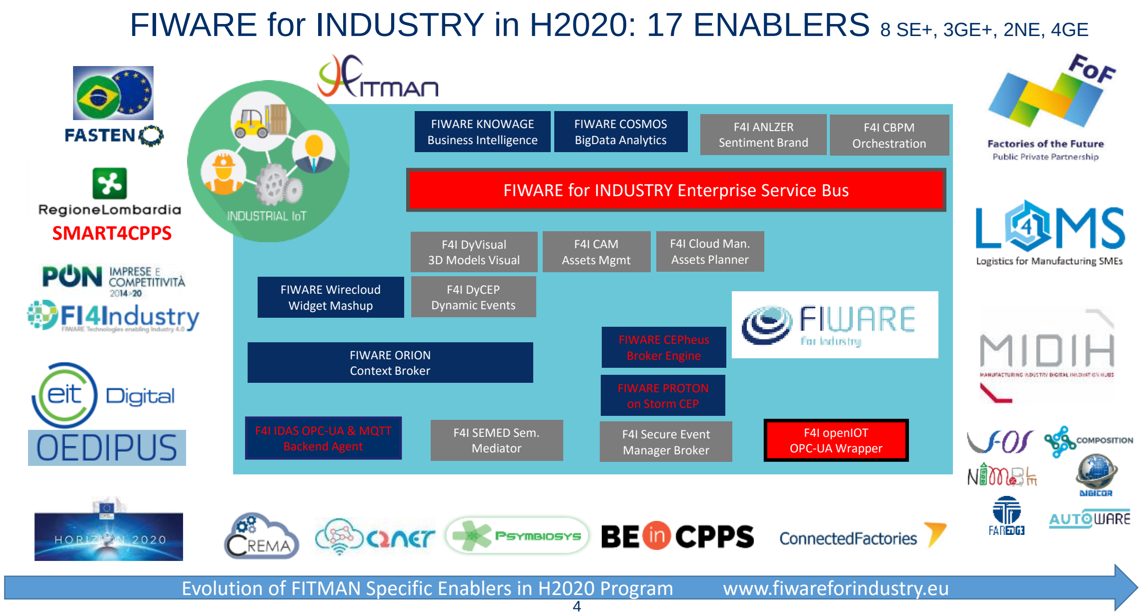


Figure 15 FIWARE for Industry H2020 ecosystem

F4I is proposing, alongside the three original FITMAN architectures for the Smart, Digital and Virtual Factory, a fourth one named **Industrial IoT Reference Architecture (IIOT-RA)**. This design follows the same approach of the previous ones: wiring together FIWARE Generic Enablers and FITMAN Specific Enablers into an integrated platform which aims at solving some key problems of the industry. The rationale behind the choice of introducing a new platform was to make good use of lessons learned from the field of FITMAN's ten industrial trials. This meant addressing more complex real-world scenarios involving multiple levels of the Enterprise, and also expanding the platform's functional portfolio with the introduction of new KETs like Big Data and Machine Learning for Complex Event Processing. Overall, IIOT-RA is a good synthesis of FITMAN's Smart, Digital and Virtual architectures, with a major focus on the Smart domain (basically, a Smart core with Digital and Virtual facets). It is the blueprint of a multi-layered, Cloud-enabled IT infrastructure with a strong support for advanced Shopfloor processes that involve IoT devices and Smart Systems. Figure 16 below, borrowed from the public F4I website, illustrates IIOT-RA's components and their mutual relationships.

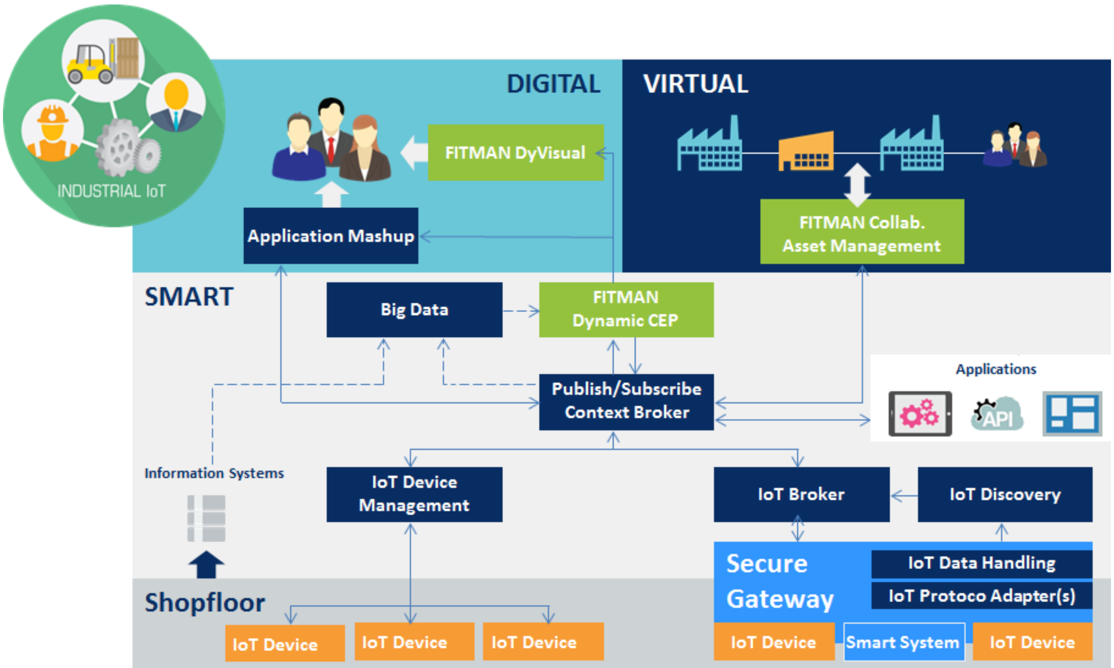


Figure 16 FIWARE for Industry IIOT Reference Architecture

The bottom layer of IIOT-RA – i.e., the Smart core – is characterized by a bi-directional, two-lane flow of events between the Shopfloor and the Cloud. The two lanes play the same role and share the same upper infrastructure, but address quite different scenarios.

On the left hand side, the Fast & Wide Lane is represented by the **IoT Device Management (IoT DM) GE** from FIWARE[8]. This is a lightweight middleware that adapts common IoT protocols like OASIS MQTT and OMA LWM2M to the FIWARE Open API for IoT cloud enablement – i.e., OMA NGSI with a REST-over-HTTP binding. The FIWARE Generic Enabler Catalogue offers an Open Source implementation of IoT DM which is based on the concept of IoT Agents: small protocol-specific modules (typically developed using the C++ language to maximize runtime performance) that do a straightforward protocol adaptation job without interfering with the data payload in any way. This very modular software architecture allows for an easy integration of new protocols on need. The Fast & Wide Lane of IIOT-RA is best suited for very large (thousands of devices) automation and monitoring scenarios with very tight time constraints (near-real-time) but not low-level event pre-processing requirements (more on this in the next paragraph).

On the opposite, right hand side, the Smart & Deep Lane puts in place a Shopfloor-deployed appliance for both protocol and data adaptation. This appliance – the Secure Gateway – is basically an Edge Node where event pre-processing can be performed in close proximity to the source. Typically, event pre-processing involves filtering, transformation and aggregation, and its main purpose is to deflate data streams running from the Shopfloor to the Cloud, lifting much of the load from the network. In addition, the appliance provides – off-the-shelf – a secure Shopfloor/Cloud communication channel. On top of the Secure Gateway, and in Cloud territory, a FIWARE **IoT Broker GE** exposes the same standard FIWARE Open API for IoT to the upper layers. The Smart & Deep Lane addresses scenarios where fewer devices produce massive (and possibly sensitive) data, and constraints allow more time for complex processing.

[8] https://catalogue.fiware.org/enablers/backend-device-management-idas

Due to their common northbound API, both Lanes plug into the same **Publish / Subscribe Context Broker** (PSCB) module[9]. This component, as its name implies, is a FIWARE GE which implements the publish / subscribe pattern for asynchronous message exchange, and is the central hub for all connected systems – i.e., the upper layers of the platform as well as those external applications and services that leverage the platform's Smart core. Similarly, to the lower layers, the PSCB hub is a FIWARE Open API for IoT (i.e., OMA NGSI) service, so that integration is straightforward using web protocols. On the other hand, PSCB can also keep historical events in a persistent storage of its own – as opposed to the IoT Broker and IoT DM components which are stateless – and make them available for inquiry. This feature helps making PSCB the optimal entry point for Shopfloor monitoring, automation and intelligence applications.

That said, the smart characterization of the platform's Smart core actually comes from the FIWARE **Big Data (BD) GE** and the FITMAN **Dynamic CEP (DyCEP) SE**. This works in close cooperation to realize an online Complex Event Processing service that can auto-adapt dynamically to changes in the working environment and in the incoming data. Online adaptation happens by means of a continuous Machine Learning process running offline in the background. The BD component supports such processes by analyzing massive historical data (extracted from persistent storage – i.e., legacy factory systems as well as the PSCB itself) in batch mode and discovering ex-post phenomena of interest (e.g., behavioural patterns); CEP logic is then updated on the fly to reflect this new knowledge. The DyCEP component, on the other hand, implements a special-purpose computing network micro-architecture supporting highly scalable distributed CEP pipelines. Overall, such dynamism represents a groundbreaking technology innovation, as the system can incrementally and autonomously improve its own capabilities. Finally, it is worth noting that DyCEP is not only a consumer of events, but a producer as well: the outcome of event processing logic is often an event stream (e.g., notification messages), that is made available to applications through the same PSCB hub from which incoming streams came from.

On top of the Smart Core, the Digital Facet of IIOT-RA is where human users connect to the platform. Two different components provide a web-based interface for users to interact with the Shopfloor. On one side, the FITMAN **DyVisual SE** is for dynamic rendering of 3D content described using the XML3D language. Models can be rotated, zoomed and virtually navigated by means of point-and-click mouse commands. In the IIOT-RA context, complex shopfloor situations (as represented by the DyCEP component) can be displayed in 3D to make them easier to understand. On the other side, the FIWARE Application Mashup GE allows user-specific cockpits to be built by assembling widgets on a web canvas. Widgets are modular UI components, selected from a library or developed for ad-hoc purposes, that leverage a common framework in order to communicate with the FIWARE Open API for IoT cloud enablement and among themselves.

Finally, the Virtual facet of IIOT-RA hosts the FITMAN **Collaborative Asset Management (CAM) SE**. This is a web-based, integrated platform for the management of virtual assets – i.e., digital representations of tangible things (e.g., devices, equipment, machinery, vehicles, infrastructure, products, people) and intangible concepts (e.g., bills of materials, SLA agreements, reference cards) that are of interest in the scope of the factory's business processes. Virtual assets are described in terms of a custom ontology (i.e., classes and properties) and stored in CAM's online repository. Virtualization is done by human operators through a simple web interface, and does not require any specific technical expertise. Once virtualized, assets become first-class citizens of the platform's IoT perspective: applications can interact over the network with them as things, using the FIWARE Open API service exposed by the PSCB hub.

[9] https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

4 A4BLUE ALIGNMENTS

A4BLUE is not the sole effort that focuses on a general purpose architecture and on an associated reference implementation of an adaptation platform improving working places conditions. Acknowledging the benefits of adaptive capabilities in the Industrie4.0 era, several other initiatives are exploring similar directions. Standards Development Organizations (SDOs), such as the **Platform Industrie 4.0** or the **Industrial Internet Consortium (IIC)** have produced Reference Architectures. Every RA outlines the structuring principles of systems for industrial applications, addressing a wide range of industrial use cases in multiple sectors, including also manufacturing.

4.1 ALIGNMENT WITH RAMI 4.0

RAMI 4.0 is an interesting architecture proposal. A visual rendering of how real world entities can be mapped to the X and Y axes of RAMI's 3D model is given in the figure below. Notably, only RAMI's Hierarchy Levels have a clear and unique mapping, while Life Cycle & Value Stream phases have a more blurred correspondence.

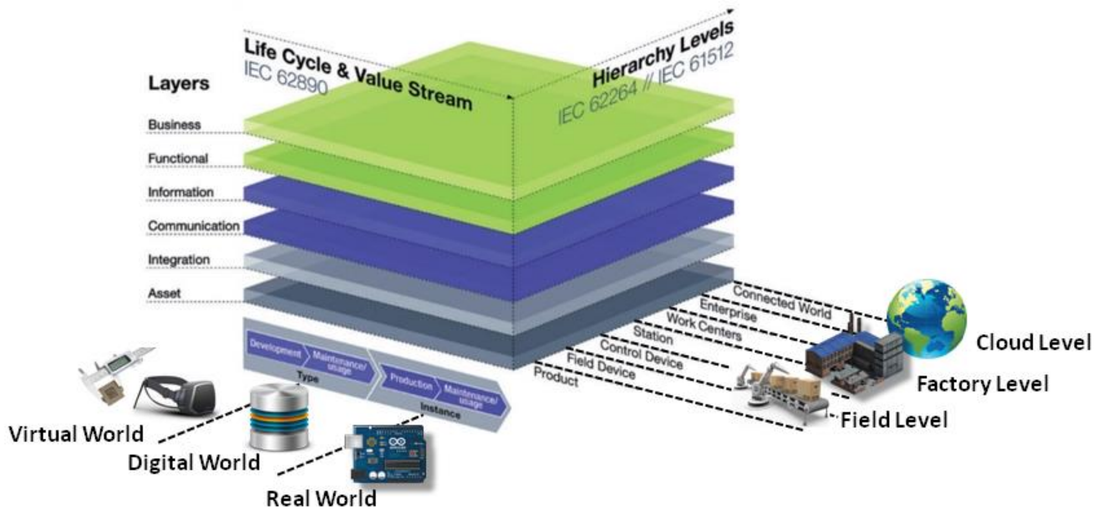


Figure 17 Mapping RAMI axis with real world entities

RAMI 4.0 has gained a significant traction in Germany, and is also driving the discussion around Industry 4.0 solutions and platforms in Europe. In particular, its glossary and its 3D structure for element mapping are increasingly used in sector-specific projects (in particular platform-building ones) and working groups as a common language. The A4BLUE RA will adopt some of the RAMI 4.0 conceptual framework as its own, simplifying communication with the external communities of developers and users. Moreover, some key design and implementation choices will ensure that the A4BLUE Platform is mostly compatible with Industrie 4.0 systems – as far as this is possible in the absence of a final specification document. RAMI also has the notion of *I4.0 Components*, which can be mapped to different kinds of A4BLUE modules as they will be described later in Chapter "6 – REFERENCE MODEL".

4.2 ALIGNMENT WITH IIRA

In A4BLUE, which deals with platforms rather than solutions, the functional and implementation viewpoints described in the IIRA are the most useful especially regarding the *control domain* and the *operations domain*, where the focus is on reading data from sensors, applying rules and logic, and exercising adapting control over the physical world and IT systems.

A4BLUE RA will exploit some conceptual organization of the components foreseen in the IIRA, by targeting a more general compliance with the layered architecture and the extensive use of databus (even if A4BLUE will prefer an Event-Driven approach).

It is quite interesting to see that RAMI's I4.0 Components, IIRA Entities and A4BLUE modules can serve the same purpose of creating a digital *live representation* of a real-world object (thing, machine or person) that can be integrated into applications. Notably, besides the obvious differences in naming the same concepts, the technical means by which this integration is achieved are different in the three perspectives. Overall, A4BLUE architectural approach is more flexible but still fully compatible with these frameworks

4.3 ALIGNMENT WITH FIWARE for INDUSTRY

The F4I RA fits well with A4BLUE since it is concerned with the integration of the real and the digital worlds, which happens with the mediation of IoT, dealing with both Data-in-Motion and Data-at-Rest. These domains roughly correspond to A4BLUE Shopfloor and Enterprise layers (further described in Section "6 – REFERENCE MODEL"). That said, the F4I RA notion of a context broker as the central hub for all information exchanges can be considered as an enabling technology for the Adaptation Framework that is proposed by A4BLUE. For these reasons, F4I RA was the primary inspiration for the design of A4BLUE modular architecture, that is described in next section.

4.4 A4BLUE RA

A *Reference Architecture* (RA) is often a synthesis of best practices having their roots in past experience. Sometimes it may represent a "vision" – i.e., a conceptual framework that aims more at shaping the future and improving over state-of-the-art design rather than at building systems faster and with lower risk. The most successful RAs are those combining both approaches. Whatever the strategy, a RA is for teamwork: its major contribution to development is to set a common context, vocabulary and repository of patterns for all stakeholders.

In A4BLUE, where we explore the business value of applying innovative adaptive patterns to the smart factory, starting from an effective RA is of paramount importance. For this reason, the A4BLUE Reference Architecture is going to be the very first outcome of the project's platform development effort, paving the way to more technical efforts in WP3, WP4 and WP5. In our research, we are going to consider some well-known and accepted *generic RAs* (as described in sub-section above) as sources of inspiration. The goal is twofold: on the one hand, to leverage valuable experience from large and respected communities; on the other, to be consistent and compatible with the mainstream evolution of the smart factory – e.g., Industrial IoT and Platform Industry 4.0. At the end of this journey we expect the A4BLUE RA to become an asset not only in the scope of the project (as the basis for the A4BLUE Platform specifications), but also in the much wider one of adapting workplaces, where it may guide the design of ad-hoc solutions having worker satisfaction as their main driver.

The prime role of the A4BLUE RA, as already stated, is to guide the engineering of the A4BLUE Platform, so that the first stakeholders to be addressed are the members of the project's internal technical team (and in some extent also part of the business team since the RA definition will target the application scenario used in the project validation phase), which amounts to a small group of people having frequent interactions: to this aim context, vocabulary and conventions must easily be shared and agreed with a minimum of effort, and the RA represents the ideal framework to address this aspect.

5 SPECIFICATIONS

This section identifies the Specifications (FEATURES) to be covered by the A4BLUE solution considering the initial results of T1.1 and T1.2 reported in D1.4 (Requirements book) and D1.5 (Use case applications) and the initial reference implementation.

5.1 ACTORS

The following Table 2 shows the main actors identified in the textual descriptions of the application scenarios as described in D1.5. In such descriptions, several names have been used for the same role, therefore the table represents a refactoring of the original actors list.

Actor Name	Business Scenarios
Assembly Operator	1,2,3,4
Auxiliary Operations Operator	2
Workshop Supervisor (tech.)	2
Quality Supervisor	1,3,4
Maintenance Technician	3
Human Resource Responsible	3
Production Supervisor	4
Assembly Planner	4
System Administrator	2,3,4

Table 2 Main actors identified in A4BLUE application scenarios

This table did not explicitly consider the system administrators that will also been involved.

The following figure groups the above-mentioned in categories that will be further used in the following sections of this document, since these actors are involved in the use case scenarios and the sequence diagrams showed below.

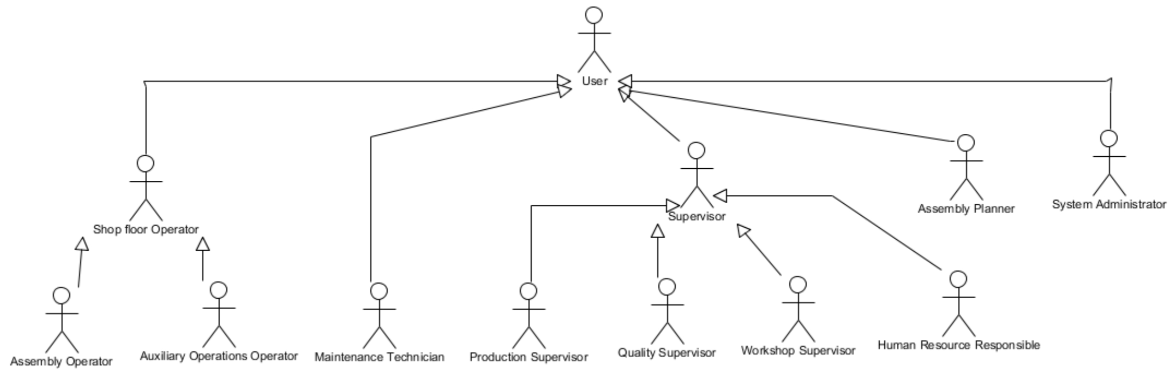


Figure 18 Involved actors

5.2 FEATURES ANALYSIS

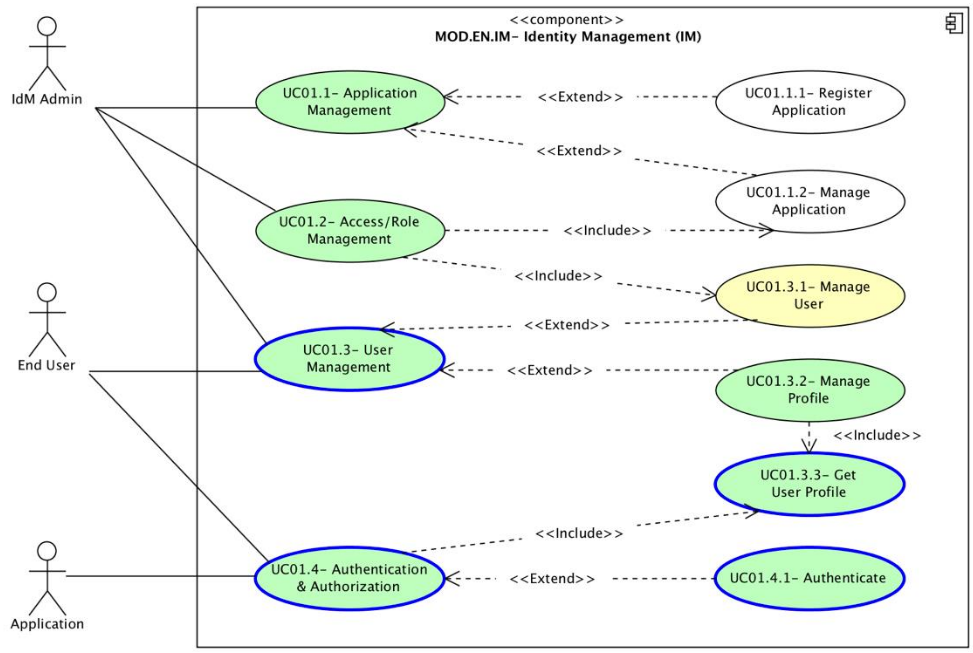
The following table represents the notation followed by the use case included in the sections below. This will be used as a criterion to prioritise functionalities and implementation effort.

(green)	*Identified as essential requirement in D1.4*
(yellow)	*Identified as desirable requirement in D1.4*
(blue border)	*Identified in at least a use case application in D1.5*

Table 3 Use case legend

5.2.1 FTR.01- Identity management

FTR.01.- Identity management	
Objective	Manage users' access to A4BLUE applications, to ensure secure and private authentication, authorization & trust management, user profile management, privacy-preserving disposition of personal data, etc.. Through the use of the provided API, every module of an A4BLUE-based system will be able to assess the user actually using the A4BLUE system, in order to start the adaptation process suitable for the specific logic provided by the requesting module.
Involved actors	ALL
Involved reference components	MOB.EN.IM
External interfaces	N/A
Addressed user level requirements	R2.7, R2.8, R2.11, R2.13, R2.15, R2.16, R4.8, R6.1
Related	SC1.1, SC2.1, SC2.2, SC3.1, SC3.2, SC4.1, SC4.2

FTR.01.- Identity management	
Application scenarios	
Related challenges	CH1.1, CH2.2, CH3.1, CH3.3, CH4.1, CH4.2
Involved Use cases	

UC.01.1 Application Management	
Objective	Register and manage applications able to share the Identities.
Initiation	On demand
Flow of Events	• IdM Administrator registers and configures a new application • IdM Administration manages existing applications • IdM Administration manages existing access/role granting at application level • IdM Administration manages existing access/role granting at user level • The system provides feedbacks on the result of the process

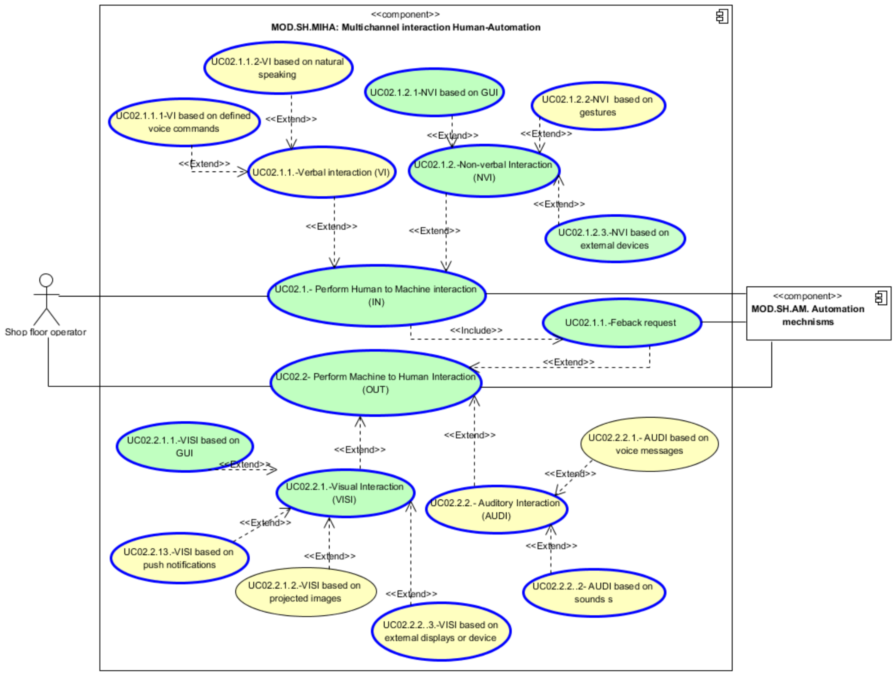
UC.01.2 Access/Role Management	
Objective	Manage granting configuration at application and user level.
Initiation	On demand
Flow of Events	• IdM Administration manages existing access/role granting at application level • IdM Administration manages existing access/role granting at user level • The system provides feedbacks on the result of the process

UC.01.3 User Management	
Objective	Manage user identity and user profile.
Initiation	On demand
Flow of Events	• IdM Administration manages existing identity • End User manages existing profile • The system provides feedbacks on the result of the process

UC.01.4 Authentication & Authorization	
Objective	Manage application authentication and authorization, providing information about the current user profile.
Initiation	On event
Flow of Events	• Application asks to verify authentication and authorization • Application retrieves information about the current user profile • The system provides granting and current user profile

5.2.2 FTR.02- Multimodal human- automation/robot Interaction

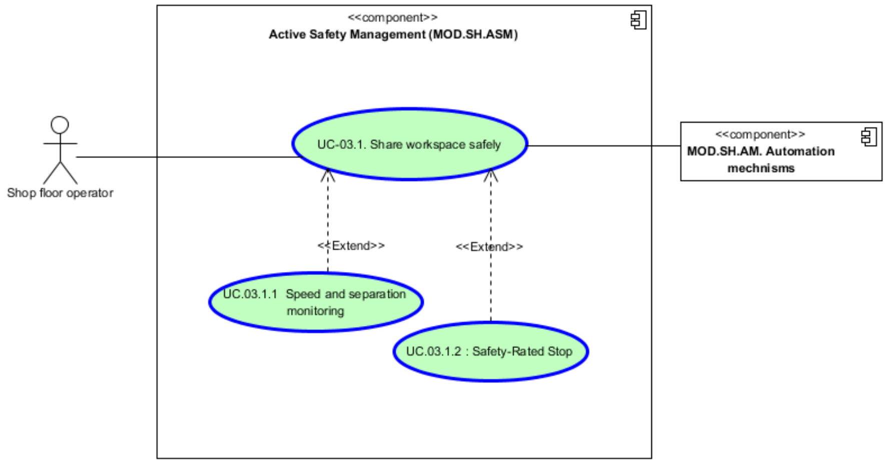
FTR.02.- Multimodal Human- automation/robot Interaction	
Objective	Provide multimodal, multichannel interaction capabilities to allow shop floor operators to communicate with the automation/robots in an easy and intuitive way.
Involved actors	Shop floor Operators
Involved reference components	MOD.SH.MHMI
External interfaces	Interaction devices, sensors and actuators
Addressed user level requirements	R2.2.20, R2.3.1,R2.3.2, R2.3.3, R2.3.4, R2.3.5, R2.3.6, R2.3.8, R2.3.9, R2.3.14, R2.3.17, R2.3.18, R2.3.19, R2.3.22,R2.4.14
Related Application scenarios	SC3.1, SC4.2
Related challenges	CH3.1, CH3.6, CH4.8, CH4.9, CH4.10

<table>
<tr><th colspan="2">FTR.02.- Multimodal Human- automation/robot Interaction</th></tr>
<tr><td>Involved Use cases</td><td>
</td></tr>
</table>

UC.02.1 Perform Human to Machine interaction	
Objective	Support the interaction of the shop floor operator with the automation/robot using verbal (e.g. voice commands, natural speaking) and/or nonverbal channels (e.g. gestures, GUI, external devices) and receive feedback regarding the command interpretation.
Initiation	On demand
Flow of Events	• The shop floor operator provides verbal and/or nonverbal input • The system interprets and fuses, if necessary, the user input channels. • The system provides feedback on instruction interpretation and request confirmation, if required. • The shop floor operator provides confirmation, if required.

UC.02.2 Perform Machine to Human interaction	
Objective	Support the interaction of the automation/robot with the shop floor operator using visual (e.g. GUI) and/or auditory (e.g. sounds, voice messages) mechanisms.
Initiation	On event
Flow of Events	• The system provides a visual and/or auditory output.

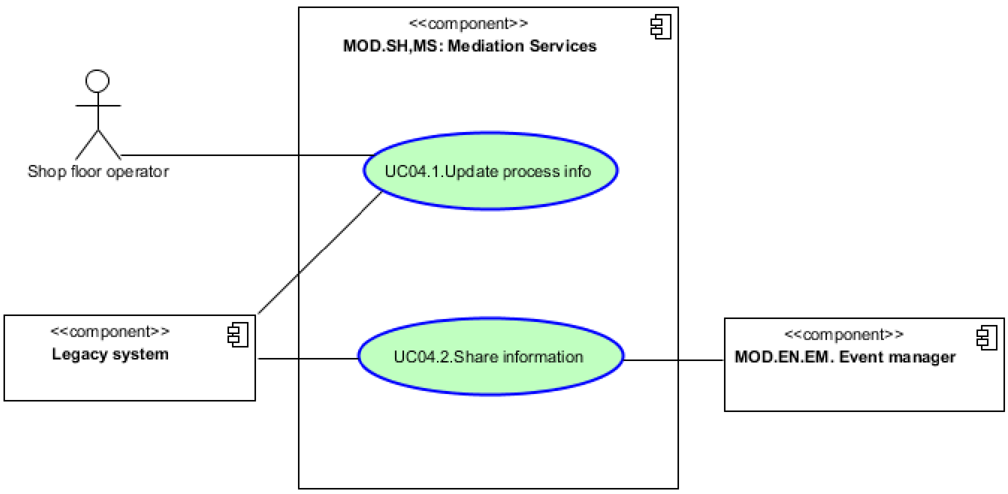
5.2.3 FTR.03- Safe Human - Automation/Robot co-existence

FTR.03.- Safe Human-Automation/Robot co-existence	
Objective	Provide capabilities to enhance the safety in environments where human and automation/robots co-exist in fenceless environments.
Involved actors	Shop floor Operators
Involved reference components	MOD.SH.AS, MOD.SH.AM
External interfaces	Automation mechanism/robot, safety related sensors
Addressed user level requirements	R2.1.23, R2.2.1, R2.2,2, R2.2,4, R2.2,8, R2.2,9, R2.2,10, R2.2,14, R2.2,16, R2.2,19
Related Application scenarios	SC2.1, SC3.1, SC3.2, SC4.2
Related challenges	CH2.2, CH3.2, CH4.8,CH4.9
Involved Use cases	

UC.03.1 Share workspace safely	
Objective	Provide active safety mechanisms to adapt the behaviour of the automation/robot, considering operators context, to enhance the safety in environments where human and automation/robots co-exist in fenceless environments. The behaviour adaptation depends on the safety mode in place and the regulations to be applied. • Safety rated stop mode: in the presence of an operator or obstruction the automation/robot should adapt its behaviour by stopping motion. • Speed & Separation monitoring mode: separation distances are monitored (e.g. by using scanners, vision systems, proximity sensors) and the robot speed reduces when an obstruction is detected and a stop condition given if direct contact proximity is attained (Safety-rated monitored stop). It can include operator motion intention prediction to anticipate adaptation.
Initiation	On event (safety related issue detected)

UC.03.1 Share workspace safely	
Flow of Events	• The system identifies a potential safety issue due to the detection of an obstacle in the automation/robot working zone. • The system adapts the automation/robot behaviour to the detected situation depending on the safety mode in place and the configuration settings based on operator's context, if available.

5.2.4 FTR.04- Integration with enterprise legacy systems

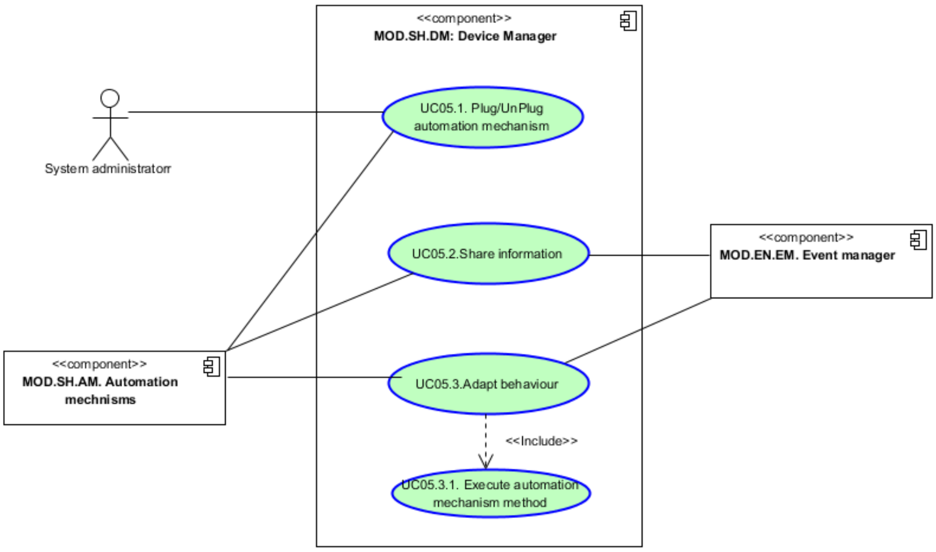
FTR.04.- Integration with enterprise legacy systems	
Objective	Enable seamless integration of the enterprise level legacy systems to support adaptation to variability.
Involved actors	N/A
Involved reference components	MOD.SH.MS, MOD.EN.EM
External interfaces	Enterprise legacy systems (e.g. MES)
Addressed user level requirements	R2.1.15, R2.1.16
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1
Related challenges	CH1.1, CH1.2, CH3.1, CH3.3, CH3.4
Involved Use cases	

UC.04.1 Update process info	
Objective	Allow the shop floor operator to interact with the legacy system to, for example start/end job order; start/end, operation operator login/logout, enter quality information, etc.
Initiation	On demand
Flow of Events	• The shop florr operator interacts with the legacy system during the process.

UC.04.2 Share information	
Objective	Share the updated information during the process.
Initiation	On event
Flow of Events	The legacy systems shares the updated information (e.g. job order start/end; operation start/end, operator login/logout, quality information, etc.) with the A4BLUE system when relevant

5.2.5 FTR.05- Plug and produce adaptive automation mechanisms

FTR.05.- Adaptive automation mechanisms	
Objective	Enable seamless integration of the automation mechanisms (e.g. robot, smart tools) and adaption to human, process or context variability.
Involved actors	System Administrator
Involved reference components	MOD.SH.DM, MOD.SH.AM
External interfaces	N/A
Addressed user level requirements	R2.1.7, R2.1.8, R2.1.12, R2.1.15, R2.1.16, R2.1.23,R2.2.6, R2.2.10, R2.2.11, R2.2.13, R2.2.14, R2.2.18, R2.2.22, R2.4.14
Related Application scenarios	SC1.1, SC2.1, SC3.1, SC3.2, SC4.2
Related challenges	CH1.2, CH1.3, CH2.1, CH2.2, CH3.1,CH3.5, CH3.6, CH4.8, CH4.9

FTR.05.- Adaptive automation mechanisms	
Involved Use cases	

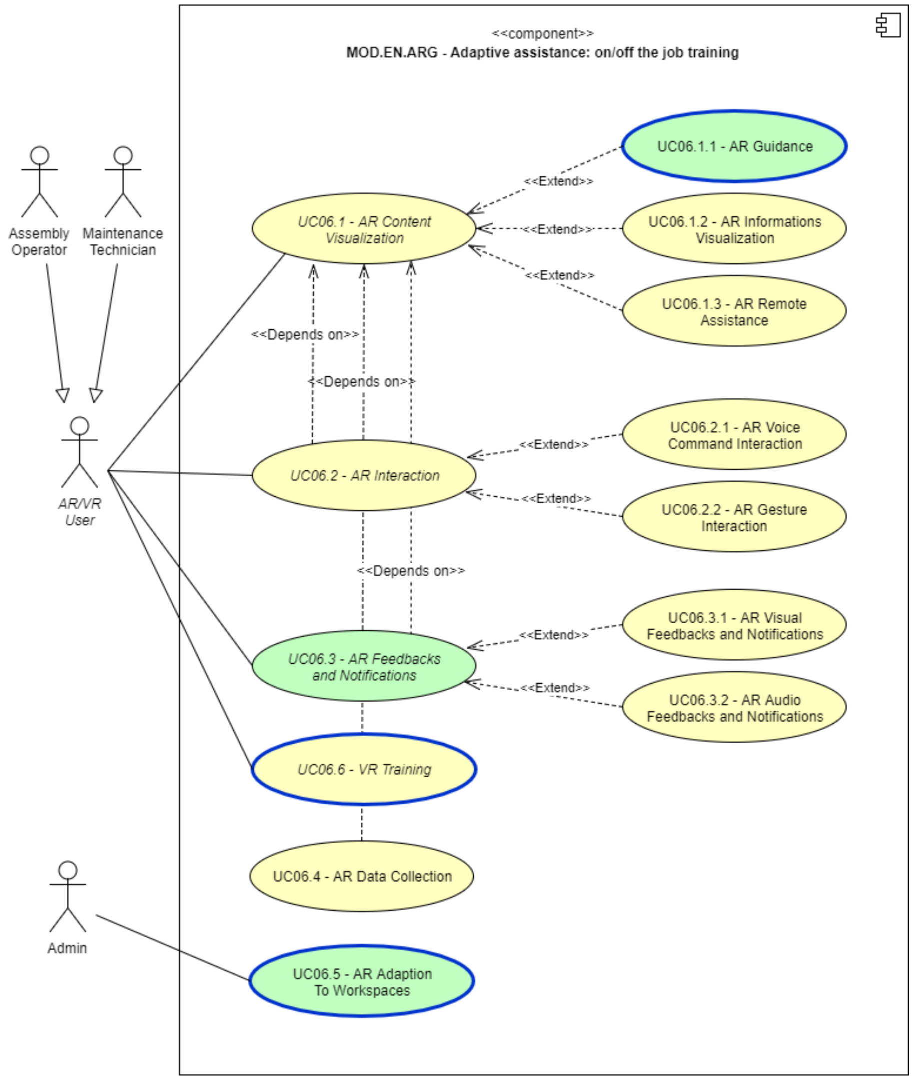
UC.05.1. Plug/ Un-plug automation mechanism	
Objective	Allow the system administrator to plug/un-plug the automation mechanism following a “plug & produce” approach.
Initiation	On demand
Flow of Events	• The system administrator plugs/un-plugs the automation mechanism • The system discovers the exposed variables and methods and registers/de-registers them. If relevant a register/ de-register event is published. • The system subscribes to the exposed variables.

UC.05.2. Share Information	
Objective	Support operation by sharing the required information (e.g. status of the automation/robot, process data,location, etc.).
Initiation	On event
Flow of Events	The automation mechanisms share the required information with the system.

UC.05.3. Adapt behaviour	
Objective	Adapt automation/robot behaviour based on the context.
Initiation	On event
Flow of Events	The system executes the appropriate method to adapt the behaviour of the automation mechanism to the context situation (i.e. considering human and process variability).

5.2.6 FTR.06- Adaptive assistance: on/off the job training

FTR.06.- Adaptive assistance: on/off the job training	
Objective	Provide adaptive assistance through VR/AR
Involved actors	ALL
Involved reference components	MOD.EN.ARG
External interfaces	N/A
Addressed user level requirements	Organisational Level: R1.1, R1.2, R1.3 Communication and Interaction Mechanisms: R3.1, R3.5, R3.7, R3.8, R3.9, R3.10 System Feedback and Assistance: R4.4, R4.10, R4.13 System Information and Instructions: R5.6, R5.7, R5.8, R5.9, R5.10
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1, SC3.2, SC4.1, SC4.2
Related challenges	CH1.1, CH2.4, CH3.3, CH4.2, CH4.3, CH4.9, CH4.10, CH4.11

FTR.06.- Adaptive assistance: on/off the job training	
Involved Use cases	

UC.06.1 Augmented Reality Content Visualization	
Objective	Provide digital content (information, instructions, multimedia, etc.) in an Augmented Reality environment in order to: • Support operators with work procedures that show how to do tasks; • Give fast, easy and immersive access to work material; • Help operators with remote assistance for real-time support.
Initiation	On demand
Flow of Events	• The operator wears his AR device • The operator selects the kind of support he needs ○ The operator chooses to display work procedures ○ The operator chooses to display a multimedia digital content ○ The operator chooses to connect to remote assistance • The operator interacts with the digital content in an immersive way

UC.06.2 Augmented Reality Interaction	
Objective	Enable the possibility to interact with AR digital content using pre-defined voice commands or gestures
Initiation	On demand

Flow of Events	• The operator starts an AR visualization session (see UC.06.1)
	• The operator provides an input
	  ○ The operator pronounces a pre-defined voice command
	  ○ The operator performs a pre-defined gesture

UC.06.3 Augmented Reality Feedbacks and Notifications	
Objective	Provide to the operator feedbacks and notifications through digital visual signals (projected messages, highlights, visual indicators) and auditory effects
Initiation	On event

Flow of Events	• The operator starts an AR visualization session (see UC.06.1)
	• An event that needs to be notified to the operator occurs
	  ○ The AR device displays a visual feedback in the virtual scene
	  ○ The AR device reproduces an audio effect

UC.06.4 Augmented Reality Data Collection	
Objective	Continuously collect data for analysis of system performance and optimisation needs.
Initiation	Automatic and continuous
Flow of Events	• The operator starts an AR visualization session (see UC.06.1) • The AR device continuously collects data about the usage and elementary events occurred on the device, without asking the operator for confirmation • The log of collected events can be extracted and exanimated later in any moment

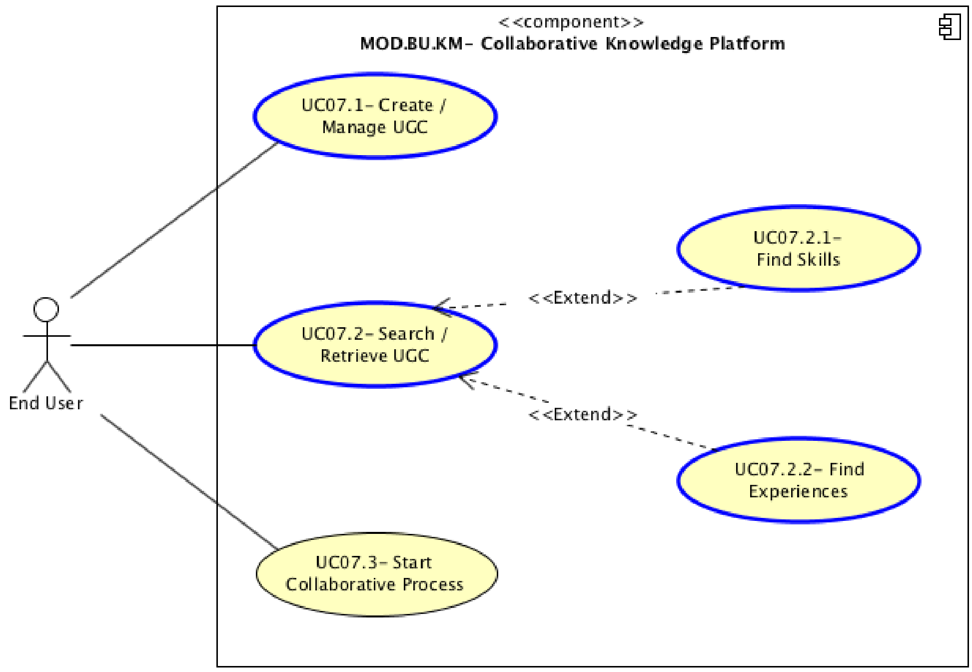
UC.06.5 Augmented Reality Adaption to Workspaces	
Objective	Provide to the system administrator the possibility to easily setup the virtual scene in order to adapt it to the real world. On introduction of new automated system or robotics the AR application receives the new configuration automatically.
Initiation	On demand
Flow of Events	• The system administrator reconfigures the workspace setup on the A4BLUE back-end platform • The operator finds the new configuration on his AR device

UC.06.6 Virtual Reality Training

UC.06.6 Virtual Reality Training	
Objective	Provide an off-the-job tool able to train the operators on working procedures. The tool will exploit 3D and Virtual Reality technologies in order to realistically recreate the workspace in a virtual world and to provide a full immersion experience.
Initiation	On demand
Flow of Events	• The operator gets to the workstation where the Virtual Reality application is installed and prepare himself for the simulation by wearing a VR headset and by grabbing the appropriate VR controllers • The operator selects the training procedures he wants to train in • The operator follows the instructions and learn how to perform the presented tasks

5.2.7 FTR.07- Adaptive assistance: collaborative knowledge management

FTR.07.- Adaptive assistance: collaborative knowledge management	
Objective	Connect human implicit knowledge with factory and production knowledge, making use of robust ICT solutions to provide an online Collaborative Working Environment (CWE) where users can discuss on uncertain environments, learn from experience of the other members of the platform, elicit tacit knowledge from the workers, and evaluate problems using Collaborative Decision Making (CDM) processes to find quick solutions.
Involved actors	Assembly Operator, Workshop Supervisor
Involved reference components	MOD.BU.KM
External interfaces	N/A
Addressed user level requirements	System Feedback and Assistance: R4.9, R4.16 System Information and Instructions: R5.5
Related Application scenarios	SC2.2
Related challenges	CH2.5

FTR.07.- Adaptive assistance: collaborative knowledge management	
Involved Use cases	

UC.07.1 Create/Manage UGC	
Objective	Create and manage User Generated Content (UGC).
Initiation	On demand
Flow of Events	• End User creates, manages and enriches his/her own UGCs • End User enriches UGCs created by other members of the MOD.BU.KM • The system provides feedbacks on the result of the process

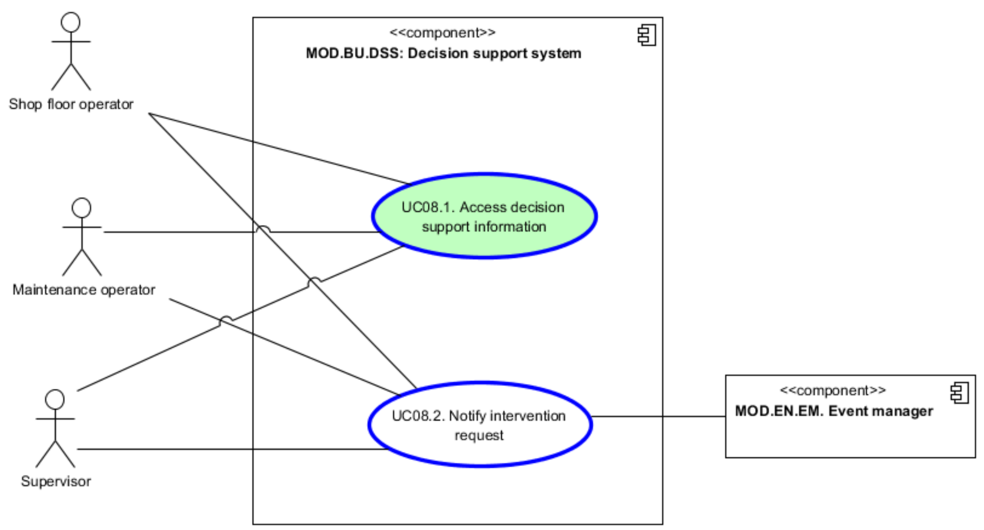
UC.07.3 Search/Retrieve UGC	
Objective	Retrieve and browse User Generated Content (UGC).
Initiation	On demand

Flow of Events	• End User browses and retrieves UGCs
	  ○ End User browses and retrieves skills
	  ○ End User browses and retrieves experiences
	• The system provides feedbacks on the result of the process

UC.07.3 Start Collaborative Process	
Objective	Start collaborative process with the community.
Initiation	On demand

UC.07.3 Start Collaborative Process	
Flow of Events	• End User selects the collaborative process to start (e.g. a collaborative decision making through a SWOT analysis) • End User engages other users from the community • The system provides feedbacks on the result of the process

5.2.8 FTR.08- Adaptive assistance: decision support

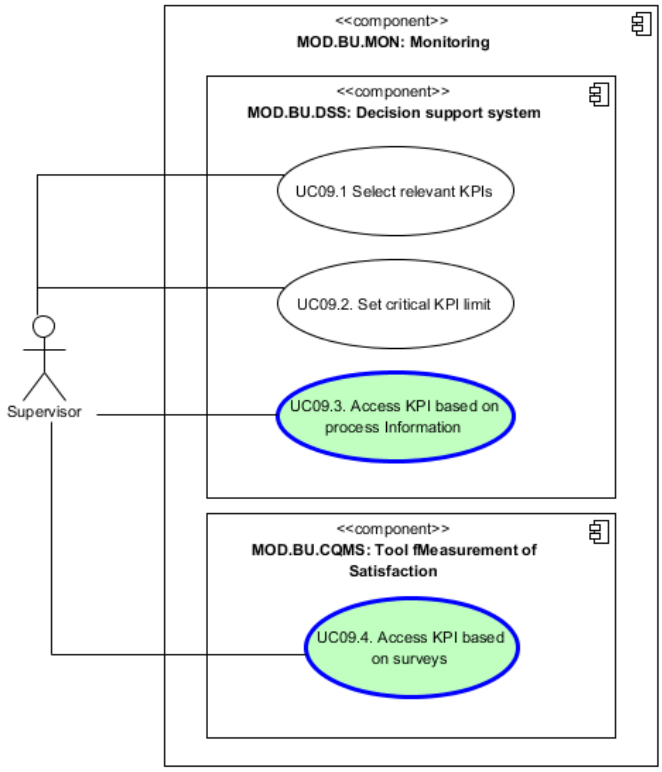
FTR.08.- Adaptive assistance: decision support	
Objective	Support decision making related to process, maintenance and inspection activities by enabling the involved workers (e.g. shop floor and maintenance operators and supervisors) to access the relevant information collected during the operation process when required (e.g. related to process parameters, defects, downtimes.).
Involved actors	Shop floor Operator, Supervisor, Maintenance technician
Involved reference components	MOD.BU.DSS
External interfaces	N/A
Addressed user level requirements	R2.1.1., R2.1.2, R2.1.6, R2.1.7, R2.1.16, R2.1.22, R2.2.3, R2.2.5, R2.2.7, R2.2.12, R2.2.13, R2.2.17, R2.3.11, R2.3.15, R2.3.22, R2.4.6,R.2.4.20, R2.5.1, R2.5, R2.5.8, R2.5.10
Related Application scenarios	SC1.1, SC3.1, SC3.2, SC4.1
Related challenges	CH1.3 CH3.5, CH4.5, CH4.6
Involved Use cases	

UC.08.1. Access decision support information	
Objective	Enable the collection and visualisation of relevant information to support the involved workers decision making process though visual analytics capabilities.
Initiation	On demand
Flow of Events	The users (e.g. shop floor operators, maintenance operator, supervisors) access visual information collected and processed by the system.

UC.08.2. Notify intervention request	
Objective	Notify the users though appropriate channels when their intervention is requested.
Initiation	On event
Flow of Events	The users (e.g. shop floor operators, maintenance operator, supervisors) are notified of intervention requests (e.g. collaboration, maintenance, inspection , quality) using appropriate notification channels (e.g. push notification, email, graphical user interface, light indicator, etc.).

5.2.9 FTR.9- Performance monitoring

FTR.09.- Performance monitoring	
Objective	Enabling the evaluation process to be performed by supporting the collection and visualisation of key performance indicators (KPIs) to assess the impact, from an economic and social perspective. Note: It involves only the information generated in theA4BLUE domain.
Involved actors	Supervisors
Involved reference components	MOD.BU.MON → (MOD.EN.DSS, MOD.BU.CQMS)
External interfaces	N/A
Addressed user level requirements	R2.1.1,R2.2.2,R2.1.21,R2.5.1,R2.5.2,R2.5.10,R2.5.11
Related Application scenarios	SC1, SC2, SC3, SC4
Related challenges	CH4.6

Involved Use cases	

UC.09.1. Select relevant KPIs	
Objective	Enable supervisors to select the relevant performance KPIs that will be measured to support decision making and to avoid manual measurement of KPIs.
Initiation	On demand
Flow of Events	• Supervisors open a list of predefined KPIs • Supervisors select the relevant KPIs

UC.09.2. Set critical KPI limit	
Objective	Set the range within the KPI value is acceptable
Initiation	On demand

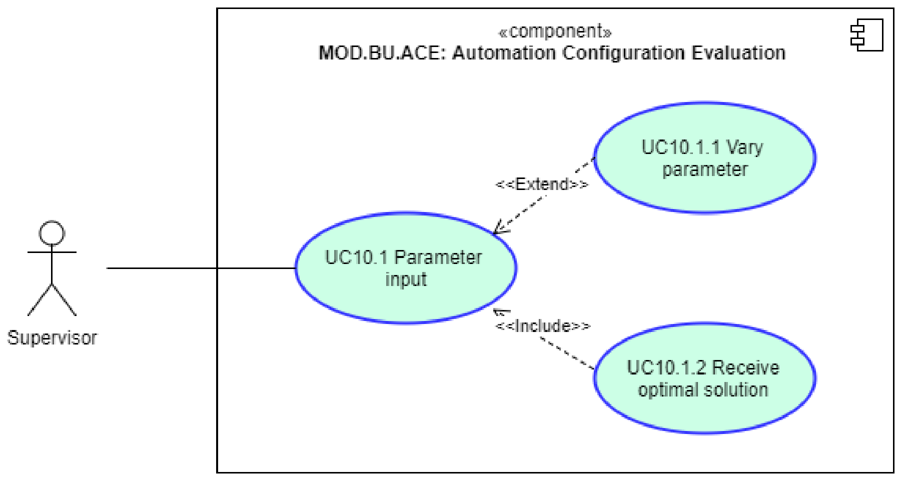
Flow of Events	• Supervisors define the range within the KPI value is acceptable without further action
	  ○ Supervisors set lower limit
	  ○ Supervisors set upper limit
	• If KPI limit is exceeded or undershot, an alert is triggered

UC.09.3. Access KPI based on process information	
Objective	Enable supervisors or Assembly Planners to access relevant performance KPIs based

UC.09.3. Access KPI based on process information	
	in information collected during the process.
Initiation	On demand
Flow of Events	Supervisors access provided performance KPIs collected and processed by the system

UC.09.4. Access KPI based on survey information	
Objective	Enable supervisors to access relevant performance KPIs based surveys.
Initiation	On demand
Flow of Events	Supervisors access provided performance based on surveys.

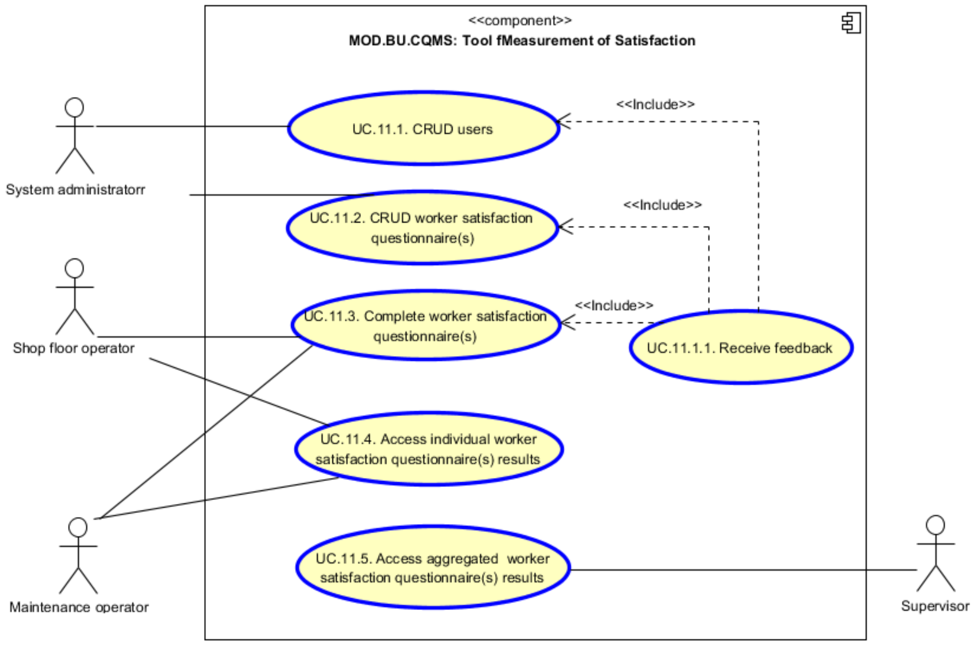
5.2.10 FTR.10- Automation level definition support

FTR.10.- Automation level definition support	
Objective	Provide a tool to identify the optimal automation configuration through varying criteria and process parameters from a socio-technical as well as from an economical perspective.
Involved actors	Assembly Planner
Involved reference components	MOD.BU.ACE
External interfaces	N/A
Addressed use level requirements	R1.10, R1.14
Related Application scenarios	SC4
Related challenges	CH4.7, CH4.12
Involved Use cases	

UC.10.1 Parameter input	
Objective	Provision of system parameters
Initiation	On demand
Flow of Events	• Assembly Planner enters the system parameters for all criteria • Varying of the parameters • Receive the recommendation of the optimal solution

5.2.11 FTR.11- Worker satisfaction assessment

FTR.11.- Worker satisfaction assessment	
Objective	Provide a web-based tool for quantitative measurement of satisfaction enabling the involved users to complete the worker satisfaction questionnaires in an easy way and evaluate the results.
Involved actors	Shop floor Operators, Supervisors
Involved reference components	MOD.BU.CQMS
External interfaces	N/A
Addressed use level requirements	R2.1.13, R2.1.14, R2.4.12
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1 SC3.2, SC4.1, SC4.2
Related challenges	CH4.10

FTR.11.- Worker satisfaction assessment	
Involved Use cases	
Other specifications	Support online (on event based) and offline (stand alone, on schedule) operation.

UC.11.1 CRUD users	
Objective	Create/Read/Update/Delete users.
Initiation	On demand
Flow of Events	• System administrator creates/ updates or deletes user data. • The system provides feedback on the result of the process (UC.11.1.1)

UC.11.2 CRUD worker satisfaction questionnaire	
Objective	Create/Read/Update/Delete worker satisfaction questionnaire(s)
Initiation	On demand
Flow of Events	• System administrator creates/ updates or deletes questionnaire. • The system provides feedback on the result of the process (UC.11.1.1)

UC.11.3- Browse worker satisfaction questionnaire	
Objective	Access to the list of pending questionnaire(s)
Initiation	On demand Pending questionnaires are defined on schedule (offline) or on event basis (online).
Flow of Events	• The system displays the list questionnaire. • Shop floor operator selects the questionnaire.

UC.11.3- Browse worker satisfaction questionnaire	
	• The system displays selected questionnaire (UC.11.1.1)

UC.11.4 Complete worker satisfaction questionnaire	
Objective	Provide the answers to the questionnaire and save it.
Initiation	On demand
Flow of Events	• Shop floor Operator completes the satisfaction questionnaire. • Shop floor Operator saves the questionnaire results. • The system checks if all the required questions have been completed. a. If yes, the questionnaire results are sent. b. If not the system asks the worker to complete the missing questions. • The system provides feedback on the result of the process (UC.11.1.1)

UC.11.5. Access individual view of worker satisfaction questionnaires results	
Objective	Access individual results of worker satisfaction questionnaire(s)
Initiation	On demand
Flow of Events	• Users ask for individual worker questionnaire results. • The system displays the collected results (UC.11.1.1). All the collected information is represented in an easy way prioritising graphical representation whenever possible.

UC.11.6. Access aggregated view of worker satisfaction questionnaires results	
Objective	Access aggregated results of worker satisfaction questionnaire(s)
Initiation	On demand
Flow of Events	• Supervisor asks for aggregated worker questionnaire results. • The system displays the collected results (UC.11.1.1). All the collected information is represented in an easy way prioritising graphical representation whenever possible.

5.3 VERIFICATION

The following traceability matrixed are intended to comply with the V-V process describes in D2.5 – “Test plans and test case specifications - Initial Version”.

5.3.1 Requirements- Feature traceability matrix

Find following the relation between the user level requirements identified in D1.4 and the defined features.

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.1.1											
2.1.2											
2.1.3											
2.1.4											
2.1.5											
2.1.6											
2.1.7											
2.1.8											
2.1.9											
2.1.10											
2.1.11											
2.1.12											
2.1.13											
2.1.14											

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.1.15											
2.1.16											
2.1.17											
2.1.18											
2.1.19											
2.1.20											
2.1.21											
2.1.22											
2.1.23											
2.1.24											
2.2.1											
2.2.2											
2.2.3											
2.2.4											
2.2.5											
2.2.6											
2.2.7											
2.2.8											
2.2.9											

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.2.10											
2.2.11											
2.2.12											
2.2.13											
2.2.14											
2.2.15											
2.2.16											
2.2.17											
2.2.18											
2.2.19											
2.2.20											
2.2.21											
2.2.22											
2.3.1											
2.3.2											
2.3.3											
2.3.4											
2.3.5											
2.3.6											

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.3.7											
2.3.8											
2.3.9											
2.3.10											
2.3.11											
2.3.12											
2.3.13											
2.3.14											
2.3.15											
2.3.16											
2.3.17											
2.3.18											
2.3.19											
2.3.20											
2.3.21											
2.3.22											
2.4.1											
2.4.2											
2.4.3											

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.4.4											
2.4.5											
2.4.6											
2.4.7											
2.4.8											
2.4.9											
2.4.10											
2.4.11											
2.4.12											
2.4.13											
2.4.14											
2.4.15											
2.4.16											
2.4.17											
2.4.18											
2.4.19											
2.4.20											
2.4.21											
2.4.22											

Requirement	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
2.4.23											
2.4.24											
2.5.1											
2.5.2											
2.5.3											
2.5.4											
2.5.5											
2.5.6											
2.5.7											
2.5.8											
2.5.9											
2.5.10											
2.5.11											
2.5.12											
2.5.13											
2.5.14											
2.5.15											
2.5.16											

Table 4 VERIFICATION: Requirements- Feature traceability matrix

5.3.2 Challenge- Feature traceability matrix

Find following the relation between the challenges identified at use case application level and reported in D1.5 and the defined features.

Challenge	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
CH1.1											
CH1.2											
CH1.3											
CH2.1											
CH2.2											
CH2.3											
CH2.4											
CH2.5											
CH3.1											
CH3.2											
CH3.3											
CH3.4											
CH3.5											
CH3.6											
CH4.1											
CH4.2											
CH4.3											
CH4.4											

Challenge	Features										
	FTR-01	FTR-02	FTR-03	FTR-04	FTR-05	FTR-06	FTR-07	FTR-08	FTR-09	FTR-10	FTR-11
CH4.5											
CH4.6											
CH4.7											
CH4.8											
CH4.9											
CH4.10											
CH4.11											
CH4.12											

Table 5 VERIFICATION: Challenge- Feature traceability matrix

6 REFERENCE MODEL

The A4BLUE Reference Model (RM) primarily supports the functional requirements—what the system should provide in terms of services to its users. The system is decomposed into a set of key abstractions, taken (mostly) from the problem domain as stated in previous Section 5. This decomposition is not only for the sake of functional analysis, but also serves to identify common mechanisms and design elements across the various parts of the system to be further details in the next sections of the document.

This section will describe the A4BLUE Reference Model, by providing a description of the elements of the A4BLUE solution at a conceptual level (starting from the Reference Implementation described in the DoA, and then updated following the specifications).

A dedicated naming convention has been used to define the unique codes used all over the document (and in further project deliverables linked to the current D2.6). Here the main rationale behind this convention is:

- “MOD” prefix is used to identify each code referring to any A4BLUE module;
- “SH/EN/BU” code is used to easily identify the level of the the RA hosting the specificed module as further specified in the rest of this section (i.e. SH for shopfloor, EN for enterprise, and BU for business);
- “XYZ” is a short memonic code coming from the full name of the module.

The rest of the section will present the mentioned components, by grouping them into three-fold breakdown structure (see Section “9 – FUNCTIONAL AND MODULAR ARCHITECTURE” for further details):

- **Shopfloor Layer**: the lower layer is intended to ease the interconnection of the A4BLUE Platform with the physical world, by hiding the complexity of dealing with shopfloor IT systems (e.g. PLC, CPS and existing legacy system) as well as dealing with human interactions (e.g. using gesture and voice commands).
- **Enterprise Layer**: the middle layer represents the core part of the A4BLUE Platform, being in charge of managing the core components needed for adaption management using an Event Driven Architecture in order to provide the assistance services. This layer will be also enhanced by tools supporting the tactical decision-making processes by producing and consuming digital information coming from the other layers.
- **Business Layer**: the upper layer is in charge of supporting strategic decision-making process (sometimes using off-line tools), targeting both blue- and white-collar workers.

The reference components are listed here below:

- SHOPFLOOR
  - MOD.SH.MS- Mediation Services
  - MOD.SH.AM- Automation Mechanisms
  - MOD.SH.DM- Device Manager
  - MOD.SH.MMI- Multimodal Interactions
- ENTERPRISE
  - MOD.EN.CAM- Collaborative Asset Manager
  - MOD.EN.EM- Event Manager
  - MOD.EN.ARG- VR/AR based training and guidance
- BUSINESS
  - MOD.BU.KM- Collaborative Knowledge Platform
  - MOD.BU.DSS- Decision Support System (DSS)

- MOD.BU.MON- MONITORING
- MOD.BU.ACE- Automation Configuration Evaluation
- MOD.BU.CQMS- Computer based tool for the measurement of worker satisfaction

6.1 SHOPFLOOR LAYER

6.1.1 MOD.SH.MS- MEDIATION SERVICES

“MOD.SH.MS- Mediation services” component supports the integration of already in place enterprise level legacy systems (e.g. Manufacturing Execution Systems-, etc.). Figure 19 shows the component diagram representing the decomposition of the MOD.SH.MS component into Functional Building Blocks (FBBs) and the relations and data flows between them.

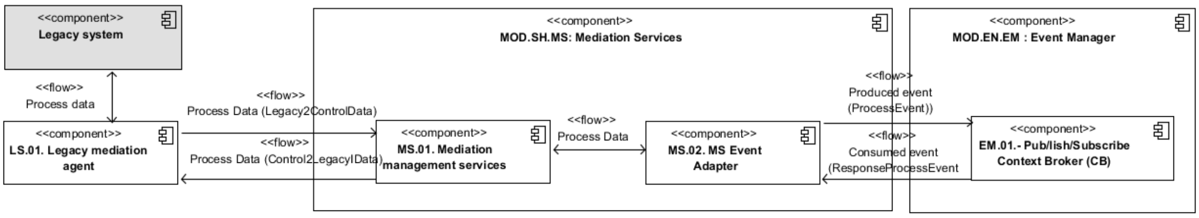


Figure 19 MOD.SH.MS. High Level module decomposition- legacy systems integration

”LS.01 Legacy mediation agent” is legacy system dependent and represents the connector that supports the bidirectional data exchange process between an already in place legacy system and the “MS.01. Mediation management services” component. The “MS.02. MS event adapter” supports publish and subscribe capabilities and adapts the process data collected through the “MS.01. Mediation management services” to the event format supported by the Event Manager (EM) component. It also transforms events coming from EM Event Manager (EM) component into process data to update the legacy system (e.g. information collected during the execution of the operations performed by the automation mechanisms).

Table 6 summarises the Functional Building Blocks (FBB) involved in MOD.SH.MS identifying the type and scope.

FBB	Software			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
LS.01. Legacy mediation agent MS.01. Mediation management services					
MS.02. MS event adapter					

Table 6 MOD.SH.MS Functional building blocks summary

6.1.2 MOD.SH.AM- AUTOMATION MECHANISMS

Automaton mechanisms in the scope of A4BLUE involve both robots and smart tools. Figure 20 shows the component diagram representing the decomposition of the MOD.SH.AM component into Functional Building Blocks (FBBs) and the relations and data flows between them. Figure 21, included in the following section, provides further details about the integration of the automation mechanisms with the A4BLUE system.

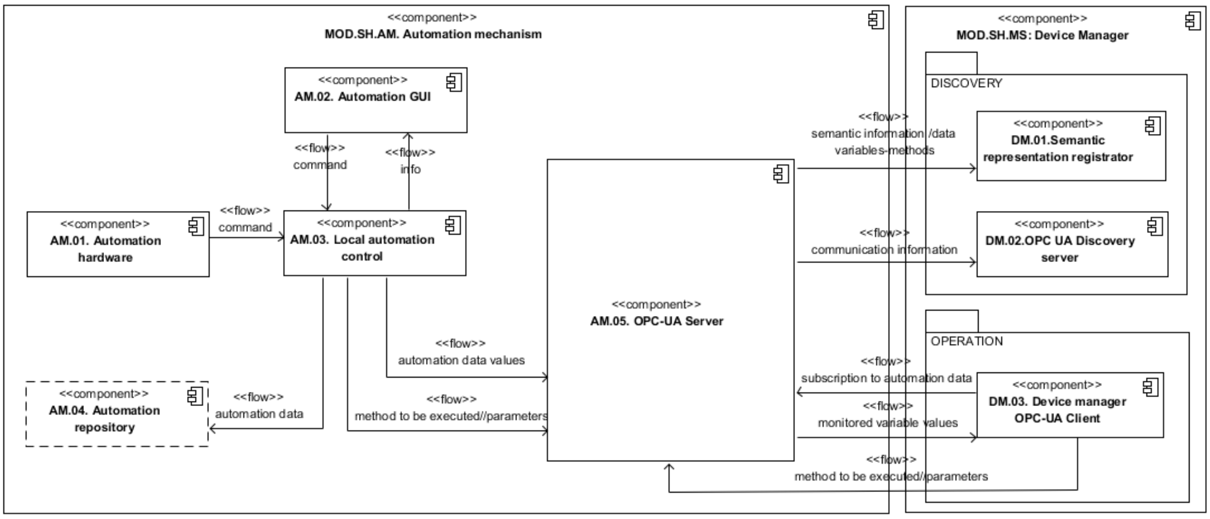


Figure 20 MOD.SH.AM. High Level module decomposition

The "AM.03. Local automation controller" controls the "AM.01. Automation hardware" (e.g. robot, smart tool) that could involve auxiliary devices (e.g. camera, force sensor, etc.) to improve the accuracy of the process in order to get a better final result, collects updated automation data values and allow remote execution of exposed methods. Furthermore, the automation mechanism can involve a graphical user interface to support user interaction (AM.02. Automation GUI).

The automation mechanism should include OPC UA technology to support standard based plug and produce integration with the A4BLUE solution. This involves AM.05. OPC-UA Server aiming to: (1) register automation semantic information (i.e involving available automation data and methods); (2) register communication information (IP, port) into the DM.02 OPC UA discovery server; (3) provide updates of the monitored automation data and execute automation methods. Optionaly it could include a repository to persist relevant data (i.e. AM.04. Automation repository).

Table 7 summarises the Functional Building Blocks (FBB) involved in MOD.SH.AM identifying the type and scope. Automation mechanisms ("MOD.SH.AM- Automation mechanism") component can work either offline (i.e. independently of the A4BLUE framework) or online (i.e. integrated in the A4BLUE framework).

FBB	Type (HW)	Type (SW)			Scope	
		Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
AM.01. Automation hardware						
AM.02. Automation GUI						
AM.03. Local automation controller						
AM.04. Automation data repository (opt)						
AM.05. OPC-UA Server						

Table 7 MOD.SH.AM Functional building blocks summary

6.1.3 MOD.SH.DM- DEVICE MANAGER

“MOD.SH.DM- Device manager” component supports plug and produce capabilities by enabling the integration of external automation mechanisms in a standardised mode (i.e. OPC UA based). It involves both the discovery and operation processes supporting the plug and produce approach.

Figure 21 shows the component diagram representing the decomposition of the MOD.SH.DM into Functional Building Blocks (FBBs) and the relations and data flows between them for the integration of automation mechanisms. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB).

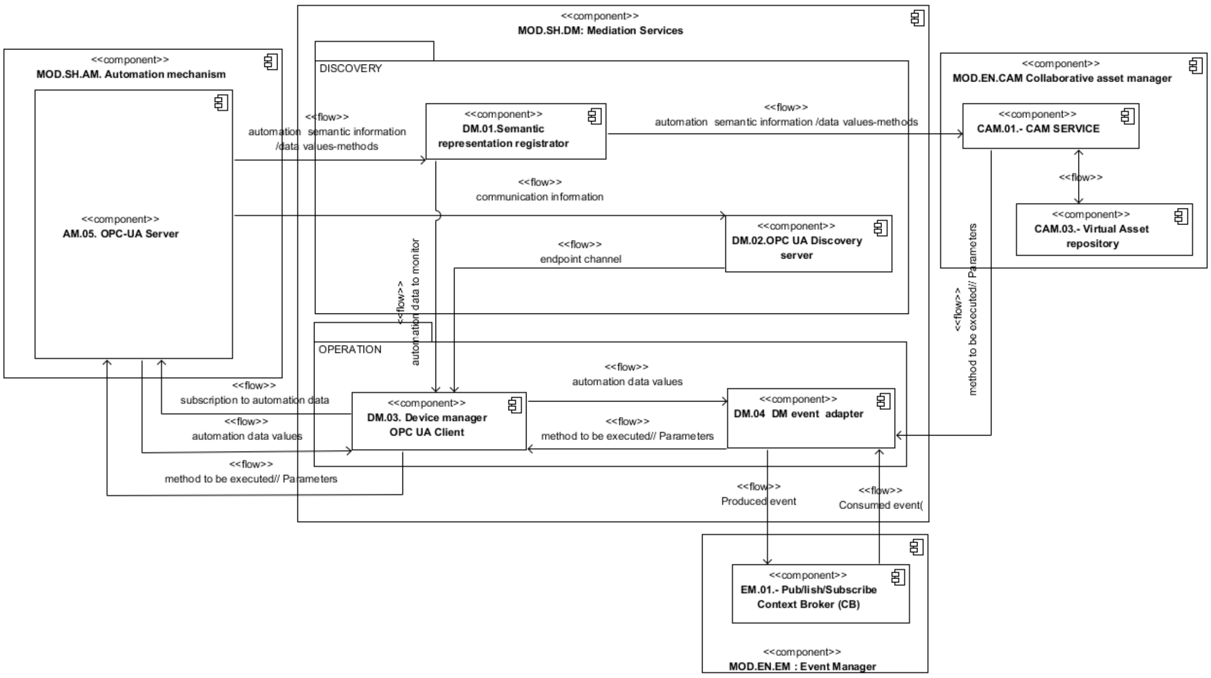


Figure 21 MOD.SH.DM. High Level module decomposition – automation mechanisms integration

Discovery process

The discovery process is initiated when a automation mechanism is plugged or unplugged. The business logic of the discovery process involves the “DM.01. Semantic representation registrator” that registers (i.e. when the automation mechanisms is plugged) or de-registers (i.e when the automation mechanisms is unplugged) the semantic representation provided by the automation mechanisms (i.e including information on the provided automation values and methods and assignation of the automation mechanism to the appropriate work centre) into the “CAM.03. Asset repository” through the “CAM.01.CAM SERVICE” of the “MOD.EN.CAM Collaborative Asset Manager”. The “DM.02 OPC UA discovery server”: maintains the list of OPC UA Servers that are registered and provides mechanisms for clients (“DM.03. Device manager OPC UA”) to find them.

Operation process

The operation process is running while the mechanism is plugged in._The business logic of the operation process is supported by the “DM.03. Device manager OPC UA client” that is subscribed to the automation values monitored by the automation mechanism and executes the appropriate methods, when required, through the right automation OPC UA server (i.e. “AM.05. Automation OPC UA server). The “DM.04. DM event adapter” supports publish and subscribe capabilities and adapts the events produced by the "DM.01. Device manager OPC UA client"" to the event format supported by the Event Manager (EM) component. It also transforms events coming from Event Manager (EM) component into commands to be executed by means of the appropiated methods exposed by the automation mechanism.

Table 8 summarises the Functional Building Blocks (FBB) involved in MOD.SH.DM identifying the type and scope.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
DM.01. Semantic representation registrator					
DM.02 OPC UA discovery server					
DM.03. Device manager OPC UA client					
DM.04. DM event adapter					

Table 8 MOD.SH.MS and MOD.SH.DM Functional building blocks summary

6.1.4 MOD.SH.MHMI- MULTICHANNEL HUMAN-AUTOMATION/ROBOT INTERACTION

"MOD.SH.MHMI- Multichannel Human-Automation/robot interaction" implements the multimodal, multichannel input/output mechanisms enabling the shop floor operator to interact with the automation mechanism (e.g robots) through gestures, voice, lights, sounds, etc. Depending on the situation a unique interaction/data source may not be enough as it can only provide partial information which may be unreliable (i.e. due to environmental conditions such as poor light or high noise.). Furthermore, the shop floor operators can have different profiles that make an interaction channel more suitable than another.

The "Multichannel Human-Automation/robot interaction" component enables both Human to Machine (H2M) and Machine to Human (M2H) interaction through multiple channels such as gestures, voice, lights, sounds, etc. It allows the system to deal with human variability as it can enhance the interaction experience and allow users with limited capabilities (e.g. sensorial limitation as limited vision or reduced hearing) to interact with the automation mechanisms in place. Furthermore, it could deal with context variability by adapting the interaction channel. H2M interaction is initiated by the shop floor operator. Shop floor operators can interact with the automation mechanisms through different interaction channels and even though several at the same time. Depending on the channel different data sources devices/sensors (e.g. MS KINECT for gestures capture, etc) or graphical user interfaces (GUI) can be involved.

Figure 22 shows the components diagram representing the decomposition of the MOD.SH.MHMI component into Functional Building Blocks (FBBs) and the relations and data flows between them. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB).

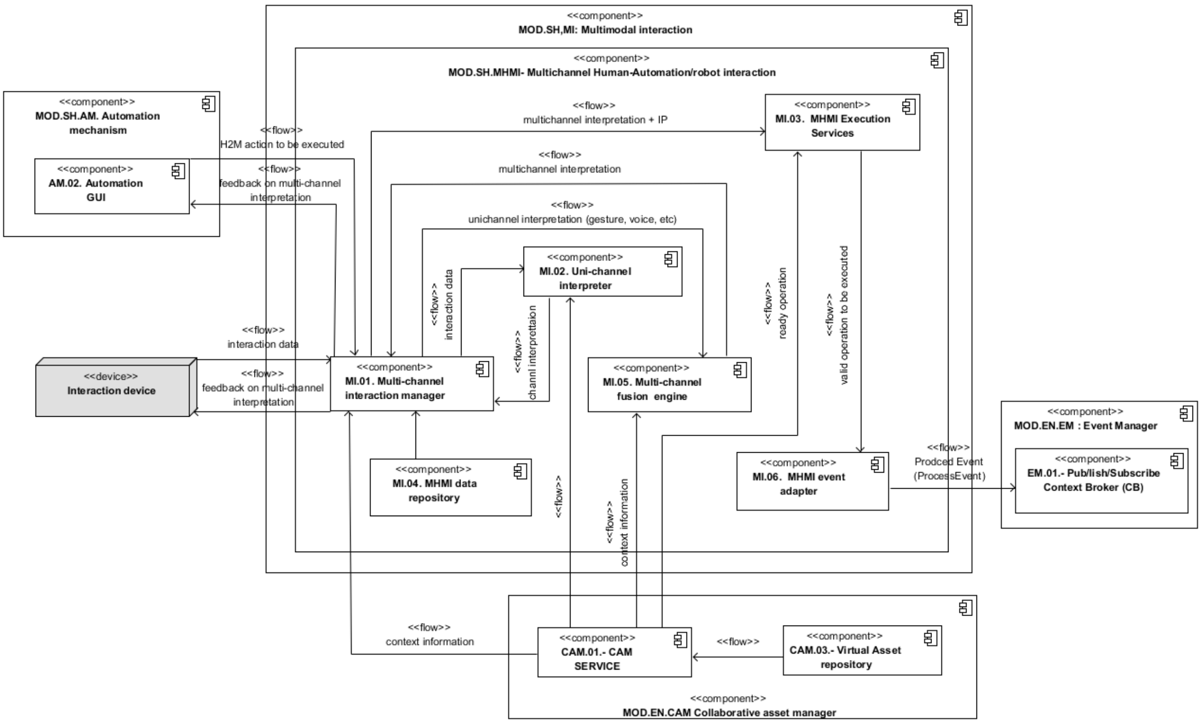


Figure 22 MOD.SHMHMI. High Level module decomposition

The "MI.01 Multi-channel interaction manager" is the sub component which orchestrates the rest of the components to perform the Human to Machine (H2M) and Machine to Human (M2H) interaction. It performs multi-channel interpretation, manages feedback, considers context information from the virtual asset repository ("CAM.03. Virtual asset repository") and, optionally, stores and retrieves information from the "MI.04. MHMI data repository" that supports data persistence for incremental learning interpretation readiness.

Specific "MI.02. Uni-channel interpreter" interprets the input data provided by the "MI.01 Multi-channel interaction manager" based on the semantic information stored in the virtual asset repository ("CAM.03. Virtual asset repository"). The "MOD.SH.MHMI- Multichannel Human-Automation/robot interaction" component includes as many uni channel interpreters as different channels are involved in the interaction.

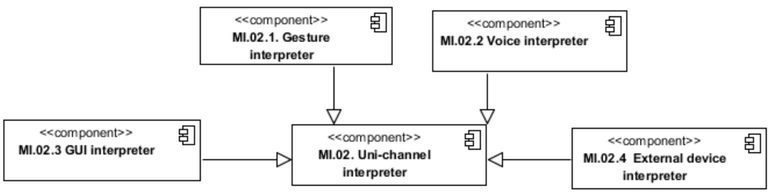


Figure 23 MOD.SH.MHMI. H2M interaction channels

The "MI.05. Multi-channel fusion engine" fuses uni-channel interpretations (i.e. partial) considering the semantic information, as well as relevant context information, stored in the "CAM.03. Virtual asset repository" and provides the complete multichannel interpretation. This interpretation is transformed by the "MI.03. MHMI execution services" into a valid command identifying both the action (e.g. EXECUTE, RESUME, STOP) and the operation to be performed (e.g. Assembly, etc.).

Finally, the "MI.06. MHMI event adapter" supports publish and subscribe capabilities and adapts the command identified by the "MI.03. MHMI execution services" to the event format supported by the Event Manager (EM) component.

Table 9 summarises the Functional Building Blocks (FBB) involved in MOD.SHMHMI identifying the type and scope. “MOD.SH.MHMI- Multichannel Human-Automation/robot interaction” component can work either in online (i.e. integrated in the A4BLUE framework) or offline (i.e. as a module independent of the A4BLUE framework) mode.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
MI.01. Multi-channel interaction manager					
MI.02. Uni-channel interpreter					
MI.03. MHMI execution services					
MI.04. MHMI data repository (opt)					
MI.05. Multi-channel fusion engine					
MI.06. MHMI event adapter					

Table 9 MOD.SH.MHMI Functional building blocks summary

6.1.5 MOD.SH.AS- ACTIVE SAFETY

MOD.SH.AS-Active safety implements active safety mechanisms to adapt the behaviour of the automation/robot considering the safety mode in place (i.e. safety rated stop mode or speed & separation monitoring mode) and the operator’s profile.

The “Active safety” component enables collision avoidance by adapting the behaviour (e.g. stopping motion, reducing speed) of the automation/robot based on human and environment perception (e.g. distance to an object, operator motion anticipation, intrusion in safety zones, etc.) as well as on operator’s context.

Figure 24 shows the components diagram representing the decomposition of the MOD.SH.AS component into Functional Building Blocks (FBBs) and the relations and data flows between them. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB) to send safety related events, if relevant.

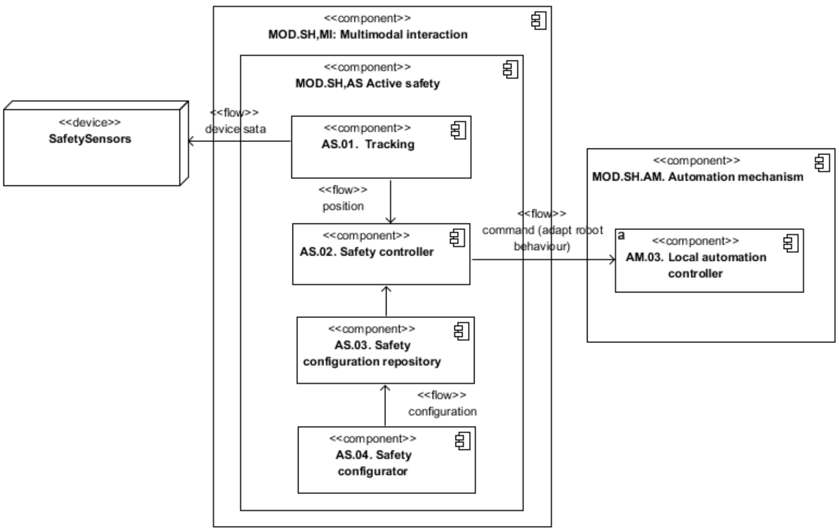


Figure 24 MOD.SH.AS. High Level module decomposition

The key sub component is the ""AS.02. Safety controller" which processes the tracking information resulting from "AS.01. Tracking"and decides how to adapt the robot behaviour considering the safety configuration stored in the "AS.03. Safety configuration repository" (e.g. safety modes, safety related actions, operator preferred modes, etc.). As safety is a critical aspect the "AS.02. Safety controller" interacts directly with the robot local control unit ("AM.03. Local automation controller").

"AS.01. Tracking" uses the data from sensors such as scanners, vision systems or proximity sensors to identify and track the position of target elements. Tracking can involve human and safety zone tracking ("AS.01.1 Human Tracking" and "AS.01.2. Zone tracking").

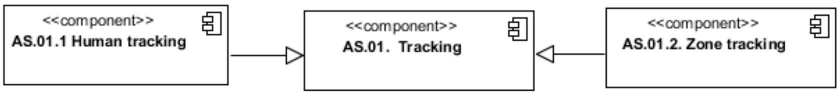


Figure 25 MOD.SH.AS. Tracking

"AS.04. Safety configurator" enables the system operator to define the safety settings.

Table 10 summarises the Functional Building Blocks (FBB) involved in MOD.SH.MI.AS identifying the type and scope.

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
AS.01. Tracking					
AS.02. Safety controller					
AS.03. Safety configuration repository					
AS.04. Safety configurator					

Table 10 MOD.SH.AS Functional building blocks summary

6.2 ENTERPRISE LAYER

6.2.1 MOD.EN.CAM- COLLABORATIVE ASSET MANAGER

In the context of the Manufacturing Industry, **Tangible Assets (TA)** are physical objects with economic value like machinery, tools, materials, workplace equipment, hardware, software, factory buildings, transportation vehicles, etc. In real-world scenarios, TAs are usually complemented by **Intangible Assets (IA)**, which are key business drivers whose essence is an idea or knowledge, and whose nature can be defined and recorded in some way – e.g., technical or scientific knowledge, worker skills and their level of adaptability, technological solutions, business relationships, etc.

Asset virtualization is the process by means of which Tangible and Intangible Assets (**T/IA**) in a Manufacturing Enterprise or Ecosystem are represented in digital format as **Virtualized Assets (VA)**. The individual steps of a typical asset virtualization process are the following:

- **Analysis** – The overall aim of T/IA virtualization is defined and documented. Individual real-world T/IA items are identified and analysed.
- **Formalization** – For each T/IA item identified in the previous step, a proper *asset model* is chosen from a domain-specific *reference ontology*. Attributes and relationships defined by the model are populated with item-specific values, resulting in a formal description of the individual T/IA item.
- **Deployment** – Each formalized T/IA item is stored in a shared *knowledge base* as a VA entity.

- **Maintenance** – VAs in the knowledge base are continuously updated to reflect real-world changes.

The A4BLUE **Collaborative Asset Manager** will be the implementation of a virtualized assets representation model (VARM) knowledge base, in charge of supporting the virtualisation and representation of Tangible Assets (TA) and Intangible Assets (IA), producing the Virtual Asset Representation (VAR) that contains the result of the adoption of the VARM model into the A4BLUE case.

The A4BLUE MOD.EN.CAM component will expose (standard) interfaces for retrieval of the assets from the knowledge base, by M2M interactions (i.e. using specific APIs) or by H2M interactions (i.e. using a dedicated GUI).

The Figure 26 here below shows the main components of the MOD.EN.CAM, describing both main Building Blocks and information flows.

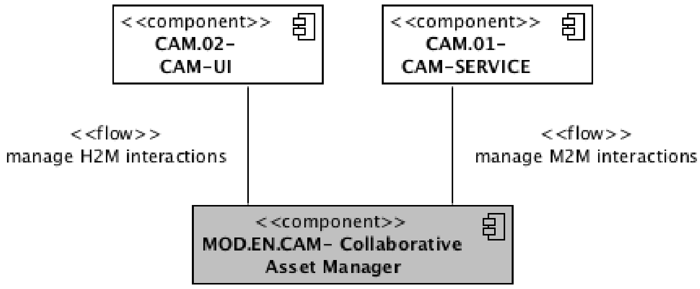


Figure 26 MOD.EN.CAM. High Level module decomposition

Table 11 summarises the Functional Building Blocks (FBB) involved in MOD.EN.CAM identifying the type and scope.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
CAM.01. CAM-SERVICE					
CAM.02. CAM-UI					

Table 11 MOD.EN.CAM Functional building blocks summary

6.2.2 MOD.EN.EM- EVENT MANAGER

Every A4BLUE-based system will be able to gather, publish, exchange, process and analyse massive data in a fast and efficient way. Events coming from the shop floor and the legacy systems (e.g. automation mechanisms, sensors and interaction devices through the Mediation Services and Device Manager) or from the other A4BLUE components must be managed in order to trigger the required adaptation or assistance actions. The Event Manager component integrated in the A4BLUE Platform will implement a Complex Event Processing in its core in order to deal with production and human variability events based on publish/subscribe services. Therefore, the Event Manager will be linked to the other assistance modules (such as AR/VR training, collaborative knowledge platform, DSS) that will be launched depending on the incoming event type.

In every data intensive scenario, you would need a component in the architecture able to mediate between entities (a physical thing or part of an application), such as data/event producers (e.g. sensors or IT systems) and the consumer applications (e.g. a smartphone application or AR tools).

In the A4BLUE context (and not only) *Events* refer to something that has happened, or is contemplated as having happened. Changes in context information can be considered as events as well. *Events Producers* can publish information in order to make them available to other entities, referred as *Events Consumers*, which are interested in processing the published information. Applications or even other components may play the role of Producers, Consumers or both. Events typically lead to creation of some data or context element describing or representing the events, thus allowing them to be processed. The creation and sending of the context element is an event, i.e., what has occurred. Since the data/context elements that are generated linked to an event are the way events get visible in a computing system, it is common to refer to these data/context elements simply as "events".

The events generated all around the system and conveyed through the MOD.EN.EM can also feed a Complex Event Processing (CEP) engine, intended to support the development, deployment, and maintenance of applications. CEP analyses event data in real-time, generates immediate insight and enables instant response to changing conditions.

A fundamental principle supported by the A4BLUE Event Manager is that of achieving a total decoupling between events Producers and Consumers. On one hand, this means that Producers publish data without knowing which, where and when Consumers will consume published data; therefore, they do not need to be connected to them. On the other hand, Consumers consume information of their interest, without this meaning they know which Producer has published a particular event: they are just interested in the event itself but not in who generated it. As a result, the EM is an excellent bridge enabling external applications to manage events related to the shop-floor in a simpler way hiding the complexity of gathering measures from on-field resources (sensors) that might be distributed or involving access through multiple low-level communication protocols.

Event Management is the process through which data/information/event are distributed and shared between components of a complex system. For efficient management, several communication schemes can be selected, with respect to the decoupling they provide such as:

- *Space Decoupling*: The interacting parties do not need to know each other. The providers publish information through an event/information service and the consumers receive information indirectly through that service. The providers and consumers do not usually hold references to each other and neither do they know how many consumers/providers are participating in the interaction.
- *Time Decoupling*: The interacting parties do not need to be actively participating in the interaction at the same time i.e., the publisher might publish some information while the subscriber is disconnected and the subscriber might get notified about the availability of some information while the original publisher is disconnected.
- *Synchronization Decoupling*: Publishers are not blocked while producing information, and subscribers can get asynchronously notified (through call-backs) of the availability of information while performing some concurrent activity i.e. the publishing and consumption of information does not happen in the main flow of control of the interacting parties.

This decoupling is important to cater for because decoupling of production and consumption of information increases scalability by removing all explicit dependencies between the interacting components. Removing these dependencies strongly reduces coordination requirements between the different entities and makes the resulting communication infrastructure well adapted to distributed environments.

The A4BLUE MOD.EN.EM component will expose (standard) interfaces for retrieval of the events and other data from the Producers to the Consumers. The consumer does not need to know where the data are located and what is the native protocol for their retrieval. It will just communicate to the EM through a well-defined interface specifying the data it needed in a defined way: on request or on subscription basis. The MOD.EN.EM will provide the data back to the consumer when queried, in case of "on-request", or when available, in case of "on-subscription" communication mode.

Furthermore, “EM.10. Status Manager” is in charge of updating the process status and keep historical records of all the events in the “CAM 03. Asset repository”.

The Figure 27 here below shows the main components of the MOD.EN.EM, describing both main Building Blocks and information flows.

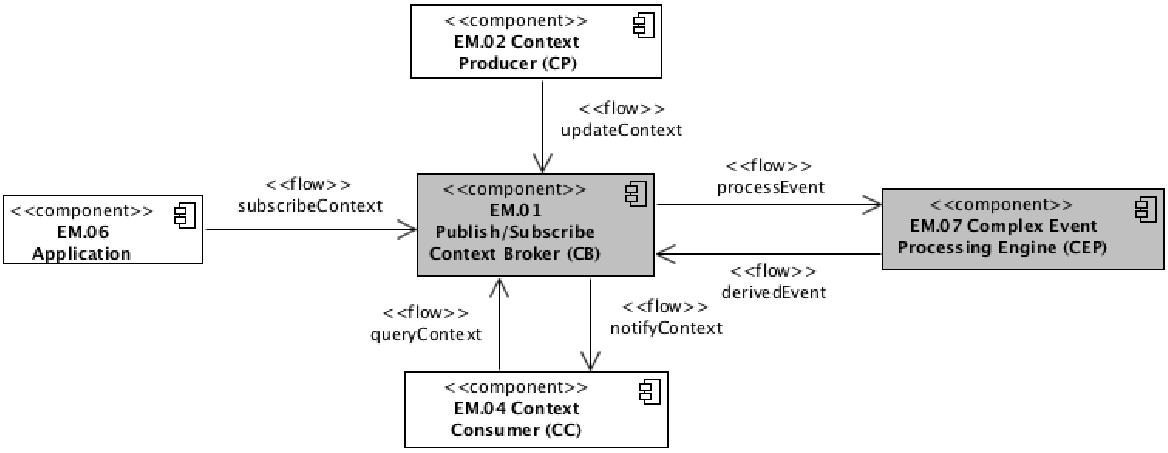


Figure 27 MOD.EN.EM. High Level module decomposition

Table 12 summarises the main Functional Building Blocks (FBB) involved in MOD.EN.EM identifying the type and scope.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
EM.01. Publish/Subscribe Context Broker (CB)					
EM.02. Context Producer (CP)					
EM.04 Context Consumer (CC)					
EM.06 Applications					
EM.07 Complex Event Processing Engine (CEP)					

Table 12 MOD.EN.EM Functional building blocks summary

6.2.3 MOD.EN.ARG- VR/AR BASED TRAINING AND GUIDANCE

The training and guidance system will be composed by two separate modules: a back-end server application and a front-end Augmented Reality player.

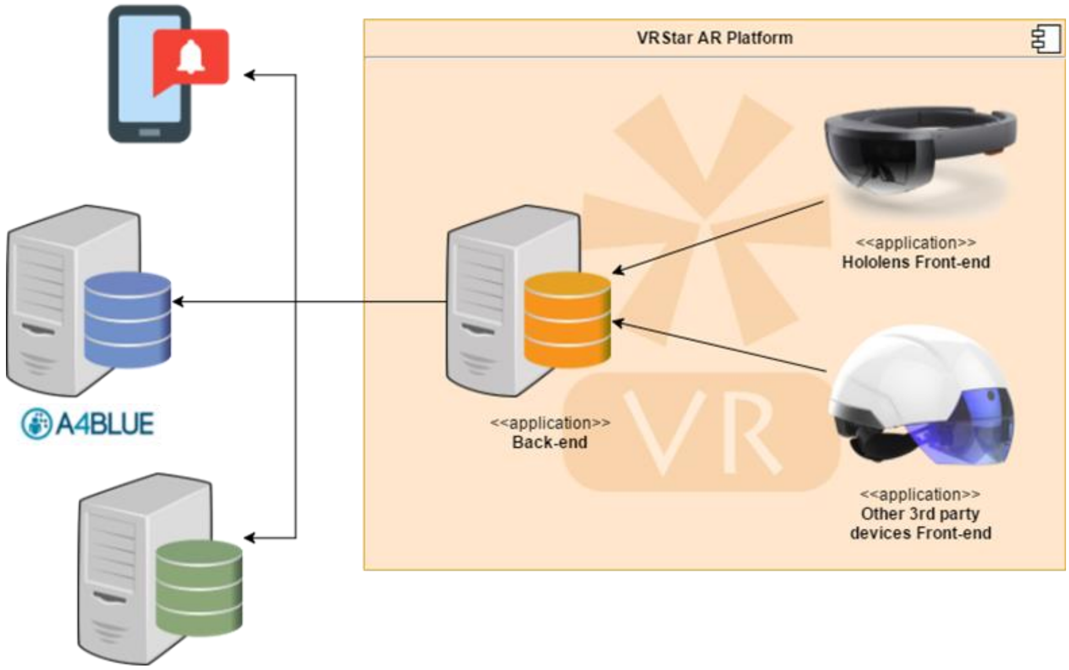


Figure 28 MOD.EN.ARG. The VRStar AR Platform and its modules

BACK-END APPLICATION

The **back-end** server application will run on a dedicated machine and it will act as a bridge between the existing A4BLUE framework and the AR-devices. Once set, the back-end application will be able to communicate with the AR device in order to send/receive data in real-time during the training session. The Event Manager and the Collaborative Asset Manager will be used as main interfaces to access A4BLUE framework resources and data. This link will enable the possibility to:

- Dynamically download 3D models and assets depending on a particular context;
- Retrieve daily tasks, assembly sheets and any other instruction for the operators;
- Save statistics about training results;
- Communicate with legacy systems through the A4BLUE framework;
- Update the subscribed workers about the job operations status through mobile notifications.

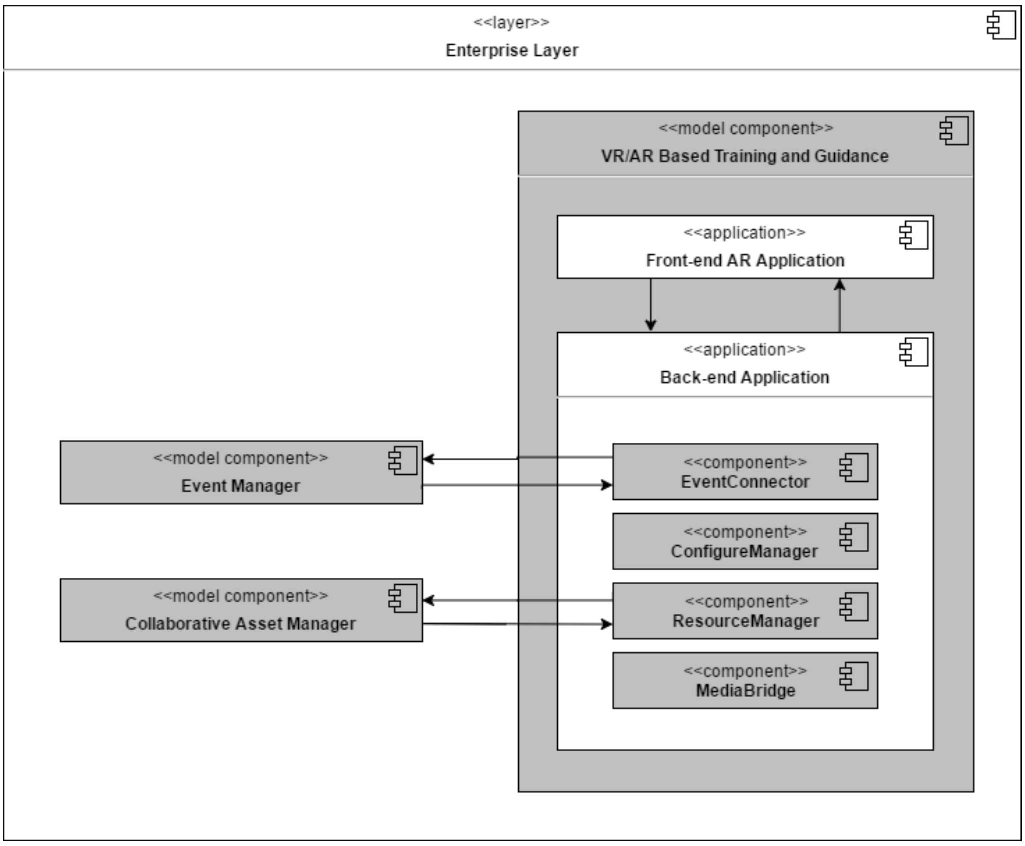


Figure 29 MOD.EN.ARG. Back-end Application Components Diagram

FRONT-END APPLICATION

The **front-end** will consists in a native application installed on the Augmented Reality HMD device that will receive data and asset from the back-end and render a 3D scene accordingly. The player will also be able to capture events such as gesture and voice commands and send them back to the server application in order to update its status in real-time and propagate this information to the A4BLUE EM.

The player will be composed by a set of components specifically designed for the management of a real-time holographic scene:

- **AnchorManager**. It retrieves, clears and stores anchors locally on the device, linking and updating an associated real transformation matrix on the scene graph.
- **MarkerManager**. It recognizes an AR marker in the scene, triggers an event when the marker is on line of sight, recognizes if the marker is a new one or already registered.
- **CursorManager**. It handles the cursor type and position in the scene. The cursor can be visualized in different forms for different interaction states.
- **GazeManager**. This component can raycast the 3D scene where the user is looking at and trigger an event on interactive object hit.
- **GestureManager**. This is an adapter that interpret and translate a subset of gestures as input events.
- **VoiceCommandManager**. This is an adapter that interpret and translate a subset of voice commands in input events
- **InputManager**. It collects all input events (gaze, gesture, voice commands) and raise events/callbacks.

- **ConfigManager**. It stores and retrieves (locally or remote) the whole client configuration. It triggers events for further load and restore context and interfaces.
- **InteractionManager**. It guides the camera using the head tracking, manage the animation timing of interactive objects, translate input events in possible commands & actions.
- **SceneManager**. It manages the consistency check, loading and unloading of all 3D objects in the scene.
- **FileManager**. It implements an abstraction layer for file system access. The virtual file system must support folders with different layer dept. It implements a cache system based on md5 hash comparison

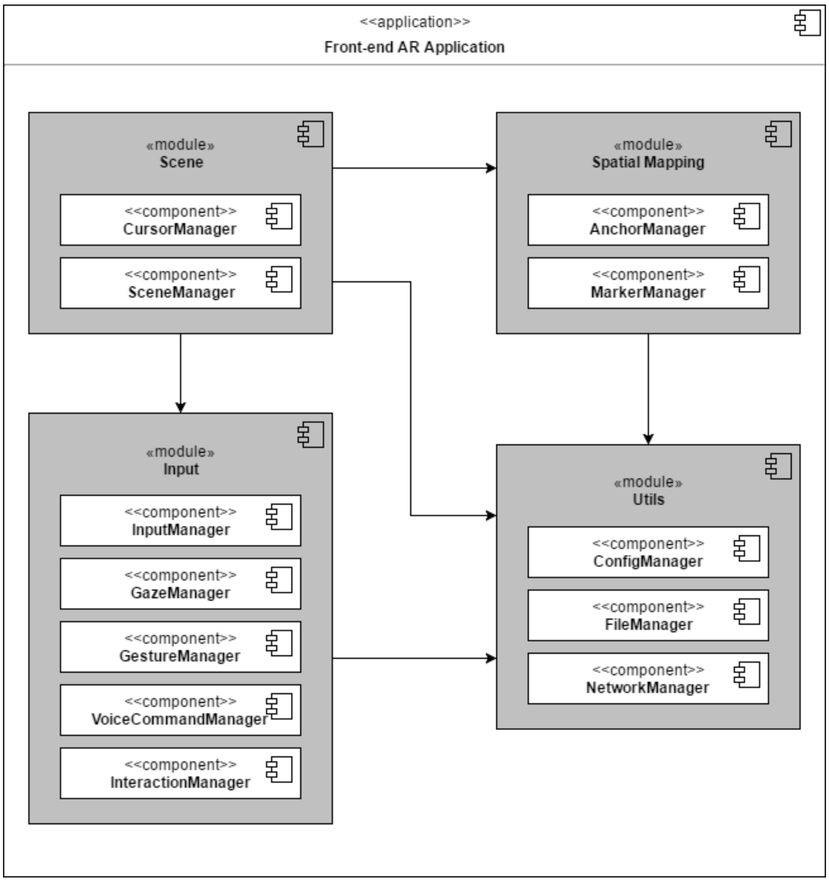


Figure 30 MOD.EN.ARG. Front-end AR Application Components Diagram

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data storage	Offline mode	Online mode
ARG.01.- Back-end					
ARG.02.- Front-end					

Table 13 MOD.EN.ARG Functional Building Blocks summary

6.3 BUSINESS LAYER

6.3.1 MOD.BU.KM- COLLABORATIVE KNOWLEDGE PLATFORM

The MOD.BU.KM module will support the definition of paradigms and methodologies aimed at fostering the cooperation and the human-oriented management of information and knowledge, making use of specific online tools for collective intelligence and cooperative open innovation. The cornerstone of such an approach will be the development of the Collaborative Knowledge Platform for Manufacturing, taking advantages from using a cloud-based management of knowledge, and controlling the huge amount of data originating from the factory floor up to the business level as a result of the increased collaboration itself. The platform will provide social networking tools and services, specifically conceived for the innovation solutions developed across the project.

The MOD.BU.KM Collaborative Platform will be built upon OPENNESS (OPEN Networked Enterprise Social Software)[10], the results of a running private research project under development at ENGINEERING R&D Department: a platform fully developed using open source technologies, and leveraging on relevant results from research fields such as Open Innovation, Collective Intelligence, Enterprise Social Software. OPENNESS integrates techniques and technologies for the concurrent extraction, derivation and determination of new knowledge, using search, clustering, data mining, automatic reasoning, and, will include recommendation systems and opinion mining techniques, etc.

The platform will be built upon three main pillars:

- **Engagement** is in a closer community of workers using the online tools
- **Intelligence** is in the people and in the decisions they make
- **Knowledge** is in the people and in their knowledge artefacts, organized and managed more efficiently

The MOD.BU.KM Collaborative platform will **support both blue and white collar workers** by reacting to sharing experiences and historical knowledge upon manufacturing processes and operations, by providing **different types of interfaces** will address the needs of different work groups.

By more readily sharing documents and quickly locating experts to answer questions, workers can complete in minutes work that had previously taken more than one week. Collaboration technologies can also provide guidance about how particular processes can be optimally performed to improve quality and increase productivity. Team members can then use the platform to hand over work, conduct discussions, share updates, review checklists and obtain approvals. The platform’s tagging, searching and messaging features make it easier to ask for help, locate documents that can be re-used and share best practices.

The Figure 31 here below shows the main components of the MOD.BU.KM, describing both main Building Blocks and information flows.

[10] http://openness.eng.it/

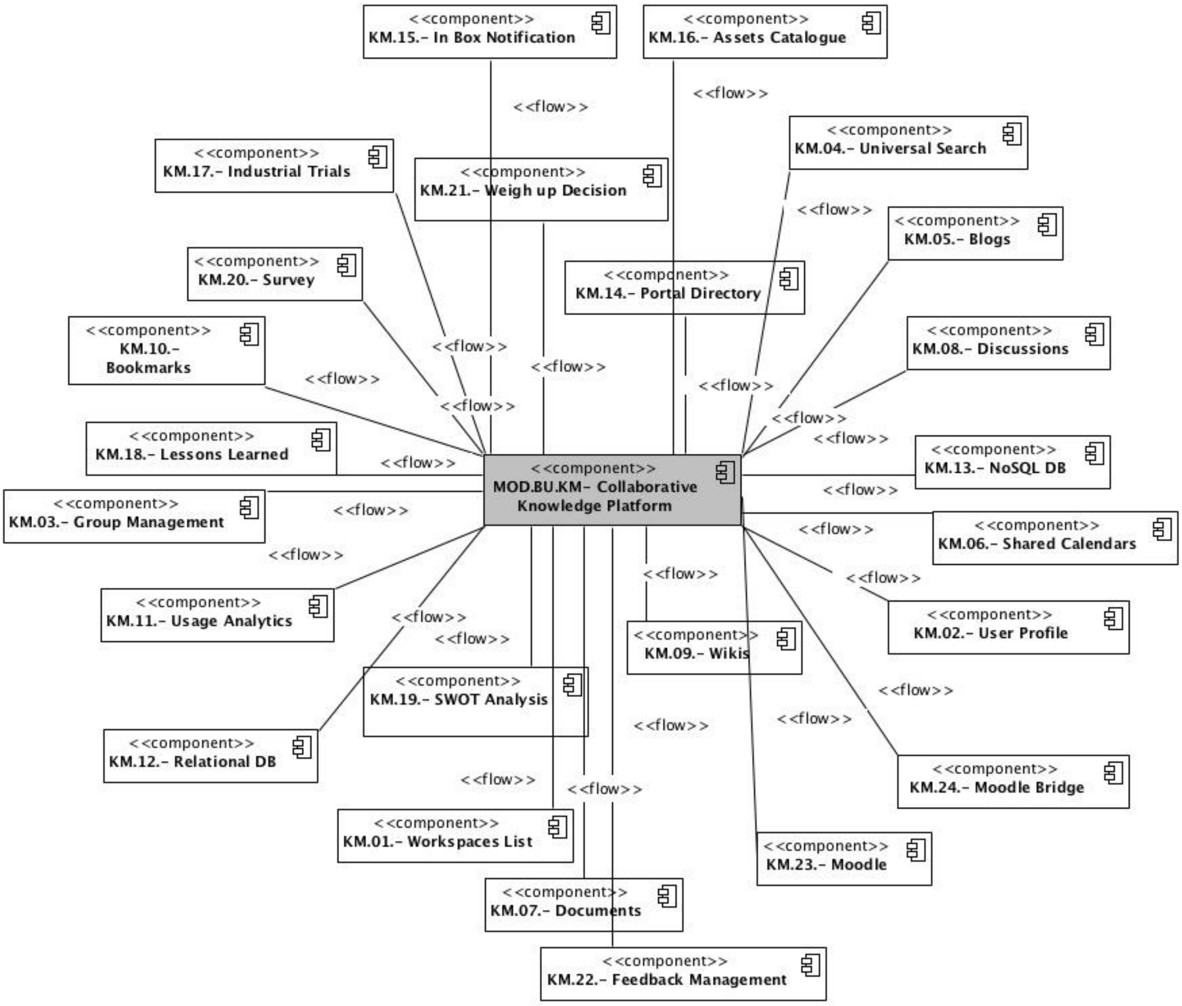


Figure 31 MOD.BU.KM. High Level module decomposition

Table 14 summarises the Functional Building Blocks(FBB) involved in MOD.BU.KM identifying the type and scope.

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data storage	Offline mode	Online mode
MOD.BU.KM- Collaborative Knowledge Platform					
KM.01.- Workspaces List					
KM.02.- User Profile					
KM.03.- Group Management					
KM.04.- Universal Search					
KM.05.- Blogs					
KM.06.- Shared Calendar					
KM.07.- Documents					
KM.08.- Discussions					

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data storage	Offline mode	Online mode
KM.09.- Wikis					
KM.10.- Bookmarks					
KM.11.- Usage Analytics					
KM.12.- Relational DB					
KM.13.- NoSQL DB					
KM.14.- Portal Directory					
KM.15.- In Box Notification					
KM.16.- Assets Catalogue					
KM.17.- Industrial Trials					
KM.18.- Lessons Learned					
KM.19.- SWOT Analysis					
KM.20.- Survey					
KM.21.- Weigh up Decision					
KM.22.- Feedback Management					
KM.23.- Moodle					
KM.24.- Moodle Bridge					

Table 14 MOD.BU.KM Functional Building Blocks summary

6.3.2 MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)

The "MOD.BU.DSS- Decision Support System" component aims to support workers on relevant decisions for the assembly, maintenance, inspection operations. It aggregates relevant information produced in the domain of the A4BLUE system and provides visual analytics capabilities to support workers in the decision-making process. Furthermore, it supports the management of multichannel notifications to notify intervention requests (e.g. maintenance, assembly collaboration, inspection, etc.).

Figure 32 shows the component's diagram representing the decomposition of the MOD.BU.DSS component into Functional Building Blocks (FBBs) and the relations and data flows between them. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB).

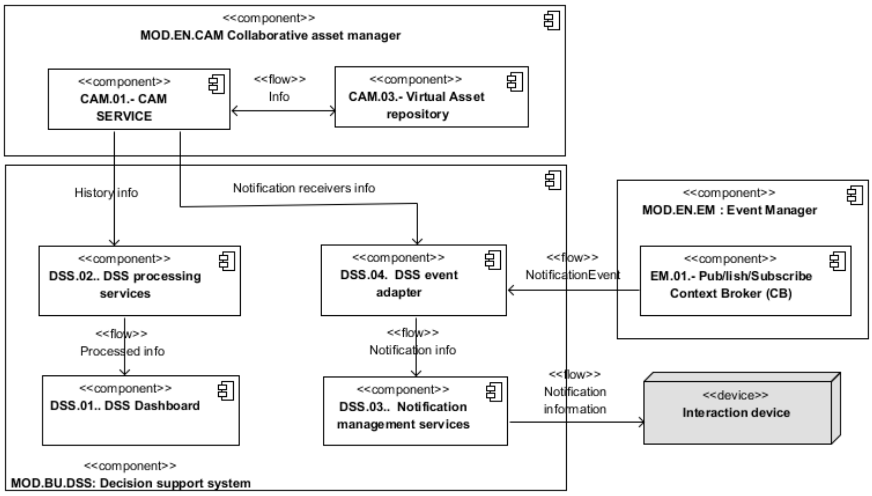


Figure 32 MOD.BU.DSS. High Level module decomposition-

Decision support

The business logic decision support features are implemented by the “DSS.02.- DSS processing services” component which retrieves and processes the required historical information (e.g. stored in the “CAM.03. Virtual asset repository” or accessed via remote service APIs). The information processed by “DSS.02.- DSS processing services” is stored in a data repository “DSS.05.- DSS data repository” (able to provide also a built-in cache mechanism to improve performances) and it is displayed through the “DSS.02.- DSS Dashboard”, which provides the graphical user interface allowing users to access to relevant information to support decision making.

Notifications management

The Event Manager (EM) component produces notification events related to operational activity to request specific interventions (e.g. collaboration request, maintenance or inspection intervention request, etc.). The “DSS.04. DSS event adapter” is in charge of adapting the notification events produced by the Event Manager (EM) and identifying the notification receivers (if required) as well as the appropriate notification channels based on the context information represented in the “CAM.03. Virtual asset repository”. “DSS.03. DSS notification management services” implements the business logic to notify the target user by using the appropiate notification channel (e.g. graphical user interface, push notification, email, sound, light indicator, etc.). In the case of pop up notifications a specific Notifier App (DSS-05. Notifier App) should be required.

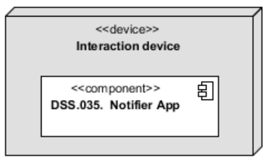


Figure 33 MOD.BU.DSS. Pop up notification channel

Table 15 summarises the Functional Building Blocks (FBB) involved in MOD.SH.DSS identifying the type and scope.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
DSS.01. DSS Dashboard					
DSS.02. DSS processing services					
DSS.03. DSS notification management services					
DSS.04. DSS event adapter					
DSS.05. DSS Notifier App					

Table 15 MOD.BU.DSS Functional Building Blocks summary

6.3.3 MOD.BU.MON- MONITORING

"MOD.BU.MON- Monitoring" is aimed to support the collection of key performance indicators (KPIs) produced in the domain of the A4BLUE solution (i.e. other KPIs are out of the scope of this module) to support the assessment of the impact of the introduction of the A4BLUE solution during the experimentation and evaluation phases.

The monitoring module (MOD.BU.MON) shows high level information using abstraction such as Key Performance Indicators (KPIs). These indicators are defined in the deliverable D2.4 – "Economic and technical assessment framework - Initial Version". The A4BLUE framework will provide a user-friendly graphical interface to allow such analysis and support whit-collar operators in both strategical monitoring and decision making process.

From the analysis of the information provided in the framework of D2.4, there are two kind of indicators that are be collected in the domain of A4BLUE: (1) indicators based on process history information and (2) indicators based on surveys (i.e. assessment questionnaires):

1) Indicators based on process history information: the indicators are calculated and displayed by taking advantage of the MOD.BU.DSS component described in Section 6.3.2.
2) Indicators based on surveys: the indicators are calculated and displayed by taking advantage of the MOD.BU.CQMS component described in Section 6.3.5.

6.3.4 MOD.BU.ACE- AUTOMATION CONFIGURATION EVALUATION

The automation configuration evaluation module (MOD.BU.ACE) is aimed to help the production planner in defining the optimal level of automation for the production processes. Figure 34 displays the components diagram that shows the decomposition of the component into Functional Building Blocks (FBBs) and also the relations and flows among them.

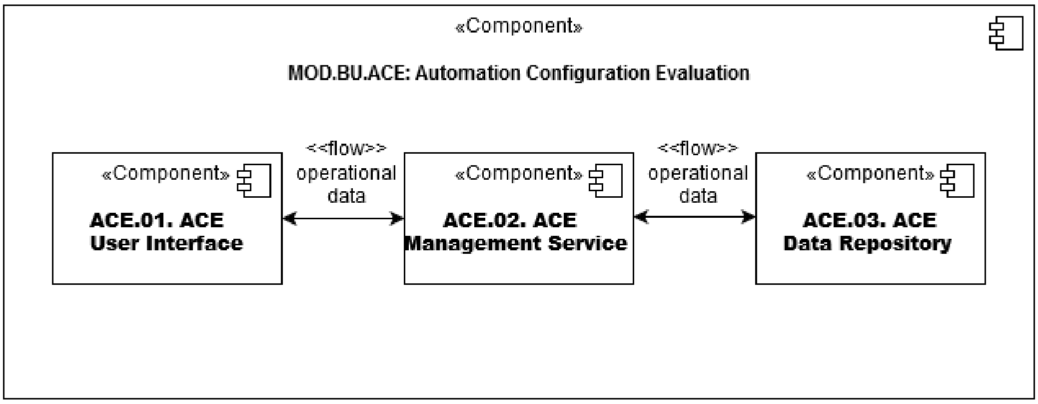


Figure 34 MOD.BU.ACE. High level module decomposition

The "MOD.BU.ACE: Automation Configuration Evaluation" component is bases upon three sub-components. The user will interact with the tool by using a graphical, web based, user interface "ACE.01. ACE User Interface". The database "ACE.03. ACE Data Repository" consists of information inputed by the user, through the user interface, and persisted for future use.

The "ACE.02. ACE Management Service" consists of the business logic of the software. It interacts with the "ACE.03. ACE Data Repository" by storing and retrieving the required information.

FBB	Type			Scope	
	Graphical User Interface	Business Logic	Data Repository	Offline Mode	Online Mode
ACE.01. ACE User Interface					
ACE.02. ACE Management Service					
ACE.03. ACE Data Repository					

Table 16 MON.BU.ACE Functional Building Blocks specification

6.3.5 MOD.BU.CQMS- COMPUTER BASED TOOL FOR THE MEASUREMENT OF WORKER SATISFACTION

"MOD.BU.CQMS- Computer-based tool for Quantitative Measurement of Satisfaction" component enables workers to complete questionnaires which assess levels of worker satisfaction in relation to human-automation systems and wider work environment characteristics. Futhermore, to allow (i.e. as described in Section 6.3.3) the assessment of additional key performance indicators (e.g. trust, usability, etc.) and alternative measures of satisfaction (to prevent repetitive questioning which will reduce data reliability) it should be generic enough to support the introduction of new questionnaires and individual questions. MOD.BU.CQMS is expected to involve both online and offline working modes**.** In the offline working mode it is used as a standalone tool and episodic assessment of worker satisfaction is supported by enabling the questionnaires to be scheduled on demand or on a periodic basis. In the online mode: it can interact with the Event Manager (EM) to support on-event assessment of worker satisfaction and automatic management of the users and roles.

Figure 35 shows the components diagram representing the decomposition of the MOD.BU.CQMS component into Functional Building Blocks (FBBs) and the relations and data flows between them. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB).

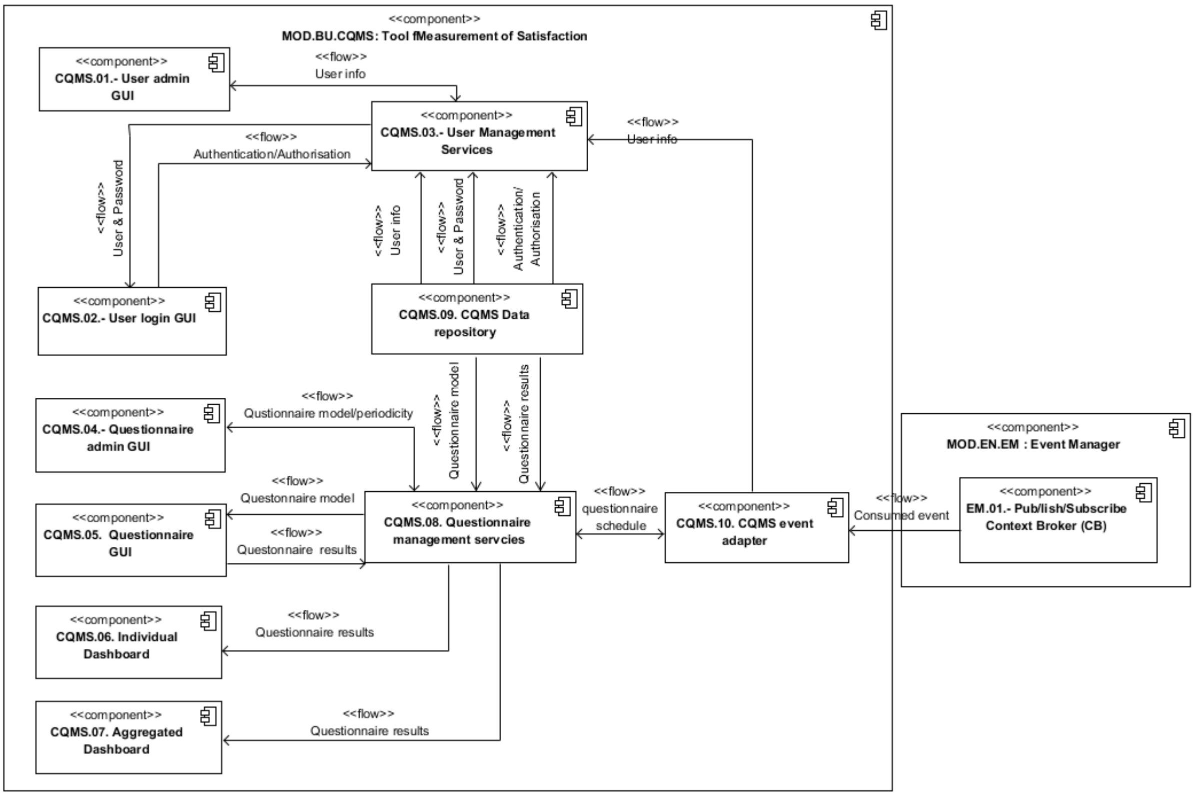


Figure 35 MOD.SH.CQMS. High Level module decomposition

In the online module users and access rights are created on an event basis while in the offline mode system administrators could access management features to manage both users and access rights (i.e. “CQMS.01.- User admin GUI”). System administrators could opt to define new questionnaires through “CQMS.04.- Questionnaire admin GUI”. All the users accessing the MOD.BU.CQMS should be authenticated and authorised (i.e. “CQMS.02.- Login GUI”). Workers can access the list of assessment questionnaires and complete them through the “Questionnaire GUI” (CQMS.05) and can access the history of the questionnaires they have completed through their “Individual dashboard” (CQMS.06). Supervisors/analysts can access aggregated questionnaire results (“CQMS.07.- Aggregated dashboard”) and export results in csv format for further analysis.

The business logic supporting user and questionnaire management features is implemented in “CQMS.03.- User management services” and “CQMS.08.- Questionnaire management services” respectively. Furthermore, these components interact with the “CQMS data repository” (i.e. CQMS.09) by storing and retrieving the required information.

To support the online working mode the “CQMS.10. CQMS event adapter” enables publish and subscribe capabilities and is in charge of adapting the events produced by the “CQMS.08.- Questionnaire management services” to the event format supported by the Event Manager (EM) component.

Table 17 summarises the Functional Building Blocks (FBB) involved in MOD.SH.CQMS identifying the type and scope.

FBB	Type (SW)			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
CQMS.01.- User admin GUI					
CQMS.02.- Login GUI					
CQMS.04.- User management services					
CQMS.04.- Questionnaire admin GUI					
CQMS.05.- Questionnaire GUI					
CQMS.06.- Individual dashboard GUI					
CQMS.07.- Aggregated dashboard GUI					
CQMS.08.- Questionnaire management services					
CQMS 09. CQMS data repository					
CQMS 10. CQMS event adapter					

Table 17 MOD.SH.CQMS Functional building blocks summary

7 BUSINESS PROCESSES

A *process* is a grouping of tasks that form an executable unit, able to realize a complex system behaviour through separate threads of control. The business processes can be described at several levels of abstraction, each level addressing different concerns. At the highest level, the process architecture can be viewed as a set of independently executing logical networks of communications, distributed across a set of resources and components, constituting the overall system.

7.1 SHOPFLOOR LAYER

7.1.1 MOD.SH.MS- MEDIATION SERVICES

The following sequence diagrams provide details of the interactions among sub-components to support "FTR.04- Integration with the enterprise legacy system" as described in the specifications section. A sequence diagram is provided for each involved use case scenario.

Update process information

The shop floor operator updates presence and process information through the legacy system graphical user interface.

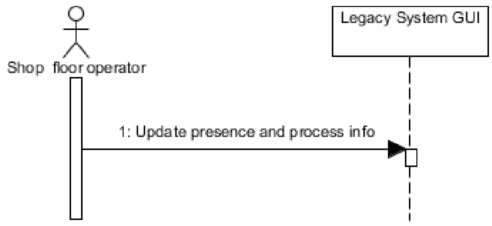


Figure 36 MOD.SH.MS. Update process information

Share information

The information collected though the legacy system graphical user interface is sent to the "MS.01. Mediation management services" component. To support it a specific legacy system dependent component can be necessary (i.e." LS.01 Legacy mediation agent").

"MS.02. MS event adapter" supports publish and subscribe capabilities and adapts the process data collected through the "MS.01. Mediation management services" to the event format supported by the Event Manager (EM) component. It also transforms events coming from the EM Event Manager (EM) component into process data to update the legacy system (e.g. information collected during the execution of the operations performed by the automation mechanisms). Furthermore, it transforms events coming from the EM Event Manager (EM) component into process data to update the legacy system (e.g. information collected during the execution of the operations performed by the automation mechanisms).

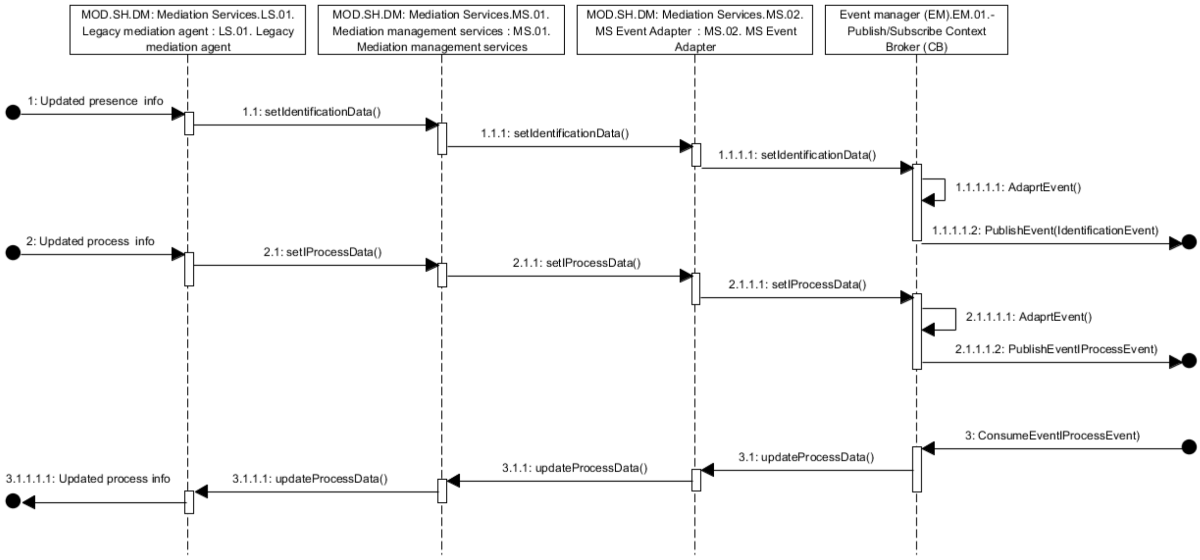


Figure 37 MOD.SH.MS. Share information Legacy Systen ←→ ABLUE

7.1.2 MOD.SH.AM- AUTOMATION MECHANISMS

The following sequence diagrams provides details of the interactions among sub-components to support “FTR.05- Plug and Produce adaptive automation mechanisms” as described in the specifications section. A sequence diagram is provided for each involved use case scenario.

Discovery process: Plug/Un-plug automation mechanism

When the automation mechanism is plugged/un-plugged “AM.05. OPC UA Server” registers the communication information (IP and Port) in the “DM.02 OPC UA discovery server.” and the automation system semantic representation (i.e. automation variables and methods) using the “MS.01. Semantic representation registrator”.

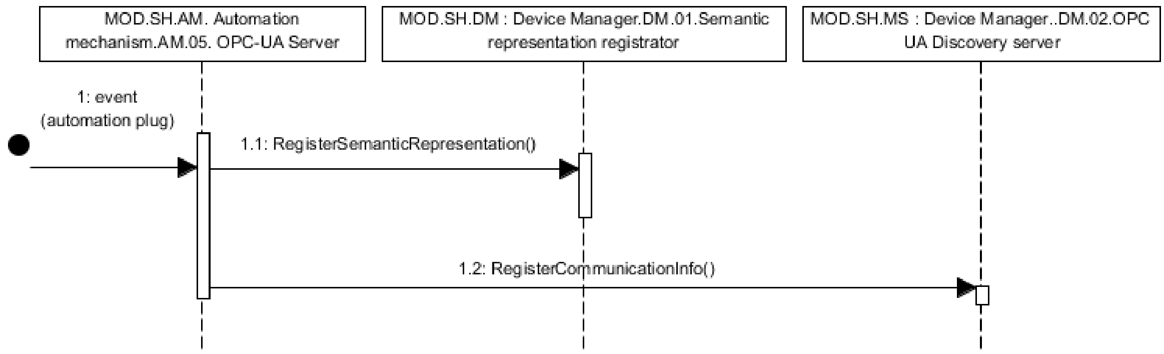


Figure 38 MOD.SH.AM. Plug/Un-plug automation mechanism

Share information (operation process)

The “AM.05. OPC-UA Server” checks “AM.01 Local automation controller” for updates in the subscribed automation variables and sends updates to the “DM.03. Device manager OPC UA client” subscribed to it.

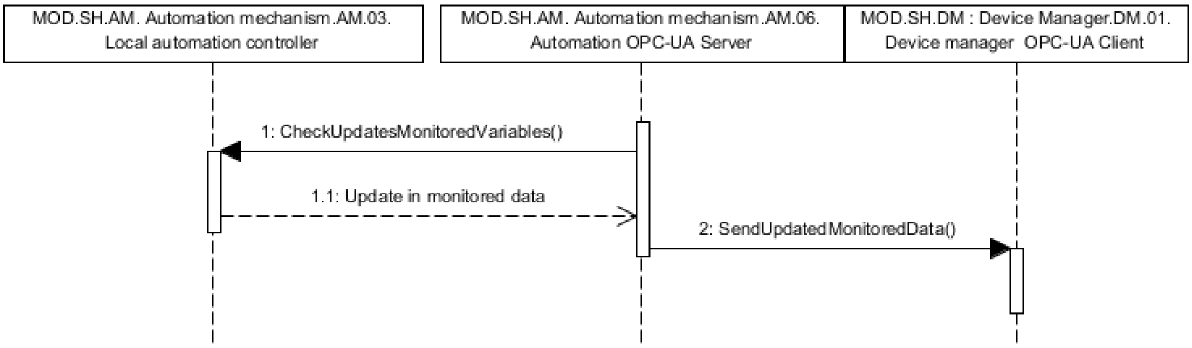


Figure 39 MOD.SH.AM. Share information with A4BLUE

7.1.3 MOD.SH.DM- DEVICE MANAGER

The following sequence diagrams provide details of the interactions among sub-components to support “FTR.05- Plug and Produce adaptive automation mechanisms” as described in the specifications section. A sequence diagram is provided for each involved use case scenario.

Discovery process: Plug/Un-plug automation mechanism

When the automation mechanism is plugged/un-plugged:

- “AM.05. OPC UA Server” registers the communication information (IP and Port) in the “DM.02 OPC UA discovery server.”
- “AM.05. OPC UA Server” registers the automation system semantic representation (i.e. automation variables and methods) described in a RDF file into the virtual asset repository (CAM.03) through the “CAM.01 CAM Services” and the identification of the data to be monitored and used by the “DM.01. Device manager OPC UA” in the “MS.01. Semantic representation registrator”. “DM.03. Device manager OPC UA client” gets the information of the available variables from the virtual asset repository (CAM.03) and subscribes to the data to be monitored.

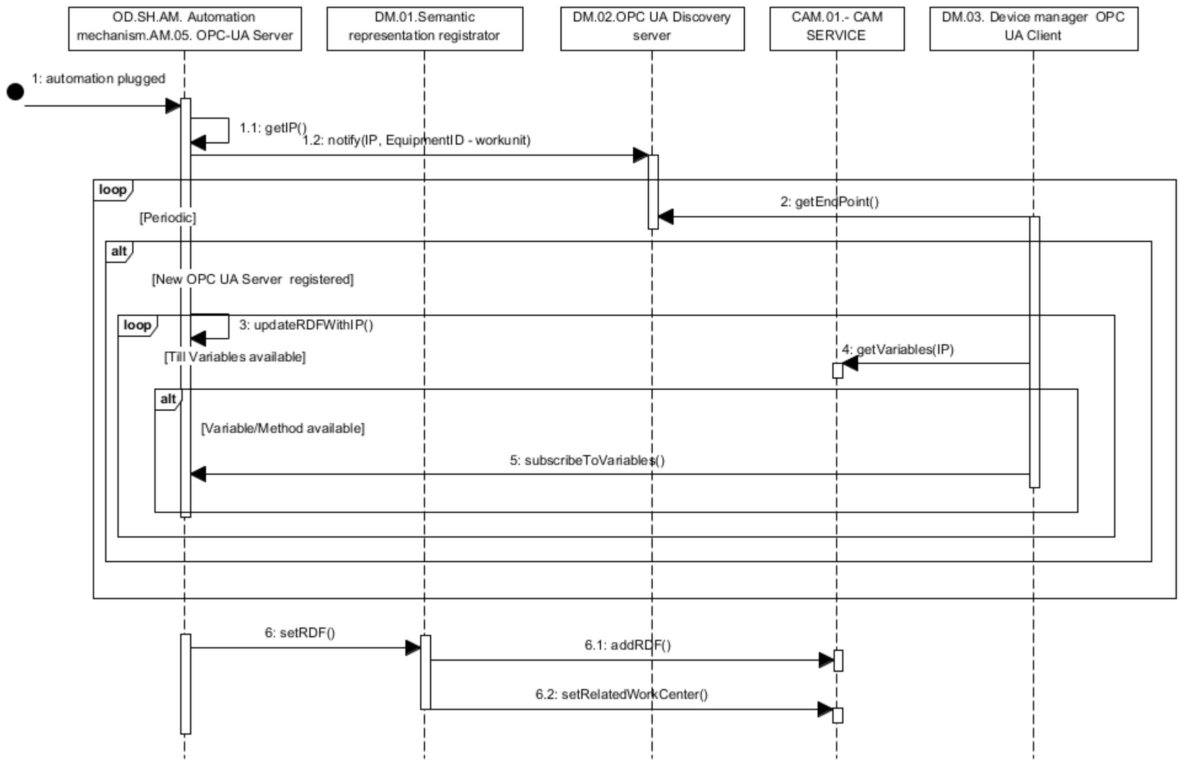


Figure 40 MOD.SH.DM. Plug/Un-plug automation mechanism

Share information (operation process)

Once a monitored piece of data is updated the “AM.05. OPC-UA Server” sends the update to the “DM.03. Device manager OPC UA client” that is in charge of publishing an event with the updated data. “DM.04. DM event adapter” adapts it to meet the event protocol supported by the Event Manager and publishs it into the Publish/Subscribe Context Broker (CB).

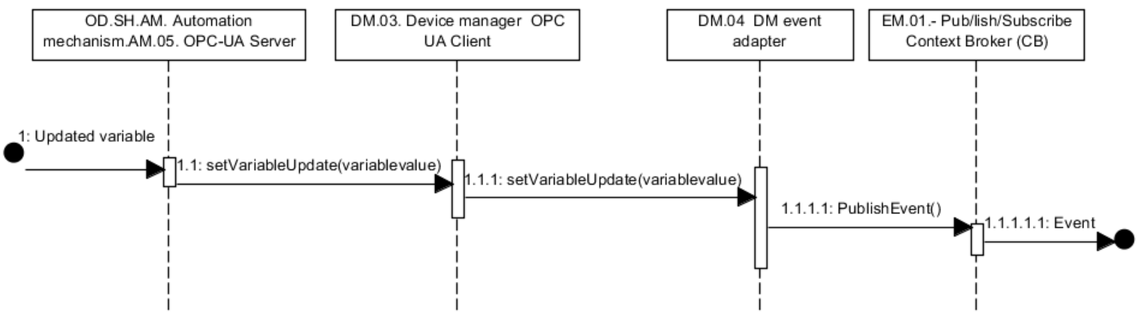


Figure 41 MOD.SH.DM. Share information with automation mechanism

Adapt behaviour (operation process)

Once the “DM.02. DM event adapter” receives an adaptation request to be executed by the automation from the “EM.01. Publish/Subscribe Context Broker (CB)” it adapts it, if necessary, to meet the event format supported by the “DM.03. Device manager OPC UA client” component and identifies the method provided by the automation that is linked to the action to be executed from the virtual asset repository (CAM.03) through the “CAM.01 CAM Services” as well as the endpoint of the "AM.05. OPC-UA Server". Then "DM.03. Device manager OPC UA client" send the method to be executed trough the "AM.05. OPC-UA Server".

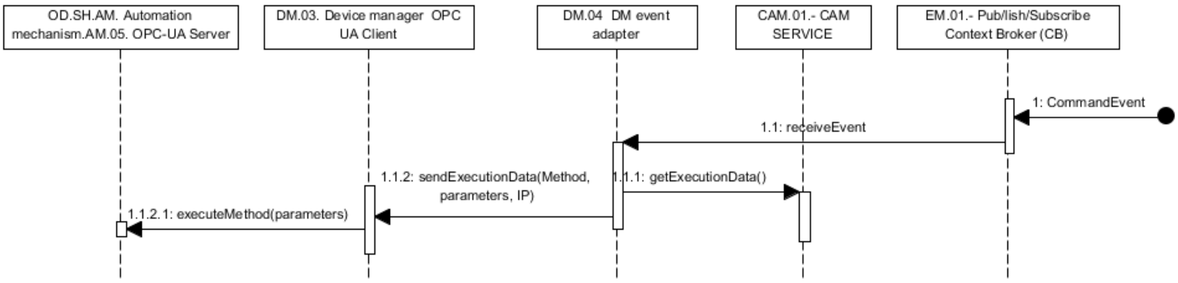


Figure 42 MOD.SH.DM. Adapt automation mechanism behaviour

7.1.4 MOD.SH.MI- MULTIMODAL HUMAN-AUTOMATION/ROBOT INTERACTIONS

The following sequence diagrams provide details of the interactions among sub-components to support FTR.02- Multimodal human- automation/robot Interaction and FTR.03- Safe Human - Automation/Robot co-existence as described in the specifications section. A sequence diagram is provided for each involved use case scenario.

Perform Human to Machine (H2M) interaction

Depending on the situation a unique interaction channel can provide the complete interaction information but, in some cases, it only provides partial information that needs to be fused to get a complete interpretation of the user interaction.

When the shop floor operator initiates the interaction the "MI.01 Multi-channel interaction manager" captures the inputs provided by the interaction sources and sends them to the appropriate interpreter ("MI.02. Uni-channel interpreter") which interprets them by using the semantic representation of the instruction and the context information (e.g. operator identification, environment status information, etc.) provided by the "Virtual Asset Repsository" (CAM.03.) though the "CAM.01. CAM Services" and sends back the channel interpretation (e.g. gesture interpretation, etc,) to the "MI.01 Multi-channel interaction manager". Then the "MI.01 Multi-channel interaction manager" stores the interpretation in the "MI.04. MHMI data repository" and sends the uni-channel interpretation, of each relevant channel, to the "MI.05. Multi-channel fusion engine". The fusion engine fuses partial interpretations and sends the multichannel interpretation back to the "MI.01 Multi-channel interaction manager" which requests feedback.

When a positive feedback is available the "MI.01 Multi-channel interaction manager" sends the multichannel interpretation along with the IP to the "MI.03. Execution services" which is in charge of identifiying the valid command (i.e. related to a process segment) to be executed. This command is sent to the "MI.06. MHMI event adapter" responsible of adapting it to the format of the process event supported by the Event Manager and published into the "EM.01. Publish/Subscribe Context Broker (CB)".

The feedback request is a Machine to Human (H2M) interaction and it is described in the Perform Machine to Human (M2H) interaction section.

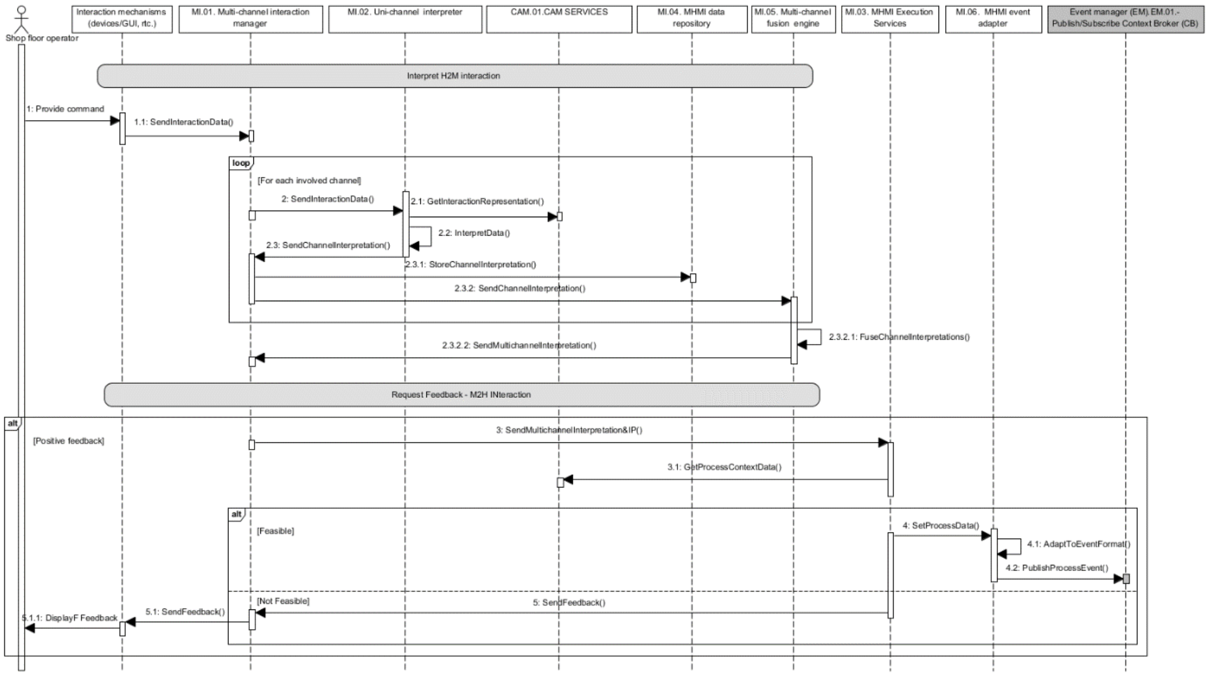


Figure 43 MOD.SH.MHMI. Perform H2M interaction

Perform Machine to Human (M2H) interaction

The figure below represents a simplified view of the feedback request loop. It can be an iterative process where the "MI.01 Multi-channel interaction manager" asks the shop floor operators to confirm that the obtained multichannel interpretation fits the operator's intention or to ask for further details though different interaction mechanisms such as a GUI, a voice message, etc. Then the shop floor operator can provide his/her feedback trough the different available interaction mechanisms. The iterative process ends when a positive feedback is obtained.

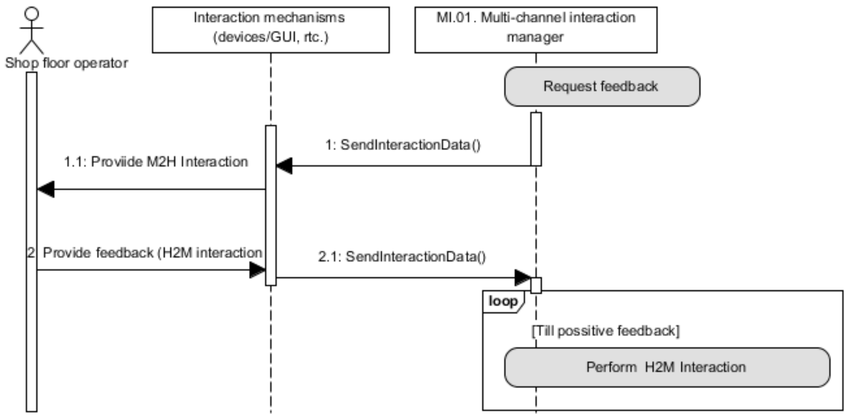


Figure 44 MOD.SH.MHMI. Perform M2H interaction- Request feedback

7.1.5 SAFE HUMAN - AUTOMATION/ROBOT CO-EXISTENCE

The following sequence diagrams provide details of the interactions among sub-components to support FTR.03- Safe Human - Automation/Robot co-existence as described in the specifications section. A sequence diagram is provided for each involved use case scenario.

Share workspace safely

The "AS.05. Safety configurator" allows system administrators to define the safety configuration. Safety devices send data to "AS.01. Human tracking" which processes it and sends to the "AS.03. Safety controller" the position of the tracked elements. The "AS.03. Safety controller" decides on automation behaviour adaptation based on this information and the configuration information (e.g. safety working mode, safety zones, context info, etc.) is stored in the "AS.04. Safety configuration repository". If a behaviour adaptation is needed the "AS.03. Safety controller" executes it directly through the "AM.03. Local automation controller".

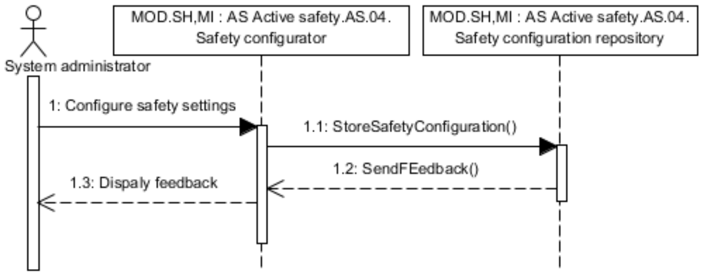


Figure 45 MOD.SH.AS. Share workspace safety configuration

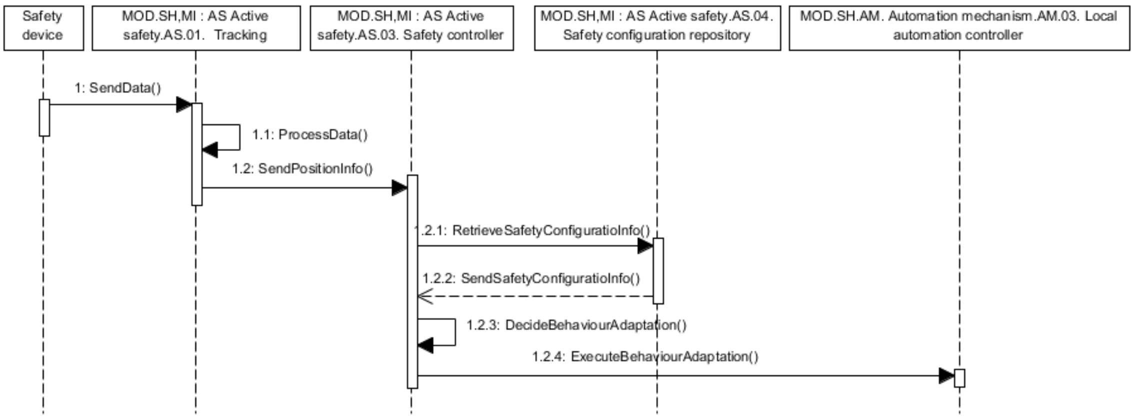


Figure 46 MOD.SH.AS. Share workspace safely operation

7.2 ENTERPRISE LAYER

7.2.1 MOD.EN.CAM- COLLABORATIVE ASSET MANAGER

Every A4BLUE-based system will be able to store, retrieve and manage information about tangible and intangible assets by using the Asset Repository capability provided by the MOD.EN.CAM component.

Both M2M and H2M interactions will be enabled by such a component, through the use of open APIs and user friendly GUIs in charge of enabling the maintenance operations on the repository itself.

The following Figure 47 shows the main business processes impacting on the A4BLUE MOD.EN.CAM component, while Figure 48 shows main interactions with the end users.

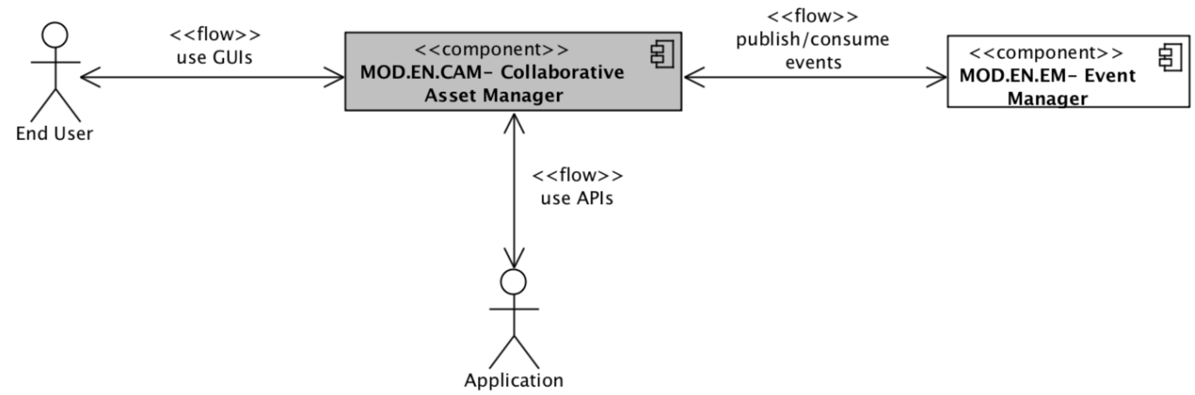


Figure 47 MOD.EN.CAM. Cross-components business processes

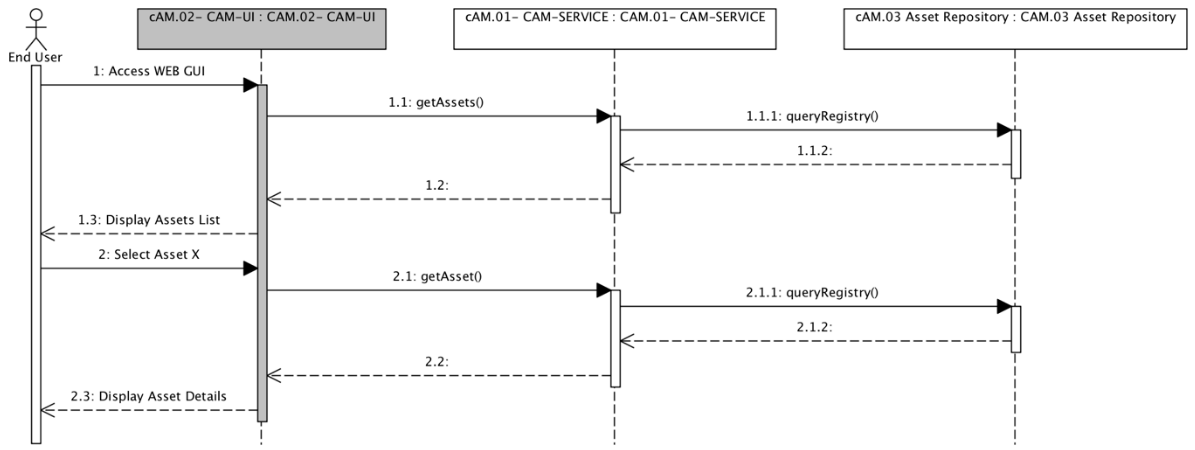


Figure 48 MOD.EN.CAM. Basic interactions and related entities

7.2.2 MOD.EN.EM- EVENT MANAGER

Every A4BLUE-based system will be able to gather, publish, exchange, process and analyse massive data in a fast and efficient way. Events will represent the information brick used to convey information among components using a decoupling approach.
Every A4BLUE component will be able to interact with the MOD.EN.EM, publishing new events or consuming events generated somewhere else in the system (e.g. from the shopfloor and the legacy systems through the Mediation Services and Device Manager, from the other assistance modules such as AR/VR training, collaborative knowledge, DSS).

The following Figure 49 shows the main business processes impacting on the A4BLUE MOD.EN.EM component.

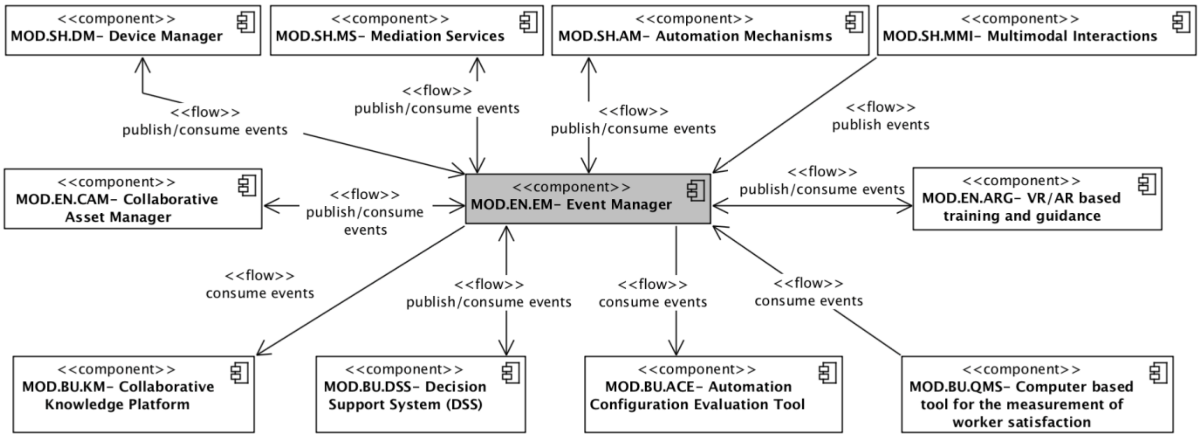


Figure 49 MOD.EN.EM. Cross-components business processes

The following Figure 50 depicts the basic interactions of the MOD.EN.EM with its natural counterparts, that are the Context Producers and the Context Consumers.

- Context Producers publish data/context elements by invoking the *updateContext* operation on a Publish/Subscribe Context Broker.
- Context Consumers can retrieve data/context elements by invoking the *queryContext* operation on a Publish/Subscribe Context Broker.
- Context data is kept persistent by Publish/Subscribe Context Brokers and ready to be queried.

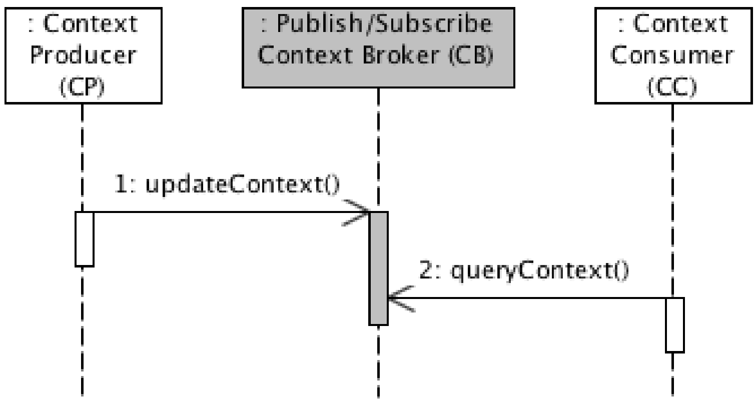


Figure 50 MOD.EN.EM. Basic interactions and related entities

Some Context Consumers can be subscribed to reception of data/context elements which comply with certain conditions, using the *subscribeContext* operation a CB exports, as shown in Figure 51.

- Subscribed consumers spontaneously receive data/context elements compliant with that subscription through the *notifyContext* operation they export.
- Note that the Application which subscribes a particular Context Consumer may or may not be the Context Consumer itself.

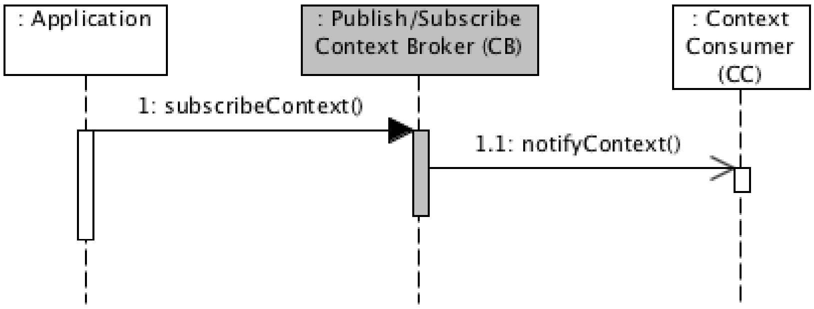


Figure 51 MOD.EN.EM. Interactions to force CCs to subscribe to specific notifications

The last example business process presented here below shows how the CB can interact with the CEP in order to start an event process (network) and then derive new event/information that feed again the CP itself.

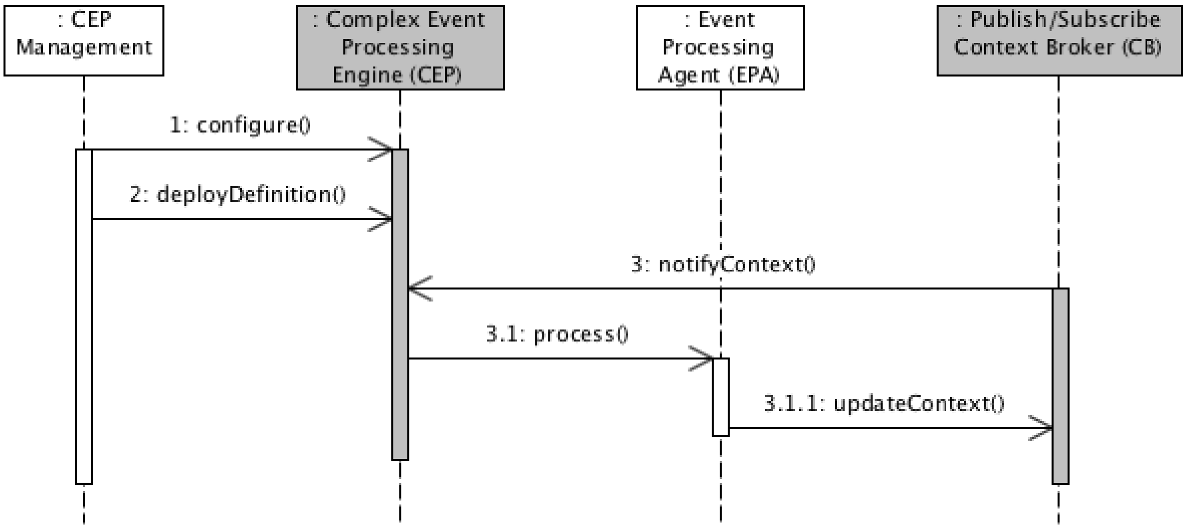


Figure 52 MOD.EN.EM. Interactions among CB and CEP

Furthermore, “EM.10. Status Manager” is in charge of updating the process status and keep historical records of all the events in the “CAM 03. Asset repository”.

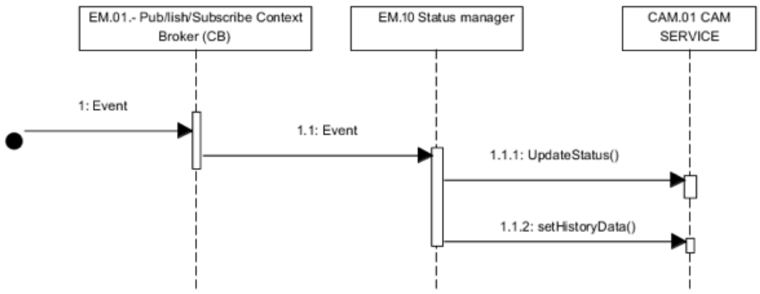


Figure 53 MOD.EN.EM. Status update

7.2.3 MOD.EN.ARG- VR/AR BASED TRAINING AND GUIDANCE

The MOD.EN.ARG component will be composed by a front-end application, viewable from a mobile AR device, and a back-end application that will act as a bridge between the AR device and the A4BLUE framework.

The operator using the AR device will be able, at any moment, to retrieve documents, 3D models, procedures and any other kind of asset available for that particular operative context. This kind of request will be forwarded from the Front-End application to the Back-End application, through the ResourceManager component. This object will have the capability to retrieve the requested asset from the Collaborative Asset Manager, create a local copy of it and send it back to the Front-End app. The copy allows the creation of a caching system and saves bandwidth, as a web server does not need to send a full response if the content has not changed.

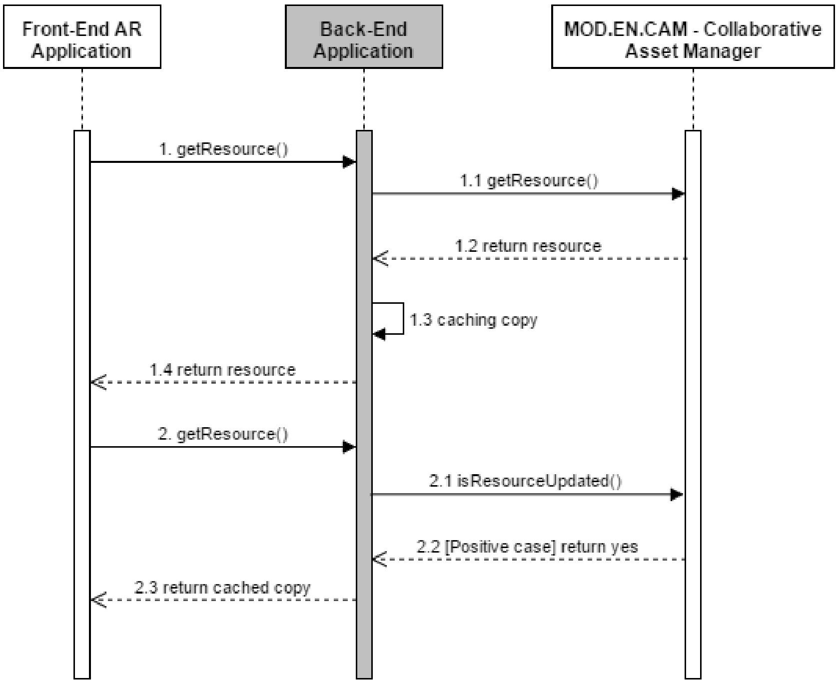


Figure 54 MOD.EN.ARG. Basic asset retrieving

The Back-end will also be able to publish content and subscribe to the Event Manager, in order to always present synchronized content in the AR Front-End.

7.3 BUSINESS LAYER

7.3.1 MOD.BU.KM- COLLABORATIVE KNOWLEDGE PLATFORM

Every worker using the MOD.BU.KM component will be able to generate content, retrieve and manage knowledge using the Asset Repository capability provided by the MOD.EN.CAM component. Both M2M and H2M interactions will be enabled by such a component, through the use of open APIs and user friendly GUIs in charge of enabling the collaboration process aimed at enriching the knowledge repository itself.

The following Figure 55 shows the main business processes impacting on the A4BLUE MOD.BU.KM component.

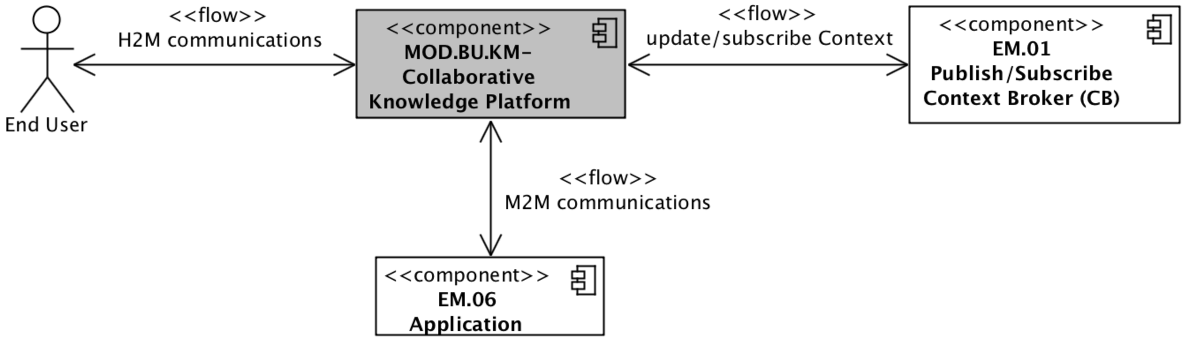


Figure 55 MOD.BU.KM. Cross-components business processes

Figure 56 and Figure 57 show example interactions with the end users, in order to add and retrieve blog entities (but similar interactions can occur for any knowledge entity managed by the platform.

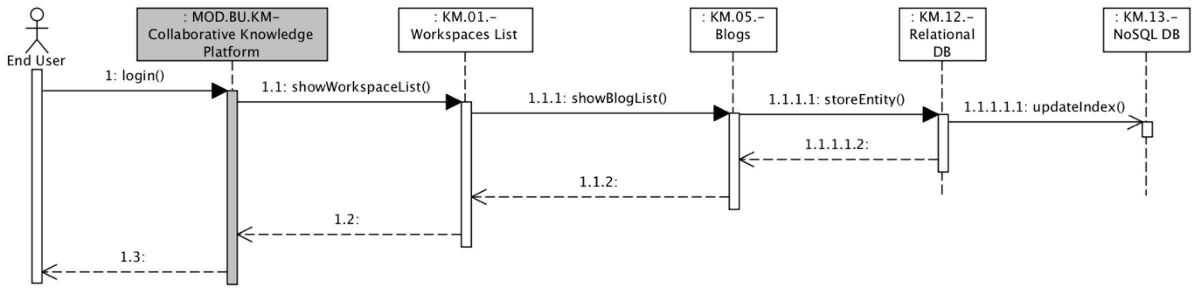


Figure 56 MOD.BU.KM. Create a new blog entry

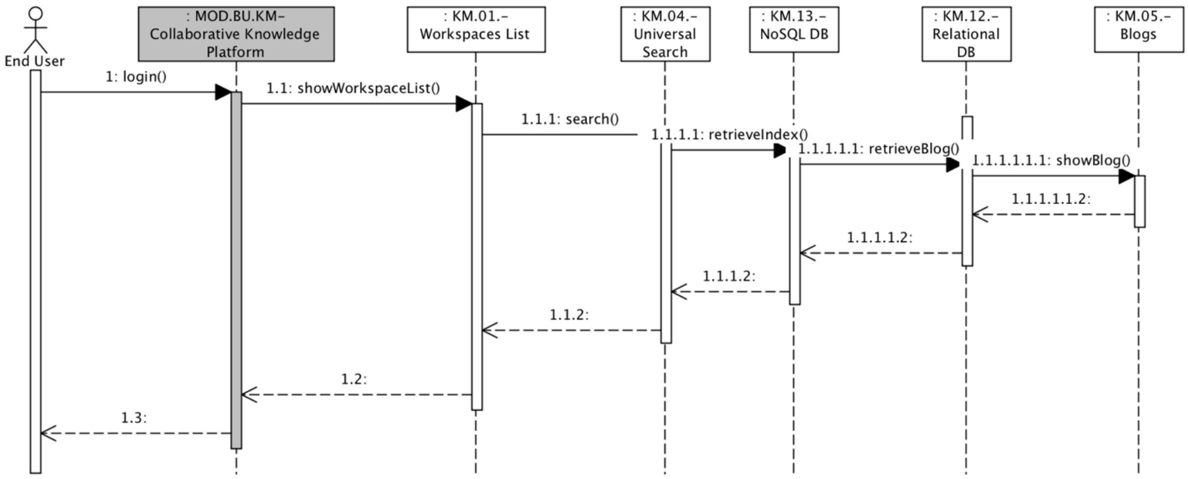


Figure 57 MOD.BU.KM. Retrieve blog entry

7.3.2 MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)

The following sequence diagrams provide details of the interactions among sub-components to support “FTR.08- Adaptive assistance: decision support” as described in the specifications section.

Access decision support information

The users can access the operational info through the graphical user interface (“DSS.01. DSS Dashboard”) and the system displays the results in the appropriate format. To this end the historical information stored in the “CAM.03. Virtual asset repository” is accessed though the “CAM.01. CAM services” and processed by “DSS.02. DSS processing services”.

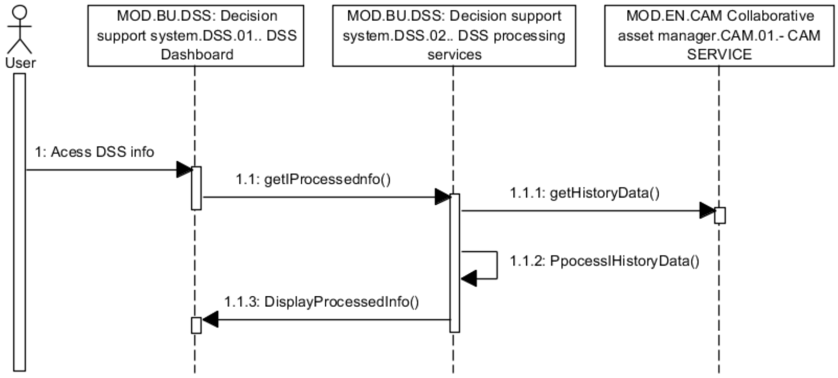


Figure 58 MOD.BU.DSS. Access decision support information

Notify intervention request

“DSS.04. DSS event adapter” receives notification events related to intervention requests (e.g. collaboration request, maintenance or inspection intervention request, etc.) and it is in charge of adapting them if required, identifying the notification receivers if required, as well as the appropriate notification channels based on the context information represented in the “CAM.03. Virtual asset repository”. “DSS.03. DSS notification management services” implements the business logic to notify the target user by using the appropiate notification channel (e.g. graphical user interface, push notification, email, sound, light indicator, etc.).

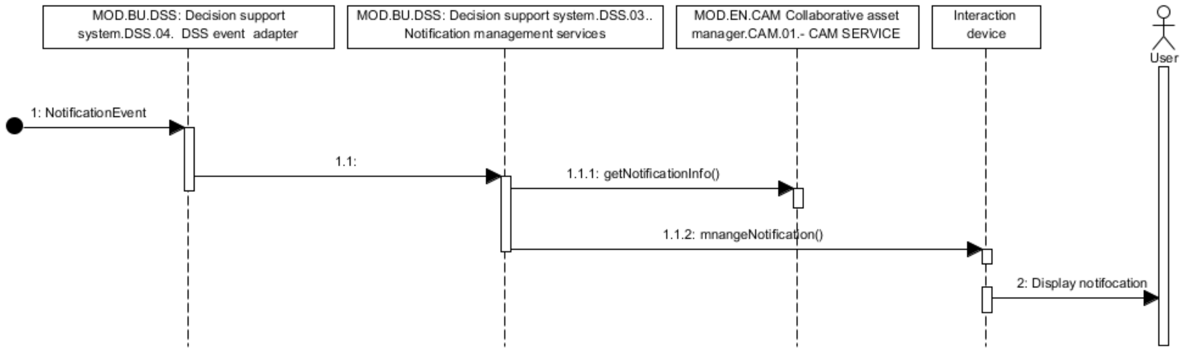


Figure 59 MOD.BU.DSS. Notify intervention request

7.3.3 MOD.BU.MON- MONITORING

As stated before indicators based on process history information are displayed by taking advantage of the MOD.BU.DSS component and the survey based indicators use the MOD.BU.CQMS component. So refer to 7.3.2 and 7.3.5for details.

7.3.4 MOD.BU.ACE- AUTOMATION CONFIGURATION EVALUATION

The automation configuration evaluation consists of two main interactions, the insertion and the access of information, both triggered by the user. The following sequence diagrams describe these interactions, by providing details among sub-components and also its connection to the user (called “Actor”).

Insertion of automation data by the user

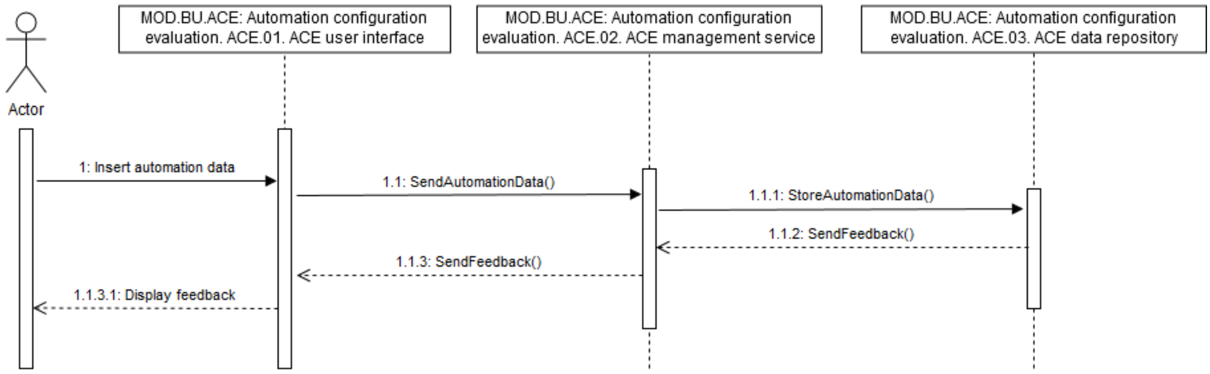


Figure 60 MON.BU.ACE. Insertion of automation data by the user

Once the user has inserted information into the “ACE.01. – ACE user interface”, it will be sent to the “ACE.02. – ACE management service”. It is responsible of executing needed calculations and of saving the information into the “ACE.03. - ACE data repository”. The system sends feedback to the user about the status of the operation.

Access of automation data

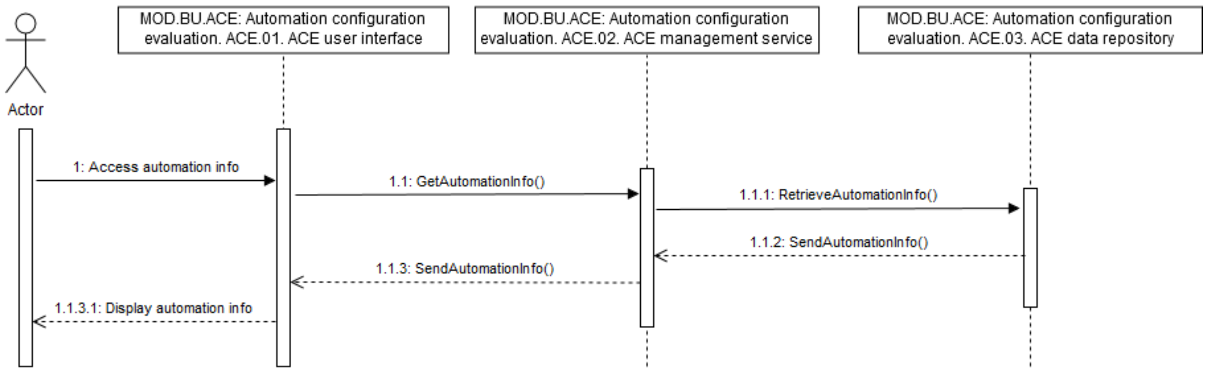


Figure 61 MON.BU.ACE. Access of automation data

When the user accesses automation information the user interface “ACE.01. – ACE user interface” will get information from the database “ACE.03. – ACE data repository” through the management service “ACE.02. – ACE management service”. The interface will then display the automation information to the user.

7.3.5 MOD.BU.CQMS- COMPUTER BASED TOOL FOR THE MEASUREMENT OF WORKER SATISFACTION

The following sequence diagrams provide details of the interactions among sub-components to support “FTR.11.- Worker satisfaction assessment” as described in the specifications section.

A sequence diagram is provided for each use case scenario involved in FTR.11. Notice that the return messages have been explicitly identified as providing feedback on the results of the process as has been identified as a requirement.

CRUD users

In the offline mode the system administrator is able to create/ update or delete user data through the user administration graphical user interface and the system provides feedback on the result of the process by displaying the appropriate message.

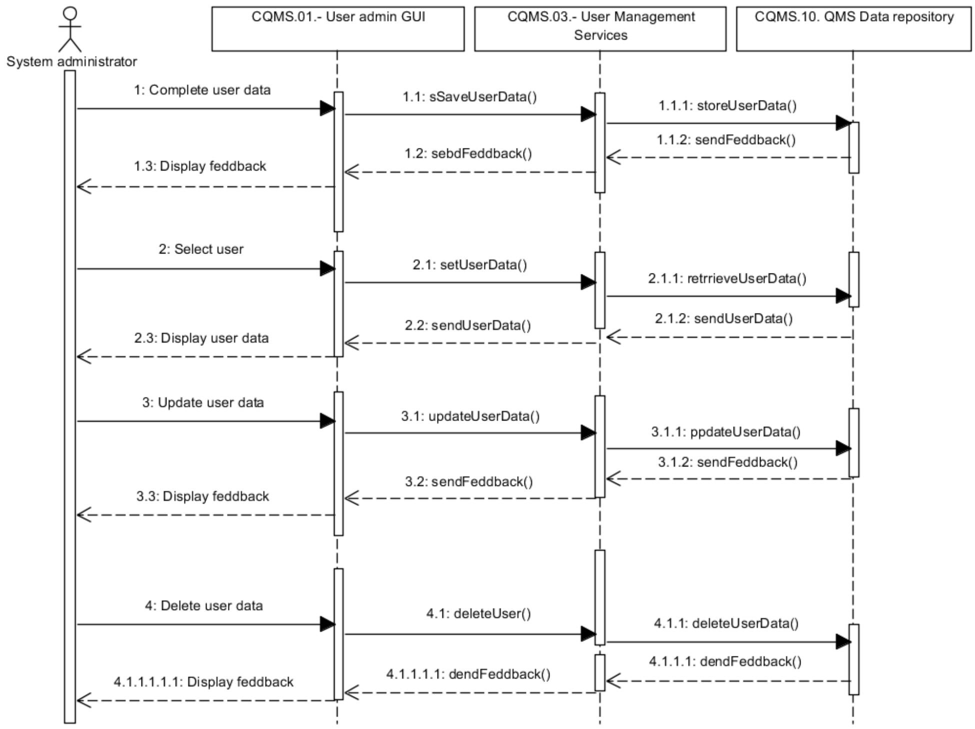


Figure 62 MOD.BU.MON. CRUD users – offline mode

In the online mode the system automatically creates/ updates or deletes user data based on the Person update event provided by the virtual asset repository (CAM.03) through "CAM.01. CAM Service" component.

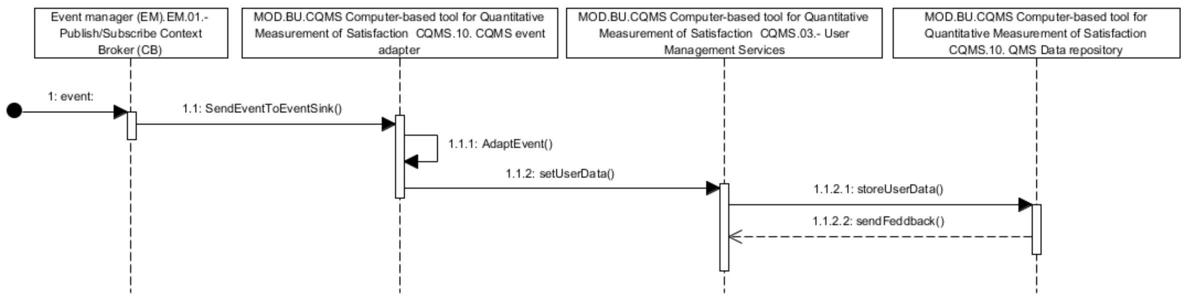


Figure 63 MOD.BU.MON. CRUD users – online mode

CRUD questionnaire

As a first approach system administrators can load questionnaires into the system using file including the questionnaire definition.

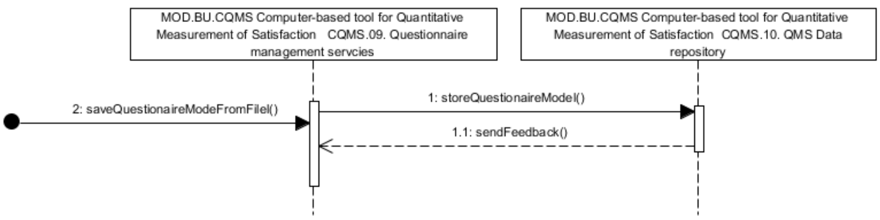


Figure 64 MOD.BU.CQM. Upload questionnaire from file

Optionally, the system administrator could create/ update or delete questionnaire model through the questionnaire administration graphical user interface and the system provides feedback on the result of the process by displaying the appropriate message.

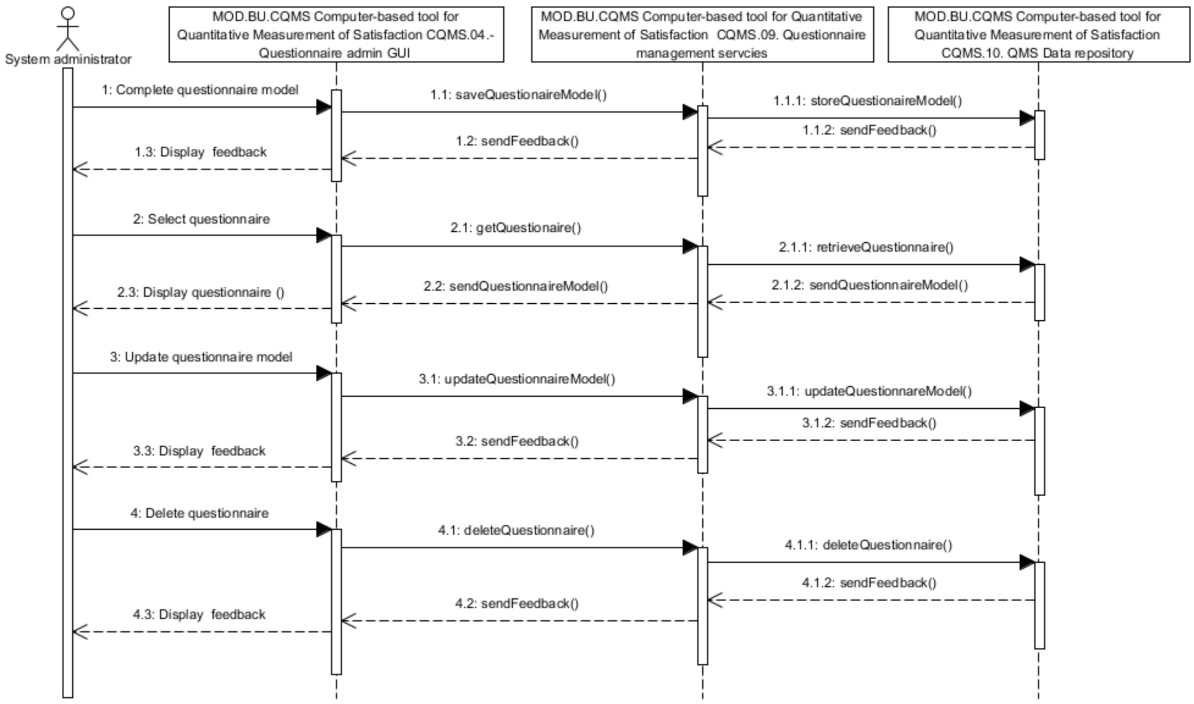


Figure 65 MOD.BU.CQM. CRUD questionnaire using GUI

Complete worker satisfaction questionnaire

The shop floor operator is able to complete and save the questionnaire. Before saving it the system checks if all the required questions have been completed. If so, the system saves the results to the database and provides direct feedback on results of the process by displaying an appropriate message. If responses are not complete the system displays a message asking the user to review them.

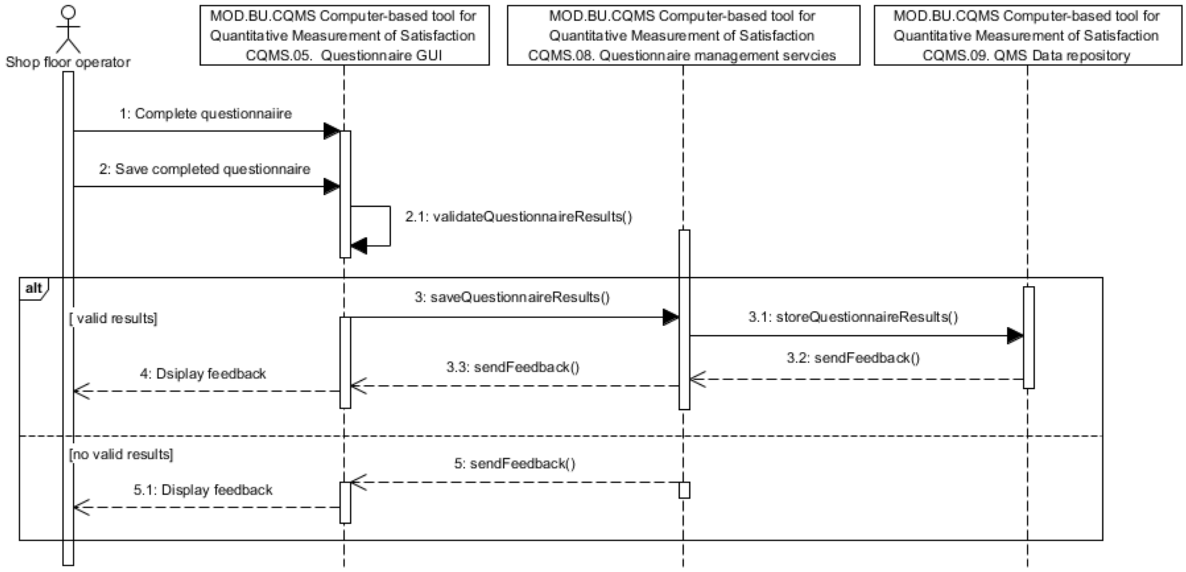


Figure 66 MOD.BU.CQM. Complete worker satisfaction questionnaire

Access individual view of worker satisfaction questionnaires results

Individual results of worker satisfaction questionnaire(s) should be accessible to both shop floor operators and supervisors. When individual worker results are requested, the system displays them prioritising graphical representation whenever possible.

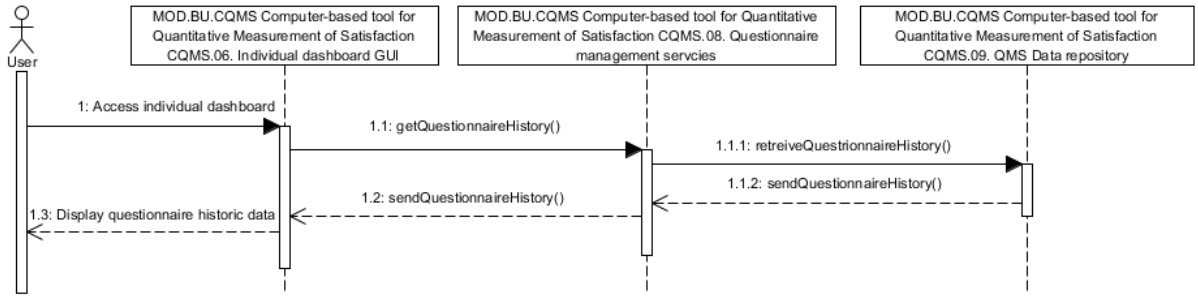


Figure 67 MOD.BU.CQM. Access individual view of worker satisfaction questionnaires results

Access aggregated view of worker satisfaction questionnaires

When Supervisors ask for aggregated worker questionnaire results the system displays the collected results prioritising graphical representation whenever possible.

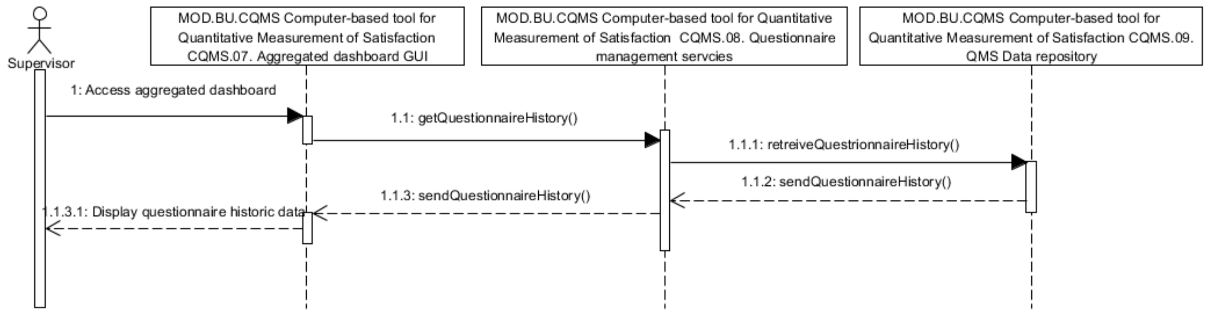


Figure 68 MOD.BU.CQM. Access aggregated view of worker satisfaction questionnaires

8 FBB SPECIFICATION

8.1 SHOPFLOOR LAYER

8.1.1 MOD.SH.MS- MEDIATION SERVICES

8.1.1.1 FBB Specification

The figure below shows the main components of the MOD.SH.MS describing both main Functional Building Blocks and information flows.

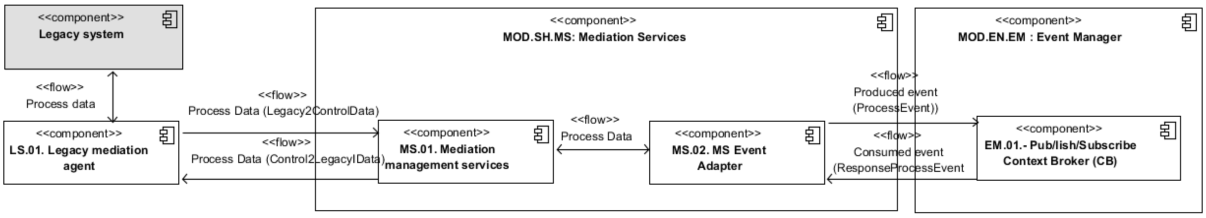


Figure 69 MOD.SH.MS. Decomposition into functional building blocks

8.1.1.1.1 MS.01. Mediation management services

“MS.01. Mediation management services” supports the exchange of the presence and process information coming from/to the legacy system. If required a specific legacy system dependent component can be necessary (i.e.” LS.01 Legacy mediation agent”).

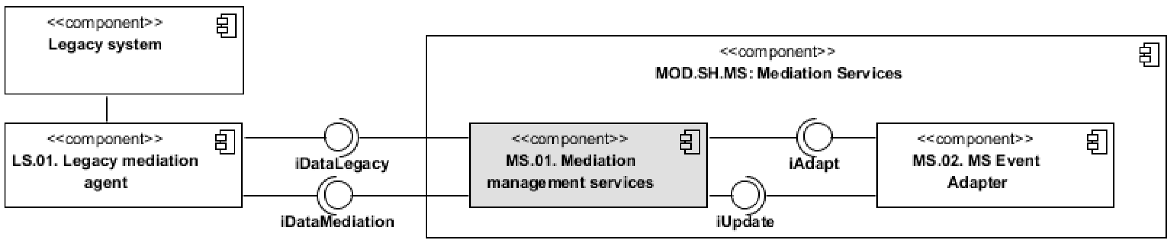


Figure 70 MS.01. Mediation management services interfaces

MS.01 exposes **iDataMediation** interface to capture the presence and process information from the legacy system (i.e through the appropiate LS.01. Legacy mediation agent” , if required) and **iUpdate** interface to send updated process information to the legacy system. MS.01 consumes the **iAdapt** interface exposed by the “M.02 MS event adapter” to publish events in the appropriate format and the *updateProcessInfo* method of the **iUpdate** interface exposed by the “MS.01. Mediation management services”.

8.1.1.1.2 MS.02. MS Event Adapter

“MS.02. MS event adapter” supports publish and subscribe capabilities and adapts the information produced by ”MS.01. Mediation management services” to the event format supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM into process data to update the legacy system (e.g. information collected during the execution of the operations performed by the automation mechanisms) through MS01.

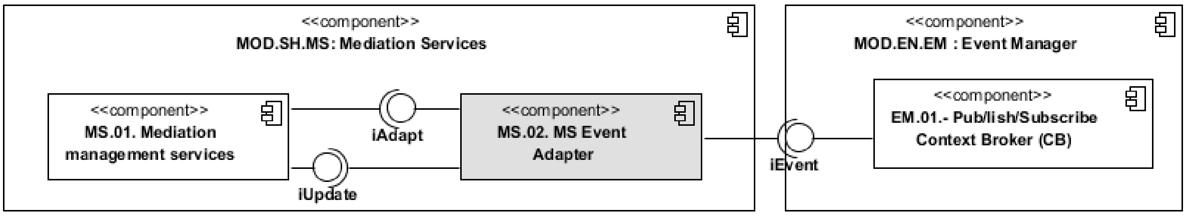


Figure 71 SM.02. MS Event Adapter interfaces

MS.02 exposes **iAdapt** interface and consumes the **iEvent (including Publish and Subscribe methods)** interface exposed by the "EM.01 Publish /Subscribe Context Broker (CB)" to publish events in the appropriate format and the *updateProcessInfo* method of the **iUpdate** interface exposed by the "MS.01. Mediation management services".

8.1.1.2 Background Assets

No background asset are used.

8.1.1.3 A4BLUE Enhancements

Completely new component.

8.1.1.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.SH:MS.

Interface	Method	Description
iDataLegacy	setIdentificationInfo	Collect presence information from legacy system .
	setProcessInfo	Collect process information from legacy system.
iUpdate	updateProcessInfo	Update process information in the legacy system.

Table 18 MS.01. Mediation management services main methods

Interface	Method	Description
iAdapt	setIdentificationData	Collects identification data.
	setProcessData	Collects process data.

Table 19 DM.02. DM event protocol adapter main methods

8.1.2 MOD.SH.AM- AUTOMATION MECHANISMS

8.1.2.1 FBB Specification

The figure below shows the main components of the MOD.SH.AM describing both main Functional Building Blocks and interfaces.

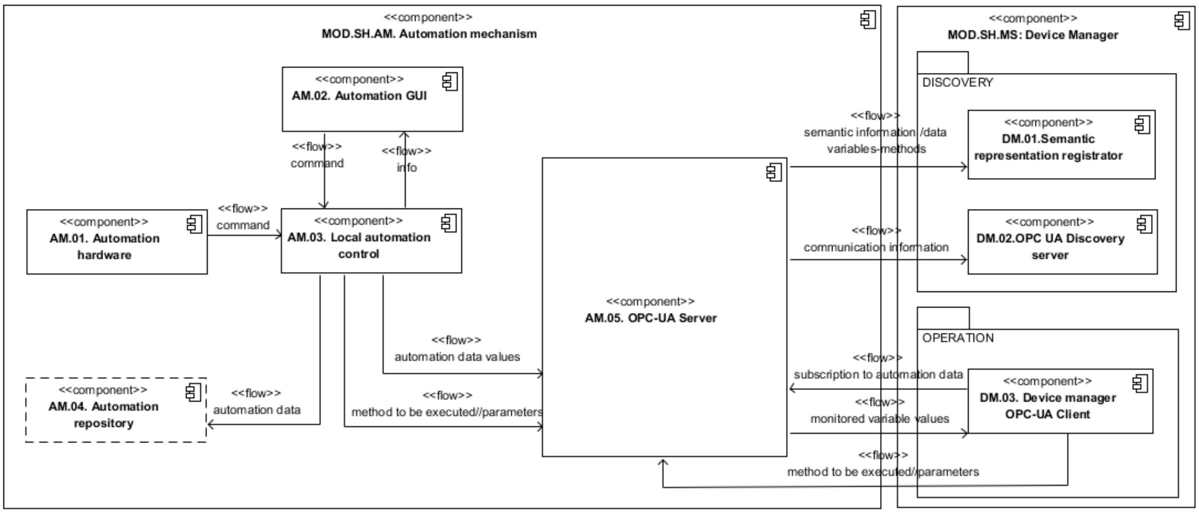


Figure 72 MOD.SH.AM. Decomposition into functional building blocks

8.1.2.1.1 AM.01. Automation hardware

The “AM.01. Automation hardware” represents the hardware part of the automation mechanism and it is capable of performing the assigned task (e.g. assembly operation, etc.). The automation hardware is controlled by the “AM.03. Local automation controller”

8.1.2.1.2 AM.02. Automation GUI

The “AM.02. Automation GUI” allows user to interact with the automation mechanisms to configure it or give specific commands.

The FBB consumes the *executeCommand* method of the **iCommand** interface exposed by the “AM.03. Local automation controller”.

8.1.2.1.3 AM.03. Local automation controller

The “AM.03. Local automation controller” controls the “AM.01. Automation hardware” (e.g. robot).

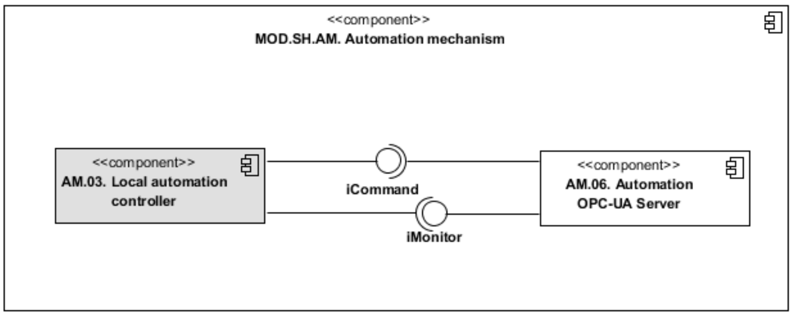


Figure 73 MOD.SH.AM.03. Local automation controller interfaces

The FBB exposes **iCommand** interface and consumes the **iMonitor** interface exposed by the “AM.06. Automation OPC-UA Server” to publish the updates of the monitored data.

8.1.2.1.4 AM.04. Automation data repository

The “AM-04. Automation data repository” enables the storage of automation mechanism related information. It is accessed through the “AM.03.- Local automation controller”.

8.1.2.1.5 AM.05. OPC UA Server

"AM.05. OPC UA server" registers (1) semantic representation of the automation through the "DM.01. Semantic registrator"; (2) the communication information (IP and Port) in the "DM.02 OPC UA discovery server" and (3) the data to be monitored in the "DM.03. Device manager OPC UA". Furthermore it provides updates of the monitored automation data to the subscribed components and executes automation methods.

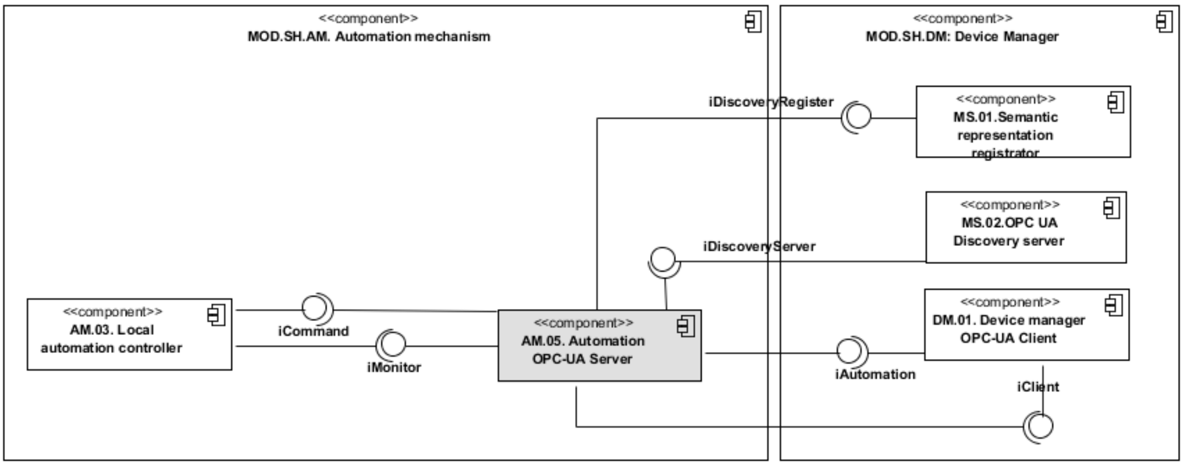


Figure 74 AM.05. OPC-UA Server interfaces

AM.05 exposes **iAutomation** interface to support the subscription to the monitored variables and execute the appropriate methods and **iMonitor** interface to support the exchange of the updated values of the monitored variables. AM.05 consumes the *registerSemanticRepresentation* method of the **iDiscoveryRegister** interface exposed by the "DM.01. Semantic representation registrator", the *registerServer* method of the **iDiscoveryServer** interface exposed by the "MS.02 OPC UA discovery server" and the *setSubscription* method of the **iClient** interface exposed by the "DM.03. Device manager OPC UA".

8.1.2.2 Background Assets

In some use case scenarios automation mechanisms are already available:(i.e. AIRBUS: Smart tools; TEK: Dual arm and mobile robot; RWTH: mobile trolley), while in CESA a new automation (i.e. robot for deburring process) will be introduced.

8.1.2.3 A4BLUE Enhancements

Where automation mechanisms are already available A4BLUE includes features to support integration of the automation and adaptation of the overall system. Furthermore, it includes a generic OPC-UA server to support plug and produce capabilities where it is not available.

8.1.2.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.SH.AMS.

Interface	Method	Description
iCommand	executeCommand	Executes the command supported by the automation mechanism.

Table 20 MOD.SH.AM.03. Local automation controller main methods

Interface	Method	Description
iMonitor	setMonitoredData	Collect updates in the monitored data.
iAutomation	subscribeToData	Subscribes to variables provided by the automation.
	executeMethod	Executes method exposed by the automation mechanism.

Table 21 MOD.SH.AM.06. Automation OPC-UA Server main methods

8.1.3 MOD.SH.DM- DEVICE MANAGER

8.1.3.1 FBB Specification

The figure below shows the main components of the MOD.SH.DM, describing both main Functional Building Blocks and information flows.

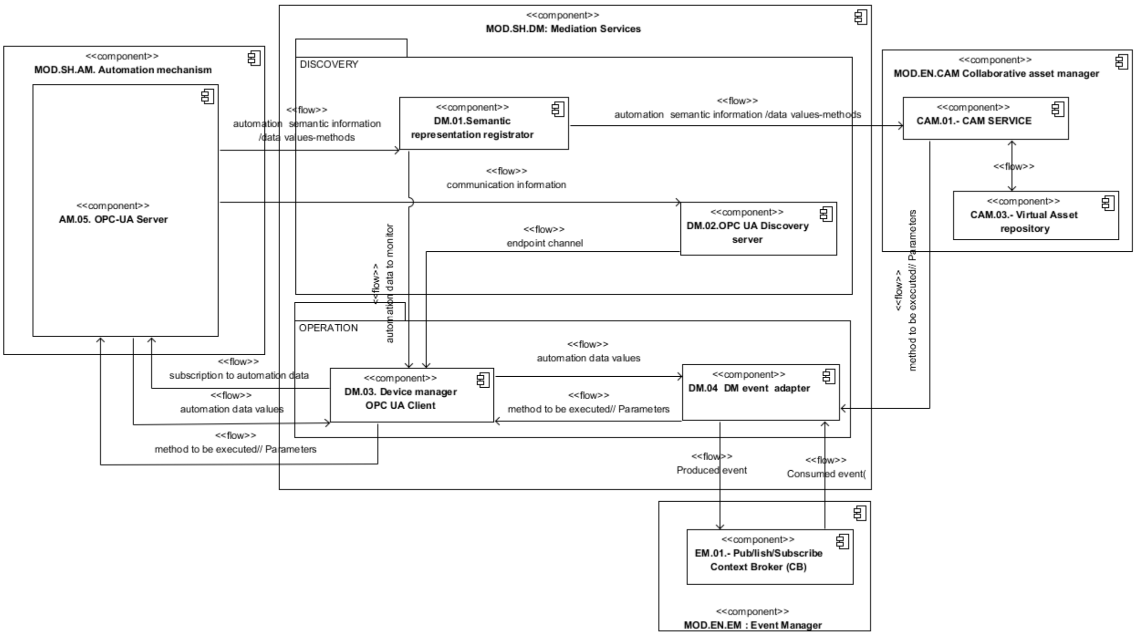


Figure 75 MOD.SH.DM. Decomposition into functional building blocks

8.1.3.1.1 DM.01. Semantic representation registrator

The “DM.01. Semantic representation registrator” registers the semantic representation provided by the automation mechanisms (i.e available variables to be monitored and methods) in the “CAM.01.Virtual Asset Repository” through the “CAM.01.CAM Service” of the MOD.EN.CAM Collaborative Asset Manager”. Furthermore, it assigns the automation mechanism to the right workcentre based on its IP.

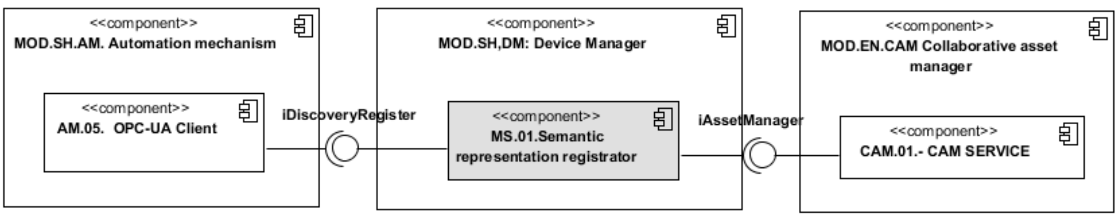


Figure 76 DM.01. Semantic representation repository interfaces

DM.01 exposes **iDiscoveryRegister** interface to support the resgistration of the semantic representation and consumes the methods exposed by the "CAM.01. CAM-Service".

8.1.3.1.2 DM.02 OPC UA discovery server

"DM.02 OPC UA discovery server" maintains the list of OPC UA Servers that are registered and provides mechanisms for clients (DM.01. Device manager OPC UA) to find them. To support a standardised approach this component must follow the specification of the Local Discovery Service (LDS) provided by the OPC Foundation[11].

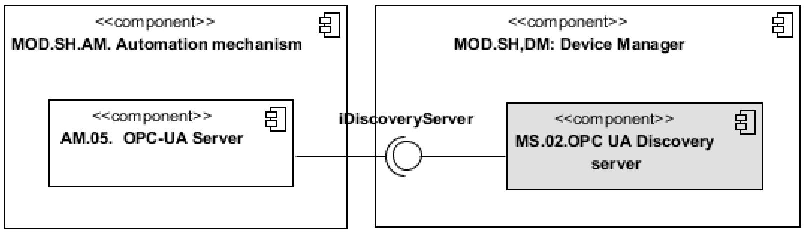


Figure 77 DM.02. OPC-UA discovery server interfaces

DM.02 exposes **iDicoveryServer** interface.

8.1.3.1.3 DM.03. Device manager OPC UA client

"DM.03. Device manager OPC UA client" supports functionalities to: (1) retrieve the endpoint of the appropriate OPC-UA Server from the "DM.02 OPC UA discovery server"; (2) subscribe to the data to be monitored through the appropriate OPC-UA Server; (3) publish variable update events through the "DM.04. DM event adapter" and (4) execute the appropriate methods when required (i.e as result of a triggered command event).

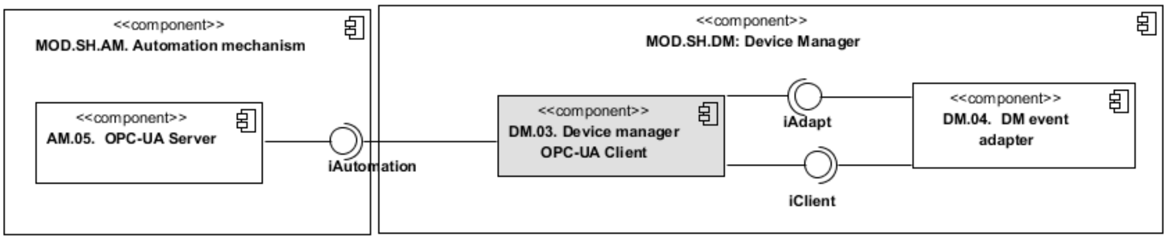


Figure 78 DM.03. Device Manager OPC-UA interfaces

DM.03 exposes **iClient** interface to support the reception of the updated data values from the automation mechanisms and to get the information of the method to be executed. Moreover, it consumes the *subscribeToData* and *executeMethod* methods of the **iAutomation** interface exposed by the "AM.05. OPC-UA Server" and the *publishEvent* method of the **iAdapt** interface exposed by the "DM.04. DM event adapter".

8.1.3.1.4 DM.04. DM event adapter

"DM.04. DM event protocol adapter" adapts the information produced by the" DM.02. Device manager OPC UA" to the event format supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the events produced by the MOD.EN.EM considering the semantic information stored in

[11] OPC foundation: https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-12-discovery

the “CAM.03. Virtual asset repository” and executes the appropriate method provided by the automation mechanism through “DM.03. Device manager OPC UA client”.

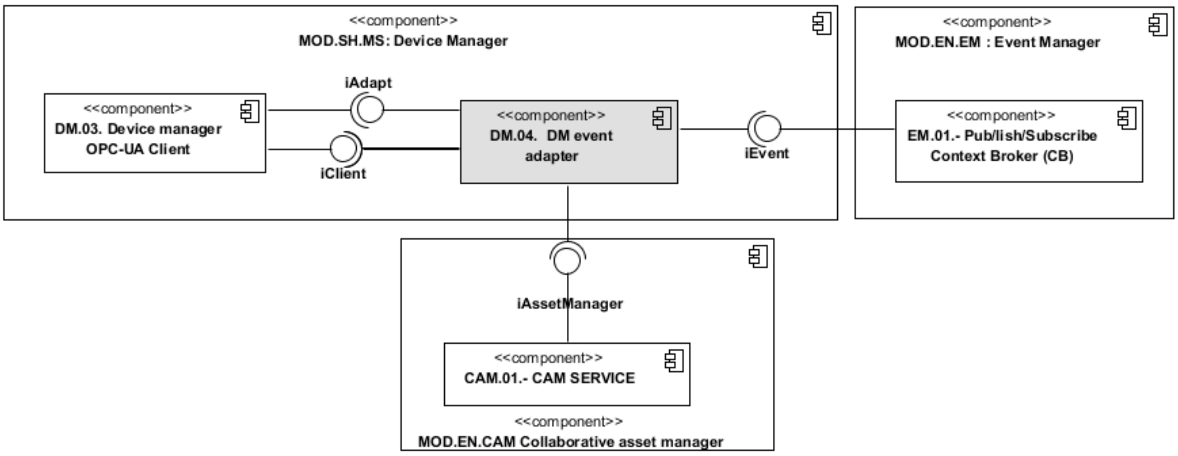


Figure 79 DM.04. DM event adapter interfaces

DM.04 exposes **iAdapt** interface to support the exchange of the updated data values and the methods to be executed and consumes the **iEvent (including Publish and Subscribe methods)** interface exposed by the “EM.01 Publish /Subscribe Context Broker (CB)” to publish events in the appropriate format, the *setMethodToExecute* method of the **iClient** interface exposed by the “DM.03. Device manager OPC UA client” and the apporpiate method of the **iAssetManager** interface “CAM.01. Services” to get the required information from the “CAM.03. Virtual asset repository” .

8.1.3.2 Background Assets

To support a standardised approach to the discovery process the “MS.02 OPC UA discovery server” is a component that must follow the specification of the Local Discovery Service (LDS) provided by the OPC Foundation[12].

8.1.3.3 A4BLUE Enhancements

The rests of the involved components will be developed in the framework of A4BLUE.

8.1.3.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.SH:DM.

Interface	Method	Description
iDiscoveryRepresentation	registerSemanticRepresentation	Registers the semantic representation including variables and methods and assigns the automation mechansms to the right workcentre.

Table 22 DM.01. Semantic representation registrator main methods

Interface	Method	Description

[12] OPC foundation: https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-12-discovery

Interface	Method	Description
iDicoveryServer	registerServer	Registers the OPC UA Server
	findServer	Retrieves the required information from the OPC UA Server

Table 23 DM.02. OPC-UA discovery server main methods

Interface	Method	Description
iClient	setmethodToExecute	Collect the information of the method to be executed
	getDataSink	Provides an endpoint to receive the information of the monitored data.

Table 24 DM.03. Device manager OPC-UA methods

Interface	Method	Description
iAdapt	setVariableValue	Collects the updated data of a subscribed variable.

Table 25 DM.04. DM event protocol main methods

8.1.4 MOD.SH.MHMI- MULTICHANNEL HUMAN-AUTOMATION/ROBOT INTERACTION

8.1.4.1 FBB Specification

The figure below shows the main components of the MOD.SH.MHMI, describing both main Functional Building Blocks and information flows.

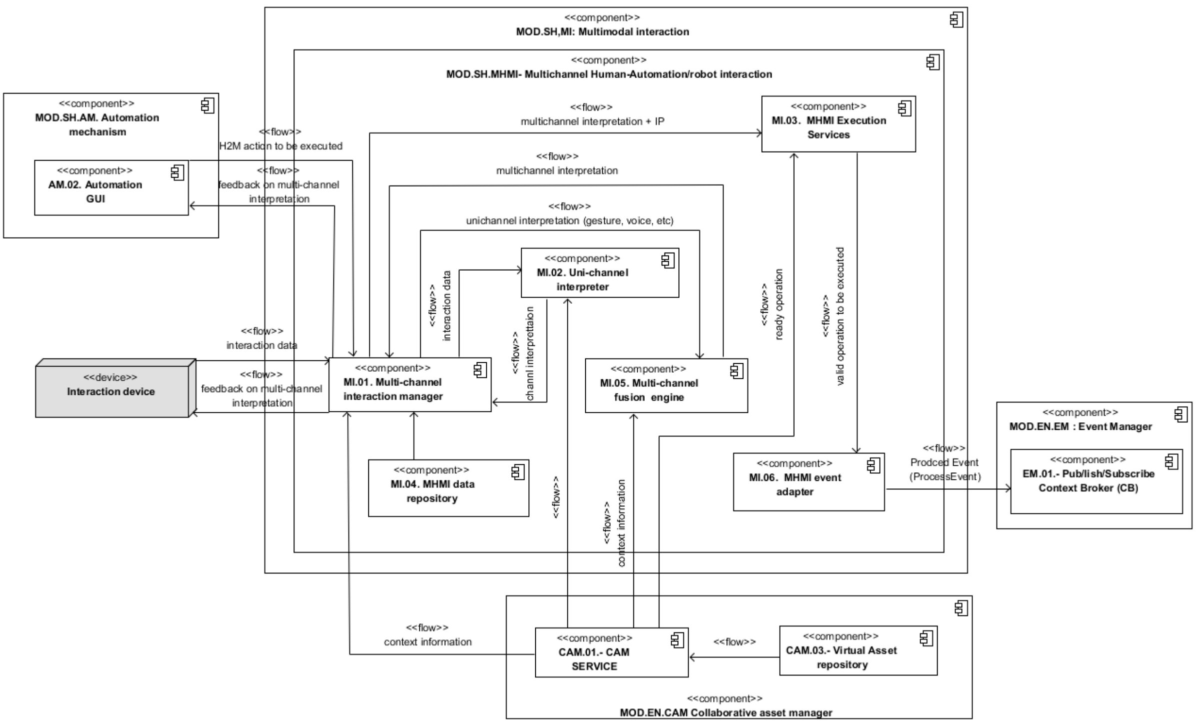


Figure 80 MOD.SH.MHMI. Decomposition into functional building blocks

8.1.4.1.1 MI.01. Multi-channel interaction manager

“MI.01. Multi-channel interaction manager” orchestrates the rest of the components to perform the Human to Machine (H2M) and Machine to Human (M2H) interaction. It supports functionalities to: (1) get the inputs provided by the interaction sources (i.e. interaction devices, GUI); (2) send collected data to the appropriate interpreter (MI.02. Uni-channel interpreter); (3) send partial interpretations to the “MI.05. Multi-channel fusion engine”; (4) publish and subscribe (e.g. context info, actions) events through the “MI.06. MHMI event protocol adapter”; (5) manage the feedback on multi-channel interpretation and (6) support data persistence to support incremental learning interpretation readiness.

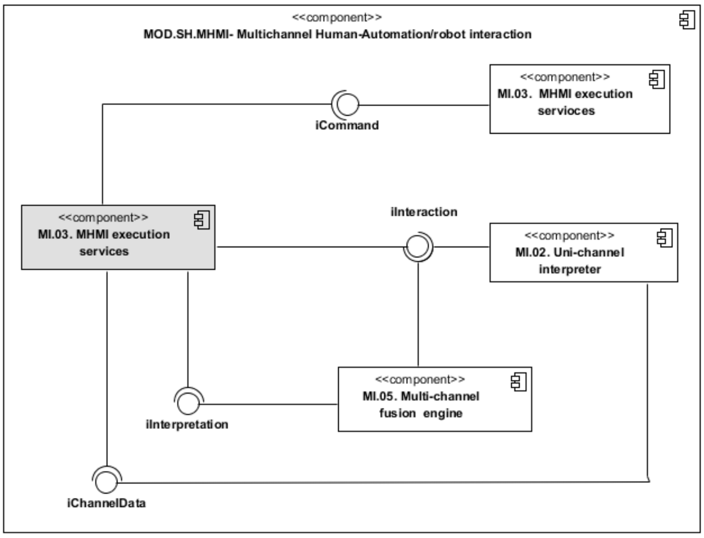


Figure 81 MI.01. Multi-channel interaction manager interfaces

MI.01 exposes **iData** and **iInteraction** interfaces and consumes the *setChannelInterpretation* method of the **iInterpretation** interface exposed by the “MI.05. Multi-channel fusion engine”, the *setChannelData* method of the **iChannelData** interface exposed by the “MI.02. Uni-channel interpreter“ and the *setCommand* method of the interface **iCommand** exposed by the “MI.03. MHMI execution services”. Furthermore, it interacts with the “MI.04. MHMI data repository” to store and retrieve the required information.

8.1.4.1.2 MI.02. Uni-channel interpreter

“MI.02. Uni-channel interpreter” interprets the input data provided by the “MI.01 Multi-channel interaction manager” by using the instruction’s semantic representation provided by the vistual asset repository through the “CAM.01. CAM Services” and sends the obtained channel interpretation to the “MI.01 Multi-channel interaction manager”

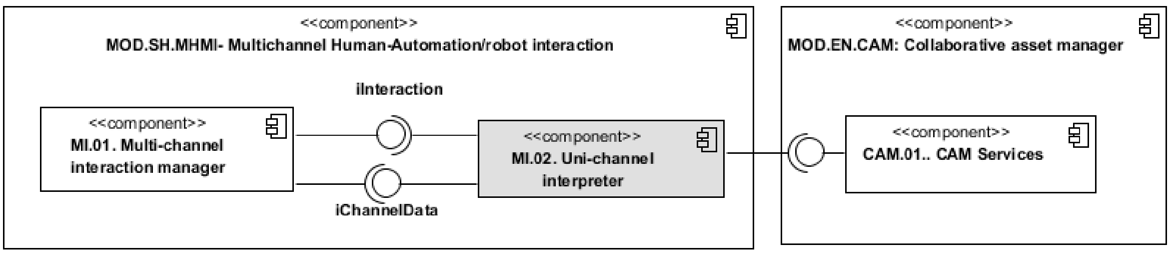


Figure 82. MI.02 Uni channel interpreter interfaces

MI.02 exposes **iChannelData** interface and consumes the *saveChannelInterpretation* method of the **iInteraction** interface exposed by the “MI.01 Multi-channel interaction manager”. Furthermore, it interacts with the vistual asset repository to retrieve the required information. The following table shows the main methods exposed by the **iChannelData** interface. The following table shows the main methods exposed by the **iChannelData** interface.

8.1.4.1.3 MI.03. MHMI Execution Services

“MI.03. MHMI execution services” transforms the interpretration resulting from the “MI.01. Multi-channel interaction manager” into a valid command and sends the information to “MI.06. MHMI event protocol adapter”.

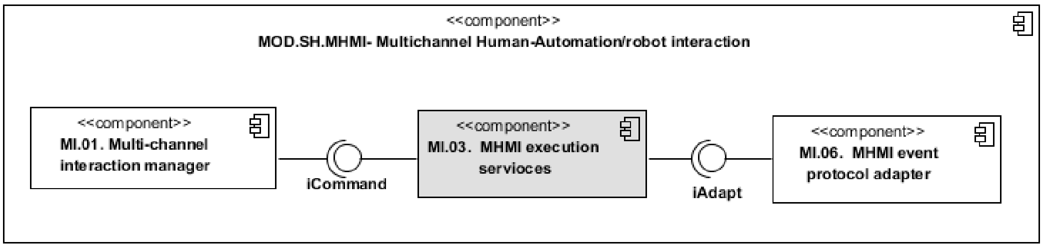


Figure 83. MI.03 .MHMI Execution services interfaces

MI.03 exposes **iCommand** interface and consumes the *publishEvent* method exposed by “MI.06. MHMI event adapter”.

8.1.4.1.4 MI.04. MHMI data repository

“MI.04. MHMI data repository” support data persistence for interpretation incremental learning readiness.

8.1.4.1.5 MI.05. Multi-channel fusion engine

“MI.05. Multi-channel fusion engine” fuses the uni-channel interpretations (i.e. partial) considering the instruction’s semantic representation provided by the by the vistual asset repository through the “CAM.01. CAM Services” and provides the complete multichannel interpretation. An additional input to the partial interpretations that could be managed by the “MI.05. Multi-channel fusion engine” to obtain the multichannel interpretation is context information (e.g. operator identification, environment status information, etc.).

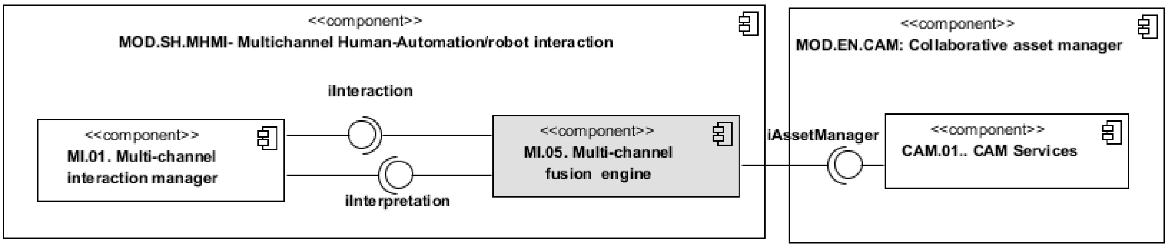


Figure 84 MI.05. Multi-channel fusion engine interfaces

MI.05 exposes **iInterpretation** interface and consumes the *saveMultichanneInterpretation* method of the **iInteraction** interface exposed by the “MI.01 Multi-channel interaction manager”. Furthermore, it consumes the methods exposed by the “CAM.01. CAM services” to store and retrieve the required information from the “CAM.03. virtual asset repository”.

8.1.4.1.6 MI.06. MHMI event adapter

“MI.06. MHMI event adapter” checks the feasibility of the command info produced by the “MI.03. MHMI Execution services” and adapts it to the event format supported by the Event Manager (MOD.EN.EM).

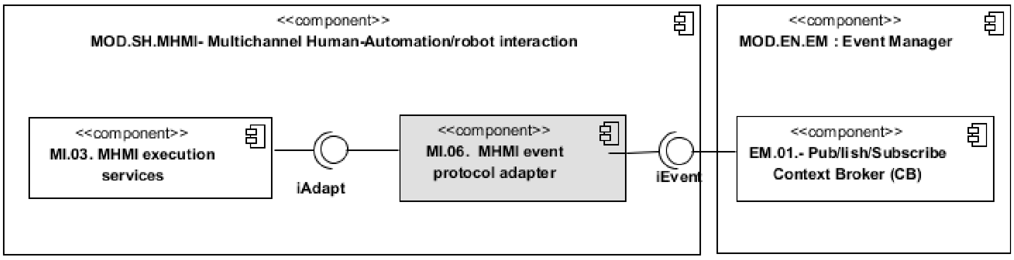


Figure 85 MI.06. MHMI event adapter interfaces

MI.06 exposes **iAdapt** interface and consumes the **iEvent (including Publish and Subscribe methods)** interface exposed by the EM.01 Publish /Subscribe Context Broker (CB) to publish events in the appropriate format and executes the appropriate method exposed by the **iCommand** interface of the “MI.03. MHMI Execution services”.

8.1.4.2 Background Assets

The **4X3 Multimodal human-robot natural communication** component developed in the scope of FourByThree project[13] is aimed to enhance collaborative scenarios where human and robots interact in a natural way. The current implementation supports voice and/or gestures. Using semantics, it is able to handle voice and gesture-based natural requests from a person, and combine both inputs to generate an understandable and reliable command for industrial robots, facilitating a safe collaboration.

For such a semantic interpretation, the component relies on four main modules, as shown in the figure below: a *Knowledge-Manager* module that describes and manages the environment and the actions that are affordable for robots, using semantic representation technologies; a *Voice Interpreter* module that, given a voice request, extracts the key elements on the text and translates them into a robot-understandable representation, combining NLP and semantic technologies; a *Gesture Interpretation* module to resolve pointing gestures and some simple orders like stopping an activity (out of the scope of the work presented in this paper); and a *Fusion Engine* for combining both mechanisms and constructing a complete and reliable robot-commanding mechanism.

[13] FourByThree: http://fourbythree.eu/

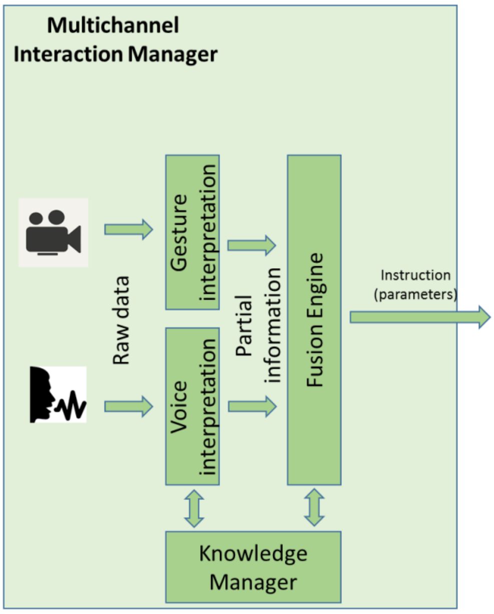


Figure 86 4X3 Multimodal human-robot natural communication component

The Multichannel Interaction Manager is the component responsible for managing the orchestration of the modules from the collection of signals (voice and gestures) to sending the interpretation of the interaction to the opportune component. The solution is configurable in the sense that the solution can operate with a single input channel (voice or gestures) or both.

In the current implementation, the four main modules work as services while the Multichannel Interaction Manager is a ROS[14] based node.

8.1.4.3 A4BLUE Enhancements

The MOD.SH.MHMI will rely on the above-mentioned background asset, especially since it provides the capability to fuse multichannel sources but it will be enhanced to increase reliability, consider context information (e.g.process and operator related info) as well as manage the feedback process and support event exchange.

8.1.4.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.SH:MHMI.

Interface	Method	Description
iData	setInteractionData	Collect the raw data from the interaction data sources.
iInteraction	saveChannelInterpretation	Save the uni-channel interpretation.
	saveMultiChannelInterpretation	Save the multichannel interpretation
	saveContextInfo	Save context information.

Table 26 MI.01. Multi-channel interaction manager main methods

[14] http://wiki.ros.org/Nodes

Interface	Method	Description
iChannelData	setChannelData	Collect interaction data related to the specific interaction channel.

Table 27. MI.02. Uni-channel interpreter main methods

Interface	Method	Description
iCommand	setCommand	Receive the identified interaction-

Table 28. MI.02. Uni-channel interpreter main methods

Interface	Method	Description
iInterpretation	setChannelIntrepretation	Receive partial (unichannel) interpretation.

Table 29 MI.05. Multi-channel fusion engine main methods

Interface	Method	Description
iAdapt	setCommandInfo	Collects command info.

Table 30 MI.06. MHMI event protocol adapter main methods

8.1.5 MOD.SH.AS-ACTIVE SAFETY

8.1.5.1 FBB Specification

The figure below shows the main components of the MOD.SH.MI.AS, describing both main Functional Building Blocks and information flows.

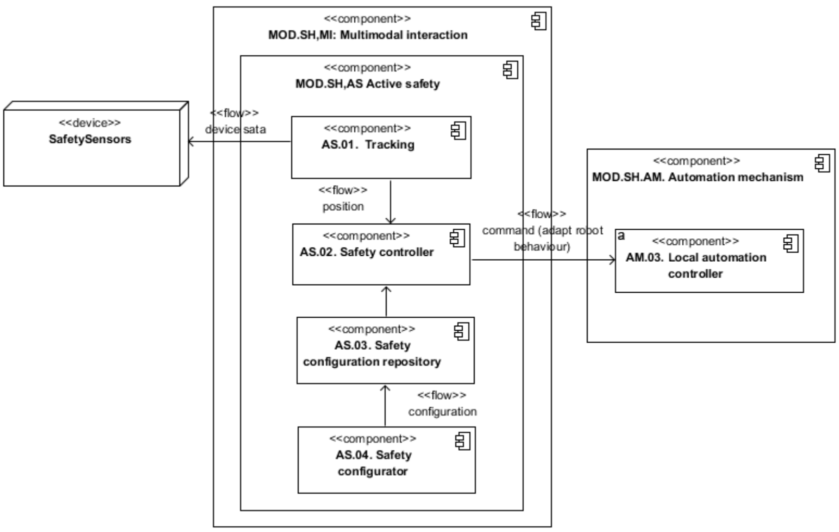


Figure 87 MOD.SH.AS. Decomposition into functional building blocks

8.1.5.1.1 AS.01. Tracking

"AS.01. Tracking" uses the data from sensors such as scanners, vision systems or proximity sensors to identify the position to obstacles and send it to the "AS.02. Safety controller". Tracking can involve human and safety zone tracking.

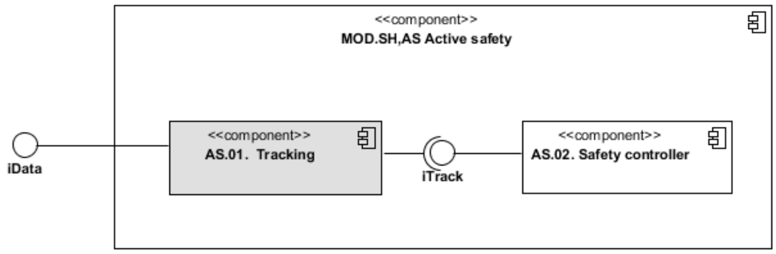


Figure 88 AS.01. Tracking interfaces

AS.01 exposes **iData** interface and consumes the *saveTracking* method of the **iTrack** interface exposed by the "AS.03. Safety controller".

8.1.5.1.2 AS.02. Safety controller

"AS.02. Safety controller" processes the tracking information and decides on how to adapt the robot behaviour considering the safety configuration.

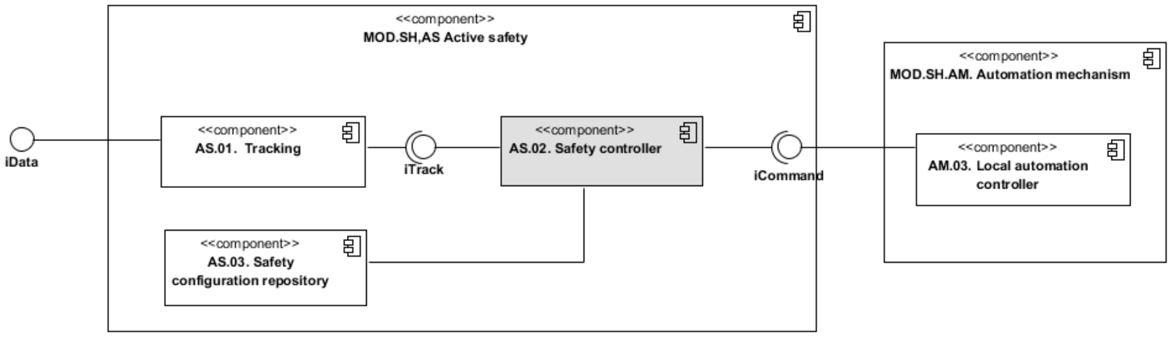


Figure 89 MI.06. MHMI event protocol adapter interfaces

AS.02 exposes **iTrack** interface exposed by the "AS.03. Safety controller" and consumes the methods of the **iCommand** interface exposed by the "AM.03. Local automation controller".

8.1.5.1.3 AS.03. Safety configuration repository

"AS.03. Safety configuration repository" supports the storage of the safety configuration (e.g. safety modes, safety related actions, operator preferred modes, etc.).

8.1.5.1.4 AS.04. Safety configurator

The "AS.04. Safety configurator" enables to define safety settings.

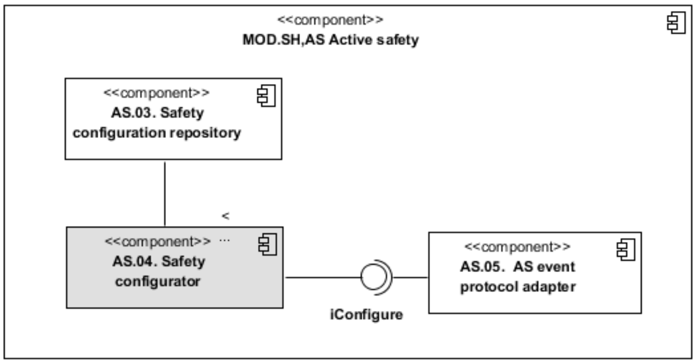


Figure 90 AS.04. Safety configurator interfaces

AS.04 exposes **iConfigure** interface to manage the configuration settings.

8.1.5.2 Background Assets

Trajectory prediction modules, developed by TEK in the scope of internal R&D and the EuroC project[15]. Trajectories can be very useful to take measures to avoid collisions between the elements in the scene, or even to be ready to take avoidance actions in advance. The trajectory prediction software uses long short-term memory (LSTM) recurrent neural networks to predict workers trajectories based on 2D range laser data. LSTMs are a category of recurrent neural networks (RNNs) which belongs to the growing field of deep learning paradigms. RNNs are artificial neural networks in which connections between units form a directed cycle. Due to this architecture, recurrent neural networks possess an internal state that stores information about past inputs. This endows the recurrent networks with the ability of processing sequences of inputs and exhibit a dynamic temporal behaviour in response to those sequences. Training RNNs to learn long-term dependencies by gradient-descent methods has proven to be difficult. LSTMs address this problem introducing gates that control how much of the past and the current state has to get through to the next time step.

8.1.5.3 A4BLUE Enhancements

MOD.SH.AS will take advantage on the above-mentioned background asset to adapt robot behaviour to the predicted trajectory considering context information.

8.1.5.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.SH.AS.

Interface	Method	Description
iData	setTrackingData	Collect the raw data from the safety sensors.

Table 31 AS.01. Main component interfaces

Interface	Method	Description
iData	saveTracking	Save tracking information.

Table 32 AS.02. Main component interfaces

Interface	Method	Description
iConfigure	setUser	Stores the configuration settings in the data repository.
	updateUser	Updates the configuration settings in the data repository.

Table 33 AS.04. Main component interfaces

8.2 ENTERPRISE LAYER

8.2.1 MOD.EN.CAM- COLLABORATIVE ASSET MANAGER

The A4BLUE Collaborative Asset Manager component represents the asset repository in any A4BLUE-based system, encompassing both GUIs and APIs, exploiting well-grounded existing background assets, coming from FIWARE (and then widely used in FITMAN and FIWARE for INDUSTRY as

[15] EUROC: http://www.euroc-project.eu/

described in Section 3.3 - FIWARE for INDUSTRY), namely the FITMAN CAM SE[16] and its evolution developed and federated to the BEinCPPS Future Internet Platform in the BEinCPPS project[17].

8.2.1.1 FBB Specification

The Figure 91 here below shows the main components of the MOD.EN.CAM, describing both main Building Blocks and information flows.

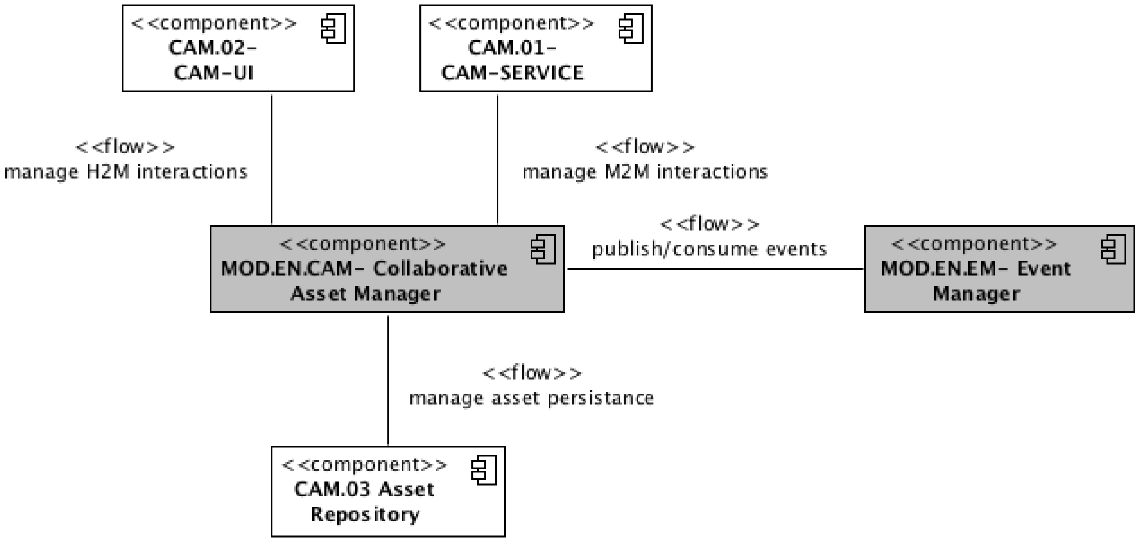


Figure 91 MOD.EN.CAM. Decomposition into Functional Building blocks

8.2.1.1.1 CAM.01. CAM Service

The CAM-Service sub-component is at the core of the MOD.EN.CAM, providing the business logic needed to interact with the asset registry and without providing a user interface. It provides a REST-based abstraction on top of the repository. By means of API calls, applications can query the reference ontology: CRUD operations are allowed on Classes, Models, Assets, Attributes, Relationships and Owners. CAM-Service provides the integration with the MOD.ENG.EM, thanks to the integration with the FIWARE Orion Context Broker.

During the development of the alpha prototypes, new interfaces have been added (to the ones already identified in D2.1) in order to support functional and technical needs emerging from the pilots: iReasonerManager, iUploadRDFManager, iSparqlManager. These interfaces enhance the background component by adding a powerful reasoning capability to the repository, as well as supporting new interactions with the RDF Repository directly through Query SPARQL.

[16] http://catalogue.fitman.atosresearch.eu/enablers/collaborative-asset-management

[17] https://github.com/BEinCPPS/fitman-cam

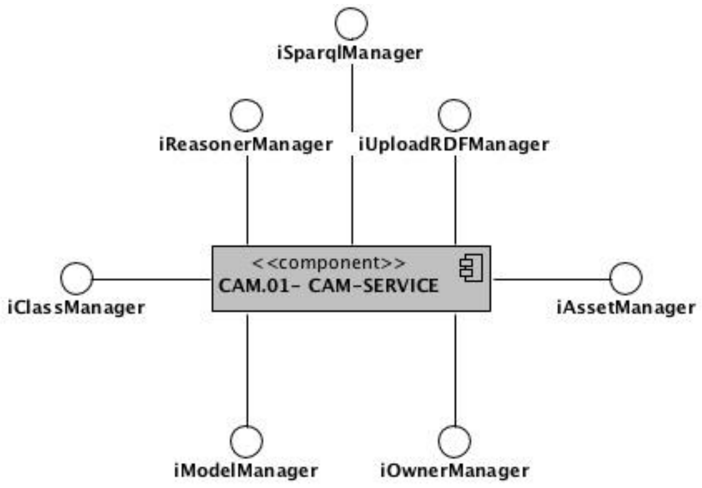


Figure 92 MOD.EN.CAM CAM Service interfaces

8.2.1.1.2 CAM.02. CAM UI

CAM UI is a web application which provides a user interface for exploiting the CAM-Service API interactively. CAM allows user to create, read, update and delete Classes, Models, Assets, Attributes, Relationships and Owners through a web interface.

8.2.1.1.3 CAM.03. Asset Repository

The Asset Repository is the physical storage of the virtualized assets, using the defined VARM ontology, as a basis for its knowledge. Once the ontology is installed, the user can use the graphical interface provided by CAM UI to manage and extend it, for example by adding custom domain concepts, and to create the VAR instances containing Tangible and Intangible Assets of the various Use Cases.

8.2.1.2 Background Assets

FITMAN CAM is the implementation of a virtualized assets (VA) knowledge base, using OWL2-based formal descriptions of any item of interest in the real world that needs to be digitally represented within an A4BLUE-based system. Within the A4BLUE project, the FITMAN CAM can also be linked to the design-time of the platform: its API can be used by engineering tools to access a common online repository of specifications of devices and systems. "FITMAN CAM exposes a web API that can be used to search and retrieve VA definitions from the knowledge base. It is worth to notice that FITMAN CAM is integrated with FIWARE Orion Context Broker (further described in next chapter §8.2.2), exposing VAs as NGSI read-only contexts.

FITMAN CAM is based on the Eclipse RDF4J ontology repository, which leverages open standards like Resource Description Framework (RDF), RDF Query Language (SPARQL) and Web Ontology Language (OWL2). On top of these, FITMAN CAM adds a web service layer exposing a REST API and a web GUI layer providing an interactive, user-friendly front-end.

FITMAN CAM allows applications and web users to define and manage Asset Classes, Asset Models and individual Assets starting from a common domain ontology that can be imported and customized in place. Once created, individual Assets can be optionally exposed as a NGSI context, in order to be integrated into FIWARE-based applications.

This CAM release contains two modules, CAM-Service and CAM:

- **CAM-Service** is a web application with no user interface. It provides a REST-based abstraction on top of the reference ontology that is stored in the RDF4J repository. By means of API calls, applications can query the reference ontology: CRUD operations are allowed on Classes, Models, Assets, Attributes, Relationships and Owners. CAM-Service also implements the integration with the FIWARE Orion Context Broker and with FIWARE security.

- **CAM** is a web application which provides a user interface for exploiting the CAM-Service API interactively. CAM allows user to create, read, update and delete Classes, Models, Assets, Attributes, Relationships and Owners through a web interface.

8.2.1.3 A4BLUE Enhancements

The background asset to be exploited to realize the A4BLUE CAM can be considered an ontology editor with a shared repository of instances of the model used to represent the assets. The actual implementation does not provide a reusable ontology to model the assets, but a new one should be selected and adapted to A4BLUE, and following the outcomes of the project Task T2.3 (i.e. the Virtual Asset Model). To this end, T2.3 has been developing the VARM in order to provide a standard definition of the the virtual representation of tangible or intangible assets (TA/IA).

Further enhancements will target the enrichments of assets representation, also taking into account other information such as worker skills and preferences, as well as the impact of soft or socio-economic factors, such as worker satisfaction, into quantifiable or monetary indicators.

To this end new APIs will be introduced in the CAM-Service, as well as new functionalities would be needed in the UI providing the web access to end users.

8.2.1.4 Interfaces

The following table presents an overview of the main interfaces exposed or used by the CAM-Service sub-system, by providing the main methods and their description. Further details on the foreseen interfaces will provided in the outcomes of WP3 (mainly in D3.4).

Interface	Method	Description
iClassManager	getClasses	Get all classes.
	getClass(className)	Get Class by className.
	createClass	Create Class.
	updateClass(className)	Update Class named className.
	deleteClass(className)	Delete Class named className.
iAssetManager	getAssets	Get all assets.
	getAsset(assetName)	Get Asset by assetName.
	getAssets(className)	Get assets of Class named className.
	getAttributes(assetName)	Get all attributes of assetName.
	getAttribute(assetName, attributeName)	Get Attribute of assetName named attributeName.

Interface	Method	Description
	getRelationships(assetName)	Get all relationships of assetName.
	getRelationship(assetName, relationshipName)	Get Relationship of assetName named relationshipName.
	createAsset	Create Asset.
	createAttribute(assetName)	Create Attribute of Asset named assetName.
	createRelationship(assetName)	Create Relationship of Asset named assetName.
	updateAsset(assetName)	Update Asset named assetName.
	updateAttribute(assetName, attributeName)	Update Attribute of assetName named attributeName.
	updateRelationship(assetName, relationshipName)	Update Relationship of assetName named relationshipName.
	deleteAsset(assetName)	Delete Asset named assetName.
	deleteAttribute(assetName, attributeName)	Delete Attribute of assetName named attributeName.
	deleteRelationship(assetName, relationshipName)	Delete Relationship of assetName named relationshipName.
iModelManager	getModels	Get all models.
	getModel(modelName)	Get Model by modelName.
	getModel(classnName)	Get models of Class named className.
	getAttributes(modelName)	Get all attributes of modelName.
	getAttribute(modelName, attributeName)	Get Attribute of modelName named attributeName.
	getRelationship(modelName)	Get all relationships of modelName.
	getRelationship(modelName, relationshipName)	Get Relationship of modelName named relationshipName

Interface	Method	Description
	createModel	Create Model.
	createAttribute(modelName)	Create Attribute of Model named modelName.
	createRelationship(modelName)	Create Relationship of Model named modelName.
	updateModel(modelName)	Update Model.
	updateAttribute(modelName, attributeName)	Update Attribute of modelName named attributeName.
	updateRelationship(modelName, relationshipName)	Update Relationship of modelName named relationshipName.
	deleteModel(modelName)	Delete Model.
	deleteAttribute(modelName, attributeName)	Delete Attribute of modelName named attributeName.
	deleteRelationship(modelName, relationshipName)	Delete Relationship of modelName named relationshipName.
iOwnerManager	getOwners	Get all owners.
	getOwner(ownerName)	Get Owner by ownerName.
	createOwner	Create Owner.
	updateOwner(ownerName)	Update Owner named ownerName.
	deleteOwner(ownerName)	Delete Owner named ownerName.
iReasonerManager	ceateOntologyModel(DL_LANG)	Create a Ontology Model
	createInfModel(reasoned, model)	Create Inference Model.
	executeInferenceQuery(sparqlQuery)	Execute an Inference Query on Inference Model
iUploadRDFManager	addRdfFileToRepo(rdfFile)	Add an RDF File to RDF4J REPO
iSparqlManager	sparqlQuery(sparqlQuery)	Execute a SPARQL Query on REPO

Table 34 MOD.EN.CAM. Main CAM-Service component interfaces

8.2.2 MOD.EN.EM- EVENT MANAGER

The A4BLUE Event Manager component will be well-grounded upon existing background assets, coming from FIWARE (and then widely used in FITMAN as described in Section 3.3 - FIWARE for INDUSTRY): FIWARE Orion Context Broker[18] and FIWARE Proactive GE[19].

8.2.2.1 FBB Specification

The Figure 93 here below shows the main components of the MOD.EN.EM, describing both main Building Blocks and information flows.

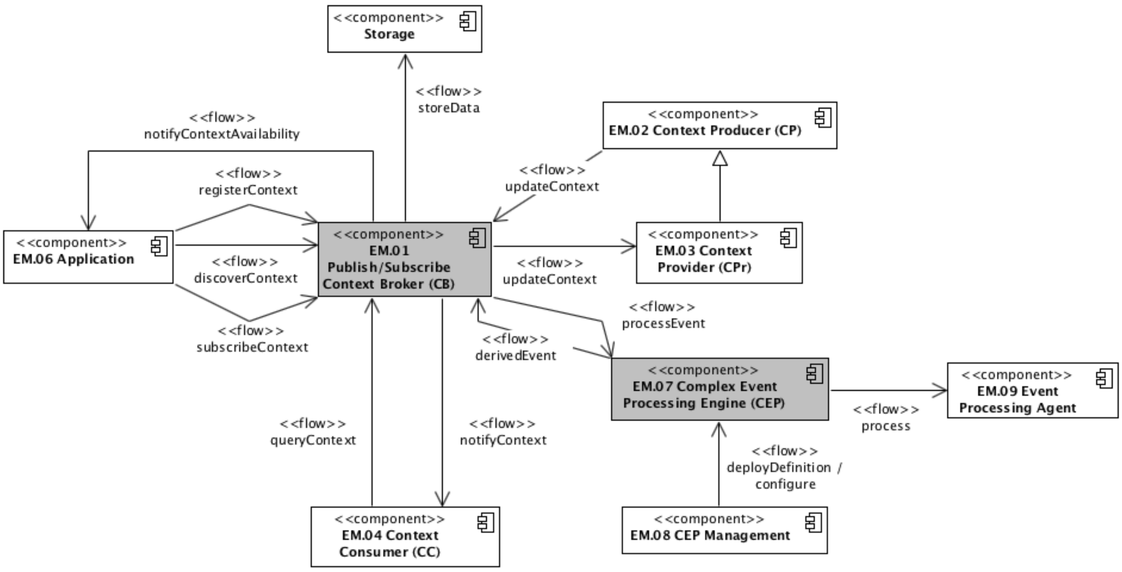


Figure 93 MOD.EN.EM. Decomposition into Functional Building blocks

8.2.2.1.1 EM.01. Publish/Subscribe Context Broker (CB)

The Publish/Subscribe Context Broker (CB) is the main actor of the MOD.EN.EM component. It works as a handler and aggregator of context data and as an interface between architecture actors. Primarily the CB has to control context flow among all attached actors; in order to do that, the CB has to know every Context Provider (CP) in the architecture; this feature is done through an announcement process. Typically, the CB provides a Context Provider Lookup Service and a Context Persistence Service.

[18] https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[19] https://catalogue.fiware.org/enablers/complex-event-processing-cep-proactive-technology-online

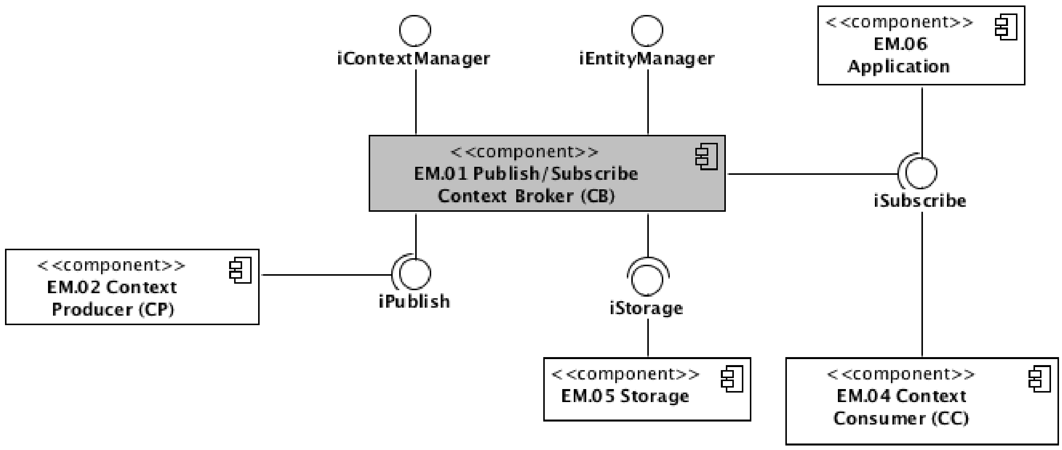


Figure 94 MOD.EN.EM. Context Broker interfaces

8.2.2.1.2 EM.02. Context Producer (CP)

A Context Producer (CP) is an actor able to generate context. The basic Context Producer is the one that spontaneously updates context information, about one or more context attributes according to its internal logic. This communication is between CS and CB is in push mode, from the CP to the CB. Context Producers can work also in pull base way, in which case they are referred as Context Providers.

8.2.2.1.3 EM.03. Context Provider (CPr)

A Context Provider (CPr) is a specialized kind of Context Producer actor, which provides context information on demand, in synchronous mode; that means that the Publish/Subscribe Context Broker or even the Context Consumer can invoke the CP in order to acquire context information. A CP provides context data only further to a specific invocation. Moreover, a CP can produce new context information inferred from the computation of input parameters; hence it is many times responsible for reasoning on high-level context information and for sensor data fusion. Every CP registers its availability and capabilities by sending appropriate announcements to the CB and exposes interfaces to provide context information to the CB and to Context Consumers.

8.2.2.1.4 EM.04. Context Consumer (CC)

A Context Consumer (CC) is an entity (e.g. a context based application) that exploits context information. A CC can retrieve context information, send a request to the CB or directly invoke a CP over a specific interface. Another way for the CC to obtain information is by subscribing to context information updates that match certain conditions (e.g., are related to certain set of entities). The CC registers a call-back operation with the subscription for the purpose, so the CB notifies the CC about relevant updates on the context by invoking this call-back function. Finally, some kinds of Context Consumer may expose updated context operations to be invoked by CB. This is mainly related with actuation capabilities, i.e. the update at Context Consumer produces a given actuation, e.g. turn on/off a lamp.

8.2.2.1.5 EM.05. Storage

Context information received by the Publish/Subscribe Context Broker (from a Context Source or as a result of a request to a Context Provider) can be stored in a context database. If another Context Consumer requests the same context information to the Publish/Subscribe Context Broker, it can be retrieved from the database. Publish/Subscribe Context Broker persistence is only for the current context snapshot (i.e. only the current status) of the system. In other words, Context Broker doesn't implement context history persistence. However, it can be achieved using a Context Consumer in charge of such persistence in any desired persistence backend.

8.2.2.1.6 EM.06. Application

Applications or even other components in the A4BLUE platform may play the role of Context Producers, Context Consumers or both. Every application can register new contexts, discover existing contexts, and subscribe to context updates (and then receive context change information).

8.2.2.1.7 EM.07. Complex Event Processing (CEP)

The Complex Event Processing (CEP) engine analyses event data in real-time, generates immediate insight and enables instant response to changing conditions. Fundamental capabilities of every CEP include event-based routing, observation, monitoring and event correlation.

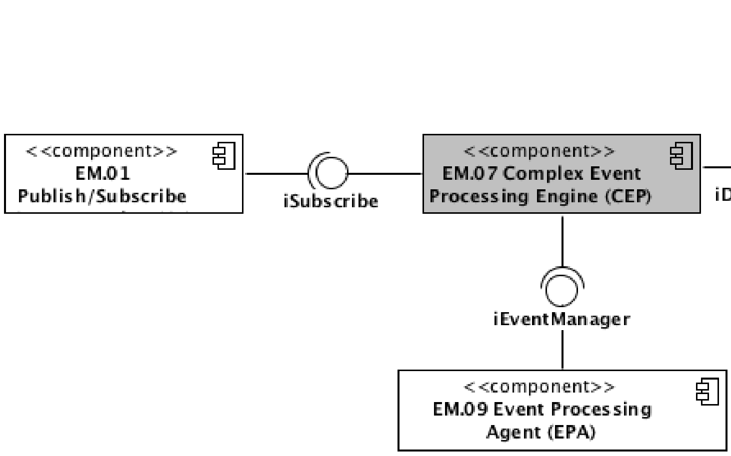


Figure 95 MOD.EN.EM. Complex Event Processing interfaces

8.2.2.1.8 EM.08 CEP Management

The CEP Management provides capabilities to manage the CEP definition repository available to the CEP engine instances at run time. These services allow putting a new definition file to the repository, getting a specific definition from the repository, updating a repository definition file or deleting a definition from the repository. In addition, there are services that allow controlling the CEP engine instances at run time (e.g. starting and stopping a CEP engine instance, updating CEP engine instance definitions and reading the state of the CEP engine instance).

8.2.2.1.9 EM.09. Event Processing Agent (EPA)

The Event Processing Agent (EPA) is an abstraction useful to allow for a flexible allocation of processing power in physical computing nodes as the entire event processing application can be executed as a single runtime artifact, or as multiple runtime artifacts according to the individual agents that make up a processing network. Therefore, the information flow is described as events originating at event producers and flowing through various event processing agents to eventually reach event consumers. Thus scalability, performance and optimization requirements may be addressed by design.

The EPAs and their assembly into a network is where most of the functions of CEP are implemented. The behaviour of an EPA is specified using a rule-oriented approach that is inspired by the ECA (Event-Condition-Action) concept and may better be described as Pattern-Condition-Action. Rules will consist of three parts:

- A pattern detection that makes a rule of relevance. A pattern is associated with event processing context
- A set of conditions (logical tests) formulated on events
- A set of actions to be carried out when all the established conditions are satisfied

It is worth noting that every pattern is associated with an Event processing context. Event processing contexts group event instances so that they can be processed in a related way. They assign each event instance to one or more processing context partitions. An event processing context can be a temporal processing context, a segmentation processing context, or a composite context that is to say one made up of other processing context specifications.

8.2.2.1.1 EM.10. Status Manager

The Status Manager is in charge of update the process status and keep historical records of all the events in the "CAM 03. Asset repository".

8.2.2.2 Background Assets

FIWARE Orion Context Broker (OCB) is the reference implementation of FIWARE's Publish/Subscribe Context Broker Generic Enabler Open Specification. You can register context producer devices/applications which will provide updates to context information, as well as context consumer devices/applications which will be notified of updates and query producers." FIWARE Orion Context Broker plays the role of a central hub for information exchange within A4BLUE-based systems, and as such it is integrated with several other components of the A4BLUE platform as shown previously in Figure 49.

The Orion Context Broker stores context information updated from applications, so queries are resolved based on that information. To make persistent context status changes in OCB, it can be connected to other FIWARE components such as Cygnus[20]. Cygnus implements a connector for context data coming from Orion Context Broker and aimed to be stored in a specific persistent storage, such as HDFS, CKAN or MySQL.

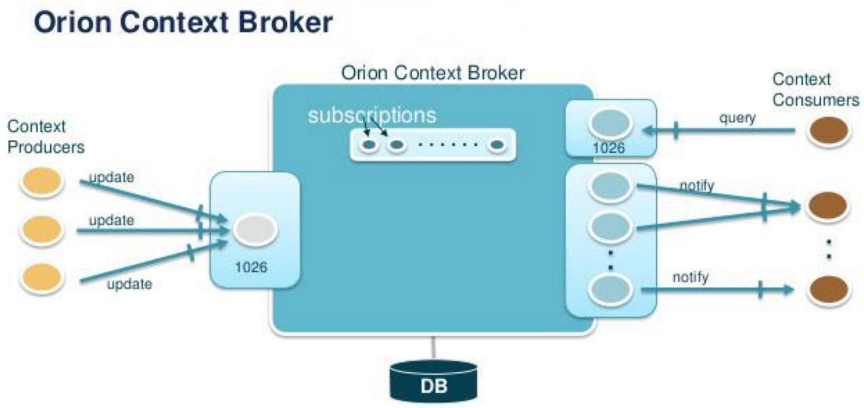


Figure 96 MOD.EN.EM. FIWARE Orion Context Broker

On the other hand, the Event Manager can be complemented by a Complex Event Processing (CEP) engine, able to analyse event data in real-time, generate immediate insight and enable instant response to changing conditions. While standard reactive applications are based on reactions to single events, the CEP reacts to situations rather than to single events. A situation is a condition that

[20] https://github.com/telefonicaid/fiware-cygnus

is based on a series of events that have occurred within a dynamic time window called processing context.

While standard reactive applications are based on reactions to single events, the CEP GE reacts to situations rather than to single events. The CEP, in fact, allows to detect patterns above contexts (triggering some action or raising some alarm), receives contexts information as input events and generates observations as output events. In certain scenarios, single events can be insignificant. A CEP engine can detect combinations of events which are meaningful, through use and detection of patterns over incoming events, and by letting other tools respond to single events with particular conditions.

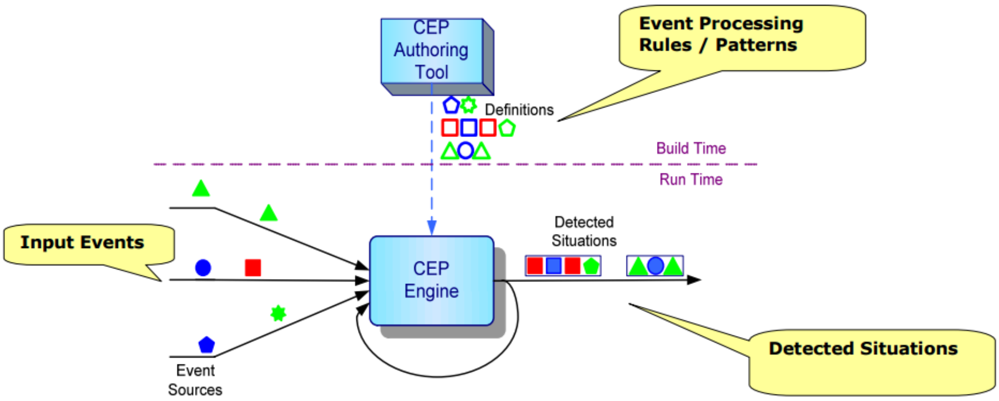


Figure 97 MOD.EN.EM. CEP execution model

D2.1 presented the FIWARE Proactive GE as the background component to be exploited in order to provide the engine processing logic. Unfortunatelly the GE owner is no longer supporting this module, there the A4BLUE Consortium has analysed several potential alternatives from the FIWARE ecosystem. This study has been closed when a new background has been selected: the Perseo CEP [21], taking in careful consideration all the features supported in order to ease its integration with the rest of the A4BLUE RA.

Perseo CEP is composed of two fundamental components: *perseo fe* (front-end) and *perseo-core* (back-end): the “front-end” CEP processes incoming events and rules, being responsible for their momentum, and for the implementation of the actions. The “back-end” CEP represents, instead, the rules processing engine able to check inbound events against rules in Event Processing Language (EPL) and invokes perseo-fe if an action has to be performed. The persistence of the rules is in memory, so it has no constant memory; the whole set of rules comes updated periodically by “perseo fe”.

So far Perseo CEP is completely integrated with FIWARE Context Broker, i.e. the source of information to be processed. The CEP uses the mechanism of Orion inscriptions to examine entity flows; the Rules which satisfies the EPL queries, analyzing the changes in status that occur in the Context Broker by triggering the corresponding actions. Perseo provides different channels of triggere action via a notification mechanism, as shown in Figure 98.

[21] https://github.com/telefonicaid/perseo-fe

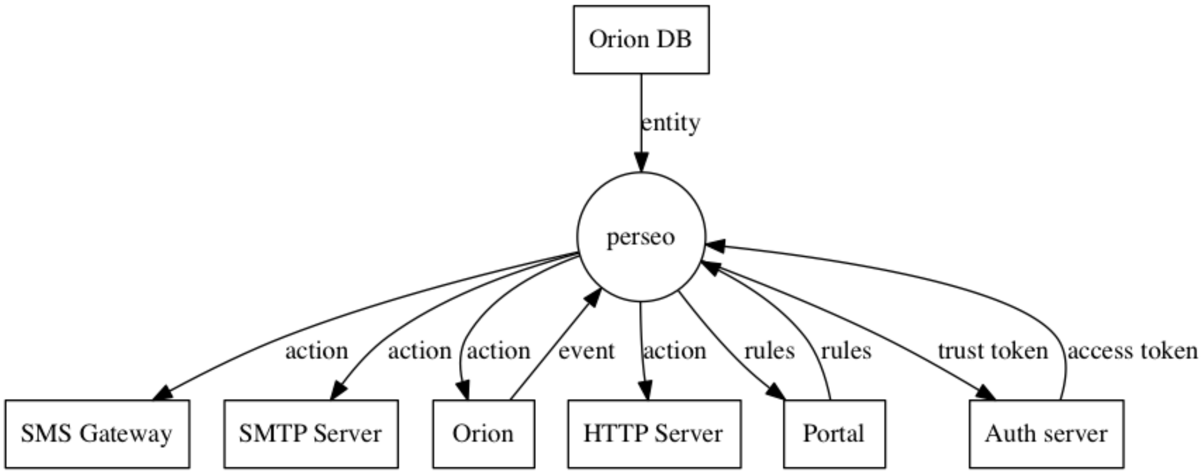


Figure 98 MOD.EN.EM. Perseo CEP – External Data Flow

8.2.2.3 A4BLUE Enhancements

The main goal of the A4BLUE Event Manager component is to establish a robust event-based adaptation manager, based on cutting edge technologies coming from the FIWARE ecosystem, and trying to ease the deploy and configuration of such complex systems (e.g. through the adoption of supporting tools to manage adaptation rules definition).

A4BLUE-based systems will considerably rely on the mentioned background assets, especially since they have a proved reliability and a wide adoption in the FIWARE ecosystem being part of the core FIWARE architecture for Data/Context Management. Furthermore, even if well-grounded assets have been selected, A4BLUE will also continuously explore new emergent solutions into the FIWARE ecosystem providing the same capabilities emerging from the FIWARE for Industry initiatives (e.g. the IoT Data Edge Consolidation GE, a recent implementation designed to provide a common access in real time to all data, for any kind of sensors and "Things", and allowing persistent exchange of information using a lightweight storage system).

8.2.2.4 Interfaces

The following table presents an overview of the main interfaces exposed or used by the Context Broker sub-system, by providing the main methods and their description. Further details on the foreseen interfaces will provided in the outcomes of WP5 (mainly in D5.1).

Interface	Method	Description
iContextManager (using NGSI APIv1)	registerContext	Register a new context in the broker.
	discoverContextAvailability	Discover wherever a specific context exists in the broker.
	updateContext	Update a specific context in the broker.
	queryContext	Retrieve context information from the broker.
iEntityManager	listEntities	Retrieve the list of entities from the broker.

Interface	Method	Description
(using NGSI APIv2)	createEntity	Create a new entity in the broker.
	retrieveEntity	Update the attributes of a specific entity in the broker.
	retrieveEntityAttributes	Retrieve the model of a specific questionnaire from the data repository.
	updateEntityAttributes	Update the attributes of a specific entity in the broker.
	removeEntity	Delete a specific entity from the broker.
iPublish	updateContext	Provide an endpoint to receive the information of a newly created event.
iSubscribe	notifyContext	Provide an endpoint to receive the information of the subscribed events.
iStorage	storeData	Store the context status change information in the persistent repository.
iUpdate	updateStatus	Update process status
	setHistoryData	Saves historical data

Table 35 MOD.EN.EM. Main CB component interfaces

The following table presents an overview of the main interfaces exposed or used by the Complex Event Processing sub-system, by providing the main methods and their description. Further details on the foreseen interfaces will provided in the outcomes of WP5 (mainly in D5.1).

Interface	Method	Description
iInstanceManager	retrieveInstanceStatus	Retrieve the status of an instance, the definition URI it is configured with and its state (stopped or started).
	startInstance	Start the instance.
	stopInstance	Stop the instance.
	configureInstance	Configure the instance with a definition file.
iDefinitionManager	listDefinitions	Retrieve all the existing definitions in the repository.

Interface	Method	Description
	addDefinition	Add a new definition in the repository.
	retrieveDefinition	Retrieve the complete definition in JSON format.
	updateDefinition	Replace content of an existing definition with new content.
	deleteDefinition	Remove the definition from the repository.
iEventManager	pushEvents	Push new events into the engine.
	pullEvents	Pull new events to feed the engine.
	sendEvent	Send a derived event to a consumer.
iSubscribe	notifyContext	Provide an endpoint to receive the information of the subscribed events.

Table 36 MOD.EN.EM. Main CEP component interfaces

8.2.3 MOD.EN.ARG- VR/AR BASED TRAINING AND GUIDANCE

The VR/AR training system will be composed by two main components: a front-end app (ARG.01) that will run on an Augmented Reality HMD device and a back-end application (ARG.02) that will act as a bridge between the AR app and the underlying A4BLUE framework layers.

In this section, more details about the ARG02-A4BLUE link will be provided.

8.2.3.1 ARG.01 Resource Manager

The Back-End application has the constant need to be synchronized with the assets, events and resources of the underlying A4BLUE framework layers. For this reason, it is necessary to have the possibility to:

- Retrieve any asset that may be useful in the AR context by a unique name;
- Retrieve information about the assets organization (i.e. directory structure, assets existence etc.);
- Retrieve information about assets' last update (i.e. version number or last modification date) without downloading the actual asset.

For these reasons, the Resource Manager component will be highly coupled with the Collaborative Asset Manager Service. The communication between the two components will happen through REST synchronous calls.

It is worth noting that the ARG.01 component will use the iAssetManager interface exposed by the CAM.01 CA-Service component; the Resource Manager will use it to perform operations on other A4BLUE assets.

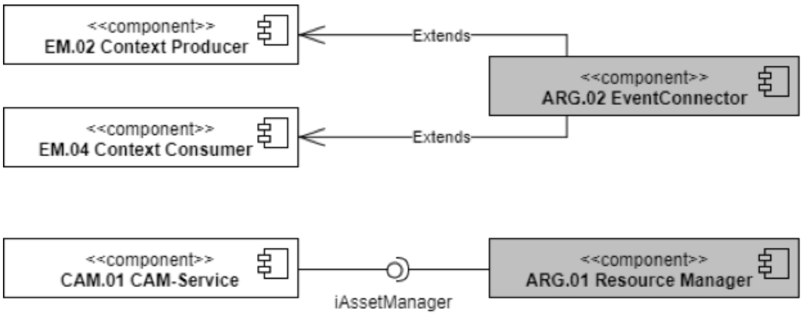


Figure 99 MOD.EN.ARG. VR/AR Training System Back-end module interactions

8.2.3.2 ARG.02 Event Connector

The Event Connector is an ARG02 module that will directly communicate with the MOD.EN.EM component in order to manage every event that could be produced or consumed during the training session. In other words, this component will act both as a Context Producer and Context Consumer.

As a Context Producer, the Event Connector will register a new context and update it with new information as soon as it will be created in the operator training/guidance.

On the other hand, as a Context Consumer, the Event Connector will be able to subscribe to specific contexts in order to be notified about new events. The Event Connector will also be able to directly query the EM Context Broker to receive the contexts' current status.

8.2.4 MOD.EN.IM- IDENTITY MANAGEMENT

Going towards the implementation of the A4BLUE solution, and following the parallel work undertaken in task T2.4 "Security risk assessment and secure middleware definition", this section will introduce the key characteristics of the main security component, even if such a component was not introduced in the Reference Implementation described in the DoA (as defined in Section 2.1).

Identity Management covers a number of aspects involving users' access to networks, services and applications, including secure and private authentication from users to devices, networks and services, authorization & trust management, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) to service domains and Identity Federation towards applications.

The A4BLUE Identity Management component will be well-grounded upon existing background assets, coming from FIWARE (and then widely used in FITMAN as described in Section 3.3 - FIWARE for INDUSTRY): FIWARE Identity Management - KeyRock [22].

8.2.4.1 FBB Specification

The Figure 93 here below shows the main components of the MOD.EN.IM, describing both main Building Blocks and information flows.

[22] https://catalogue.fiware.org/enablers/identity-management-keyrock

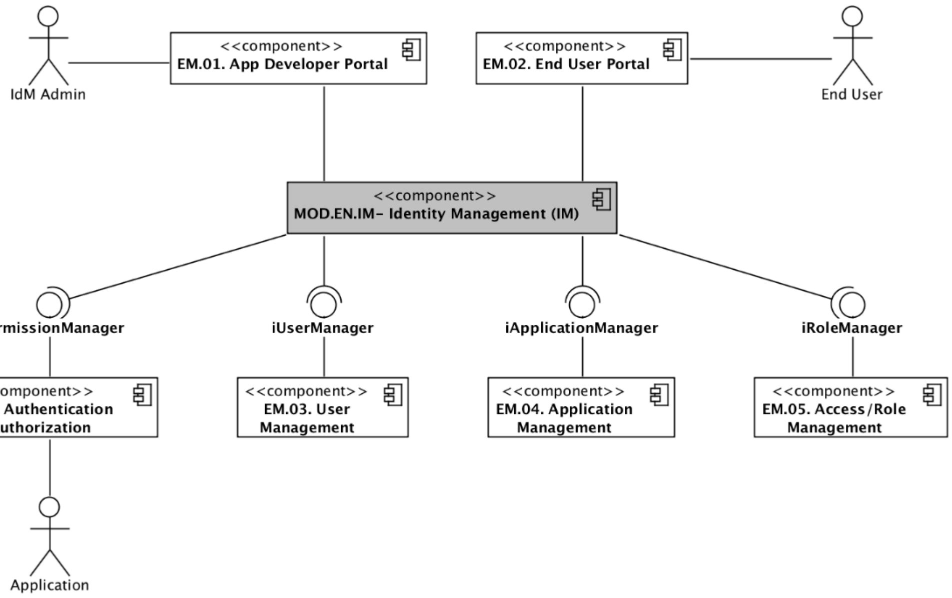


Figure 100 MOD.EN.IM. Decomposition into Functional Building blocks

8.2.4.1.1 IM.01. Application Developer Portal

The Application Developer Portal is where developers can register and manage their applications, especially the client applications, including the application credentials. With such credentials, the application is able to authenticate to the IdM and participate in an authentication and authorization process to get access to a protected service. The developer can also manage access for his application, and in particular, define application-specific roles.

8.2.4.1.2 IM.02. End User portal

This is where end users self-register in the IdM with email address, password, etc. This is typically implemented as a Web user interface. End users may also review and modify their personal account data and maintain their privacy settings using this portal.

8.2.4.1.3 IM.03. User Management

Provides a REST API to create user accounts, retrieve and modify user attributes, delete user accounts. The user management API is typically used by web applications (or any kind of service provider), to retrieve extra information about their users.

8.2.4.1.4 IM.04. Application Management

REST API for managing applications (registering the application, retrieving and modifying application data such as credentials, deleting the application).

8.2.4.1.5 IM.05. Access Management

REST API to manage roles globally or for a specific application. There are two aspects of role management involved here: defining the role permissions and assigning the roles to the users. The role permissions make up an authorization policy that may be pushed to the Authorization PDP GE[23].

[23] https://catalogue.fiware.org/enablers/authorization-pdp-authzforce

8.2.4.1.6 IM.06. Authentication and Authorization

This is the core component of the IdM, supporting the two most common security standards in web-based applications: SAML2.0[24] and OAuth2[25].

SAML 2.0 provides federated identity, more specifically federated Single Sign-On and user attribute exchange between IdM systems (FIWARE IdM GE and other Identity Providers).

The OAuth standard is an authentication and authorization framework that addresses the core scenario where you have to allow a website or application (Consumer) to access protected resources of an End User from a web service (Service Provider) via an API, without requiring this End User to disclose their Service Provider credentials to the Consumer.

8.2.4.2 Background Assets

Identity Management is key on any architecture. A4BLUE IdM will rely on the FIWARE "Identity Management – KeyRock" GE to provide the support and extend its functionalities.

Identity Manager (IdM) GE API specifications comply with existing standards for authentication and user and provide access information. IdM offers tools for administrators to support the handling of user life-cycle functions. It reduces the effort for account creation and management, as it supports the enforcement of policies and procedures for user registration, user profile management and the modification of user accounts. Administrators can quickly configure customized pages for the inclusion of different authentication providers, registration of tenant applications with access to user profile data and the handling of error notifications. For end users, the IdM provides a convenient solution for registering with applications since it gives them a means to re-use attributes like address, email or others, thus allowing an easy and convenient management of profile information. Users and administrators can rely on standardized solutions to allow user self-service features. As it is possible to configure several applications that shall be linked to his IdM, the main benefit for users is a single sign-on (SSO) to all these applications. The IdM offers hosted user profile storage with specific user profile attributes. Applications do not have to run and manage their own persistent user data storages, but instead, can use the IdM user profile storage as a Software as a Service (SaaS) offering.

The IdM is composed of two independent components, a RESTful back-end and web front-end. If you want to see the code for each of the components of the IdM and more specific documentation please head to each component's repository:

- Horizon based front-end ging/horizon
- Keyrock, a Keystone based back-end ging/keystone

8.2.4.3 A4BLUE Enhancements

The main goal of the A4BLUE Identity Manager component is to establish a robust and secure environment where distributed application can trust both users and applications consuming its own services.

The A4BLUE solution will be based on cutting edge technologies coming from the FIWARE ecosystem, and trying to ease the deploy and configuration of such complex systems even in systems not fully based on FIWARE. A4BLUE-based systems will considerably rely on the mentioned background assets, especially since they have a proved reliability and a wide adoption in the FIWARE ecosystem being part of the core FIWARE architecture for Security[26].

[24] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#samlv20

[25] https://oauth.net/

[26] https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Security_Architecture

8.2.4.4 Interfaces

The following table presents an overview of the main interfaces exposed or used by the A4BLUE IM component, by providing the main methods and their description. Further details on the foreseen interfaces will provided in D2.8 ("Secure Middleware definition for EDA- SOA Integration - Final Version").

Interface	Method	Description
iRoleManager	List Roles	Get all entries.
	Read Role details	Get a specific entry.
	Update a Role	Update a specific entry.
	Create a Role	Add a new role.
	Delete a Role	Delete a specific role.
iApplicationManager	List Applications	Get all applications.
	Create an Application	Add a new application.
	Read Application details	Get a specific application.
	Update an Application	Update a specific application.
	Delete an Application	Delete a specific application.
iPermissionManager	List Permissions	Get all permissions.
	Create a Permission	Add a new permission.
	Read Permission details	Get details of a specific permission.
	Update a Permission	Update a specific permission.
	Delete a Permission	Delete a specific permission.
iUserManager	List Users	Get all users.
	Create a User	Add a new user.
	User	Get a specific user.
	Update a User	Update a specific user.
	Delete a User	Delete a specific user.

Table 37 MOD.EN.IM. Main platform interfaces

8.3 BUSINESS LAYER

8.3.1 MOD.BU.KM- COLLABORATIVE KNOWLEDGE PLATFORM

8.3.1.1 FBB Specification

The Figure 101 here below shows the main components of the MOD.BU.KM, describing both main Building Blocks and information flows.

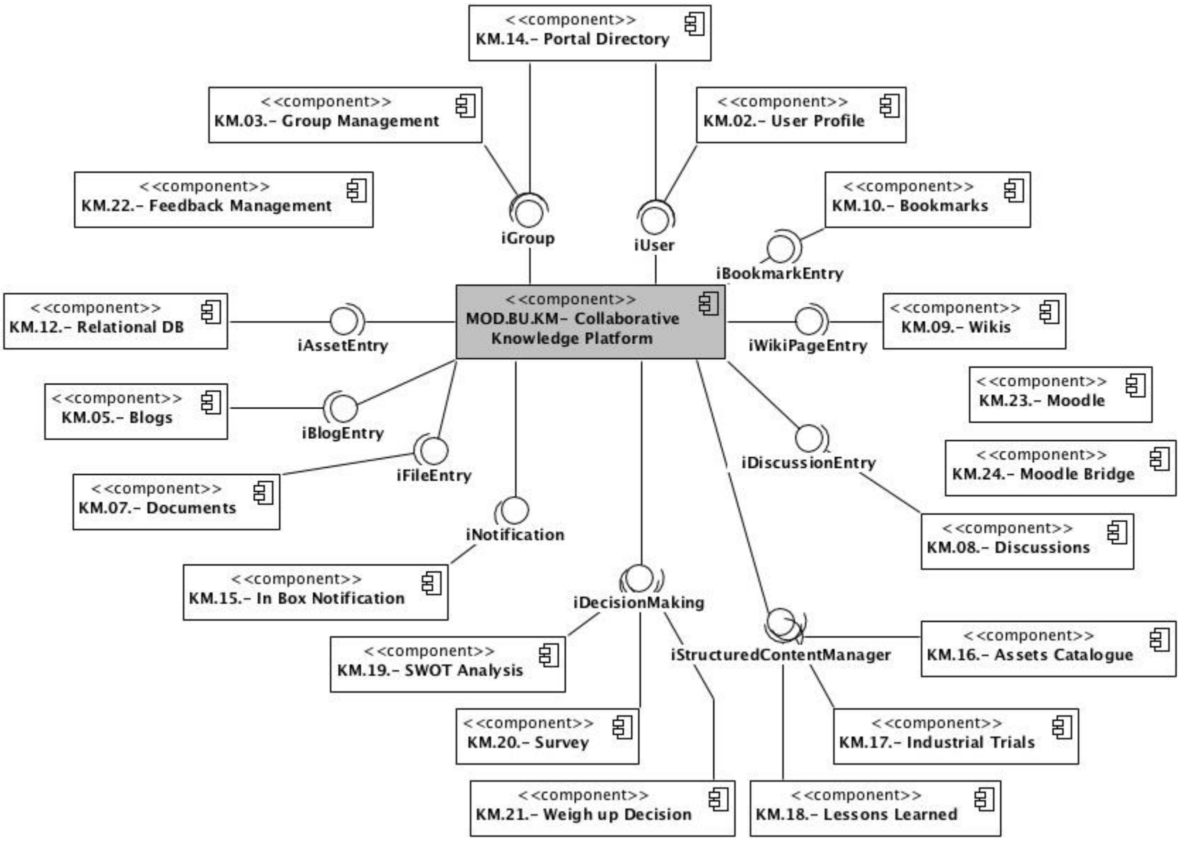


Figure 101 MOD.BU.KM. Decomposition into Functional Building blocks

8.3.1.1.1 KM.01.- Workspaces List

Workspaces List is an application providing the list of workspaces, which are working areas where users share the knowledge, participate to discussions and decisions, annotate resources and contents to find better and follow the live evolution of the knowledge.

8.3.1.1.2 KM. 02.- User Profile

User Profile is a service by which a user of a platform can know colleagues, find experts and collaborators. User profile shows information about a user with respect to activities performed, reputation, skills, following and followers, latest contents, subscriptions, similar people and suggested contents.

8.3.1.1.3 KM. 03.- Group Management

The Group Management application aims to provide the full list of workspace participants. For each member, it is possible to start actions like getting the view of individual profiles and starting to follow. This application shows all the teams of which they are a membership and enables them to create new teams and circles.

8.3.1.1.4 KM. 04.- Universal Search

The MOD.BU.KM includes several search features to help a user quickly find what they are looking for, or just to browse through content, people, and teams.

Universal Search manages interactive search by inserting filters on the search bar. The users can search for specific words and simply enter the search terms to see content containing all the specified words in any order. They can find resources by person, title and content also benefit from the suggestions shown in real-time in the search box when typing the query if they decide to search for a resource from a term. Furthermore, it also shows a set of tips to enrich the query that take into account the user's activities in the community.

8.3.1.1.5 KM. 05.- Blogs

Blog is a service that allows members of a group to share thoughts and ideas through articles. They are also an excellent way to bring out the widespread tacit knowledge in the organization on strategic issues. Currently, corporate blogs are used for internal communications about general issues, events, new product launches. Interesting then are the possible uses of blogs for training, for example in support of learning and self-empowerment paths.

8.3.1.1.6 KM. 06.- Shared Calendar

Shared Calendar is a service that allows you to define shared calendars at community level, in addition to staff, allowing the management of events through the incorporation, the association of a date of beginning and end, the assignment of a lifetime, an indication of a location, the tag association and sending invitations and reminders. The application allows you to view events by day, week, month and year.

8.3.1.1.7 KM. 07.- Documents

Documents supports document management among members of the group that can upload files, view and download those ones of the others members. Offline access can be ease using WebDav clients or using a Dropbox-like local folder synchronization mechanism.

8.3.1.1.8 KM. 08.- Discussions

Discussion is a service to manage online discussions on topics of community interest. Discussion allows to put questions to get qualified answers from colleagues and internal experts, develop new ideas, discuss the pros and cons of new products and services.

8.3.1.1.9 KM. 09.- Wikis

Wiki service allows users to enter and edit in real time the content of the pages they face. The usage of the wiki will allow the members of the platform to consolidate emerging knowledge from users in order to share a common point of view on media and convergence information

8.3.1.1.10 KM. 10.- Bookmarks

Bookmarks is a service that allow users to store and share Internet bookmarks. They can be organized and categorized using the same collaboration process as for the other applications.

8.3.1.1.11 KM. 11.- Usage Analytics

The Usage Analytics application lets you visualize statistics on how users are working within the platform, who is most active, the workspace producing more knowledge and so on. Intuitive and customizable charts can be used by the administrator to select the KPI to monitor and their progress over time.

8.3.1.1.12 KM. 12.- Relational DB

Structured knowledge, containing mainly User Generated Content, is made persistent by using a classical relational database, in order to ease the management along the overall platform taking advantages of well-known data storage solutions.

8.3.1.1.13 KM. 13.- NoSQL DB

Indexing and fast search are implemented using NoSQL engines in order to fit the requirements of performance and semantic reasoning needed to organize existing knowledge and support the Universal Search service.

8.3.1.1.14 KM.14.- Portal Directory

The Portal Directory application allows users to search for other users and organizations, looking them up from the registered entities of the portal, and retrieving the full descriptions and details. The most important characteristcs of users are their skills, while organizations are organized by competencies.

8.3.1.1.1 KM.15.- In Box Notification

This service is part of the core functionalities of the KM Platform, aiming at managing web notifications and showing the main activities realized within the platform using a user centric approach. It is possible to set up what things you would like to be notified about and where those notifications are sent (e.g. using web or email).

8.3.1.1.2 KM.16.- Assets Catalogue

It simplifies the access to a catalogue of the main technologies adopted by the main Organization targeted for different attributes. Attributes include “product/service” characteristics, user manual, related documentation, availability conditions of the assets, best practices, software license used, and contact information.

8.3.1.1.3 KM.17.- Industrial Trials

A database containing a collection of interesting experiments developed both interanally but also externally to the main Organization, detailing business goals, obtained results, used technologies, actors involved and implementations characteristics.

8.3.1.1.4 KM.18.- Lessons Learned

Lessons learned system is a key component of knowledge management, and a primary driver for continuous performance improvement. This application provies a collection of lessons learned from contributors across the main Organization and other partners.

8.3.1.1.5 KM.19.- SWOT Analysis

A service to encourage each member of the group to consider other points of view for a well-balanced decision. This application provides a set of tools to collaborative build a SWOT analysis, using both resources from the KM Platform as well as other user generated items.

8.3.1.1.6 KM.20.- Survey

This application supports a group of people to evaluate an opportunity, providing a converging rating system and a comparison engine intended to provide the best choice among the evaluated items.

8.3.1.1.7 KM.21.- Weigh up Decision

This easy-to-use application prvide a quick collaborative decision making tool, intended to highlight PROs (strengths) and CONs (weaknesses) between different alternatives, trying to build a quick converging path toward a shared choice.

8.3.1.1.8 KM.22.- Feedback Management

This application allows to collect and centrally manage feedbacks and data from their employees, especially related to work order or shift. The component transforms feedback into actionable information and enables the distribution of that information throughout the main Organization. Data can easily be filtered by multiple criteria (e.g. work order, SOI, …). Dedicated notification mechanisms could be used to send the right information to the right operator in a timely and context-aware manner (e.g. using the AR device developed in MOD.EN.ARG component).

8.3.1.1.9 KM.23.- Moodle

Moodle Learning Management System is a flexible, open source, and free to download learning management solution able to provide online learning personalized environments. This system will be integrated with the main KM Platform in order to provide a complete environment intended to provide a complete support to formal up- and re- skilling training courses.

8.3.1.1.10 KM.24.-Moodle Bridge

This application realizes a bridge between the KM core Platform and Moodle, offering three main functionalities:

- Courses: a catalogue organized by categories and shared in two sections, one containing the courses you are CURRENTLY enrolled, a second containing the suggested courses based on skills profiling and matching;
- Badges: a digital badge is essentially an online recognition of your achievements and skills. With the ability to track a recipient's communities of online interaction, a Moodle badge will show the work completed, and the outcomes learned to achieve said badge;
- Activities: a chronological ordered view upon configurable Moodle activities streams available intended to provide the highlights of the activities completed within the LMS system.

8.3.1.2 Background Assets

OPENNESS (OPEN Networked Enterprise Social Software) is a Web-based platform (enriched with a suite of mobile applications) supporting the development and management of collective knowledge and intelligence and collaborative working relying on social networking and semantic-based management of content resources and information.

The platform and its services support and stimulate the usage and creation of knowledge (often implicitly) which exists within social networks (both internal and external), with the aim of promoting open innovation-driven processes, and improving problem solving and decision making in groups and networks. The "wisdom of the crowds" and the collective intelligence emerging from the collaboration of multiple actors become the tools to perform activities ranging from routine processes to strategic decisions, innovation, and change management.

During the development of OPENNESS suite, ENGINEERING has designed to also integrate in the suite a Learning Management System leveraging on **Moodle**, i.e. a learning platform designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalised learning environments. Merging formal (in Moodle) and informal (in OPENNESS) approaches has been validated by the delivery of a powerful set of learner-centric tools and collaborative learning environments that empower both teaching and learning.

8.3.1.3 A4BLUE Enhancements

OPENNESS platform will be extended in the scope of the A4BLUE project in order to better assist blue and white collar workers in Collaborative Decision Making (CDM) processes, adding new services to handle conflict resolutions and to support knowledge take up from the workers, in order to feed a lesson learned database built upon structured and unstructured knowledge. A special focus will be given to the feedback management (collection and provisioning) and to informal/formal learning).

8.3.1.4 Interfaces

The following table presents an overview of the main interfaces exposed or used by the KM-Service sub-system, by providing the main methods and their description. Further details on the foreseen interfaces will provided in the outcomes of WP5 (mainly in D5.4).

Interface	Method	Description
iAssetEntry	getEntries	Get all entries.
	getEntry	Get a specific entry.
	updateEntry	Update a specific entry.
iGroup	addGroup	Add a new group.
	deleteGroup	Delete a specific group.
	getGroup	Get a specific group.
	getGroups	Get all groups.
	updateGroup	Update a specific group.
iUser	addUser	Add a new user.
	deleteUser	Delete a specific user.
	getUser(emailAddress)	Get a specific user by email address.
	getUser(id)	Get a specific user by email id.
	getUser(screenname)	Get a specific user by email screen name.

Interface	Method	Description
	updateUser	Update a specific user.
iBlogEntry	addEntry	Add a new Blog entry.
	deleteEntry	Delete a specific Blog entry.
	getEntries	Get all Blog entries for a specific group.
	getEntry	Get a specific Blog entry.
	subscribe	Subscribe to Blog entries updates.
	updateEntry	Update a specific Blog entry.
iBookmarkEntry	addEntry	Add a new Bookmark entry.
	deleteEntry	Delete a specific Bookmark entry.
	getEntries	Get all Bookmark entries for a specific group.
	getEntry	Get a specific Bookmark entry.
	subscribe	Subscribe to Bookmark entries updates.
	updateEntry	Update a specific Bookmark entry.
iDiscussionEntry	addMessage	Add a new Discussion entry.
	deleteMessage	Delete a specific Discussion entry.
	getCategoryMessages	Get all Discussion entries for a specific category.
	getMessage	Get a specific Discussion entry.
	getThreadMessages	Get all Discussion entries for a specific thread.
	subscribe	Subscribe to Discussion entries updates.
	updateMessage	Update a specific Discussion entry.
iFileEntry	addFileEntry	Add a new File entry.

Interface	Method	Description
	copyFileEntry	Copy a specific File entry.
	deleteFileEntry	Delete a specific File entry.
	getFileEntries	Get all File entries for a specific folder.
	getFileEntry	Get a specific File entry.
	revertFileEntry	Revert a specific File entry to a previous version.
	subscribe	Subscribe to File entries updates.
	updateFile	Update a specific File entry.
iWikiPage	addPage	Add a new Wiki page
	changeParent	Change parent to a specific Wiki page.
	deletePage	Delete a specific Wiki page.
	getChildren	Get children pages of a specific Wiki page.
	getOrphans	Get orphan pages in a Wiki.
	getPage	Get a specific Wiki page.
	movePage	Move a specific Wiki page to another position.
	subscribe	Subscribe to Wiki updates.
	updatePage	Update a specific Wiki page.
iNotification	sendNotification	Send a notification on a portal event to a registered user.
	updateNotificationPreferences	Update the User preferences upon notification configuration.
iDecisionMaking	createDecision	Create a new collaborative decision making process.

Interface	Method	Description
	updateDecision	Update existing decision making process.
	deleteDecision	Delete existing decision making process.
iStructuredContentManager	manageAsset	Manage a specific Asset entity from the existing catalogue (allowing CRUD operations).
	manageLessonLearned	Manage a specific Lesson Learned entity from the existing catalogue (allowing CRUD operations).
	manageTrial	Manage a specific IndustrialTrial entity from the existing catalogue (allowing CRUD operations).

Table 38 MOD.BU.KM. Main platform interfaces

8.3.2 MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)

8.3.2.1 FBB Specification

The Event Manager (EM) component produces events related to operational activity (e.g. start/ end operations, start/ end work orders, defects, downtimes, etc.) that are relevant for decision making. The "DSS.04. DSS event protocol adapter" is in charge of adapting the events produced by the Event Manager (EM) and store operational data into the "DSS.03.- DSS data repository" through the "DSS.02. DSS management services".

Users interact with the MOD.BU.DSS though web based graphical user interfaces (DSS.01.- DSS GUI).

The business logic supporting they decision support features is implemented in the. "DSS.02.- DSS management services" component which interacts with the "DSS.03.- DSS data repository" by storing and retrieving the required information.

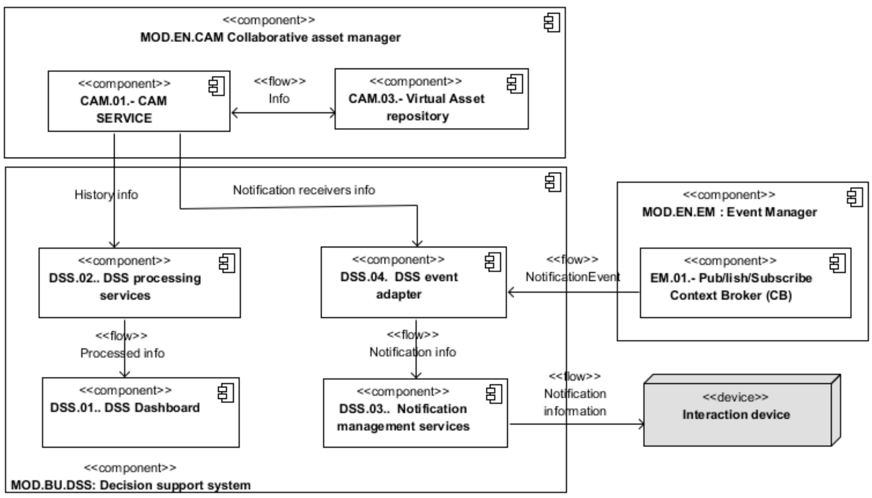


Figure 102 MOD.BU.DSS Decomposition into functional building blocks

8.3.2.1.1 DSS.01.- DSS GUI

The "DSS.01.- DSS GUI" allows the users to access an aggregated, graphical view of relevant operational results to support decision making. DSS.01 consumes the *getProcessedInfo* method of the **iDSS** interface exposed by the "DSS.02.- DSS processing services".

8.3.2.1.2 DSS.02.- DSS processing services

The "DSS.02.- DSS processing services" to get the history data from the virtual asset repository and process it to provide the right information supporting decision making process.

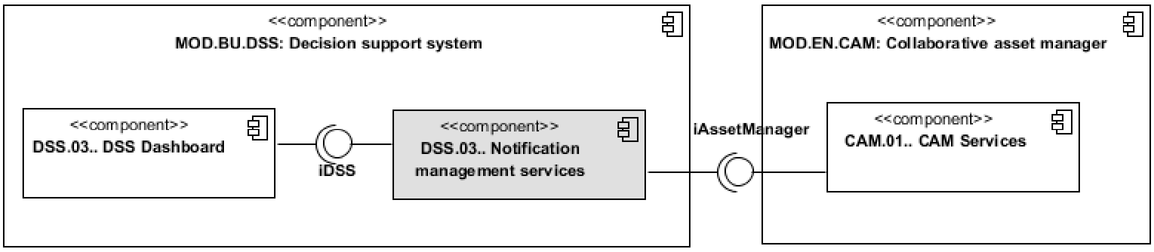


Figure 103 DSS.02. DSS processing services interfaces

DSS.02 exposes **iDSS** interfaces and uses the interfaces provided by the "CAM.01. CAM services" to access information from the virtual asset repository.

8.3.2.1.3 DSS.03.- Notification management services

The "DSS.03.- Notification management services" provides the functionalities to collect the notification information coming from the "DSS.04.- DSS event protocol adapter".

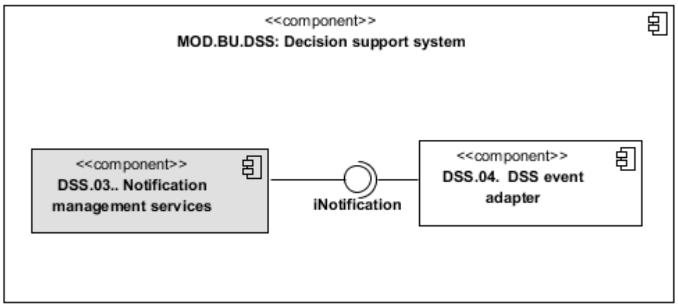


Figure 104 DSS.03. Notification management services interfaces

DSS.03 exposes **iNotification** interface to manage the notifications.

8.3.2.1.4 DSS.04. DSS event adapter

DSS.04 processes the notification event produced by the MOD.EN.EM and executes the *setOperationalData* method exposed by the **iDSS** interface of the "DSS.03.- Notification management services" component.

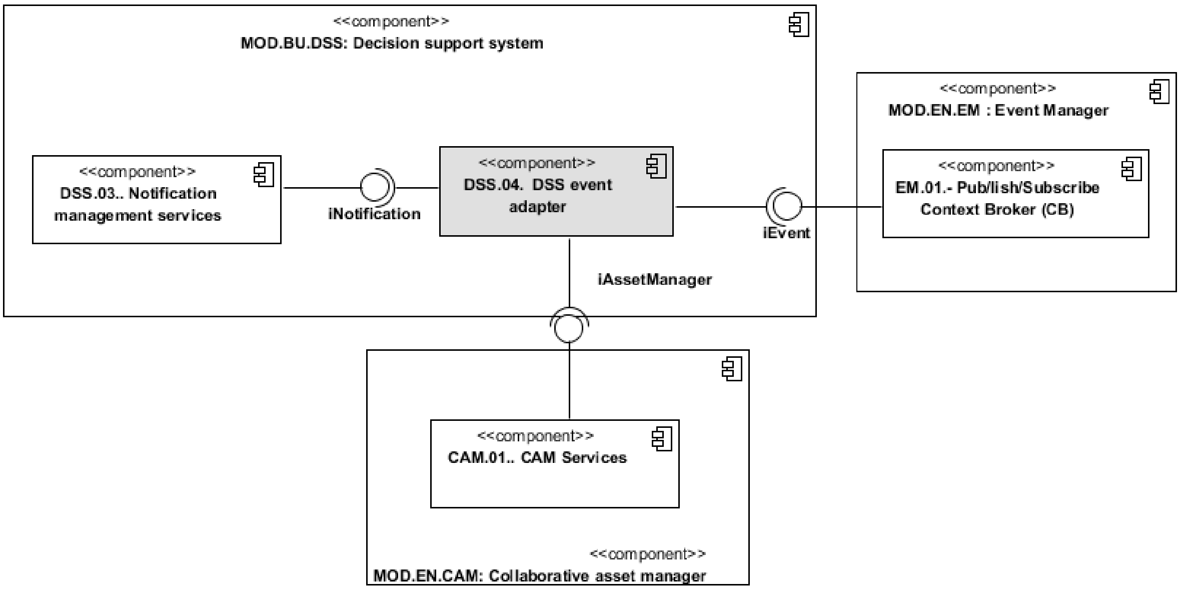


Figure 105 DSS.04. DSS event protocol adapter interfaces

DSS.04 consumes the **iEvent (including Publish and Subscribe methods)** interface exposed by the EM.01 Publish /Subscribe Context Broker (CB) to publish events in the appropriate format and execute the appropriate method of the **iNotification** interface.

8.3.2.2 Background Assets

As an initial approach, several potential background asset candidates were identified in D2.1 to support some of the identified functionalities. Initial background assets were evaluated in the scope of Task 5.3 (Assistance: Decision Support System) and further ones were identified.

The selected tools involve: R (to support "DSS.02.- DSS processing services") KNOWAGE (to support "DSS.01.- DSS dashboard") and Firebase Cloud Messaging (to support push notifications management from "DSS.03.- Notification management services").

R[27]

R is a language and environment for statistical computing providing a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

R is able to perform SPARQL queries, a capability useful for seamless integratation with the CAM (i.e. using its semantic reasoning capabilities, exposing a SPARQL endpoint).

[27] R: https://www.r-project.org/

KNOWAGE[28]

KNOWAGE is an open source business analytics suite developed by ENGINEERING that combines traditional data and big data sources into valuable and meaningful information. The suite is composed of several modules, each one conceived for a specific analytical domain. They can be used individually as a complete solution for a certain task, or combined with one another to ensure full coverage of users' requirements, allowing to build a tailored product. All KNOWAGE modules are developed on the same architecture, sharing more than technology: they use the same metadata and analytical layer, share security, life cycle and general capabilities.

KNOWAGE can aggregate data in custom and high-view performance dashboards, properly composing and orchestrating business intelligence and data mining/analytics primitives. Data need a pre-processing step before they are submitted for the visualization, so a big data analysis approach is supported by integrating a processing engine (e.g. R and Python scripts can be used to pre-process dataset before starting visualization and reporting tasks).

KNOWAGE is available in two versions: Community and Enterprise editions. The Community Edition is available, including the whole set of analytical capabilities, entirely free and under an open source license.

Firebase Cloud Messaging[29]

Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection allowing to deliver and receive messages and notifications on Android, iOS, and the web at no cost.

8.3.2.3 A4BLUE Enhancements

The tools identified in the previous section will be used by a newly developed DSS module able to provide the required funtionality and interact with the A4BLUE plataform.

8.3.2.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.BU.DSS.

Interface	Method	Description
iDSS	getProcessedInfo	Provides access to processed info.

Table 39 DSS.02. DSS management services component interfaces

Interface	Method	Description
iNotification	setNotificationData	Provides notification data.

Table 40 DSS.03. Notification management services main component interfaces

8.3.3 MOD.BU.MON- MONITORING

MOD.BU.MON functionality is covered by taking advantage of the MOD.BU.DSS component for indicators based on process history information and by MOD.BU.CQMS component for indicators based on surveys.

[28] https://www.knowage-suite.com

[29] https://firebase.google.com/docs/cloud-messaging/

8.3.4 MOD.BU.ACE- AUTOMATION CONFIGURATION EVALUATION

8.3.4.1 FBB Specification

Figure 106 shows the main components of the MOD.BU.ACE, describing both main Functional Building Blocks and information flows among them.

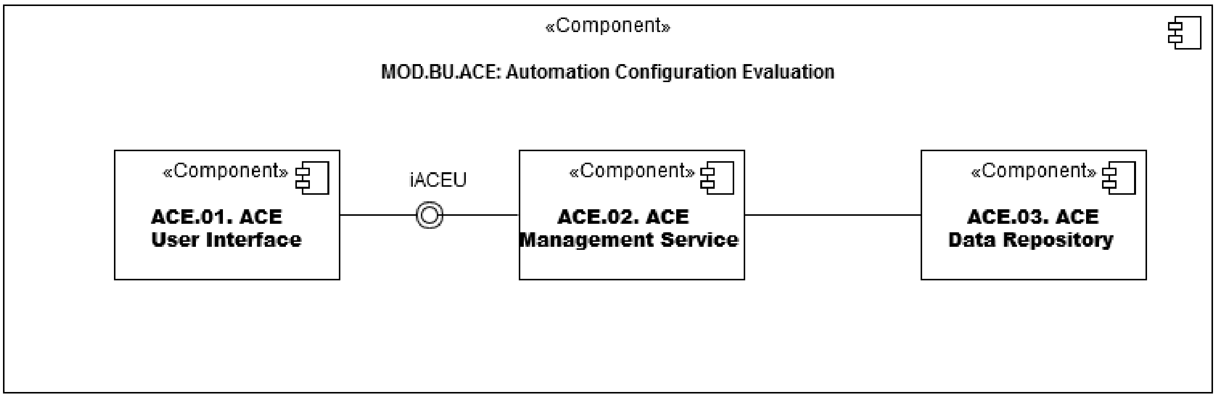


Figure 106 MOD.BU.ACE. Interfaces

8.3.4.1.1 ACE.01. ACE User Interface

The “ACE.01. – ACE User Interface” allows the production planner to insert and view information related to the automation configuration evaluation. It will also display a graphical representation of the data to the end user.

The FBB consumes and expose the methods of the “iACEU” interface. The next table shows an initial approach to the main methods exposed and consumed by the interface.

Interface	Method	Description
iACEU	getAutomationResults	Retrieve automation data and calculation to the user
	sendAutomationData	Sends automation data
	setAutomationData	Save automation data set by the user

Table 41 ACE.01. ACE User Interface main methods

8.3.4.1.2 ACE.02. ACE Management Service

It provides functionalities to:

- Process data coming through the “ACE.01. ACE User Interface”;
- Calculate automation results;
- Make available the raw and processed automation information to the “ACE.01. ACE User Interface”;
- Store and retrieve data in/from the “ACE.03. ACE Data Repository”.

The FBB exposes and consumes iACEU described in the previous section.

8.3.4.1.3 ACE.03. ACE Data Repository

The “ACE.03. Data Repository” enables storage of automation data coming from the “ACE.01. ACE User Interface”, it is accessed through the “ACE.02. ACE Management Service”.

8.3.4.2 Background Assets

The MOD.BU.ACE is a newly designed tool, specifically targeting the A4BLUE needs as far described in the methodological approach contained in D3.1 - “Methodology for socio- economically sustainable design of optimal automation - Initial version”. For this reason no existing backgrounds are available to support the development of this module, resulting in a brand new exploitable asset of the A4BLUE project.

8.3.4.3 A4BLUE Enhancements

The methodology described in D3.1 - “Methodology for socio- economically sustainable design of optimal automation - Initial version” provides the basis for the Automation Configuration Evaluation module of the A4BLUE Adaptive Framework.

This methodology has been specifically designed to support the objectives of the A4BLUE project, therefore a brand new tool is needed to foster the use of this methodology. To this end two different tools have been foreseen.

A first attempt to build a testing tool (to refine and improve the teorethical approach used while designing the methodlogy) has been realized using a Microsoft Excel Tool, taking advantages of its lower complexity and the possibility of using the programming language Visual Basic for Applications (VBA). With the help of this tool, the methodology for determining the optimal degree of automation can be applied and validated with concrete examples. In this way, occurring problems could already be detected and solved in the development phase of the underlying methodology.

A more complete and user friendly tool will be, instead, developed during the beta phase of the project, by delivering a web based tool to ease the workflow associated to the evaluation methodology, as well as the integration with the rest of the A4BLUE Framework. This web tool will be based on the “template” provided by the VBA tool, in order to speed up the development process itself. In addition, this tool will be further connected to the monitoring module that consists in the main KPIs of the specific industry. These indicators will support the industry planner in deciding if an upgrade or downgrade in the automation level is needed, and who will be able to simulate new developments in this matter.

With the help of the developed tools, the methodology for determining the optimal degree of automation will be applied and further validated in the pilot scenario in WP6.

8.3.5 MOD.BU.CQMS- COMPUTER BASED TOOL FOR THE MEASUREMENT OF WORKER SATISFACTION

8.3.5.1 FBB Specification

The figure below shows the main components of the MOD.BU.CQMS, describing both main Functional Building Blocks and information flows.

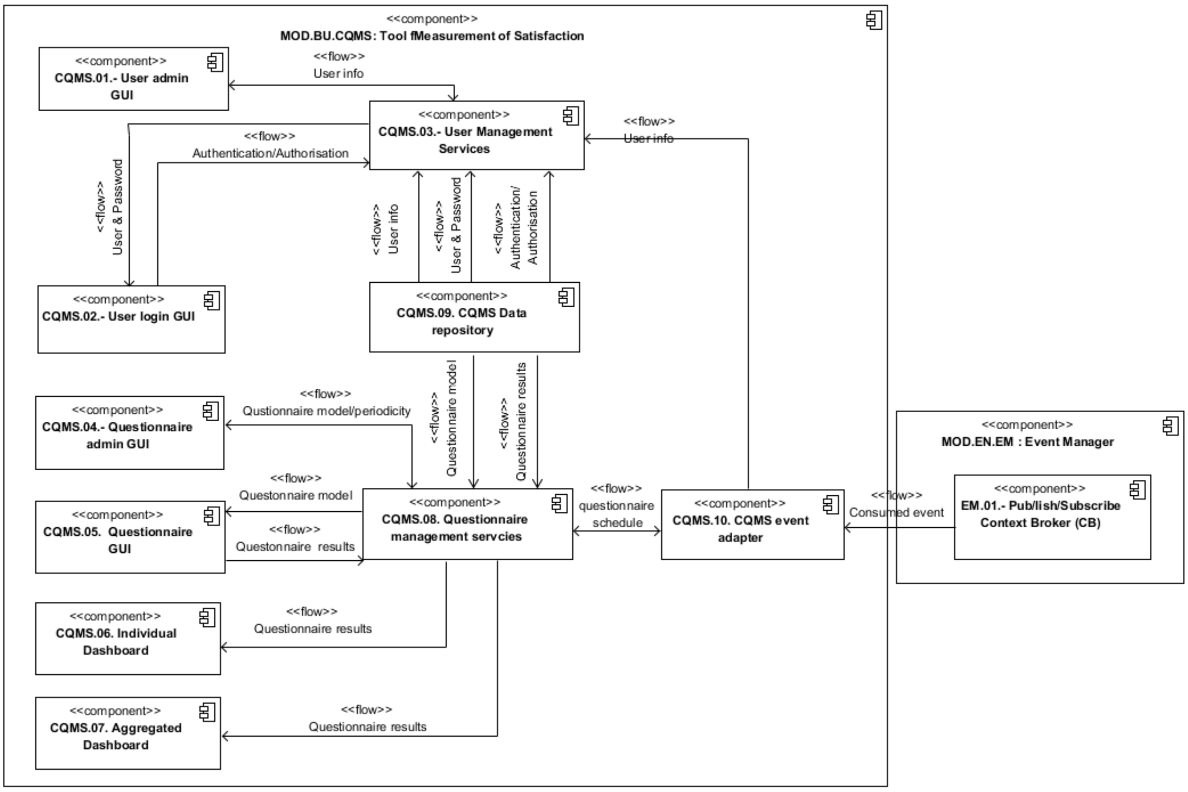


Figure 107 MOD.BU.CQMS. Decomposition into functional building blocks

8.3.5.1.1 CQMS.01. User Admin GUI

The CQMS.01 enables the system administrator to create/ read/update user and manage user access by using the CQMS.08. CQMS.01 consumes the *setUser*, *updateUser*, *deleteUser* and *getUserList* methods of the **iUserAdmin** interface exposed by the CQMS.08.

8.3.5.1.2 CQMS.02. Login GUI

The CQMS.02 allows to introduce the user's identifier and password to support user authentication and authorisation process through the CQMS.08. CQMS.02 consumes the *getAuthorisation* method of the **iUser** interface exposed by the CQMS.03.

8.3.5.1.3 CQMS.03. User Management Services

The CQMS.03 provides functionalities to: (1) create (i.e. including access credentials), read, update and delete questionnaires; (2) generate user/token; (3) check user access authorisation.

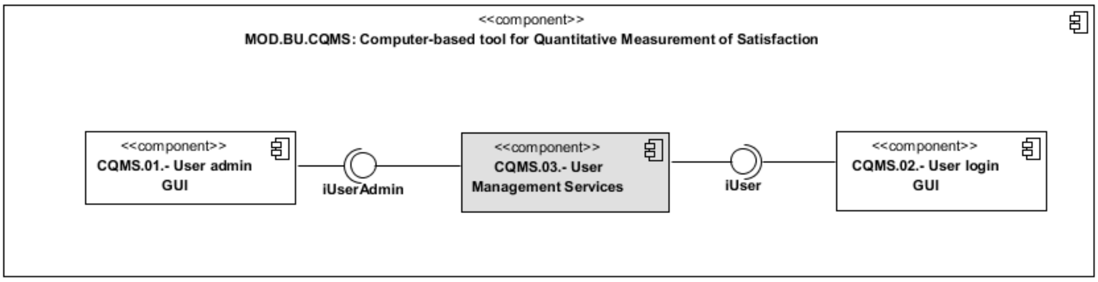


Figure 108 CQMS.03. Main interfaces

8.3.5.1.4 CQMS.03 exposes iUserAdmin and iUser interfaces. CQMS.04. Questionnaire Admin GUI

CQMS.04 enables the system administrator to model the appropriate questionnaires by using the CQMS.08. CQMS.04 consumes the *setQuestionnaire*, *updateQuestionnaire* and *deleteQuestionnaire* methods of the **iQuestionnaireAdmin** interface exposed by the CQMS.08.

8.3.5.1.5 CQMS.05. Questionnaire GUI

CQMS.05 displays the selected questionnaire which is built online based on the retrieved model and allows to save the collected results through the CQMS.08. CQMS.05 consumes the *getQuestionnaireList*, *getQuestionnaire* and *saveQuestionnaireResults* methods of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.5.1.6 CQMS.06. Individual questionnaire dashboard GUI

CQMS.06 allows both the shop floor operators and supervisors to access an individual view of the results of the worker satisfaction questionnaire prioritising, whenever possible, graphical representation by using the CQMS.08 CQMS.06 consumes the *getIndividualResults* method of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.5.1.7 CQMS.07. Aggregated questionnaire dashboard UI

CQMS.07 allows the supervisors to access an aggregated view of the results of the worker satisfaction questionnaire prioritising, whenever possible, graphical representation by using the CQMS.08. CQMS.07 consumes the *getAggredatedResults* method of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.5.1.8 CQMS.08. Questionnaire Management Services

CQMS.08 provides functionalities to: (1) create (i.e. including periodicity), read, update and delete questionnaires; (2) collect the results from the questionnaires and store them into the data repository; (3) calculate the score of the questionnaire based on the scores assigned to the potential answers related to each question involved in the questionnaire; (4) schedule questionnaire on periodicity basis; (5) publish and subscribe to events to enable online operation.

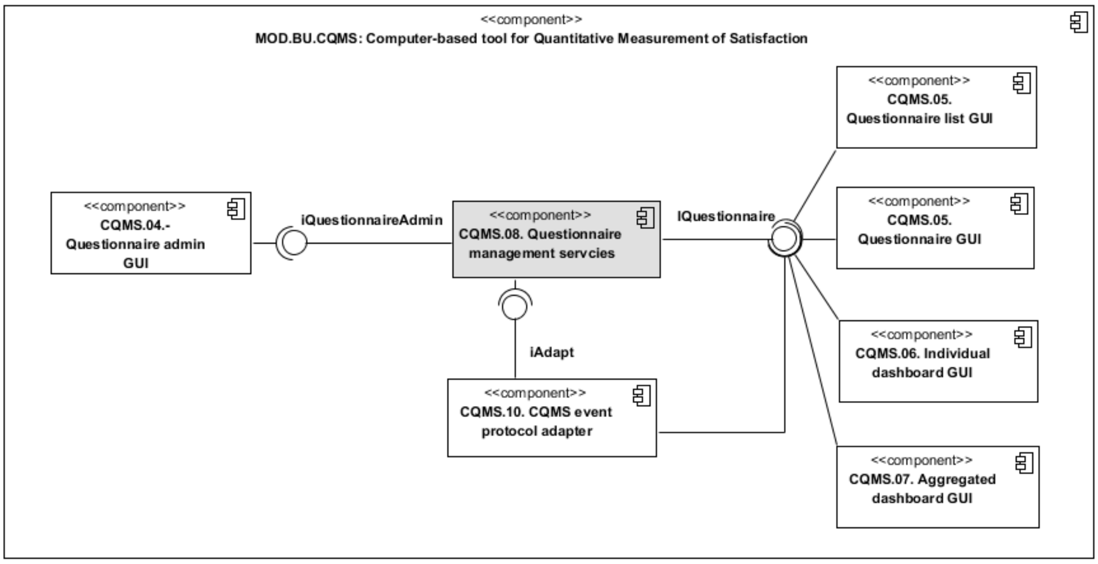


Figure 109 CQMS.08. Questionnaire management services interfaces

CQMS.08 exposes **iQuestionnaireAdmin** and **iQuestionnaire** interfaces and consumes the **iAdapt** interface exposed by the CQMS.10 to adapt the questionnaire event to the appropriate event protocol.

8.3.5.1.9 CQMS.09. CQMS data repository

"CQMS.09. CQMS data repository" enables the storage of questionnaire related information. It is accessed through the "CQMS.03.- User management services" and "CQMS.08.- Questionnaire management services".

8.3.5.1.10 CQMS.10. CQMS event adapter

CQMS.10 adapts the information produced by the "CQMS.08.- Questionnaire management services" to the event format supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM to update the users.

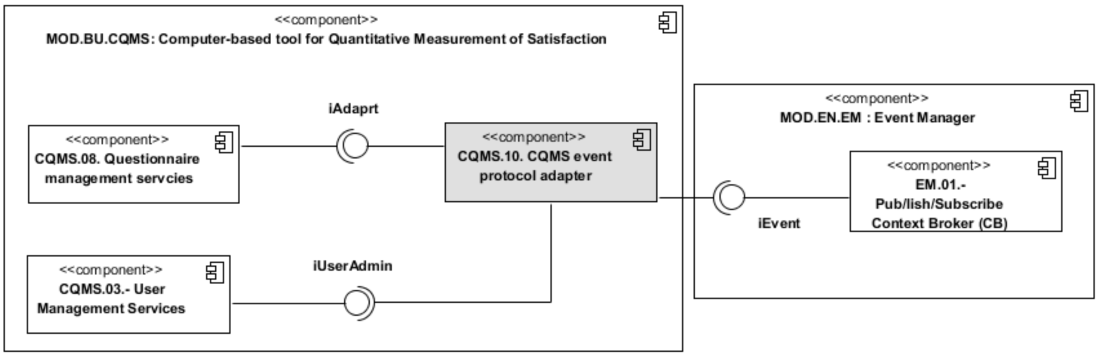


Figure 110 CQMS.10. CQMS event protocol adapter interfaces

CQMS.10 exposes **iAdaptSubscription** interface and consumes the **iEvent (including Publish and Subscribe methods)** interface exposed by the EM.01 Publish /Subscribe Context Broker (CB) to publish events in the appropriate format and the appropriate method of the **iUserAdmin** interface to update the authorised users.

8.3.5.2 Background Assets

FIS-TAR Health Questionnaire (HQ-SE)[30] is a specific enabler (SE) developed in the scope of the FISTAR project[31]. The HQ-SE provides a RESTFUL API that allows: creation of a simple assessment (on-demand) and monitoring (periodicity based) of questionnaires from csv files (no graphical user interface is provided); questionnaire monitoring and results capture; (optionally) storage of questionnaire models and results in the repository based on configuration options; keep audit trail (i.e. log of actions as read, accomplish, identifying the user that accesses the information and the accessed user information); (optional) to publish new questionnaire context event into the FIWARE Publish/Subscribe Context Broker GE.

8.3.5.3 A4BLUE Enhancements

The main goal of the A4BLUE Computer-based tool for Quantitative Measurement of Satisfaction is to develop a robust web-based tool enabling the involved users to complete the worker satisfaction questionnaires in an easy way and evaluate the results.

The MOD.BU.CQMS will rely on the above mentioned background asset, especially since it provides the capability to define and manage questionnaires. HQ-SE will be integrated in the in the new web based tool, that will include new history data visualisation and export capabilities. Furthermore, new publish/subscribe capabilities will be enhanced to support online working mode.

[30] Health Questionnaire Specific Enabler: http://fistarcatalogue.fiware.eng.it/enablers/health-questionnaire-service/documentation

[31] FI-STAR Project: https://www.fi-star.eu/fi-star.html

8.3.5.4 Interfaces

The following tables describe the interfaces exposed and consumed in the framework of MOD.BU.CQMS.

Interface	Method	Description
iUserAdmin	setUser	Stores the user and access data in the data repository.
	updateUser	Updates the model of a specific user in the data repository.
	deleteUser	Deletes a specific user from the data repository.
	getUserList	Retrieves the list of users from the data repository.
iUser	getAuthorisation	Checks if the selected user is authorised to access the MOD.BU.CQMS.

Table 42 CQMS.03. Main methods

Interface	Method	Description
iQuestionnaireAdmin	setQuestionnaire	Stores the questionnaire model in the data repository.
	updateQuestionnaire	Updates the model of a specific questionnaire in the data repository.
	deleteQuestionnaire	Deletes a specific questionnaire from the data repository.
iQuestionnaire	getQuestionnaireList	Retrieves the list of questionnaires from the data repository.
	getQuestionnaire	Retrieves the model of a specific questionnaire from the data repository.
	saveQuestionnaireResults	Updates the model of a specific questionnaire in the data repository.
	getIndividualResults	Retrieves individual history data
	getAggredatedResults	Retrieves aggregated history data
	setQuestionnaireSchedule	Schedules a selected set of questionnaires, per user and time interval, and stores them in the data repository.
	getQuestionnaireSchedule	Retrieves the list of scheduled questionnaires, per user and time interval, from the data repository.

Table 43 CQMS.08. Questionnaire management services main component interfaces

Interface	Method	Description
iAdapt	setQuestinnaireInfo	Collects questionnaire info

Table 44 CQMS.10. CQMS event protocol adapter main component interfaces

8.4 SUMMARY FUNCTIONAL DESIGN TRACEABILITY MATRIX

The following table maps the identified Functional Building Blocks (FBB) with the features identified in section “5 – SPECIFICATIONS”.

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
AM.01. Automation hardware											
AM.02. Automation GUI											
AM.03. Local automation controller											
AM.04. Automation data repository (opt)Automation semantic registrator											
AM.05. OPC-UA Server											
LS.01. Legacy mediation agent											
MS.01. Mediation management services											
MS.02. MS event adapter											
DM.01. Semantic representation registrator											
DM.02 OPC UA discovery server											
DM.03. Device manager OPC UA client											
DM.04. DM event adapter											
MI.01. Multi-channel interaction manager											

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
MI.02. Uni-channel interpreter											
MI.03. MHMI execution servicesMI.03. Interaction semantic repository											
MI.04. MHMI data repository (opt)											
MI.05. Multi-channel fusion engine											
MI.06. MHMI event protocolevent adapter											
AS.01. Tracking											
AS.02. Safety controller											
AS.03. Safety configuration repository											
AS.04. Safety configurator											
AS.05. AS event protocol adapter											
CAM.01. CAM Service											
CAM.02. CAM UI											
CAM.03. Asset Repository											
EM.01. Publish/Subscribe Context Broker (CB)											
EM.02. Context Producer (CP)											

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
EM.03. Context Provider (CPr)											
EM.04. Context Consumer (CC)											
EM.05. Storage											
EM.06. Application											
EM.07. Complex Event Processing (CEP)											
EM.08 CEP Management											
EM.09. Event Processing Agent (EPA)											
ARG.01 Resource Manager											
ARG.02 Event Connector											
IM.01. Application Developer Portal											
IM.02. End User portal											
IM.03. User Management											
IM.04. Application Management											
IM.05. Access Management											
IM.06. Authentication and Authorization											
KM.01. Workspaces List											

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
KM.02. User Profile											
KM.03. Group Management											
KM.04. Universal Search											
KM.05. Blogs											
KM.06. Shared Calendar											
KM.07. Documents											
KM.08. Discussions											
KM.09. Wikis											
KM.10. Bookmarks											
KM.11. Usage Analytics											
KM.12. Relational DB											
KM.13. NoSQL DB											
KM.14. Portal Directory											
KM.15. In Box Notification											
KM.16. Assets Catalogue											
KM.17. Industrial Trials											

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
KM.18. Lessons Learned											
KM.19. SWOT Analysis											
KM.20. Survey											
KM.21. Weigh up Decision											
KM.22. Feedback Management											
KM.23. Moodle											
KM.24. Moodle Bridge											
DSS.01. DSS Dashboard											
DSS.02. DSS processing services											
DSS.03. DSS notification management services											
DSS.04. DSS event adapter											
DSS.05. DSS data repositoryNotifier App											
CQMS.01. User admin GUI											
CQMS.02. Login GUI											
CQMS.04. Questionnaire admin GUI											
CQMS.05. Questionnaire list GUI											

Functional Building blocks	FTR.01	FTR.02	FTR.03	FTR.04	FTR.05	FTR.06	FTR.07	FTR.08	FTR.09	FTR.10	FTR.11
CQMS.05. Questionnaire GUI											
CQMS.06. Individual dashboard GUI											
CQMS.07. Aggregated dashboard GUI											
CQMS.08. Questionnaire management services											
CQMS 09. CQMS data repository											
CQMS 10. CQMS event protocol adapter											

Table 45 Specifications (Features) – Functional Design (FBBs) traceability matrix

9 FUNCTIONAL AND MODULAR ARCHITECTURE

This section aims at providing an overall and integrated viewpoint on the A4BLUE RA, as a combination of the modules described in previous sections. The functional viewpoint focuses, in fact, on the functional components, their interrelation and structure, the interfaces and interactions between them, and the relation and interactions of the system with external elements in the environment.

This viewpoint will be further extended (using a different angle) in the following activities within WP3, WP4 and WP5 scope in order to better detail the implementation viewpoint of an A4BLUE-based system; in the end, a further refinement will be overcome in WP6 to define the deployment viewpoint in order to further describe how this RA will be derived and adapted to deploy the real use cases under piloting activities in the four demonstration cases.

According to the A4BLUE RA, the functionalities of an adaptive assembly system can be decomposed into three high-level *Functional Domains* - Shopfloor, Enterprise and Business. Domains are very simple and straightforward: they define a coarse mapping of system elements to either the factory – Shopfloor Domain- or the broader world of corporate IT - Enterprise or Business Domains. Examples of elements in Shopfloor Scope are machinery, field devices, workstations, SCADA and MES systems, and any software running in the factory data centre. To the Enterprise Domain belong systems and applications shared across the factory (e.g. shopfloor and topfloor), while to the Business Domain belong applications and services dedicated to decision making and business value creation.

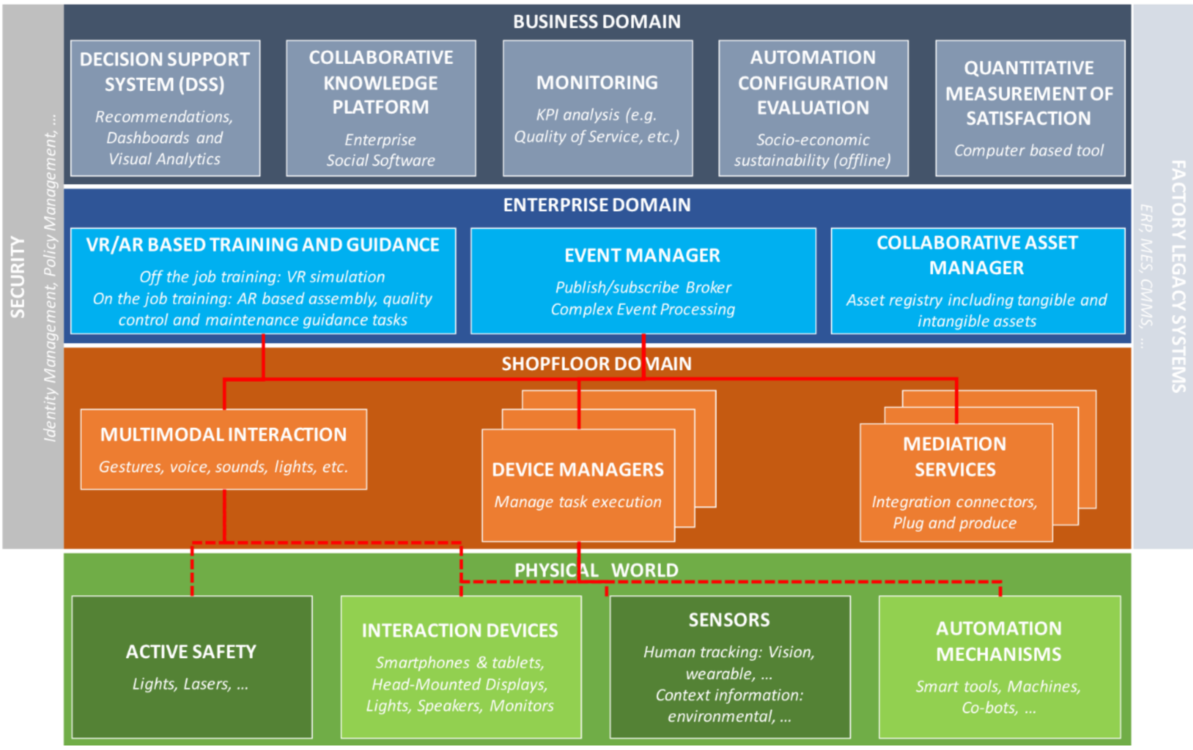


Figure 111 A4BLUE F&M Architecture

9.1 SHOPFLOOR DOMAIN

The A4BLUE Shopfloor Domain is the bottom layer of A4BLUE RA and is populated by any kind of device that is connected to the *digital world* on one side and to the *real world* to the other. As a consequence, the Shopfloor Domain includes functionalities supporting *automated control* and *automated adaptation* of physical production processes. While the meaning of “control” in this context is straightforward, “adaptation” is worth a few additional words. Automated adaptation is the enabler of plug-and-play factory equipment (better known as *plug-and-produce*), as well as new operator-aware capabilities into the system in order to make it able to adapt for instance to worker skills and preferences, or to the process/product variability. The Shopfloor domain requires a bidirectional monitoring/control communication channel with the Field and its physical world.

The Shopfloor domain partially maps to the Control domain of the IIRA (see section 3.2). The main difference is that Control is also responsible for decoupling the real word from the digital world, as it includes the functionality for shopfloor communication, entity abstraction, modelling and asset management. In other words, Control mediates all shopfloor access from other domains like Information, Operations, etc. In the A4BLUE RA, instead, the Shopfloor domain is only focused on its main role, while auxiliary concerns are dealt with by Data Models and Field Abstraction are managed by functions in the Enterprise domain (e.g. using the MOD.EN.CAM).

In other words, the Shopfloor Domain is also populated by entities of the real world – i.e., those physical elements of production processes that are not directly connected to the network, such as: **Things, People** and **Environments**. These are represented in the digital world by some kind of *wrapper* or *adaptor*.

9.2 ENTERPRISE DOMAIN

The A4BLUE Enterprise Domain is the middle layer of A4BLUE RA and represents the core part of any A4BLUE-based system. To this end, this domain is populated by several components in charge of managing the logic for adaption management, using an Event Driven Architecture in order to provide the assistance services.

To be able to respond to variable demands (both on process and on product level), the operators need dynamic, distributed and adaptive decision support extracting value from historical data and monitoring the evaluation phase, helping them to distinguish decision options and maximizing productivity based on virtual asset representation, while continuously interacting with the operator (e.g. using AR devices). To manage and handle counterproductive situations, adaptability must be a significant variable of the production system (not only at shopfloor level, but at a wider scope at the enterprise level). Using event data in combination with a distributed control system facilitates adaptive decision-making and dynamic control capability enabling the operators (with a variable level of skills) to handle uncertainty in a safer and more productive way.

This layer will be also enhanced by tools supporting digital twin, by producing and consuming digital information coming from the other layers (e.g. creating a digital representation of the behavior of a physical object or process from the Shopfloor Domain, that helps optimize business performance using components from the Business Domain).

9.3 BUSINESS DOMAIN

The A4BLUE Business Domain is the upper layer, in charge of supporting strategic decision-making process (using both online and offline tools), targeting both blue- and white-collar workers.

All of the components in this domain aim to extract and take benefits from an improved knowledge of the running automation and adaptation processes, to gain competitive advantages from the shopfloor processes to the business insights, and make use of all the information gathered around the adaptive system. Knowledge plays an important role in the life of organizations in the current economy, considered as an economic asset which helps companies to overcome constant changes in business environment. Knowledge Management tools should support the easy deliver of the right knowledge to the right persons at the right time and to help people to share information and use them in the way which can lead to improving of organizational performance, developing a mid-long term strategy.

10 CONCLUSIONS

This deliverable completes the beta iteration in T2.1 and T2.2 by introducing the A4BLUE RA, providing the high-level design of the A4BLUE Platform and outlining its implantation (to be further refined in WP3, WP4 and WP5) while developing the piloting and validation activities within the WP6 scope. These results have been driven by earlier work and results from WP1, notably the analysis of requirements and reference applicative business scenario. The main highlights include:

- A4BLUE is one of the current efforts (within the FOF-04-2016 cluster) to specify an adaptive framework for assembly, in-line with recently introduced reference architectures for the manufacturing industry.
- The A4BLUE RA addresses functionalities in three distinct, yet interrelated and complementary domains: shopfloor, enterprise and business.
- The A4BLUE RA envisions automation and adaptation functionalities in close proximity to factory physical processes, including real-time operations and taking into account process/product/operator variability.
- The A4BLUE Platform design has some innovative characteristics that mark a difference with respect to other similar on-going efforts: software components from the FIWARE for Industry ecosystem are used as a key enabler of distributed adaptation capabilities in several assembly scenarios.
- The A4BLUE Platform enables a wide range of business scenario, including their mixing with socio-economic evaluation and analysis for decision support.
- The baseline of the A4BLUE implementation can be existing open source frameworks, platforms and tools, including background assets belonging to project partners.

Overall, this document provides a sound basis for development and integration activities that will be performed as part of technical work packages, notably WP3, WP4 and WP5. In particular, it defines the main components and structuring principles of the A4BLUE Platform, also in terms of implementation tasks: this will ensure that inter-dependant activities can be streamlined in the best of ways. Hence, the document will be a valuable input for all partners engaged in technical design and software development.

The A4BLUE RA and Platform design will also drive the implementation of use cases in the scope of WP6. As a general rule, use cases might require some customization, however not impacting on Platform components described in this document. Identified components might be extended, provided they are backward-compatible – i.e. systems using the standard interfaces are not affected.

Note however that development and integration activities are likely to introduce revisions to this Platform design, resulting from:

- new findings and technological choices made during the detailed design and implementation of individual components;
- changes in requirements and use cases (as already outlined in deliverable D1.4 and D1.5 and their evolutions, the project is ready to embrace changes in technical and business requirements);
- external evolution of the project's KETs and endorsed standards.

All these revisions will be further documented in the technical reports resulting as outcomes of the other WPs (mainly WP3, WP4, WP5 and WP6).

REFERENCES

[1] VDI/VDE GMA, "Reference Architecture Model Industrie 4.0 (RAMI4.0)" (July 2015)

[2] FDI Cooperation, "FDI - Field Device Integration Technology", January 2012 (http://www.fdi-cooperation.com/tl_files/images/content/Publications/FDI-White_Paper.pdf)

[3] AutomationML consortium, "Whitepaper AutomationML Part 1 - Architecture and general requirements", October 2014 (https://www.automationml.org/o.red/uploads/dateien/1417686950-AutomationML%20Whitepaper%20Part%201%20-%20AutomationML%20Architecture%20v2_Oct2014.pdf)

[4] Kruchten, "Architectural blueprints - The "4+ 1" view model of software architecture," IEEE Software, 1995

[5] Matthias Hartmann, Bastian Halecker, June 2015, Management of Innovation in the Industrial Internet of Things, pp. 1, 7-8

[6] The Reference Architectural Model Industrie 4.0 (RAMI 4.0), http://www.zvei.org/Downloads/Automation/ZVEI-Industrie-40-RAMI-40-English.pdf

ANNEX A REQUIREMENTS BOOK: USER LEVEL (FROM D1.4)

Please find here below an extract of D1.4, listing the main requirements at user level, to ease the understanding of codes used in Section 5. Any updated version of the list could be published as part of WP1, therefore the following list could suffer some misalignments.

1. Organisational Level		
	ESSENTIAL	
Code		Description
Alpha	Beta	
1.1	2.1.1	Continuous data collection for analysis of system performance and optimisation needs
1.6	2.1.2	Monitoring work station performance for future process improvement
1.16	2.1.3	Constant logging of production waste data for the purposes of future planning
	2.1.4	On-the-job work instructions for workers
1.17	2.1.5	Constant recording of automation / robot usage data to a central system to manage maintenance activities
1.5	2.1.6	Constant recording of tool usage data to a central system to monitor maintenance activities
1.3	2.1.7	Reconfigurability for new automation or robotics, e.g. 'plug & produce' capabilities
	2.1.8	Automated systems and robots that operate in a standardardised way
1.8	2.1.9	Continuous interaction of all systems in the organisation for resource allocation
	2.1.10	Abilities for evaluating and selecting optimal levels of automation to meet economic requirements
	2.1.11	Automatic responses to answer manual requests from operators
	DESIRABLE	
	2.1.12	Ability to assess and adapt to optimal levels of automation for individual workers
	2.1.13	Abilities for determining optimal levels of automation to maintain operator satisfaction
1.19	2.1.14	Capabilities for evaluating workers' levels of satisfaction
1.15	2.1.15	Self-adjusting to compensate for reduced technical capabilities, e.g. older technologies, functional
	2.1.16	Functions that can adapt to suit new / different workforce requirements
	2.1.17	Systems that can detect and adjust to suit the requirements of different operators e.g. training and experience levels
1.7	2.1.18	Autonomous assessment of requirements and adaptation to new products
1.7	2.1.19	Autonomous assessment of requirements and alteration to new processes
	2.1.20	Capabilities that adapt to workers' levels of satisfaction
1.9	2.1.21	Automated system or robot capabilities for utilising operators' expertise/ knowledge
	2.1.22	Direct connection to organisational systems for post-production product service and support
1.4	2.1.23	Direct connection to internal control systems (e.g. Enterprise Resource Planning) to adapt the assembly process
	2.1.24	Standardised off the job training for all workers

Table 46 User Requirements: Organisational Level

2. Automation and Robotics		
	ESSENTIAL	
Code		Description
Alpha	Beta	
2.1	2.2.1	Collision avoidance detection and stop functions
2.10	2.2.2	Mechanisms that alert operators of safety status and functioning
	2.2.3	Automatic updates on information concerning process / production
	2.2.4	Mechanisms for making operators aware of safety status
	2.2.5	Automatic feedback to operators of updates to process / production information
2.12	2.2.6	Autonomous adaptation to varying production demands
	2.2.7	Mechanisms for providing information when requested by operators
2.3	2.2.8	Autonomous detection and adjustment of speed to suit the distances and / or speeds of operators
2.14	2.2.9	Autonomous adaptation to varying environmental conditions, e.g. light and noise levels
	2.2.10	Standardised system programs / consistent behaviour, e.g. speed, procedure
	DESIRABLE	
2.8	2.2.11	Adaptability of position and configuration to suit operators' physical characteristics, e.g. height, reach
	2.2.12	Mechanisms for maintaining operators' awareness of process status
	2.2.13	Automatic detection of workforce / operator profiles
2.5	2.2.14	In-built functions to enable collaborative work (on shared tasks) with operators without physical guarding
	2.2.15	Ability to distinguish people from other system features and adapt behaviour
	2.2.16	Personalisable functions to satisfy individual operators' preferences, e.g. working methods, speed etc.
	2.2.17	Manually controllable functions that can be used by operators
	2.2.18	Autonomous detection and adaptation of functions in response to workers characteristics, e.g. skill, age, experience
2.18	2.2.19	In-built functions to enable co-existing work (on separate tasks but near to an operator) without physical guarding
2.17	2.2.20	Automatic feedback to management on process status / task completion
	2.2.21	Functions for teaching workers how to perform tasks
	2.2.22	Autonomous adaptation of programs to correspond with operators' capabilities and preferences

Table 47 User Requirements: Automation and Robotics

3. Communication and Interaction Mechanisms		
	ESSENTIAL	
Code		Description
Alpha	Beta	
3.1	2.3.1	Feedback mechanisms that show if the system has understood a command
	2.3.2	Visual display of messages to operators providing text or graphic notifications / work instructions
	2.3.3	Mechanisms for operators to provide feedback on process / task status
	2.3.4	Interaction capabilities that enable real-time query-response communications
	2.3.5	Interactive systems that enable operators to interrogate information / instructions more deeply when needed
	2.3.6	Interactive systems that allow operators to control functions and automation / robots
	2.3.7	Mechanisms for operators to report personal circumstances and concerns
	DESIRABLE	
3.11	2.3.8	Adaptable systems where workers can choose the mode of interaction they use
	2.3.9	Speech / voice messages to operators for providing auditory notifications / work instructions
	2.3.10	Mobile devices for communications and receiving notifications
	2.3.11	Mechanisms for operators to share practices/ problem solving solutions informally
	2.3.12	Wearable control devices
3.6	2.3.13	Systems that can be controlled using handheld mobile devices, e.g. tablet, smartphone
3.5	2.3.14	Combined visual and auditory messages for feedback and notifications
	2.3.15	Wearable devices for communications and receiving notifications
	2.3.16	Interactivity that enables operators to verify each completed task step / retrieve information on the next step
3.3	2.3.17	Fixed / static controls, e.g. mounted console / tablet
	2.3.18	Traditional computer based format for providing work instructions
3.10	2.3.19	Gesture control interaction systems
	2.3.20	Augmented reality devices that provide controls, e.g. 'Google glasses'
	2.3.21	Personalisable communication devices for sending / receiving individual updates and information
3.4	2.3.22	Non-defined voice command interaction systems (using natural speech with unlimited vocabulary)

Table 48 User Requirements: Communication and Interaction Mechanisms

4. Work System Feedback, Training and Assistance		
	ESSENTIAL	
Code		Description
Alpha	Beta	
4.10	2.4.1	Assistance provided when requested by operators
	2.4.2	Personalised ergonomic assessment from the system to identify individual operator needs
4.11	2.4.3	Automatic detection of / guidance for emergency and/or unexpected situations
	2.4.4	Standardised assistance and guidance provided by the system to suit all operators
4.9	2.4.5	Mechanisms for operators to directly input knowledge and process improvement ideas
	2.4.6	Provision of specific feedback when requested by operators
4.2	2.4.7	Constant availability of tools and equipment
	2.4.8	Automatic provision of tools / equipment when system detects need
	2.4.9	Generic ergonomic / postural guidance from the system
	DESIRABLE	
4.4	2.4.10	Augmented reality devices to provide remote assistance from other personnel / more experienced operators
	2.4.11	Automatic provision of assistance when system detects ergonomic needs of an operator
	2.4.12	Automatic assessment of operators' levels of satisfaction
	2.4.13	Augmented reality training methods that provides off- and on–the-job training for operators
	2.4.14	Tools / equipment provided bit the system in response to specific requests from operators
4.6	2.4.15	Personalised assistance provided by the system to suit operators' individual capabilities
5.8	2.4.16	Virtual reality simulation training methods to develop operator competencies 'off–the-job'
	2.4.17	Automatic detection and provision of assistance
4.1	2.4.18	Provision of tools / equipment at specific stages of assembly via system monitoring of progress
5.7	2.4.19	Augmented reality that provides information and instructions to workers while they are working
	2.4.20	Feedback from the work system to keep operators aware of overall system performance
	2.4.21	Feedback from the work system that keeps the operator aware of their own work progress
	2.4.22	Provision of ergonomic training from the work system
	2.4.23	Assistance / feedback from the system to keep operators satisfied as they work
	2.4.24	Continuous automatic feedback from the system

Table 49 User Requirements: System Feedback and Assistance

5. System Security and Data Management		
	ESSENTIAL	
Code		Description
Alpha	Beta	
	2.5.1	Systems automatically capture system performance data
	2.5.2	Systems automatically analyse data
	2.5.3	All workstations are password protected
6.1	2.5.4	Security mechanisms are required to prevent attacks from external sources
	2.5.5	Data is captured equally across all workstations and roles
	2.5.6	Identity cards are needed for operators to log on before using a workstation
6.2	2.5.7	Managers to have access to system data (on process, performance, etc.)
6.2	2.5.8	System data (on processes, performance, etc.) can be accessed by information technology maintenance personnel
	2.5.9	Access to operators' personal data is only available to personnel who work on the maintenance of information technology systems
	2.5.10	Systems automatically capture individual operator performance data
	DESIRABLE	
	2.5.11	Systems automatically report performance data to managers
	2.5.12	Operator data is retained for predefined limited period of time
	2.5.13	System data is retained for a predefined limited period of time
	2.5.14	Operator data for all workers is kept on the 'cloud' so that individual profiles can be downloaded when they log on to a workstation
	2.5.15	Data is only be captured / retained for specific operators / specific workstations
6.4	2.5.16	Operator data only includes ergonomic information, e.g. the height they set the workbench

Table 50 User Requirements: System Security and Data Management

ANNEX B APPLICATION SCENARIOS AND CHALLENGES (FROM D1.5)

Please find here below an extract of D1.5, listing the application scenarios and their challenges as identified in D1.5, in order to ease the understanding of codes used in Section 5. Any updated version of the list could be published as part of WP1, therefore the following list could suffer some misalignments.

Scenario Code	Scenario Name
SC1.1	Hydraulic system assembly
SC2.1	Deburring (auxiliary)
SC2.2	Main landing gear assembly
SC3.1	Collaborative assembly
SC3.2	Logistics (auxiliary)
SC4.1	Wing and bonnet assembly
SC4.2	Mobile tooling supply (auxiliary)

Table 51 Application scenarios (from D1.5)

Challenges	Scope
CH1.1- Adapted on the job guidance	Human, process & context variability
CH1.2- Adaptation of the tools involved in the assembly process	Process variability
CH1.3·- Decision support	Process variability
CH2.1- Including automation mechanisms in manual deburring process	Process variability
CH2.2- Safety and ergonomics	Safety and worker satisfaction
CH2.3- Information and documentation fragmentation	Process variability
CH2.4- Training	Human & Process variability
CH2.5- Knowledge management	Human & Process variability
CH3.1- Adaptation to worker's profile	Human variability
CH3.2- Adaptation of the robot behaviour based on	Context variability

Challenges	Scope
safety related criteria	
CH3.3- Adaptation of on the job guidance	Context & human variability
CH3.4- Integration with legacy systems	Process variability
CH3.5- Integration of auxiliary activities (i.e. maintenance, quality)	Process variability
CH3.6- Adaptation to new trends	Process variability
CH4.1- Adaptation to worker's experience	Human variability/ worker satisfaction
CH4.2- Training of inexperienced workers	Human variability
CH4.3- Picking individual variant parts	Process variability/ worker satisfaction
CH4.4- Picking joining parts	Process variability
CH4.5- Checking rear light function	Process complexity
CH4.6- Integration of auxiliary activities (i.e. maintenance, quality, quality assurance)	Process variability
CH4.7- Determining the optimal level of automation	Process variability/ Automation
CH4.8- Minimizing non-added value activities	Process efficiency
CH4.9- Improving ergonomics	Worker satisfaction, ergonomics
CH4.10- Enhancing area efficiency	Process efficiency
CH4.11- Error prevention	Process efficiency/ Worker satisfaction
CH4.12- Transparent decision on adaptive automation	Automation/ Worker satisfaction

Table 52 Challenges identified in use case scenarios (from D1.5)