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

GA number: 723828

Deliverable <2.1>

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

Partner responsible	ENGINEERING - INGEGNERIA INFORMATICA SPA
Contributors	TEK, RWTH, ILL, KOM
Type	Report (R)
Dissemination level	Public (PU)
Date:	28/07/2017
Version:	1.0
Due Date	M9
Status of the document	Final version
Total number of pages:	149
Suggested readers:	All
Keywords:	Reference Architecture, FBB Decomposition, Business Processes

Document History			
Ver.	**Date**	**Changes**	**Author**
0.1	17/02/2017	First draft of the ToC	ENG
0.2	24/02/2017	Added further descriptions in the sections; Added draft content in Section 3; Added Section 5	ENG, TEK
0.3	29/04/2017	Started filling Abbreviations and Glossary of terms sections; Main responsibilities assignment	ENG
0.4	29/05/2017	Updated section descriptions; Integrated TEK contributions; Added Annex A on "Requirements book" from D1.1	ENG
0.5	31/05/2017	Addressed main comments from partners	ENG, TEK, ILL
0.6	01/06/2017	Added Event Manager component descriptions; Added Annex B on "Challenges" from D1.2	ENG
0.7	08/06/2017	Addressed main comments from partners; Integrated contributions received so far	ENG, TEK, ILL
0.8	23/06/2017	Added new content as ENG contributions; Reversed order in FBB Specification sub-sections; Integrated TEK contributions on paragraphs 5.2.11, 6.3.4, 7.3.4, 8.3.4; Integrated ILL contributions on paragraphs 6.2.3 and 7.2.3.	ENG, TEK, ILL
0.9	10/07/2017	Added new content as ENG contributions; Completed Sections 2, 3 and 4; Integrated ILL contributions on paragraphs 5.2.6, 6.2.3, and 8.2.3 Integrated TEK contributions on Sections 5, 6, 7 and 8.	ENG, ILL, TEK
0.9.8	19/07/2017	Added minor contributions and conclusions	ENG, ILL, TEK, CESA, RWTH
0.9.9	27/07/2017	Minor changes following reviewers' feedbacks	ENG
1.0	28/07/2017	Final version ready for submission	ENG

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	Yvan Baudin	Minor improvements and text polishing
TEK	Francisco Javier Diez	General comments on module distribution into the three different domains and re-organization of F&M Architecture visual diagram

Executive Summary

The current deliverable (i.e. D2.1: Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system - Initial 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). However, to provide on time (i.e. D2.1 is due Month 9), relevant information to the starting or ongoing work on other technical work-packages (i.e. WP3, WP4, and WP5) it has been decided to include only high-level specification of the A4BLUE Platform in this initial version of the deliverable. Tasks T2.1 and T2.2 follow the incremental-iterative approach (alpha-beta loop) identified for the overall project, so it is still an ongoing activity, thus only initial results are included in D2.1 and the final results will be reported in M19 in the scope of D2.6 (Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system – Final Version).

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)"
- D1.1 - Requirements book and human factors best practice guidance - Initial 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).
- D1.2 - Scenarios definition - Initial Version: D1.2 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 D2.1 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 D2.1 were put in place along the duration of the initial iteration (M4-M9) 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
CWE	Collaborative Working Environment
DoA	Description of Actions
DyCEP	Dynamic Complex Event Processing
ECA	Event-Condition-Action
EFFRA	European Factories of the Future Research Association
EPA	Event Processing Agent
F2F	Face-to-Face
FBB	Functional Building Block
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
M2M	Machine-to-Machine
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
SDO	Standards Development Organization
SE	Specific Enablers
TA	Tangible Asset
ToC	Table of Contents
UGC	User Generated Content
UI	User Interface
VA	Virtualized Asset

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. D2.1: Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system - Initial 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). However, to provide on time (i.e. D2.1 is due Month 9), relevant information to the starting or ongoing work on other technical work-packages (i.e. WP3, WP4, and WP5) it has been decided to include also high-level specification of the A4BLUE Platform in this initial version of the deliverable, foreseeing further details in the outcomes of the other WPs.

Tasks T2.1 and T2.2 follow the incremental-iterative approach (alpha-beta loop) identified for the overall project, so it is still an ongoing activity, thus only initial results are included in D2.1 and the final results will be reported in M19 in the scope of D2.6 (Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system – Final Version).

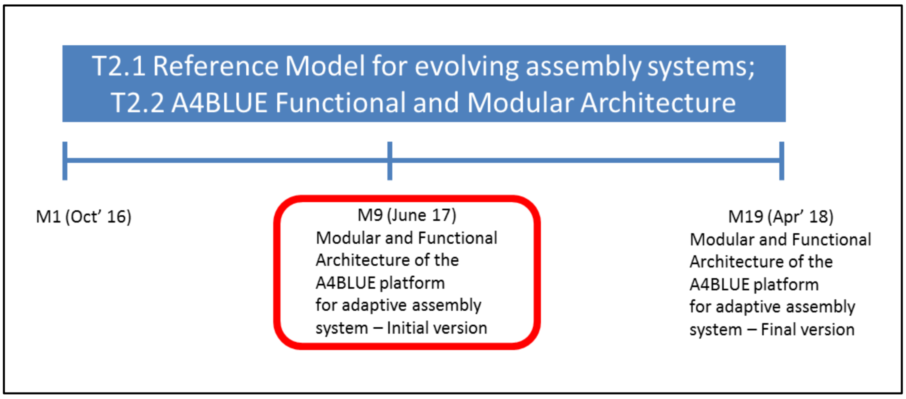


Figure 1 Tasks T2.1 and T2.2 timeframe and related deliverables

The purpose of both tasks T2.1 and T2.2 is to identify key challenges to be overcome by the A4BLUE as well as common and application specific requirement 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.

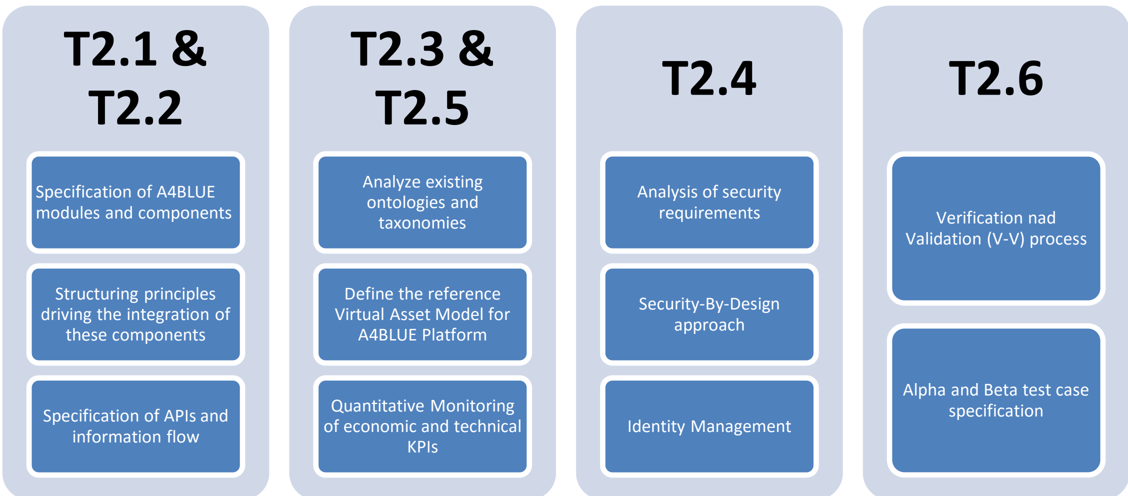


Figure 2 Relation of the tasks T2.1 and T2.2 with the rest of tasks in WP2

1.2 RELATIONSHIP WITH OTHER TASKS

D2.1 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 D2.1.
- **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 support 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 below.

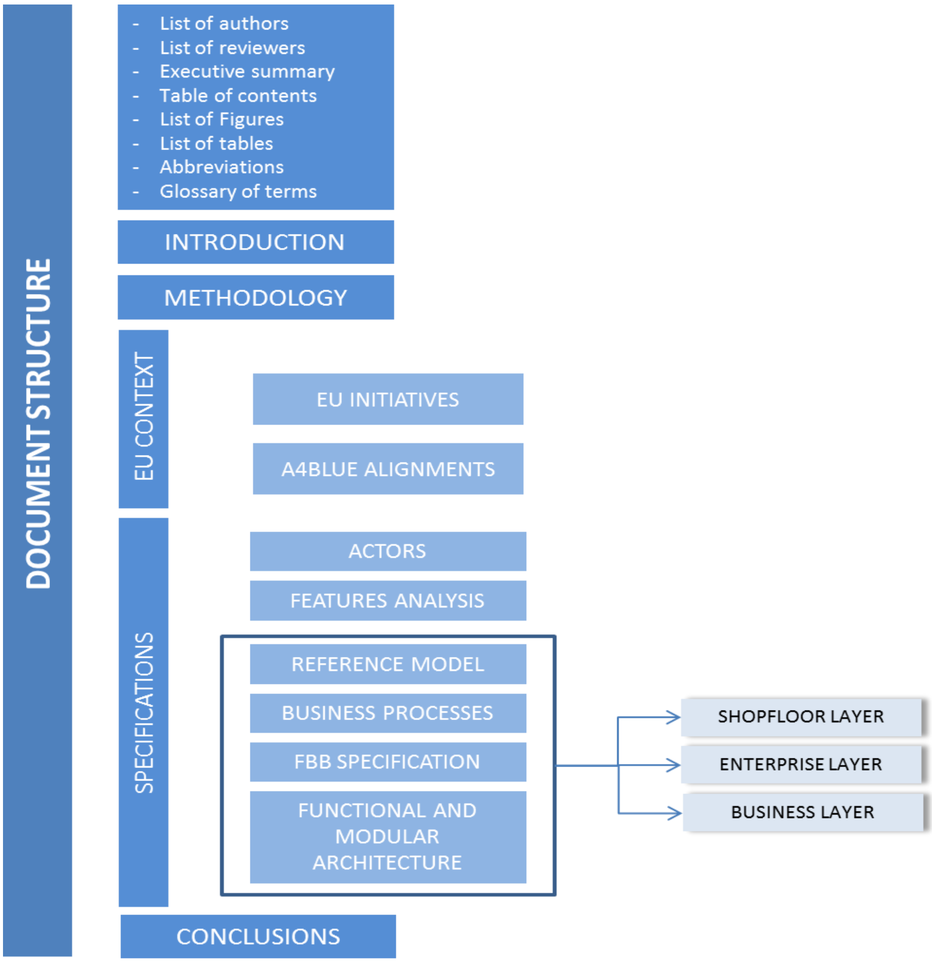


Figure 3 Diagram Document Structure

2 METHODOLOGY

2.1 REFERENCE IMPLEMENTATION

The A4BLUE Adaptive Framework will be designed 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.

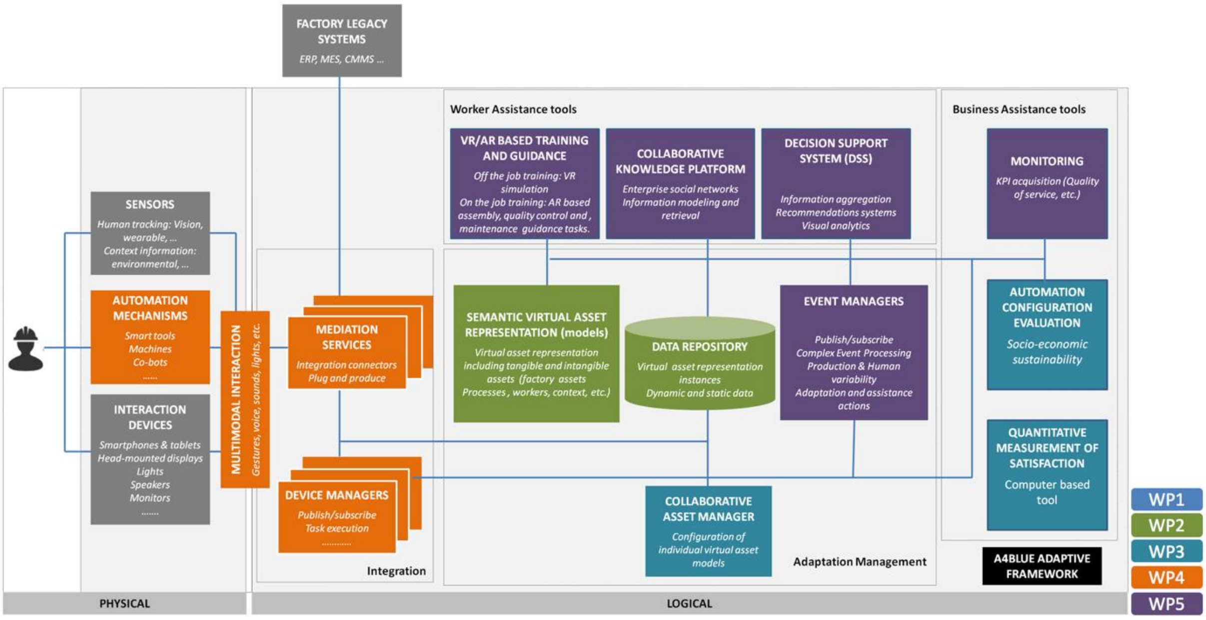


Figure 4 A4BLUE Reference Implementation (from DoA)

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 (i.e. robots) through gesture, voice, lights, sounds, 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 training (i.e. taking place in at the workplace in the real working situation) by using appropriate AR hardware and software components.
	Collaborative Knowledge Management	Facilitating the transfer of knowledge, especially implicit and informal ones, from skilled workers (e.g. older workers) to young or less experienced workers to help them taking 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. Itis aimed only to support the collection of the relevant performance metrics produced in the domain of the A4BLUE solution other kind of KPIs should be out of the scope of this component.
	Automation	An evaluation tool able to assess the levels of automation

Pillars	Reference Modules	Description
	Configuration Evaluation	(assembly system configurations/assembly process) from a socio-technical as well as from an economical perspective. The results socioeconomic evaluation 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 the involved shop floor operators to complete the worker satisfaction questionnaires in an easy way and evaluates 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. 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"

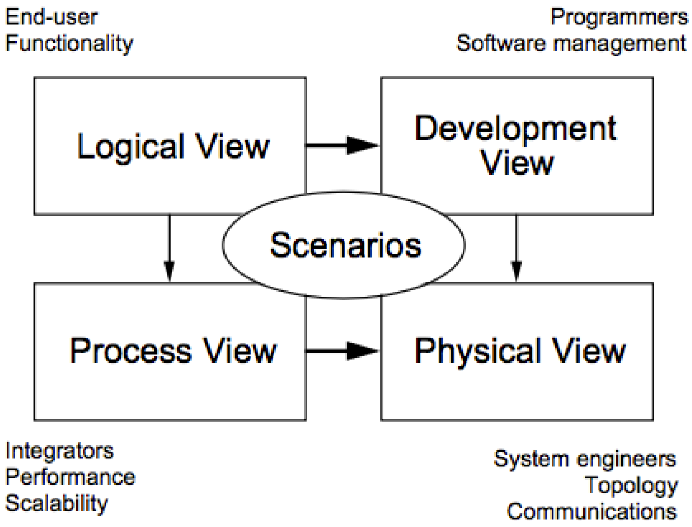


Figure 5 The "4+1" view model

Apart from adherence to the 4+1 view approach, the present deliverable has taken account prior WP1 developments (notably the requirements described in D1.1 and the use cases described in D1.2) 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 D1.1, 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 to notice how Different levels of an architecture description could include both design and implementation aspects, as summarized in the following Figure 6.

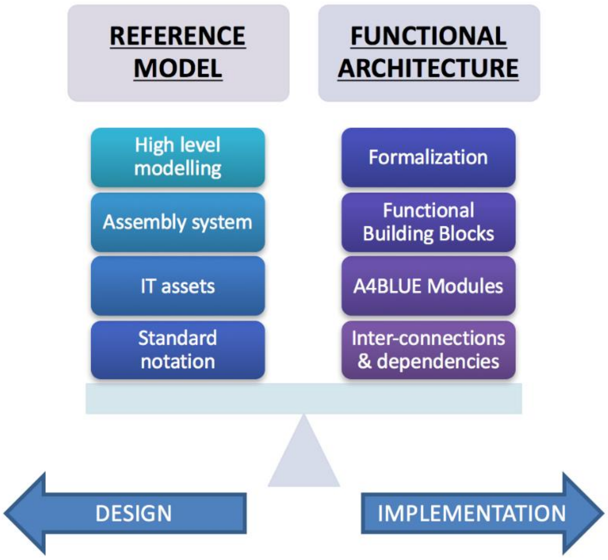


Figure 6 Architecture description: Design vs. Implementation phases

The Reference Model will ideally connect the domain of the problem (as described in WP1 outcomes, namely D1.1 and D1.2) 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 process envisioned to derive from the Reference Model a FBB Specification, the following Figure 7 shows the suggested iterative and incremental approach.

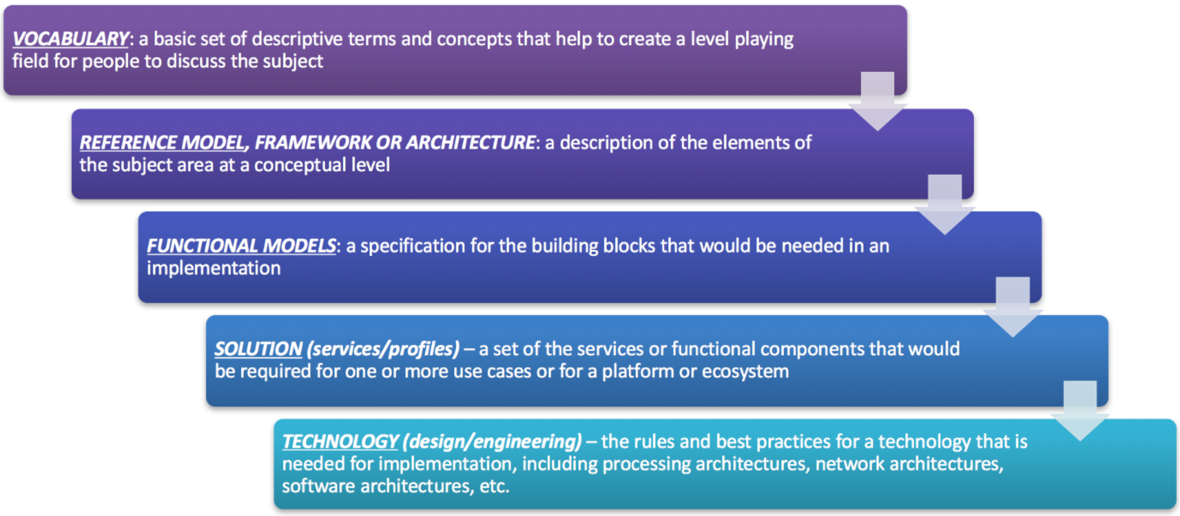


Figure 7 Architecture description: iterative and incremental approach

Starting from defining a grounding 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 block needed for the implementation stage (namely in Section 8 – "FBB SPECIFICATION"). Here we pass to the domain of the solution, by starting the definition of a set of services that would be required by one or more application scenario identified in WP1, to be further described in terms of background assets and technologies to be used for implementing such service, by describing their decomposition in sub-components and 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 problem 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 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).

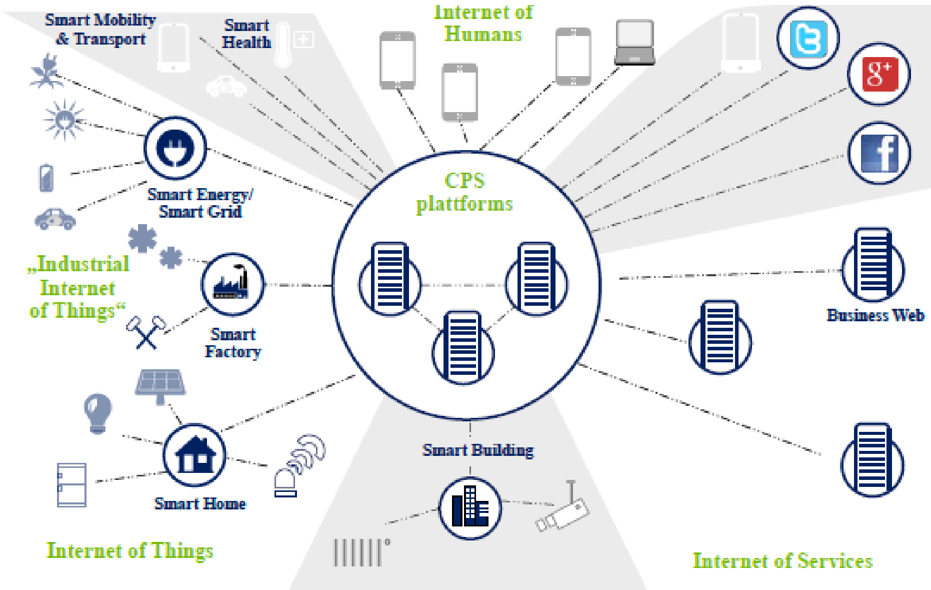


Figure 8 Cyber-physical systems: the core of IoT and IIoT

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, 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 speed up [1].

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

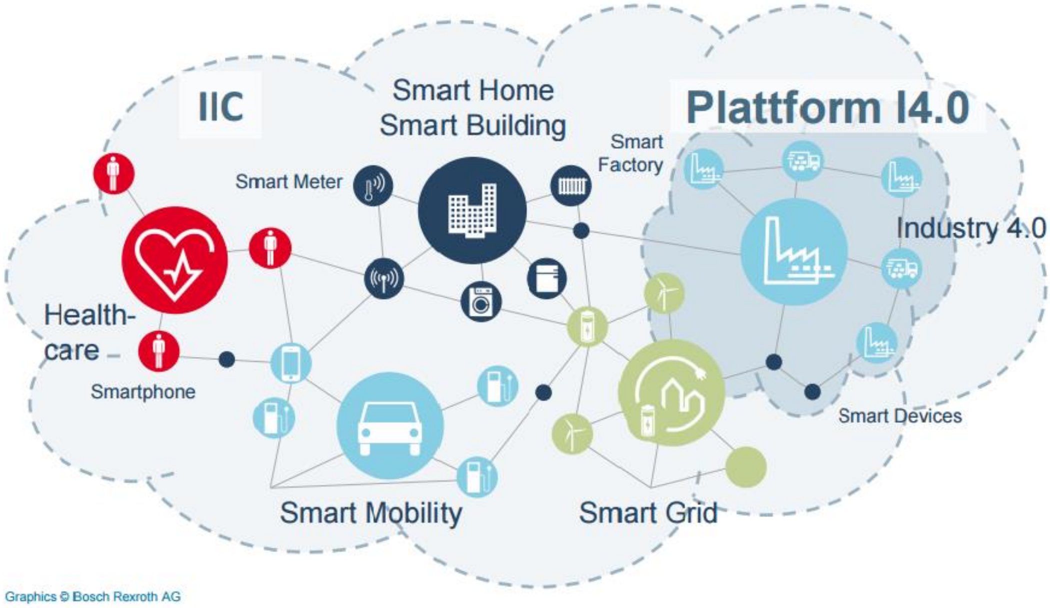


Figure 9 Industrie 4.0 in the Internet of Things and Services

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's 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

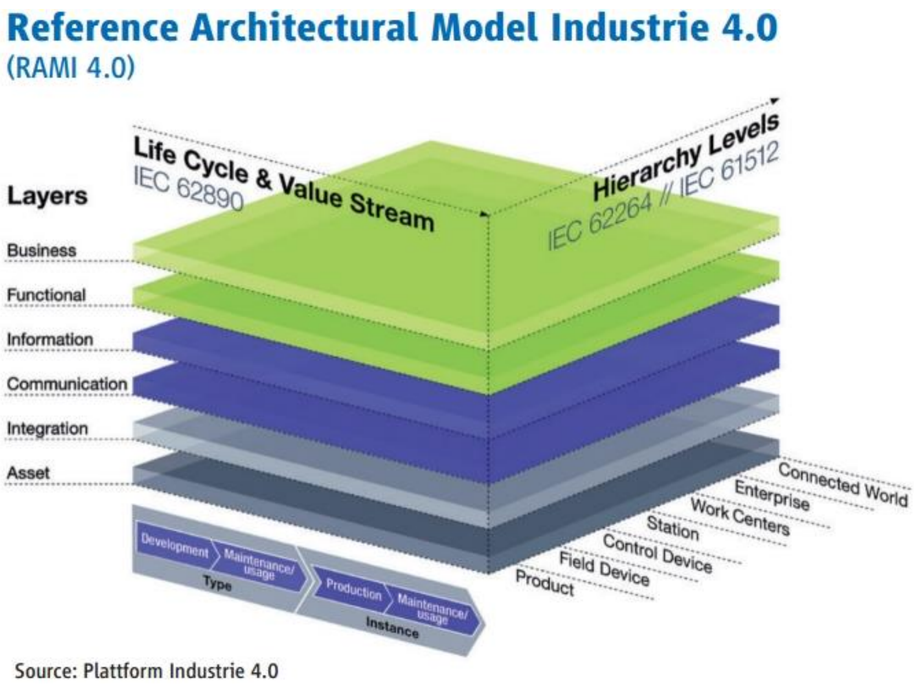


Figure 10 RAMI 4.0 Reference Architecture

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).

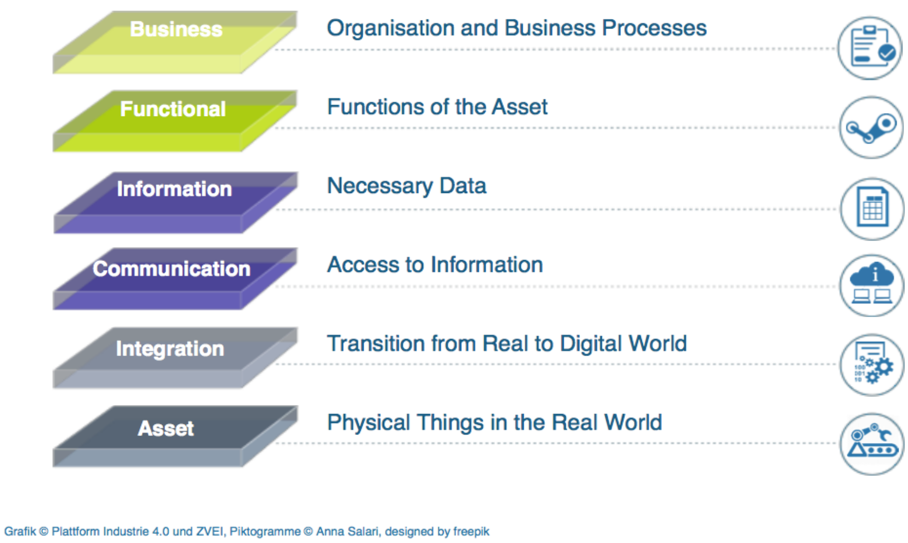


Figure 11 RAMI 4.0 Layers

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 objective of RAMI is to provide an end-to-end (i.e., since the inception of the product's idea, till its dismantling or recycling) framework able to connect and consistently correlate all technical, administrative and commercial data so 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 at defining 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*. Even if not necessarily an *I4.0 component* physically embeds the *object(s)* it represents, logically the relationship between an *I4.0 component* and its *object(s)* is the one represented in Figure 12.

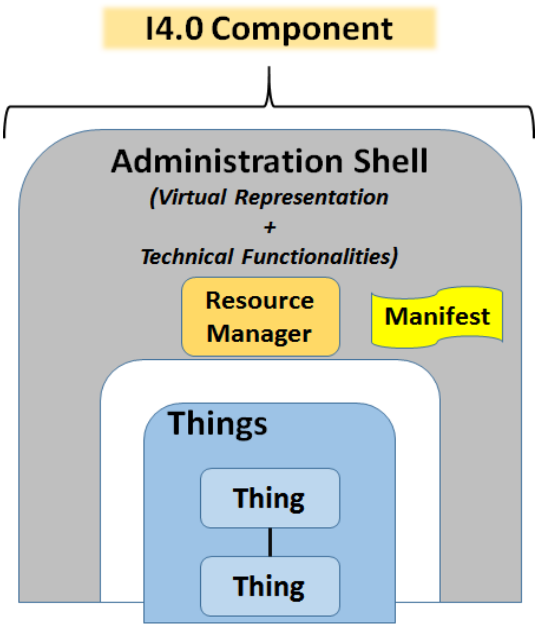


Figure 12 I4.0 Component

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 reuse 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.
- 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.

[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 *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.

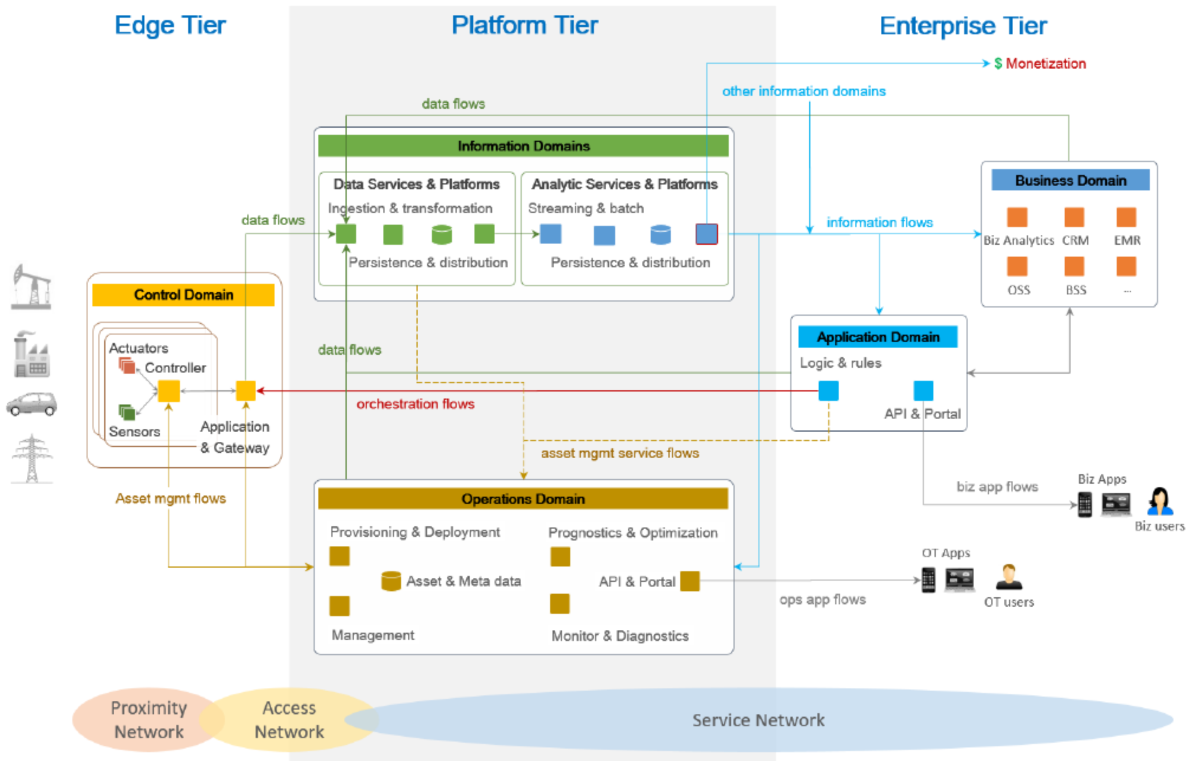


Figure 13 IIRA Three-Tier Architecture Pattern (implementation + functional viewpoint)

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.

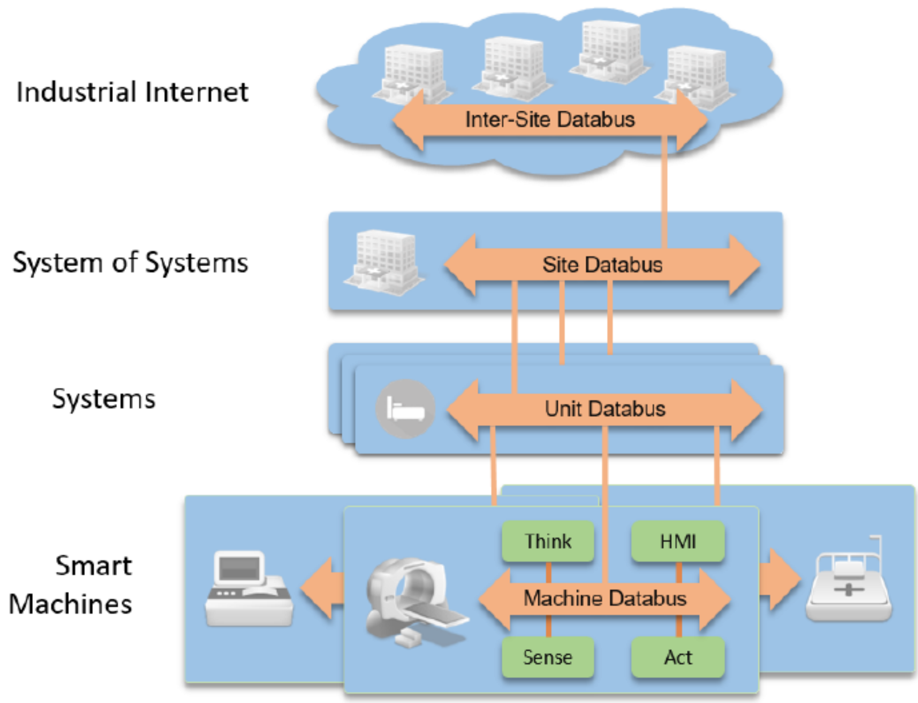


Figure 14 IIRA Layered Databus Architecture Pattern (implementation viewpoint)

3.3 FITMAN

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 (FW4I)** 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. FW4I 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 15 below, borrowed from the public FW4I website, illustrates IIOT-RA's components and their mutual relationships.

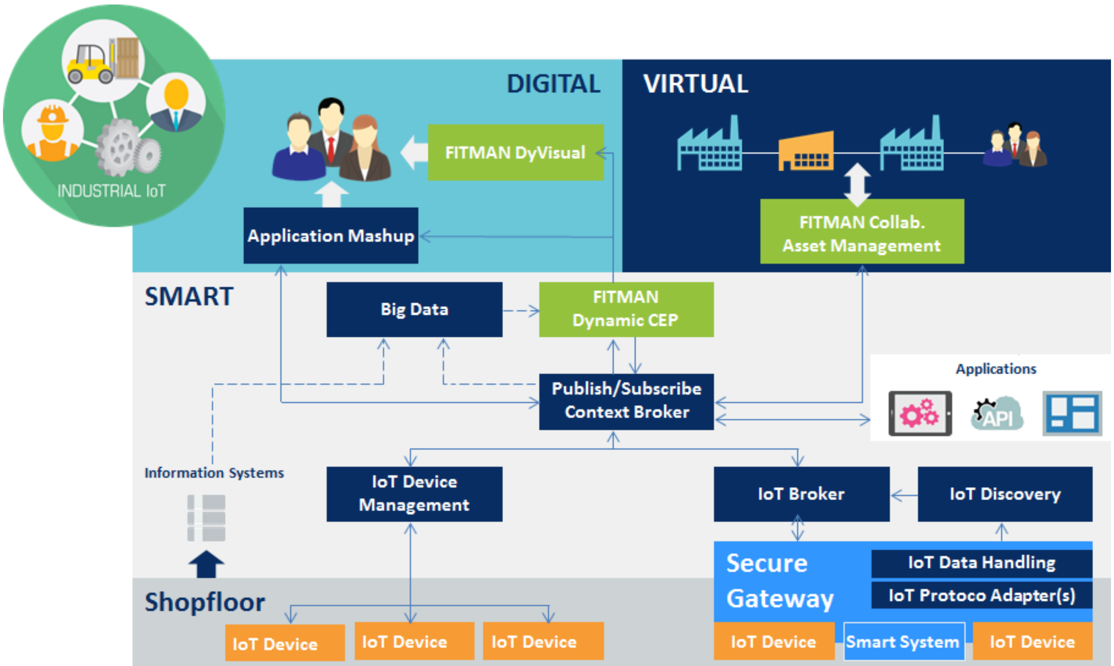


Figure 15 FIWARE for Industry IIOT Reference Architecture

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

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

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 no 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.

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 dynamicity 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

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

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.

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.

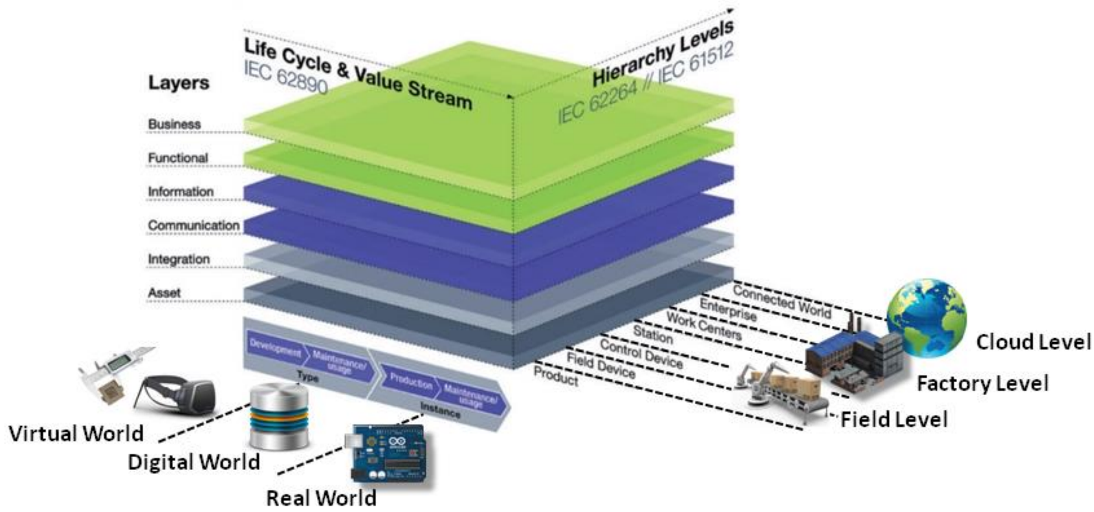


Figure 16 Mapping RAMI axis with real world entities

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 is 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 FITMAN IIOT-RA

The FITMAN IIOT-RA fits well with A4BLUE since IIOT-RA is concerned with the integration of the real and the digital worlds, which happens with the mediation of IoT. IIOT-RA's Smart and Digital domains roughly correspond to A4BLUE Shopfloor and Enterprise layers (further described in Section "6 – REFERENCE MODEL"). However, the Virtual domain in IIOT-RA has a completely different scope with respect to A4BLUE Business layer: the former is about collaborative enterprise networks / extended supply chains / cloud manufacturing – all scenarios that go under the collective name of Virtual Factory – while the latter focuses on the digital representation of business processes and systems supporting business decisions. That said, the IIOT-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, IIOT-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 D1.1 (Requirements book) and D1.2 (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 D1.2. 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

Table 2 Main actors identified in A4BLUE application scenarios

This table did not consider explicitly 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.

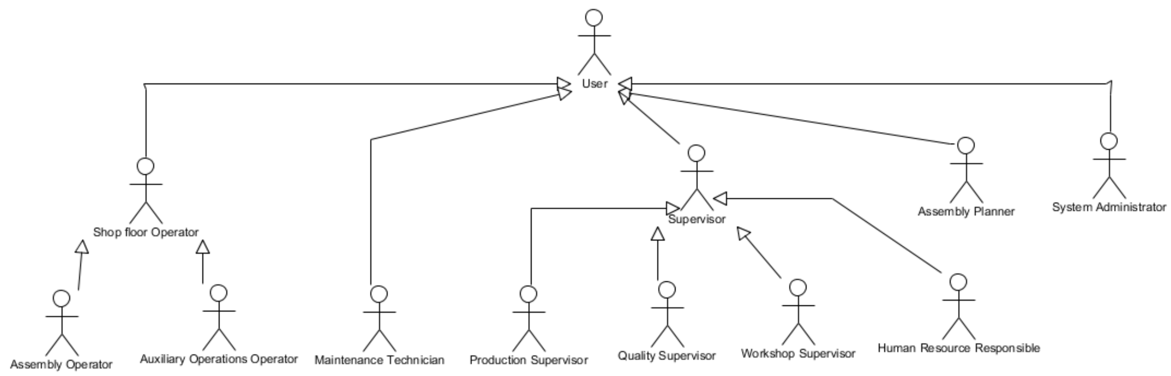


Figure 17 Involved actors

5.2 FEATURES ANALYSIS

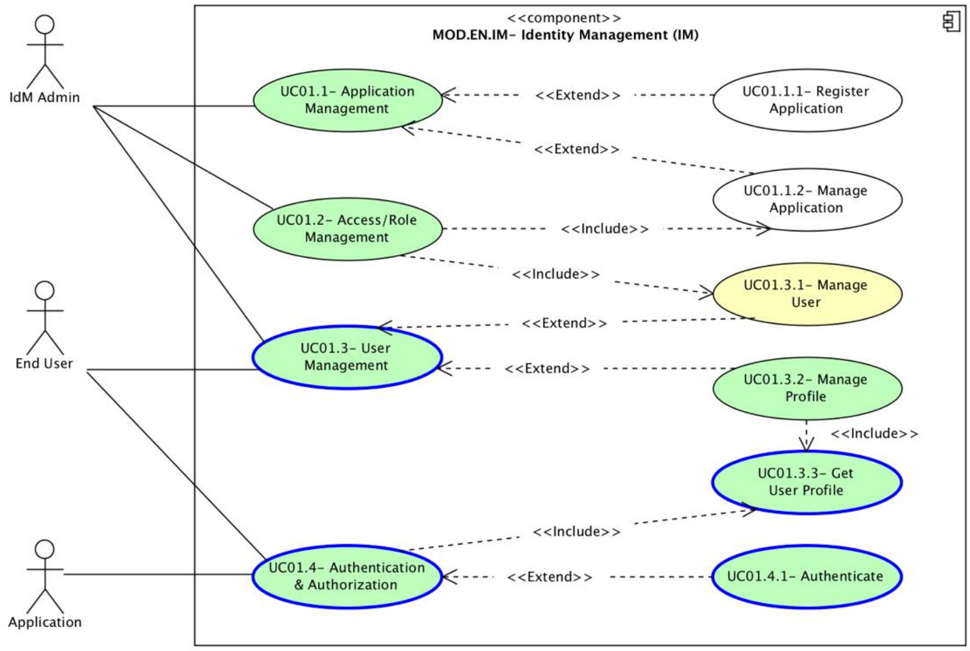
The following table represents the notation followed by the use case included in the sections below. This will be used as a criterion to prioritise functionalities and implementation effort.

(green)	*Identified as essential requirement in D1.1*
(yellow)	*Identified as desirable requirement in D1.1*
(blue border)	*Identified in at least a use case application in D1.2*

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 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 Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1, SC3.2, SC4.1, SC4.2
Related challenges	CH1.1, CH2.2, CH3.1, CH3.3, CH4.1, CH4.2

FTR.01.- Identity management	
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

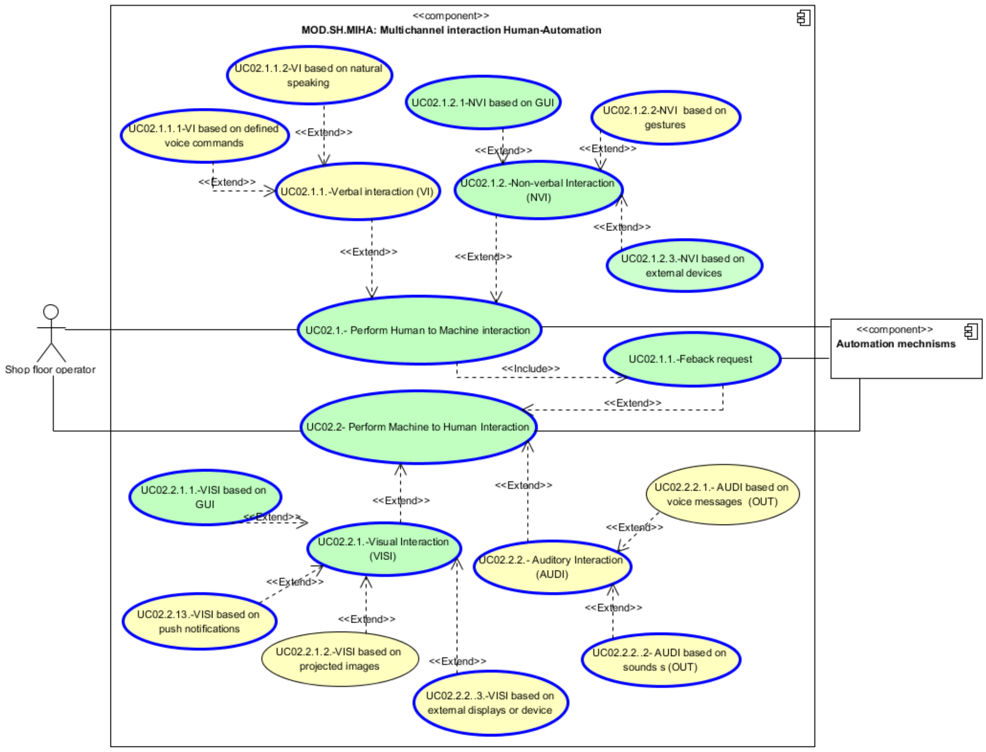
UC.01.3 User Management	
Objective	Manage user identity and user profile.
Initiation	On demand

UC.01.3 User Management	
Flow of Events	• End User 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

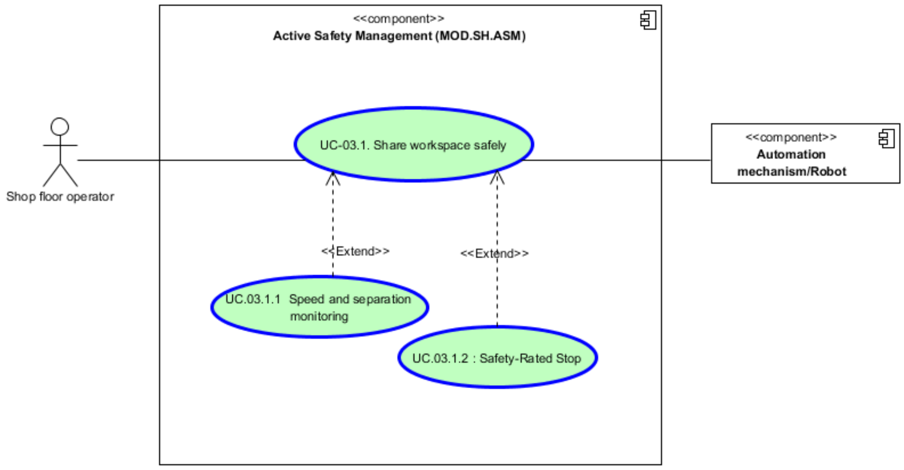
FTR.02.- Multimodal Human- automation/robot Interaction	
Objective	Provide multimodal, multichannel interaction capabilities to allow shop floor operators to communicate with the automation/robots in an easy and intuitive way.
Involved actors	Shop floor Operators
Involved reference components	MOD.SH.MHMI
External interfaces	Interaction devices, sensors and actuators
Addressed user level requirements	R3.1, R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R3.8, R3.9, R3.10, R3.11
Related Application scenarios	SC1.1, SC2.1, SC3.1, SC4.2
Related challenges	CH2.1, CH3.1, CH3.6, CH4.1, CH4.2, CH4.6, CH4.8, CH4.9, CH4.10

FTR.02.- Multimodal Human- automation/robot Interaction	
Involved Use cases	

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.

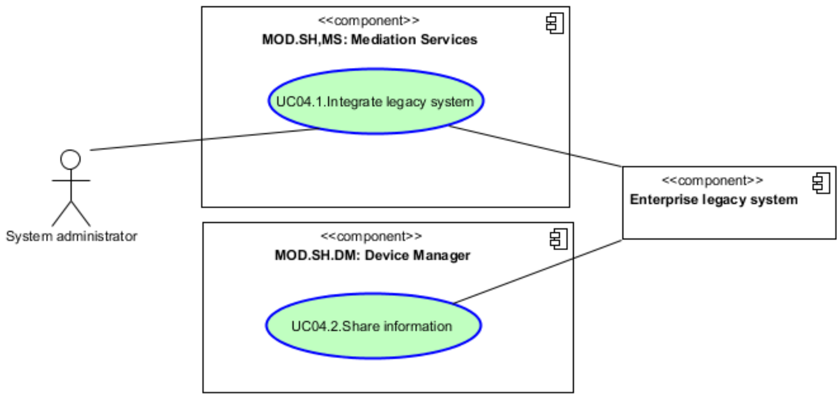
5.2.3 FTR.03- Safe Human - Automation/Robot co-existence

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, R2.2, R2.3, R2.4, R2.5, R2.9, R2.15
Related Application scenarios	SC2.1, SC3.1, SC3.2, SC4.2
Related challenges	CH2.2, CH3.2
Involved Use cases	

UC.03.1 Share workspace safely	
Objective	Provide active safety mechanisms to adapt the behaviour of the automation/robot, considering operators profile, 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)
Flow of Events	• The system identifies a potential safety issue due to the detection of an

UC.03.1 Share workspace safely	
	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 profile, if available.

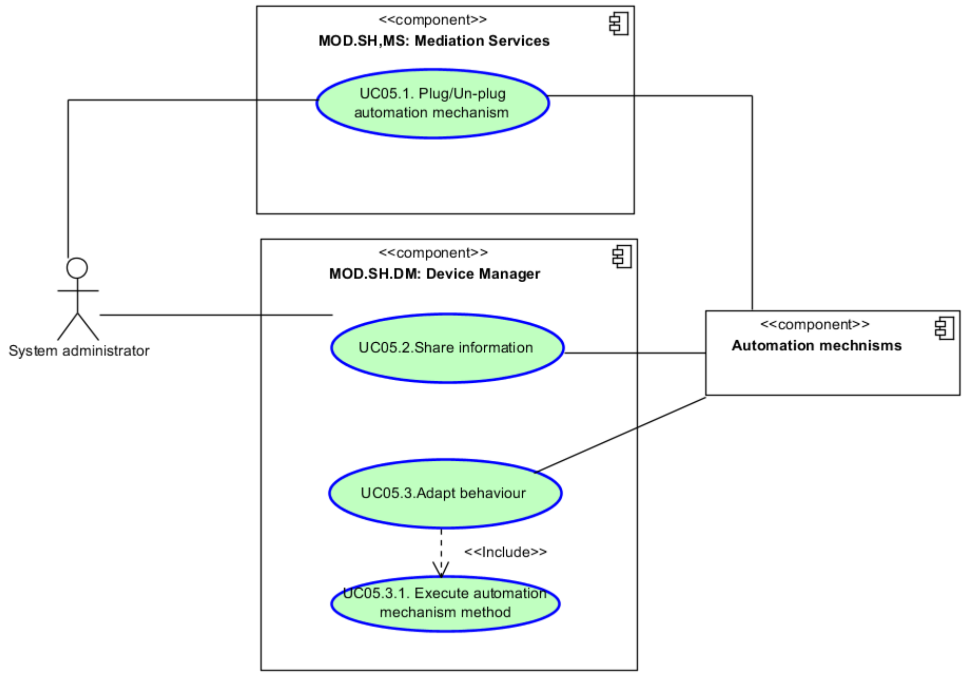
5.2.4 FTR.04- Integration with enterprise legacy systems

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.SH.DM
External interfaces	Enterprise legacy systems (e.g. MES)
Addressed user level requirements	R1.4
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1, SC4.1
Related challenges	CH3.4
Involved Use cases	

UC.04.1 Integrate legacy system	
Objective	Allow the system administrator to integrate the legacy system.
Initiation	On demand
Flow of Events	• The system administrator integrates the legacy system • The system discovers the exposed variables and methods and registers/de-registers them. If relevant a register/ de-register event is published.

UC.04.2 Share information	
Objective	Share the required information (i.e. data to be monitored: e.g. work order start/end; operation validation, etc.) during the operation.
Initiation	On event
Flow of Events	The legacy systems share the required information with the A4BLUE system when available.

5.2.5 FTR.05- Adaptive automation mechanisms

FTR.05.- Adaptive automation mechanisms	
Objective	Enable seamless integration of the automation mechanisms (e.g. robot, smart tools, etc.) and adaption to human, process and context variability
Involved actors	System administrator
Involved reference components	MOD.SH.DM, MOD.SH.AM
External interfaces	N/A
Addressed user level requirements	R1.3, R1.7, R1.12, R4.10
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1, SC3.2, SC4.2
Related challenges	CH1.3, CH2.1, CH2.2, CH3.5, CH3.2
Involved Use cases	

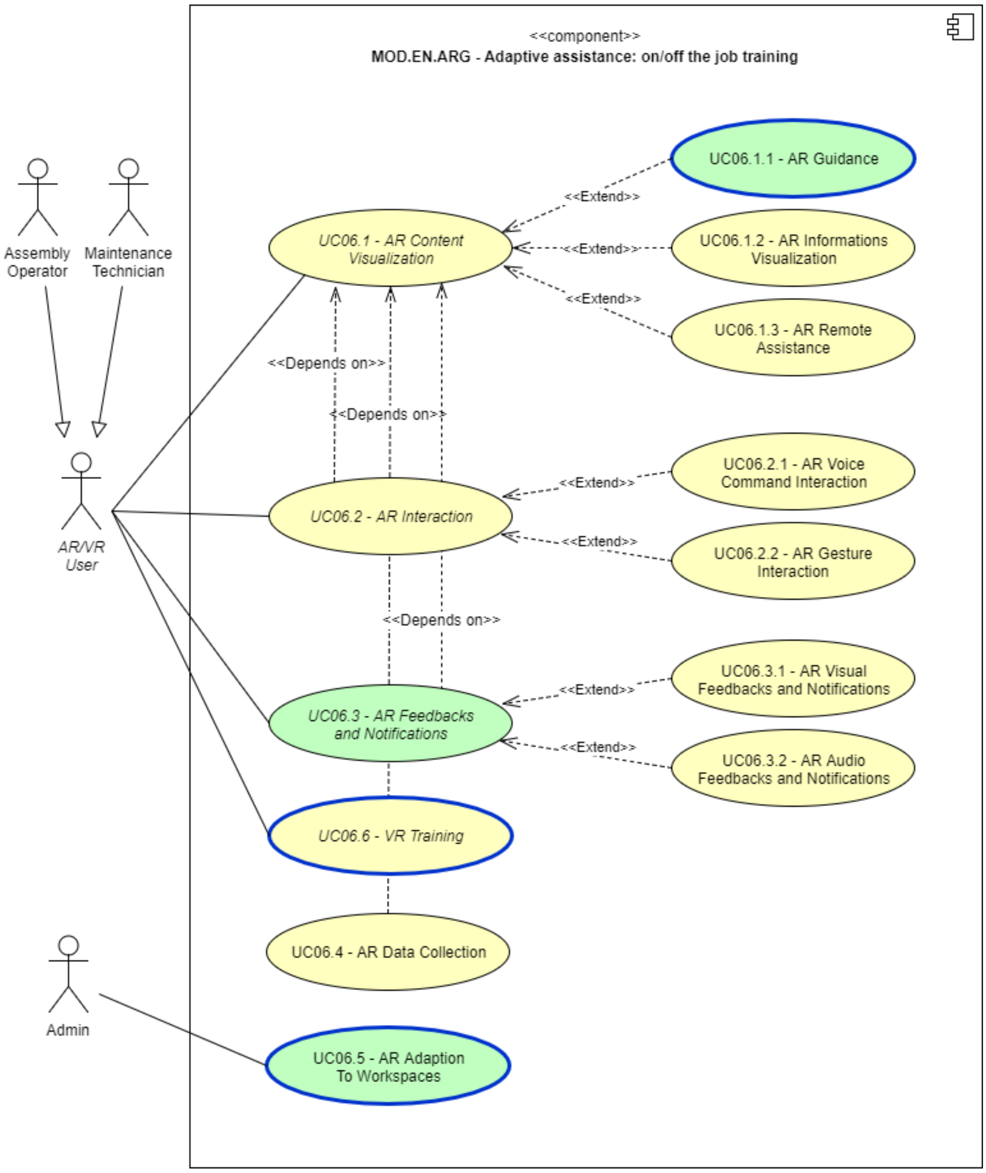
UC.03.1. Plug/ Un-plug automation mechanism	
Objective	Allow the system administrator to plug/un-plug the automation mechanism following a “plug & produce” approach.
Initiation	On demand
Flow of Events	• The system administrator plugs/un-plugs the automation mechanism • The system discovers the exposed variables and methods and registers/de-registers them. If relevant a register/ de-register event is published.

UC.03.2. Share Information	
Objective	Support operation by sharing the required information (e.g. status of the automation/robot, process data, etc.).
Initiation	On event
Flow of Events	• The system and the automation mechanisms share the required information (bi-directional).

UC.03.3. 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. including human and process status).

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

UC.06.1 Augmented Reality Content Visualization	
	• 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

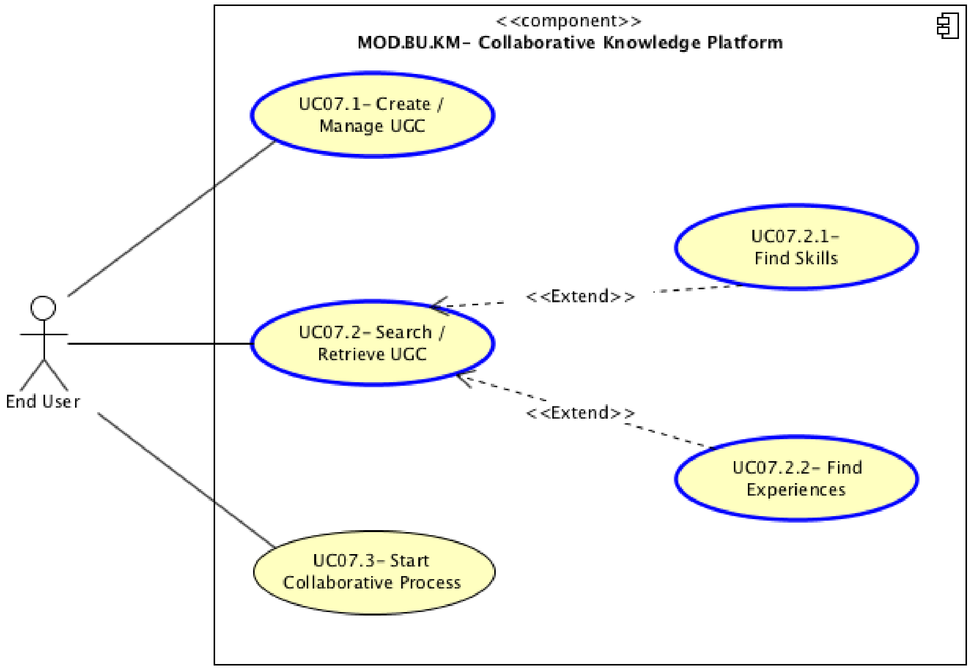
UC.06.4 Augmented Reality Data Collection	
Objective	Continuously collect data for analysis of system performance and optimisation needs.
Initiation	Automatic and continuous
Flow of Events	• The operator starts an AR visualization session (see UC.06.1) • The AR device continuously collects data about 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	
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	

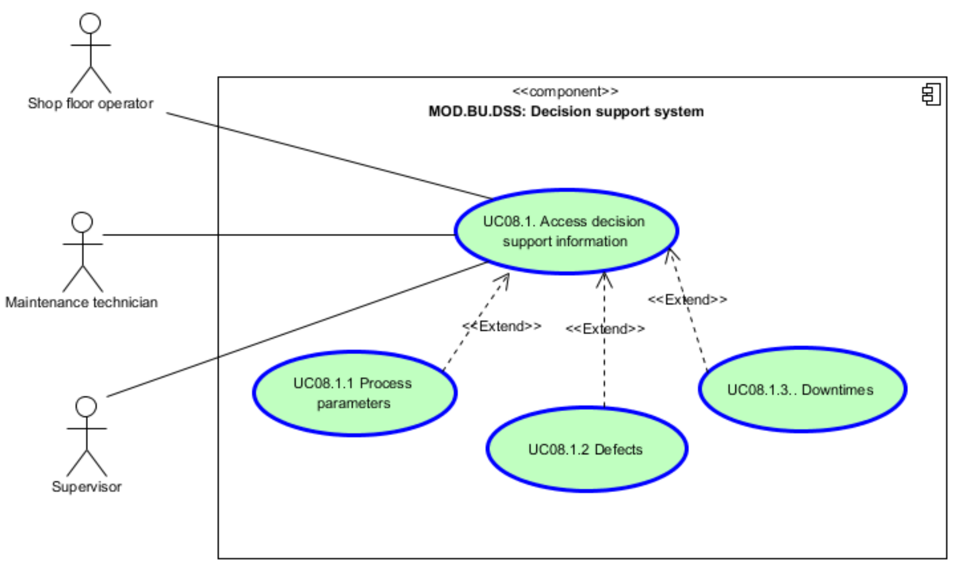
UC.07.1 Create/Manage UGC	
Objective	Create and manage User Generated Content (UGC).
Initiation	On demand
Flow of Events	• End User creates, manages and enriches his/her own UGCs • End User enriches UGCs created by other members of the MOD.BU.KM • The system provides feedbacks on the result of the process

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 o End User browses and retrieves skills o 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

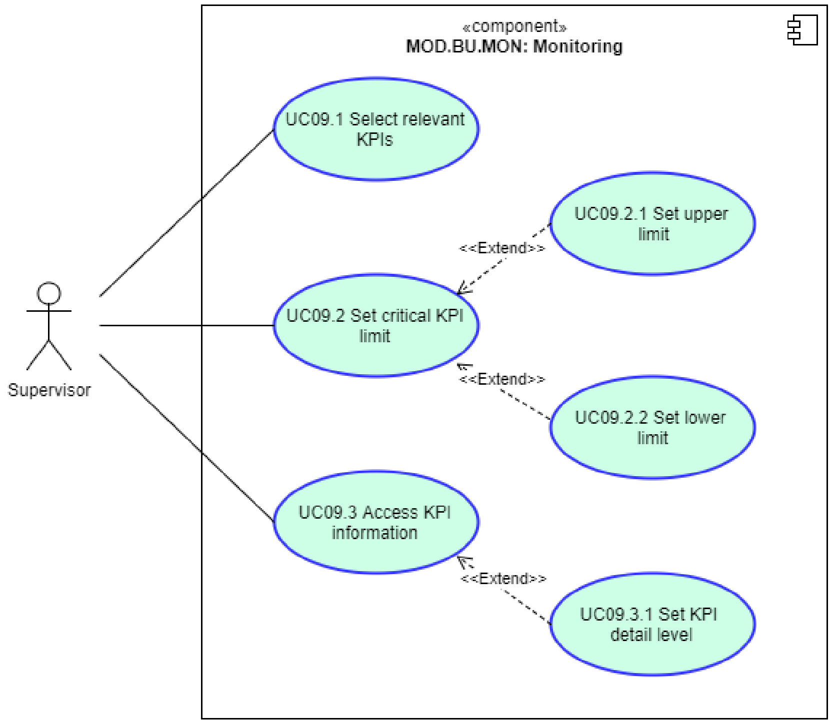
5.2.8 FTR.08- Adaptive assistance: decision support

FTR.08.- Adaptive assistance: decision support	
Objective	Support decision making related to assembly, 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, supervisors, maintenance technician
Involved reference components	MOD.BU.DSS
External interfaces	N/A
Addressed user level requirements	R1.5, R1.7
Related Application scenarios	SC1.1, SC3.1, SC3.2, SC4.1
Related challenges	CH1.3 CH3.5, 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

UC.08.1. Access decision support information	
	workers decision making process though visual analytics capabilities.
Initiation	On demand
Flow of Events	• The users access visual information collected and processed by the system,

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, Assembly Planner
Involved reference components	MOD.BU.MON
External interfaces	N/A
Addressed user level requirements	R1.1, R1.6, R4.9, R4.16, R6.2
Related Application scenarios	SC1, SC2, SC3, SC4
Related challenges	CH1.3, CH4.7
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 • The system measures and provides the selected KPIs at the KPI dashboard

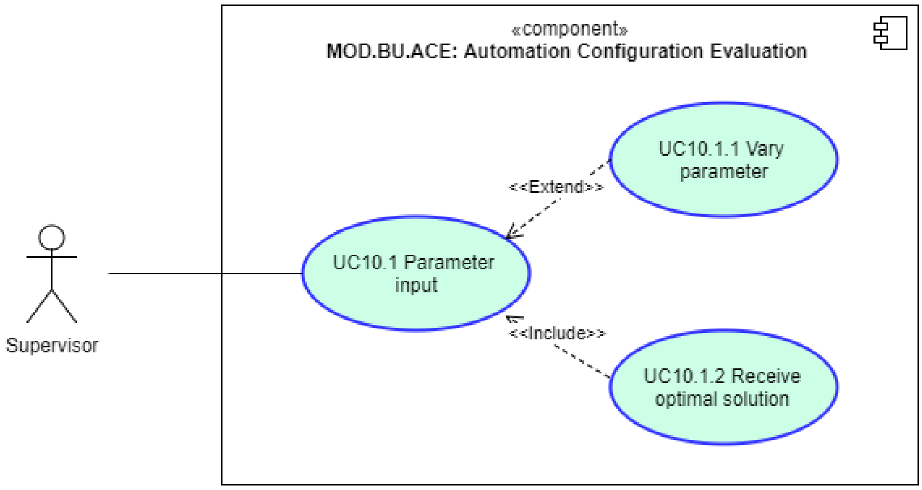
UC.09.2. Set critical KPI limit	
Objective	Set the range within the KPI value is acceptable
Initiation	On demand

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 information	
Objective	Enable supervisors or Assembly Planners to access relevant performance KPIs.
Initiation	On demand
Flow of Events	• Supervisors or Assembly Planners access provided performance KPIs collected and processed by the system ○ Supervisors set required KPI level of detail

5.2.10 FTR.10- Automation level definition support

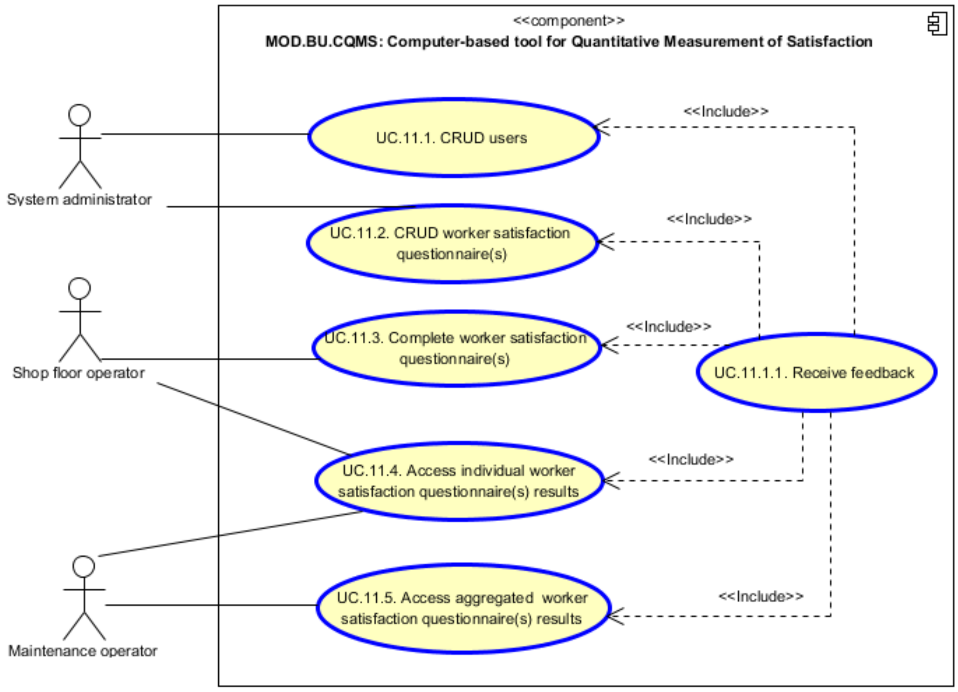
FTR.10.- Automation level definition support	
Objective	Provide a tool to identify the optimal automation configuration through varying criteria and process parameters from a socio-technical as well as from an economical perspective.
Involved actors	Assembly Planner
Involved reference components	MOD.BU.ACE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

FTR.10.- Automation level definition support	
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	R1.19, R3.1
Related Application scenarios	SC1.1, SC2.1, SC2.2, SC3.1, SC4.2
Related challenges	CH2.1, CH2.2, CH2.3, CH2.4, CH2.5, CH3.1, CH3.6, CH4.1, CH4.2, CH4.6, CH4.8, CH4.9, CH4.10

<table>
<tr><th colspan="2">FTR.11.- Worker satisfaction assessment</th></tr>
<tr><td>Involved Use cases</td><td>
</td></tr>
<tr><td>Other specifications</td><td>Support online (on event based) and offline (stand alone, on schedule) operation.</td></tr>
</table>

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. • 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.

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).

The reference components are listed here below:

- MOD.SH.DM- Device Manager
- MOD.SH.MS- Mediation Services
- MOD.SH.AM- Automation Mechanisms
- MOD.SH.MMI- Multimodal Interactions
- MOD.EN.CAM- Collaborative Asset Manager
- MOD.EN.EM- Event Manager
- MOD.EN.ARG- VR/AR based training and guidance
- MOD.BU.KM- Collaborative Knowledge Platform
- MOD.BU.DSS- Decision Support System (DSS)
- MOD.BU.ACE- Automation Configuration Evaluation Tool
- MOD.BU.CQMS- Computer based tool for the measurement of worker satisfaction

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.

6.1 SHOPFLOOR LAYER

6.1.1 MOD.SH.AM- AUTOMATION MECHANISMS

Automaton mechanisms in the scope of A4BLUE involve both smart tools and robots Figure 18 shows the components diagram representing the decomposition of the MOD.SH.AM component into Functional Building Blocks (FBBs) and the relations and data flows between them.

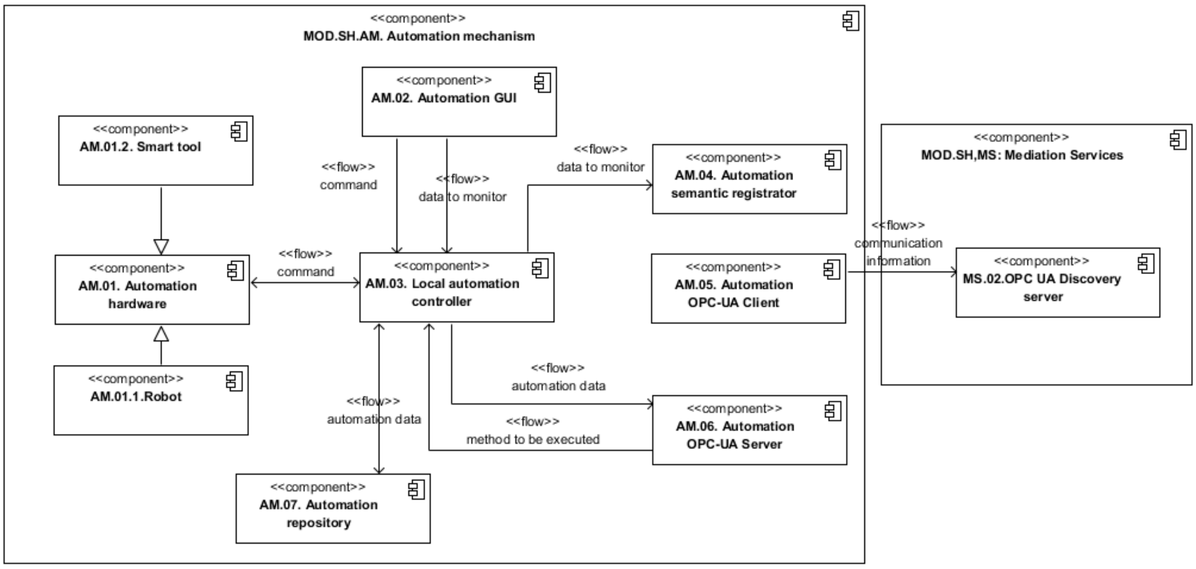


Figure 18 MOD.SH.AM. High Level module decomposition

Figure 20 included in the following section, provides further details about the integration of the automation mechanisms with the A4BLUE system.

The “AM.03. Local automation controller” controls the “AM.01. Automation hardware” (e.g. robot). AM.01 can involve auxiliary devices (e.g. camera, force sensor, etc.) to improve the accuracy of the process in order to get a better final result.

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 integration with the A4BLUE solution. This involves:

- AM.04. Automation semantic registrator: registers automation semantic information and identifies the automation data to be monitored.
- AM.05. Automation OPC-UA Client: registers communication information (IP, port) into the MS.02 OPC UA discovery server.
- AM.06. Automation OPC-UA Server: provides updates of the monitored automation data and executes automation methods.

Table 4 summarises the Functional Building Blocks (FBB) involved in MOD.SH.AM identifying the type and scope.

FBB	Type			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 semantic registrator					
AM.05. Automation OPC-UA Client					

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
AM.06. Automation OPC-UA Server					
AM.07. Automation data repository					

Table 4 MOD.SH.AM Functional building blocks summary

6.1.2 MOD.SH.MS- MEDIATION SERVICES

The Mediation Services and Device Manager components closely interact to support the plug & produce approach involving both the discovery and operation processes of automation mechanisms and legacy systems. Due to this strong coupling, it is not easy to describe them separately; therefore, only high level information on MOD.SH.MS will feed this section, while further descriptions will be part of next section.

The MOD.SH.MS- Mediation services component includes plug-and-produce capability to connect the automation as CPS into the A4BLUE framework. It involves the functional capabilities of the automation mechanisms to be recognized and used by the productive system (i.e. discovery process). Nowadays legacy systems as Manufacturing Execution systems (e.g. SAP MII, MES in SC3, etc.) involve OPC-UA capabilities to integrate with external systems. As an initial approach, to be revised in the scope of Task 4.1 (CPS enabling connectivity and management) once detailed inputs on the legacy systems to be integrated is available from Task 1.2 (Scenarios definition), the integration of the legacy systems will follow an analogous approach to the integration of automation mechanisms.

6.1.3 MOD.SH.DM- DEVICE MANAGER

The MOD.SH.DM- Device manager component supports the operation process by monitoring data, executing methods and managing the events from the automation mechanisms and legacy systems through publish/subscribe features.

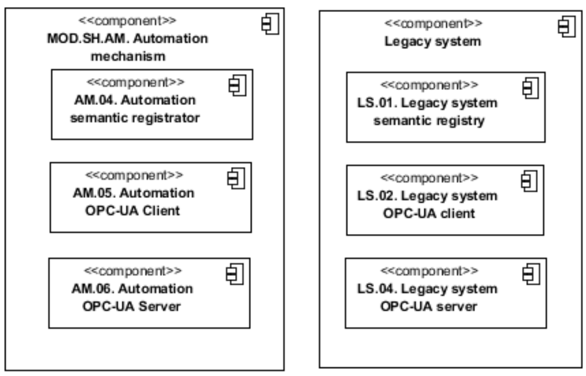


Figure 19 MOD.SH.MS & MOD.SH.DM. Automation and legacy systems OPC-UA approach

Figure 20 shows the components diagram representing the decomposition of the MOD.SH.MS and MOD.SH.DM into Functional Building Blocks (FBBs) and the relations and data flows between them for the integration of automation mechanisms. The integration of legacy systems should follow and analogous approach. Furthermore, it includes the interaction with the Event Manager (MOD.EN.EM), through the EM.01 Publish/Subscribe Context Broker (CB).

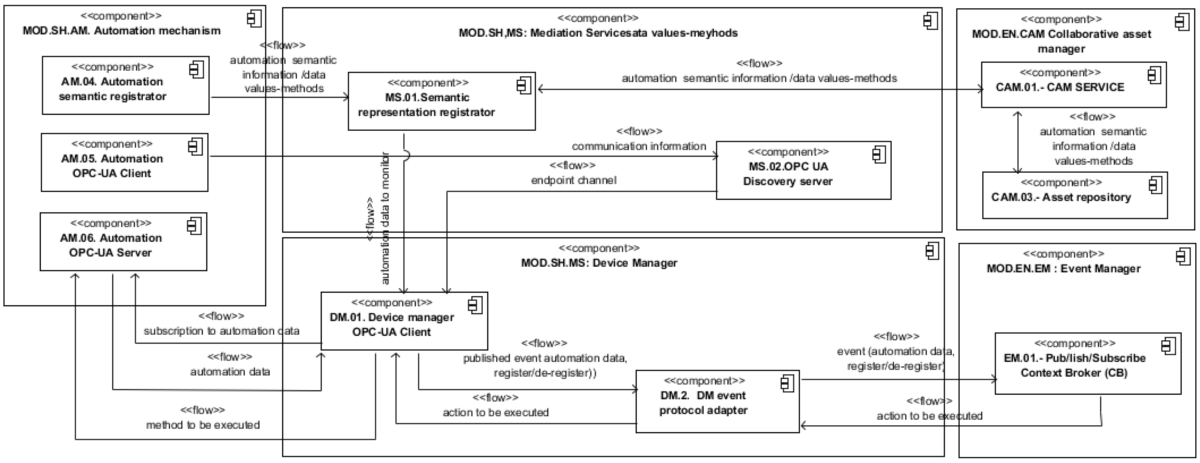


Figure 20 MOD.SH.MS & MOD.SH.DM. High Level module decomposition – automation mechanisms integration

The business logic of the discovery process of both automation mechanisms and legacy systems is mainly supported by the “MOD.SH.MS- Mediation services “component and involves:

- MS.01. Semantic representation registrator: stores the semantic representation provided by the automation mechanisms and legacy systems as well as the data to be managed into the “CAM.03. Asset repository” through the “CAM.03.CAM SERVICE” of the MOD.EN.CAM Collaborative Asset Manager”.
- MS.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.

The discovery process is initiated only when integrating the automation mechanisms or legacy systems.

The business logic of operation process is supported by the “MOD.SH.DM-Device manager “component and involves:

- DM.01. Device manager OPC UA client: subscribes to the data to be monitored and executes the appropriate methods, when required, through the appropriate OPC UA server (i.e. “AM.06. Automation OPC UA server “or “L.03. Legacy system OPC UA server”.
- DM.02. DM event protocol adapter: supports publish and subscribe capabilities and it adapts the events produced by the “DM.01. Device manager OPC UA client”” to the event protocol supported by the Event Manager (EM) component.

Table 5 summarises the Functional Building Blocks (FBB) involved in both MOD.SH.MS and MOD.SH.DM identifying the type and scope.

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
MS.01. Semantic representation repository				N/A	
MS.02 OPC UA discovery server				N/A	
DM.01. Device manager OPC UA				N/A	
DM.02. DM event protocol adapter				N/A	

Table 5 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 for the shop floor operator interaction with the automation mechanisms (i.e. robots) through gesture, voice, lights, sounds, etc.

Depending on the situation a unique interaction/data source could not be enough as it can provide only partial information even unreliable (i.e. due to environmental conditions such as poor light or high noise.). Furthermore, the shop floor operates 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 or graphical user interfaces (GUI) can be involved (e.g. MS KINECT for gestures capture, etc).

Figure 21 shows the components diagram representing the decomposition of the MOD.SHMHMI 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).

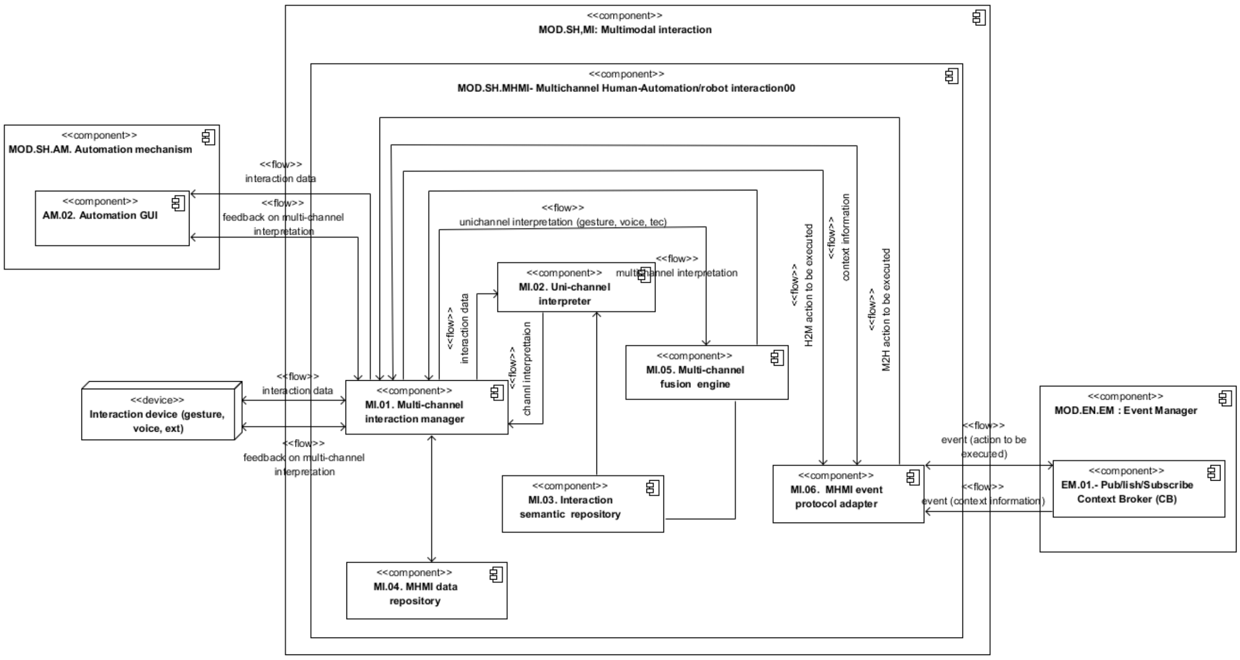


Figure 21 MOD.SHMHMI. High Level module decomposition

As shown in the figure below different interactions channel can be envisages.

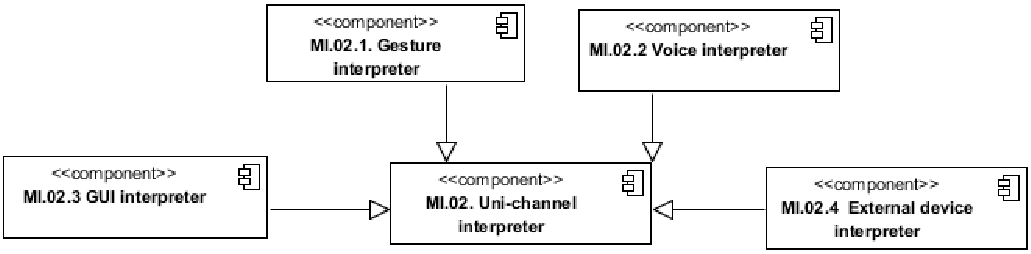


Figure 22 MOD.SH.MHMI. H2M interaction channels

The key sub component is the "MI.01 Multi-channel interaction manager" which oversees orchestrating the rest of components to perform the Human to Machine (H2M) and Machine to Human (M2H) i, the "MI.01. Multi-channel interaction manager" manages the feedback on multi-channel interpretation and interacts with the "MI.04. MHMI data repository" by storing and retrieving the required information to support data persistence for interpretation incremental learning readiness.

The rest of relevant sub components involved are

- MI.02. Uni-channel interpreter: interprets the input data provided by the "MI.01 Multi-channel interaction manager" supported on semantic information provided by "MI.03. Interaction semantic repository". The system involves as many interpreters as channels are involved in the interaction.
- MI.05. Multi-channel fusion engine: fuses the uni-channel interpretations (i.e. partial) considering Semantic information provided by the "MI.03. Interaction semantic repository and provides the complete multichannel interpretation.
- MI.06. MHMI event protocol adapter: supports publish and subscribe capabilities and it adapts the events produced by the "MI.01. Multi-channel interaction manager" to the event protocol supported by the Event Manager (EM) component.

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

FBB	Type			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. Interaction semantic repository					
MI.04. MHMI data repository					
MI.05. Multi-channel fusion engine					
MI.06. MHMI event protocol adapter					

Table 6 MOD.SHMHMI 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 profile (e.g. working mode preferences).

Figure 23 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.

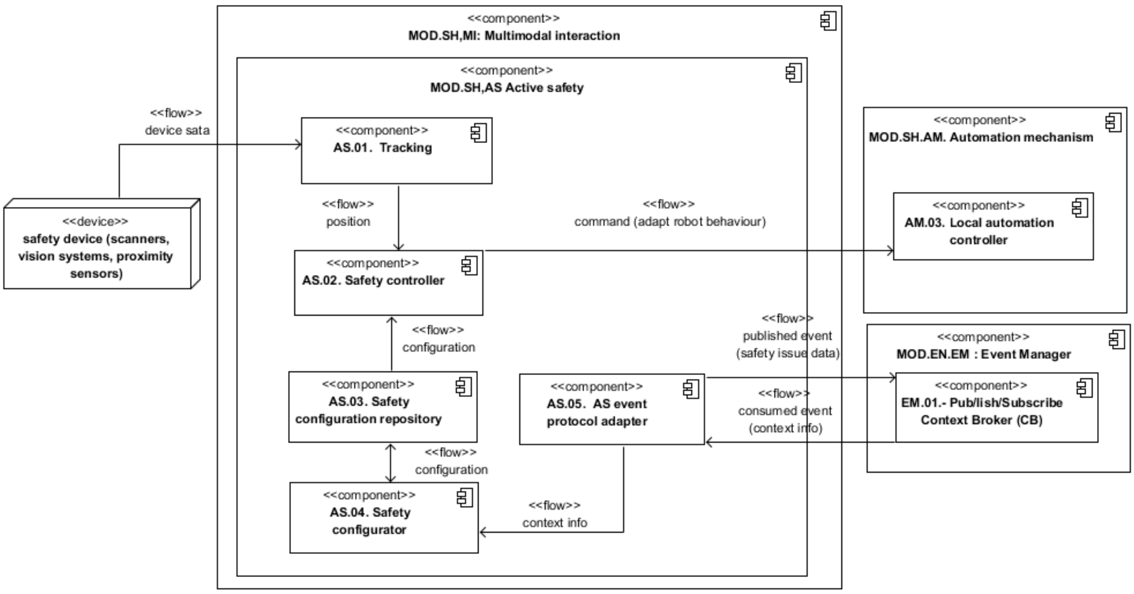


Figure 23 MOD.SH.AS. High Level module decomposition

The key sub component is the ""AS.02. Safety controller" which processes the tracking information resulting from "AS.01. Tracking" decides on 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" use 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").

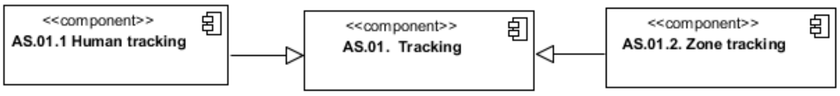


Figure 24 MOD.SH.AS. Tracking

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

If relevant, the "AS.04. AS event protocol adapter" supports publish and subscribe capabilities and adapts the events produced by the "AS.02. Safety controller" to the event protocol supported by the Event Manager (EM) component. Furthermore, it allows collecting information related to the operator from the A4BLUE system.

Table 7 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					
AS.05. AS event protocol adapter					

Table 7 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 (VA) knowledge base, in charge of supporting the virtualisation and representation of Tangible Assets (TA) and Intangible Assets (IA).

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

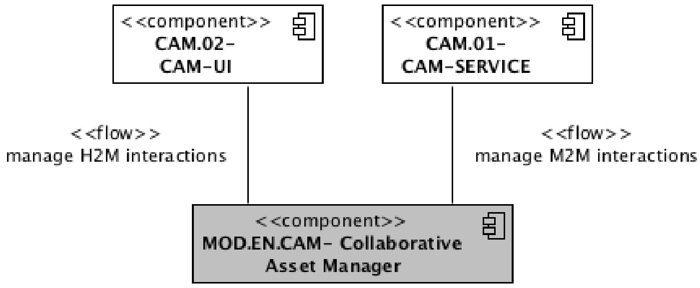


Figure 25 MOD.EN.CAM. High Level module decomposition

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, 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. Some functional requirements this technology addresses include event-based routing, observation, monitoring and event correlation.

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 doesn’t 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.

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

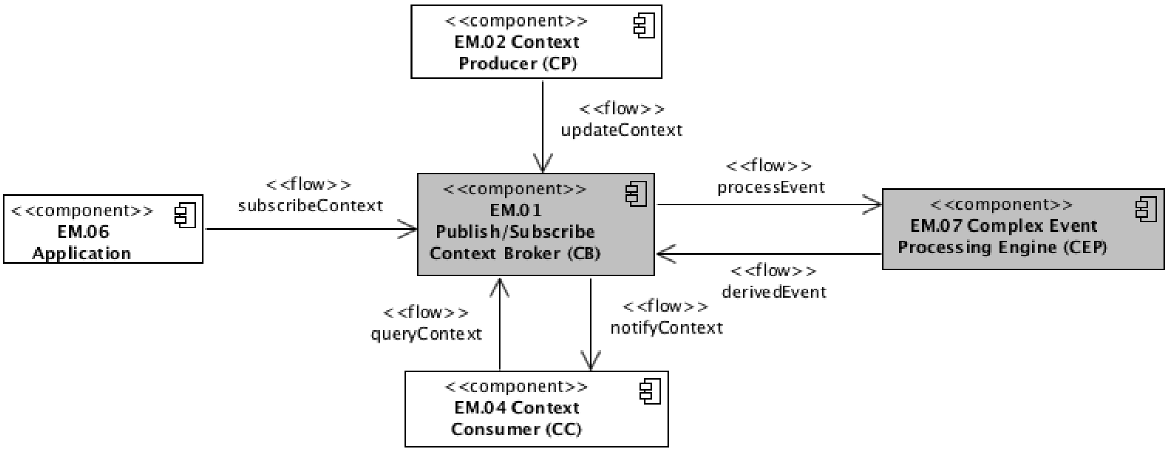


Figure 26 MOD.EN.EM. High Level module decomposition

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

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

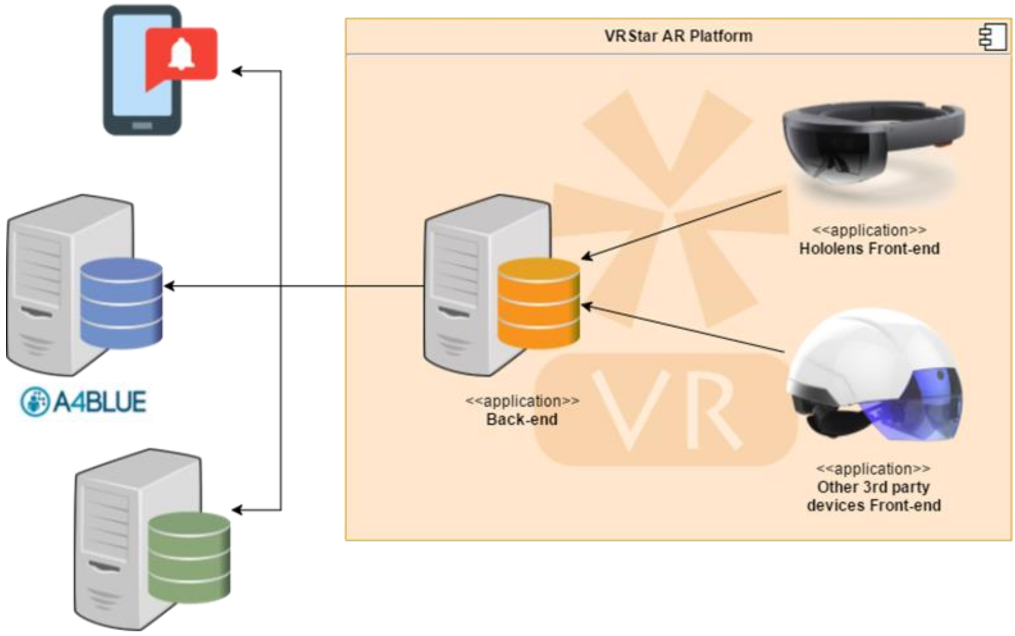


Figure 27 MOD.EN.ARG. The VRStar AR Platform and its modules

6.2.3.1 MOD.EN.ARG.01 – 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.

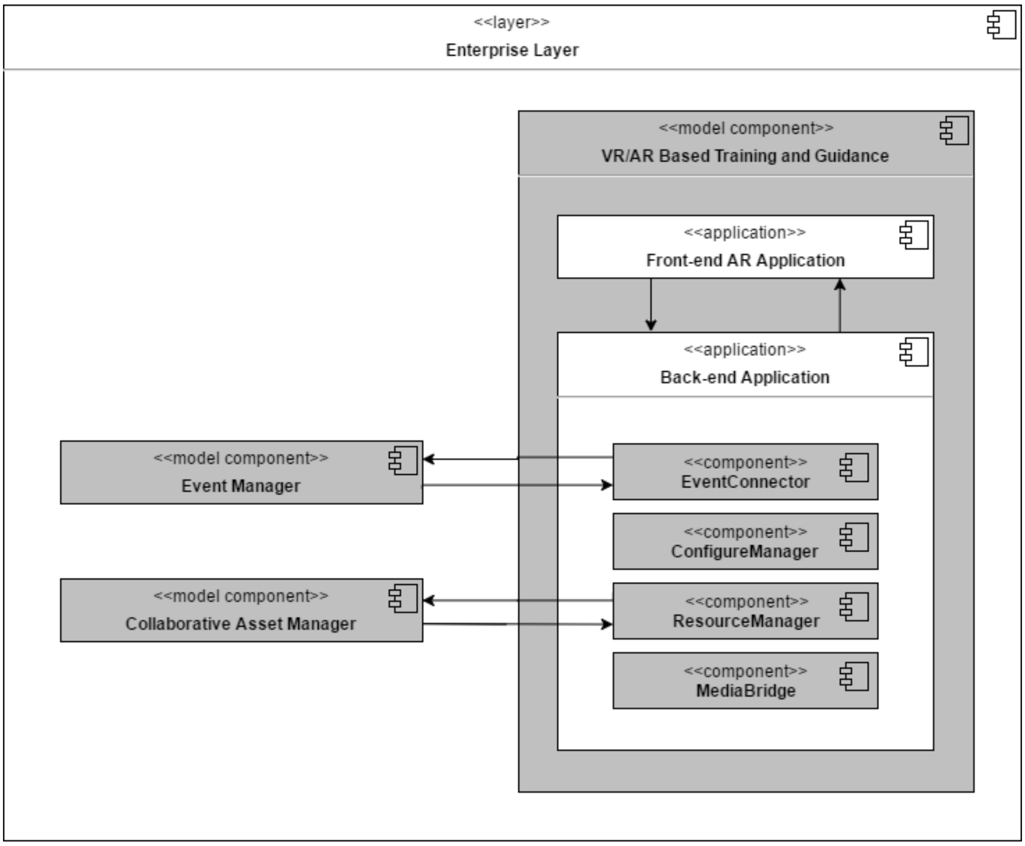


Figure 28 MOD.EN.ARG. Back-end Application Components Diagram

6.2.3.2 MOD.EN.ARG.02 – 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 optimization, 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

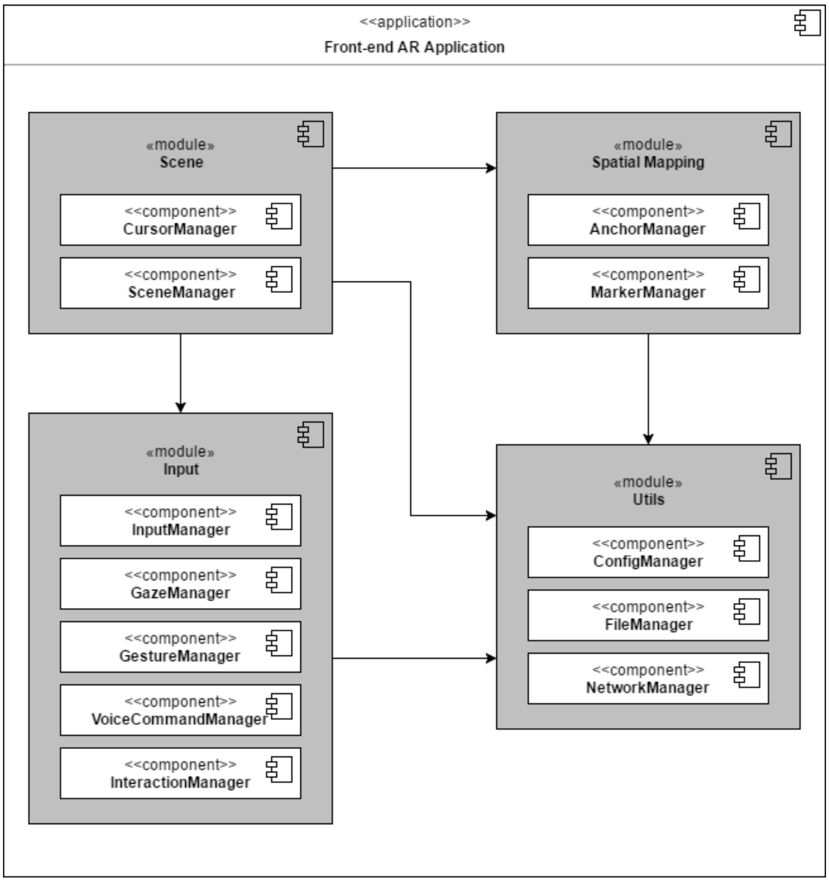


Figure 29 MOD.EN.ARG. Front-end AR Application Components Diagram

FBB	Type		
	Graphical User Interface	Business logic	Data storage
MOD.EN.ARG- VR/AR BASED TRAINING AND GUIDANCE			
ARG.01.- Back-end			
ARG.02.- Front-end			

Table 8 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 in all 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 be able to 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 off 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 reused and share best practices.

The Figure 30 here below shows the main components of the MOD.BU.KM, describing both main Building Blocks and information flows.

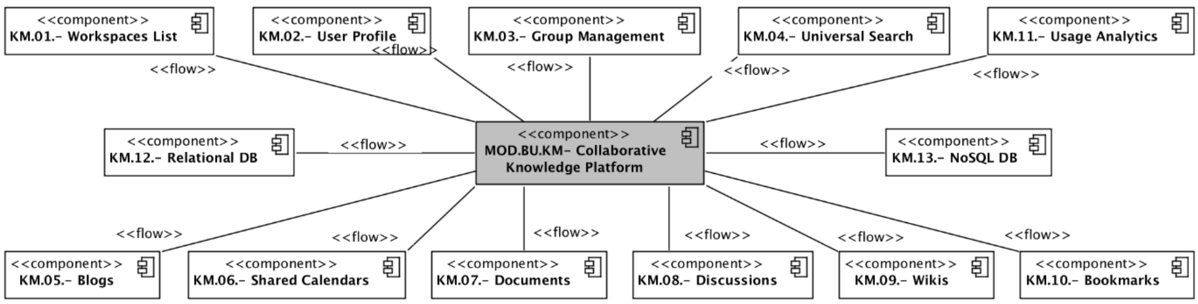


Figure 30 MOD.BU.KM. High Level module decomposition

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

Table 9 summarises the Functional Building Blocks(FBB) involved in MOD.BU.KM identifying the type and scope.

FBB	Type		
	Graphical User Interface	Business logic	Data storage
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			
KM.09.- Wikis			
KM.10.- Bookmarks			
KM.11.- Usage Analytics			
KM.12.- Relational DB			
KM.13.- NoSQL DB			

Table 9 MOD.BU.KM Functional Building Blocks summary

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

The Decision Support module (MOD.BU.DSS) 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.

Figure 31 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).

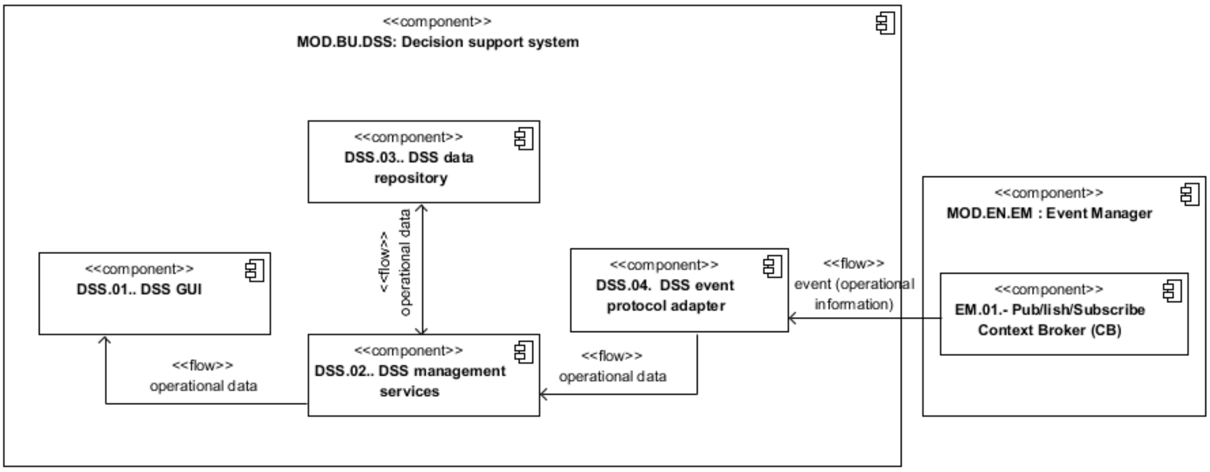


Figure 31 MOD.BU.DSS. High Level module decomposition

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.

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

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
DSS.01. DSS GUI				N/A	
DSS.02. DSS management services				N/A	
DSS.03. DSS data repository				N/A	
DSS.04. DSS event protocol adapter				N/A	

Table 10 MOD.BU.DSS Functional Building Blocks summary

6.3.3 MOD.BU.MON- MONITORING

The usage of performance indicators (KPI: Key Performance Indicator) allows evaluating the appropriate execution of the production from a sustainability (economic and social) perspective.

Task 2.5 (Formalisation of the economic and technical assessment framework) is identifying the KPIs to be used for evaluating the performance of the A4BLUE solution along with the methods to gather them. The Monitoring component is aimed only to support the collection of the relevant performance metrics produced in the domain of the A4BLUE solution. Other kind of KPIs should be out of the scope of this component.

MOD.BU.MON is expected to follow an analogous approach to the one identified in the previous section ("6.3.2 – MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)").

Potential updates could come from the preliminary results of Task 2.5 – "Formalisation of the economic and technical assessment framework)" and should be reported in the main outcome of the related task (i.e. D2.4 due at M13) but also in the final release of this deliverable (i.e. D2.6 due at M19).

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

The model for the quantitative measurement of satisfaction involves the development of a questionnaire with verified satisfactory levels of reliability and validity (i.e. psychometric instrument) which can be used to assess levels of worker satisfaction in relation to human-automation systems and wider work environment characteristics.

The Computer-based tool for Quantitative Measurement of Satisfaction (MOD.BU.CQMS) enables the involved shop floor operators to complete the worker satisfaction questionnaires in an easy way and evaluates the results. MOD.BU.CQMS should be generic enough to support re-usability and allow the update and introduction of new questionnaires, if required.

Furthermore, MOD.BU.CQMS is expected to involve both online and offline working modes.

- **Offline mode**: MOD.BU.CQMS is used as a standalone tool involving all the required components. In the offline working mode, episodic assessment of worker satisfaction is supported by enabling to schedule the questionnaires on periodicity basis.
- **Online mode**: MOD.BU.CQMS interacts with the Event Manager (EM) to support on event based assessment of worker satisfaction.

MOD.SH.CQMS 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).

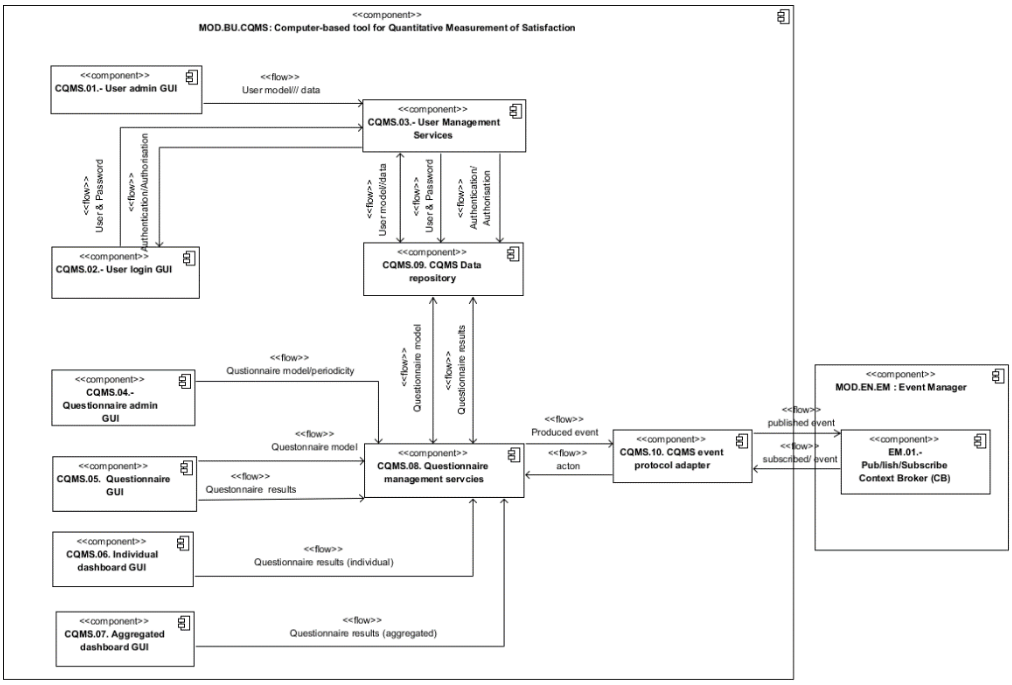


Figure 32 MOD.SH.CQMS. High Level module decomposition

Users (i.e. system administrators, shop floor operators and supervisors) interact with the CQMS though web based graphical user interfaces (GUI). In particular:

- All the users should be authenticated and authorised to access the MOD.BU.CQMS (i.e. “CQMS.02.- Login GUI”)
- System administrators access management features to create/read/update/delete both users and access rights (i.e. “CQMS.01.- User admin GUI”) and questionnaires (i.e. “CQMS.04.- Questionnaire admin GUI”);
- Shop floor operators can complete satisfaction related questionnaire(s) (i.e. “CQMS.05.- Questionnaire GUI”) and access the history of scored questionnaire results (i.e. “CQMS.06.- Individual dashboard GUI”);
- Supervisors can access aggregated and individualised scored questionnaire results (“CQMS.06.- Individual dashboard GUI” and “CQMS.07.- Aggregated dashboard GUI”);

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 a generic approach to the online working mode the “CQMS.10. CQMS event protocol adapter” supports publish and subscribe capabilities and it is in charge of adapting the events produced by the “CQMS.08.- Questionnaire management services” to the event protocol supported by the Event Manager (EM) component.

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

FBB	Type			Scope	
	Graphical User Interface	Business logic	Data repository	Offline mode	Online mode
CQMS.01.- User admin GUI					
CQMS.02.- Login GUI					
CQMS.04.- Questionnaire admin GUI					
CQMS.05.- Questionnaire list 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 protocol adapter					

Table 11 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.AM- AUTOMATION MECHANISMS, MOD.SH.MS- MEDIATION SERVICES and MOD.SH.DM- DEVICE MANAGER

The following sequence diagrams provide details of the interactions among sub-components to support “FTR.04- Integration with enterprise legacy systems” and “FTR.05- Adaptive automation mechanisms” as described in the specifications section. A sequence diagram is provided for each involved use case scenario. As, initially, an analogous approach is foreseen for the integration of both legacy systems and automation mechanisms only the diagrams related to the automation mechanisms are provided.

Discovery process: Plug/Un-plug automation mechanism/integrate legacy system

When the automation mechanism is plugged/un-plugged:

- the “AM.04. Automation semantic registrator” registers: the automation system semantic representation 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” that sends to the “DM.01. Device manager OPC UA client” the order to subscribe to the data to be monitored.
- the “AM.05. Automation OPC-UA client” registers the communication information (IP and Port) in the “MS.02 OPC UA discovery server.”
- the “DM.01. Device manager OPC UA client” gets the information of the endpoint of the automation OPC-UA server from the “MS.02 OPC UA discovery server”, and subscribes to the data to be monitored through the “AM.06. Automation OPC-UA Server”.

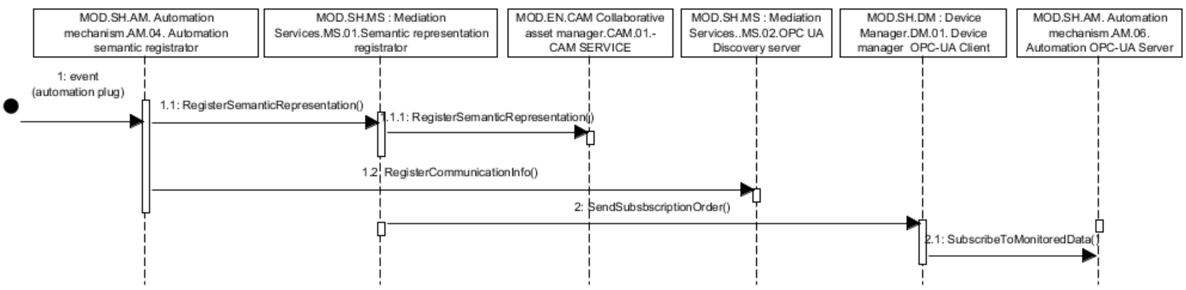


Figure 33 MOD.SH.AM. Plug/Un-plug automation mechanism

Furthermore, it can publish a register/ de-register event that is adapted through the “DM.02. DM event protocol adapter” to meet the event protocol supported by the Event Manager and published into the Publish/Subscribe Context Broker (CB).

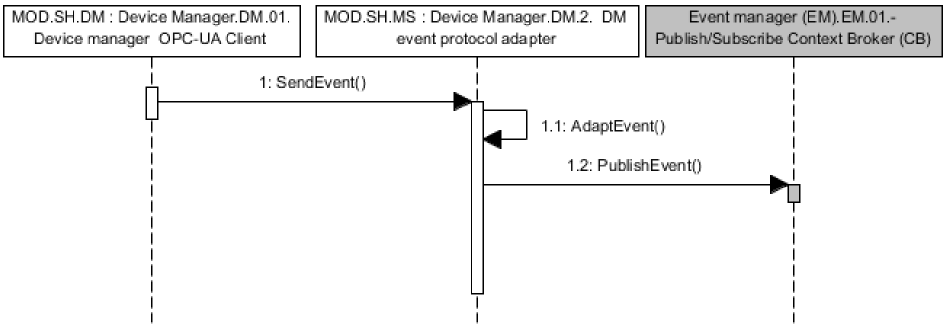


Figure 34 MOD.SH.AM. Event production

<u>Share information (operation process)</u>

Once a monitored data is updated the "AM.06. Automation OPC-UA Server" sends the update to the "DM.01. Device manager OPC UA client" that is in charge of publishing an event with the updated data. the into the "DM.02. DM event protocol adapter" that adapts it to meet the event protocol supported by the Event Manager and published into the Publish/Subscribe Context Broker (CB).

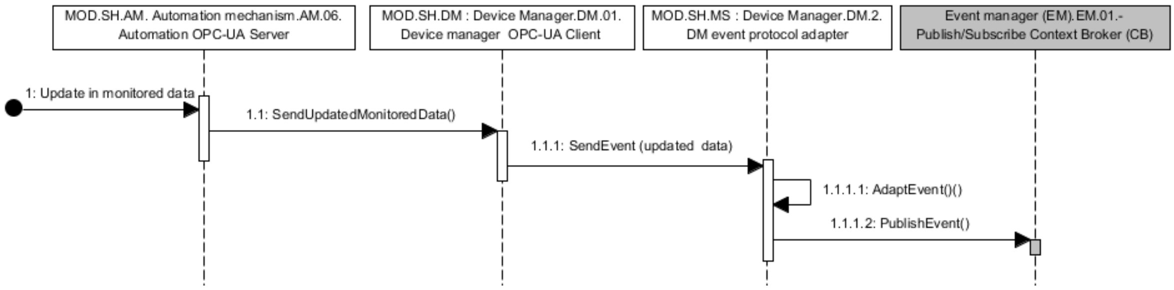


Figure 35 MOD.SH.AM. Share information with automation mechanism

<u>Adapt behaviour (operation process)</u>

Once the "DM.02. DM event protocol 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 protocol supported by the "DM.01. Device manager OPC UA client" component and notifies it.

When the "DM.01. Device manager OPC UA client" receives the notification of the event it retrieves the semantic representation information from the "MS.01. Semantic representation repository" to identify the method provided by the automation that is linked to the action to be executed and the endpoint of the "AM.06. Automation OPC-UA Server" from the "MS.02 OPC UA discovery server" and then executes the appropriate method trough the "AM.06. Automation OPC-UA Server".

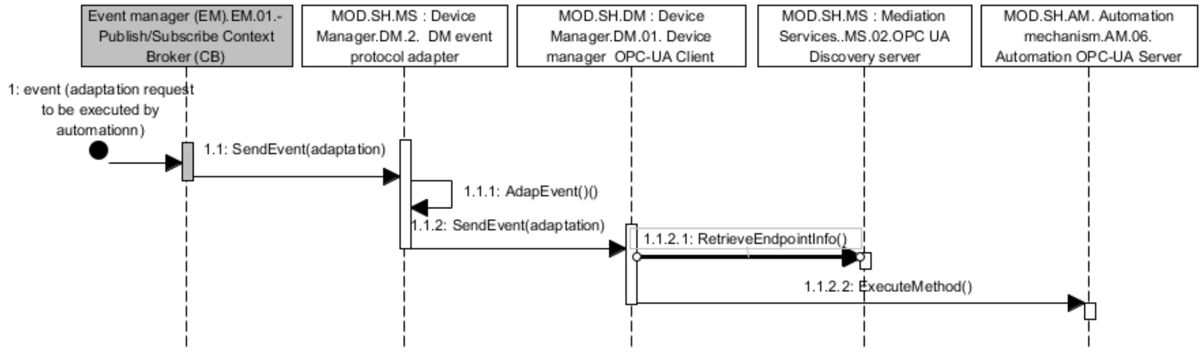


Figure 36 MOD.SH.AM. Adapt automation mechanism behaviour

7.1.2 MOD.SH.MI- MULTIMODAL 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.

7.1.2.1 Multimodal Human- Automation/Robot Interaction

Perform 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 send them to the appropriate interpreter (“MI.02. Uni-channel interpreter”) which interprets them by using the semantic representation of the instruction provided by the “MI.03. Interaction semantic repository “and sends back the channel interpretation (e.g. gesture interpretation). Then the “MI.01 Multi-channel interaction manager” stores the interpretation in the “MI.04. MHMI data repository” and sends (i.e. for each relevant channel) uni-channel interpretation to the “MI.05. Multi-channel fusion engine”. The fusion engine fuses partial interpretation and sends back the multichannel interpretation.

Once the multichannel interpretation is available the “MI.01 Multi-channel interaction manager” requests feedback through the available interaction mechanisms (e.g. interaction devices, GUI) to the shop floor operator to check that the obtained interpretation is the right one or to request more details. The feedback provided by the operator follow the above described sequence till a positive feedback is available.

When a positive feedback is available it publishes an event with the H2M interaction information (i.e. action to be executed) into the “MI.06. MHMI event protocol adapter” that adapts it to meet the event protocol supported by the Event Manager and published into the “EM.01. Publish/Subscribe Context Broker (CB)”.

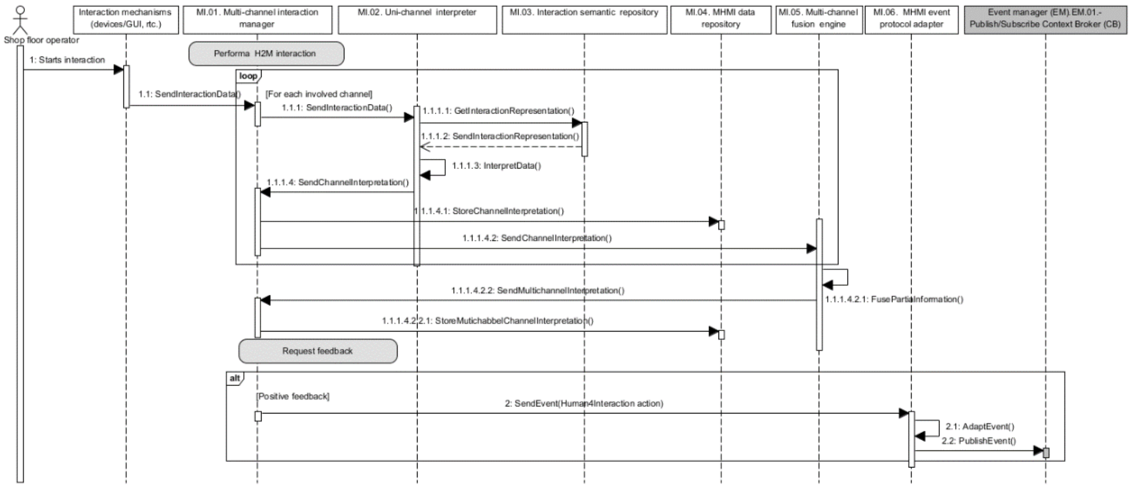


Figure 37 MOD.SH.HMMI. Perform H2M interaction

The figure below represents a simplified view of the feedback request loop. It is can be an iterative process where the "MI.01 Multi-channel interaction manager" can ask 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 operator can provide his/her feedback trough the different available interaction mechanisms following the above described sequence diagram. The iterative process ends when a positive feedback is obtained.

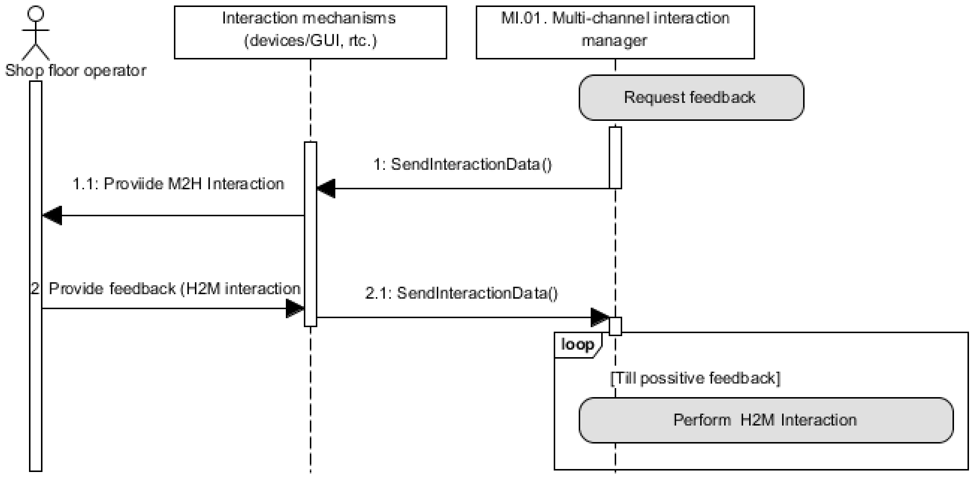


Figure 38 MOD.SH.HMMI. Request feedback

Context information (e.g. operator identification, environment status information, etc.) is an additional input to the partial interpretations that could be managed by the "MI.05. Multi-channel fusion engine" to obtain the multichannel interpretation. Context information can come from the A4BLUE system though the "MI.06. MHMI event protocol adapter" and "MI.01 Multi-channel interaction manager" as described in the following sequence diagram.

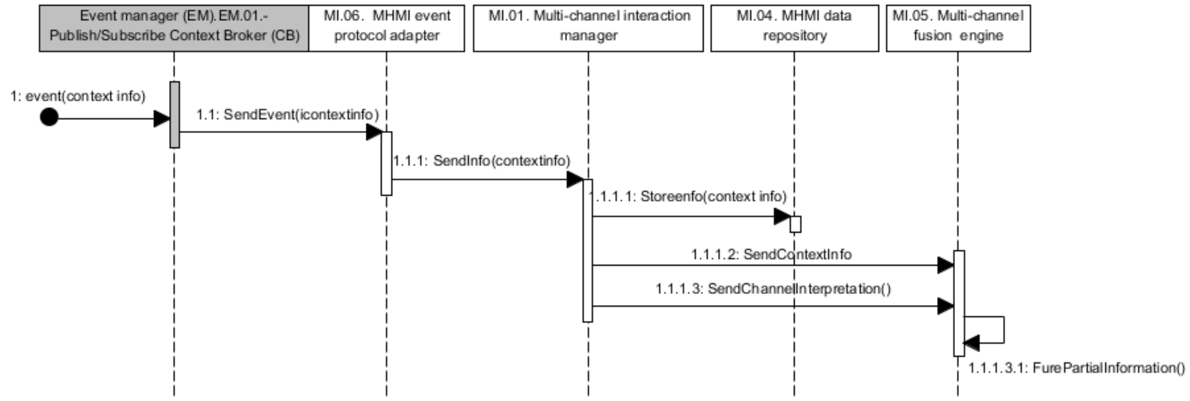


Figure 39 MOD.SH.HMMI. Context information

Perform M2H interaction

Once the “MI.06. MHMI event protocol adapter” receives an action execution event from the “EM.01. Publish/Subscribe Context Broker (CB)” it identifies the action and adapts it, if necessary, to meet the event protocol supported by the “MI.01. Multi-channel interaction manager” component and notifies it. When the “MI.01. Multi-channel interaction manager” receives the notification of the event it stores the action in the “MI.04. MHMI data repository” and send the required data to the integration mechanism so they can perform the expected Machine to Human (M2H) interaction (e.g. display message in the GUI, turn on a light, make a sound, etc.).

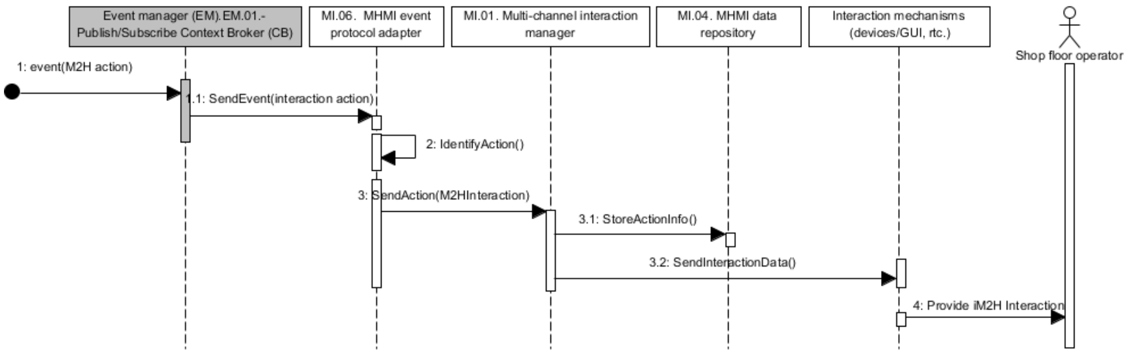


Figure 40 MOD.SH.HMMI. Perform M2H interaction

7.1.2.2 Safe Human - Automation/Robot Co-existence

Share workspace safely

The “AS.05. Safety configurator” allows system administrators to define the safety configuration.

Safety device send data to “AS.01. Human tracking” that 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.) 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”.

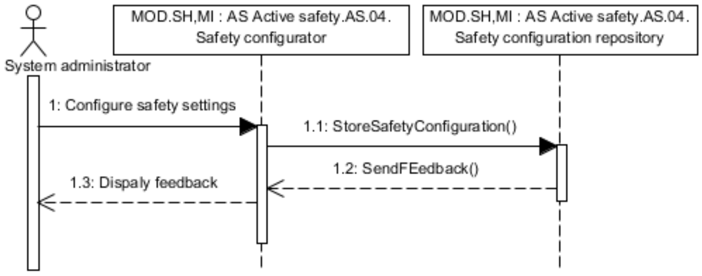


Figure 41 MOD.SH.AS. Share workspace safely configuration

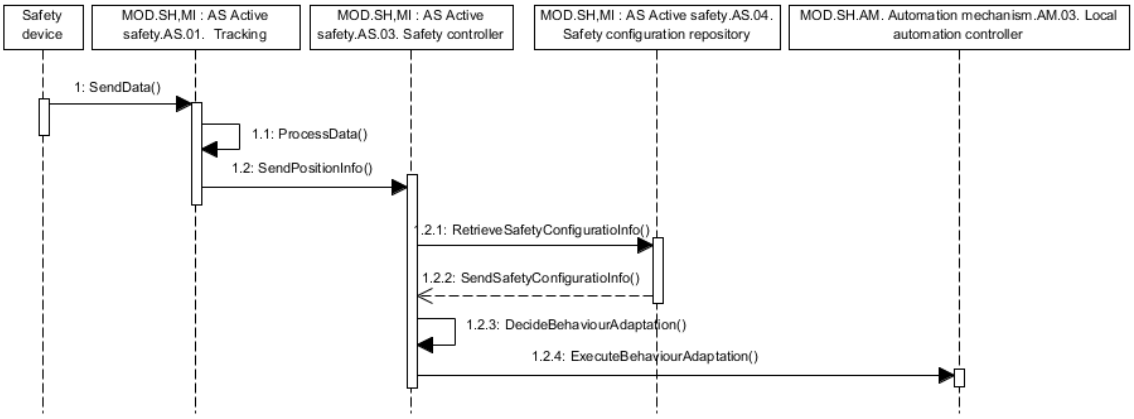


Figure 42 MOD.SH.AS. Share workspace safely operation

If relevant the "AS.03. Safety controller" could send events to the through "AS.06. AS event protocol adapter" that adapts it to meet the event protocol supported by the Event Manager and published into the "EM.01.Publish/Subscribe Context Broker (CB)".

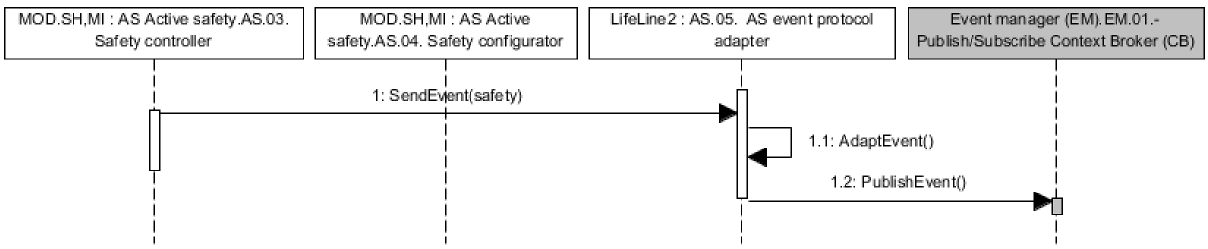


Figure 43 MOD.SH.AS. Online operation- safety issue event

Furthermore, context info coming from the "EM.01. Publish/Subscribe Context Broker (CB)". through "AS.06. AS event protocol adapter" can update the safety configuration.

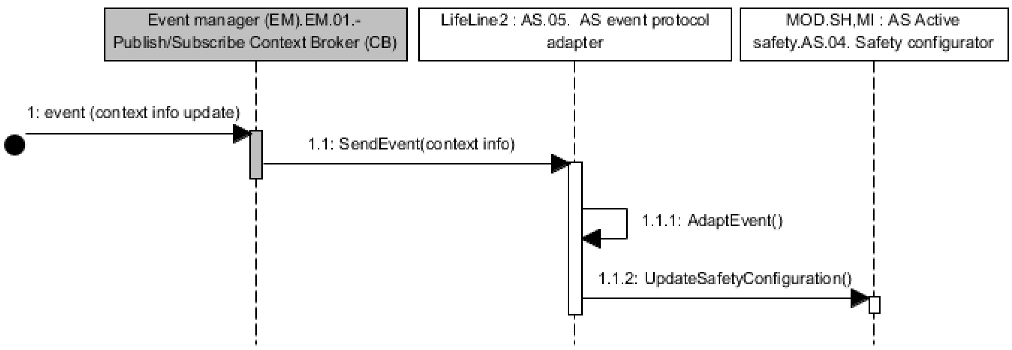


Figure 44 MOD.SH.AS. Online operation- context info event

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 enables 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 45 shows the main business processes impacting on the A4BLUE MOD.EN.CAM component, while Figure 46 shows main interactions with the end users.

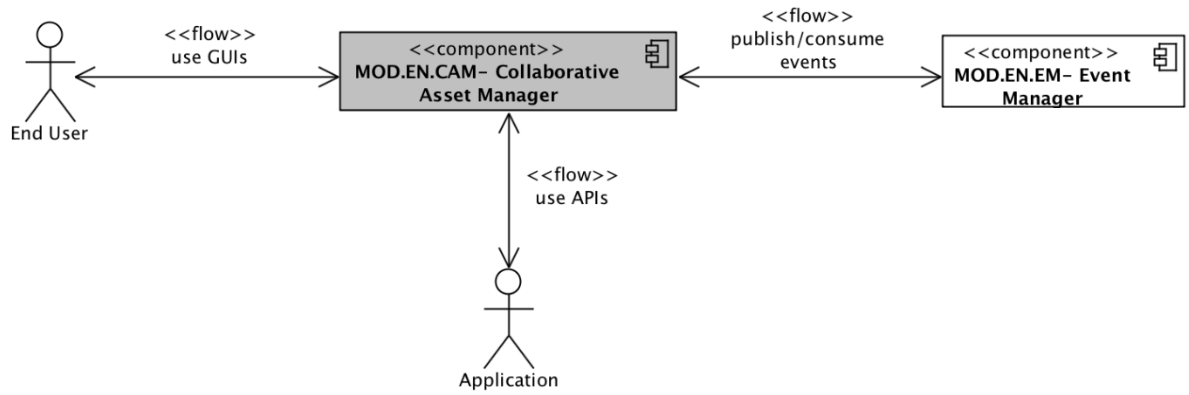


Figure 45 MOD.EN.CAM. Cross-components business processes

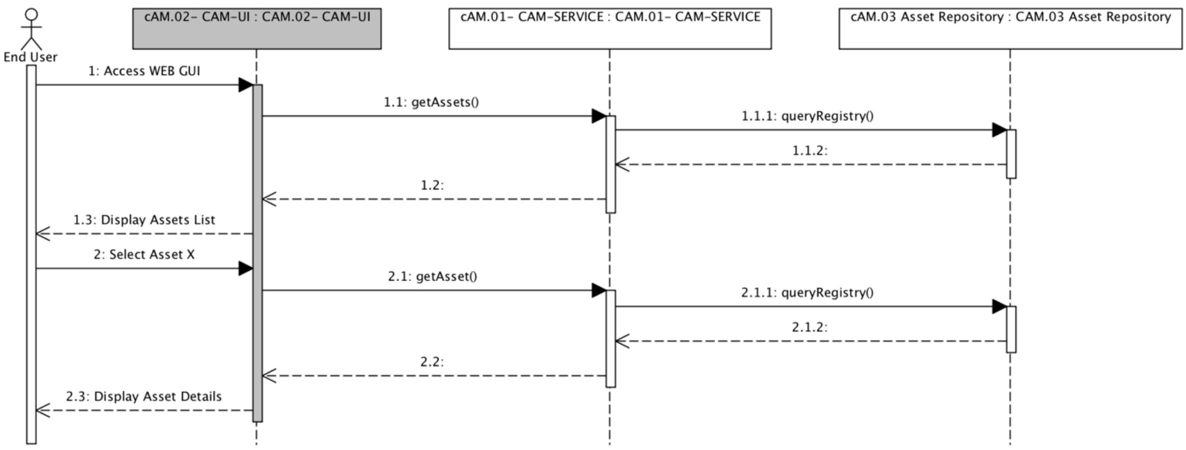


Figure 46 MOD.EN.CAM. Basic interactions and related entities

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 covey 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 47 shows the main business processes impacting on the A4BLUE MOD.EN.EM component.

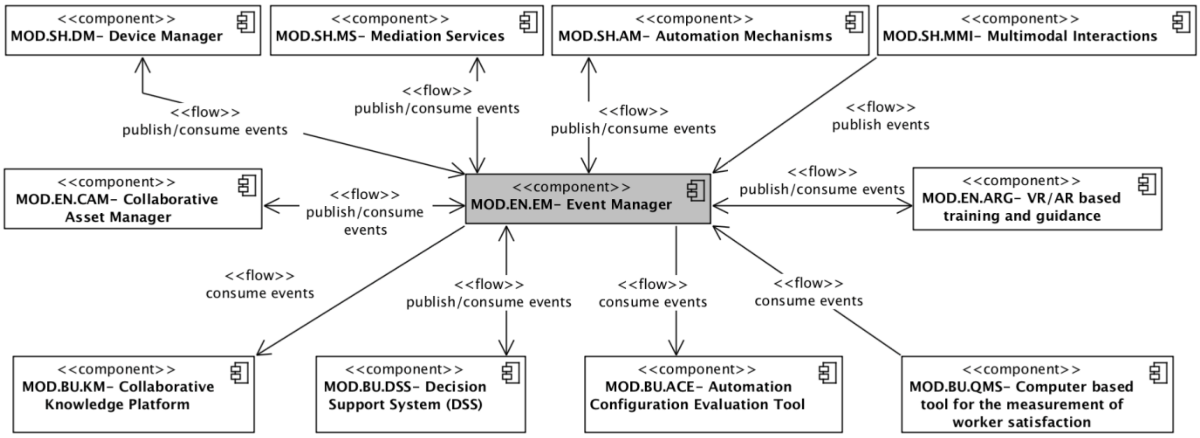


Figure 47 MOD.EN.EM. Cross-components business processes

The following Figure 48 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.

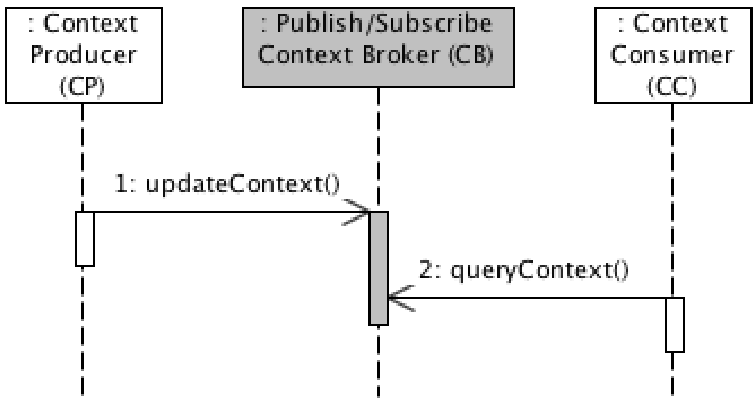


Figure 48 MOD.EN.EM. Basic interactions and related entities

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 49.

- 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.

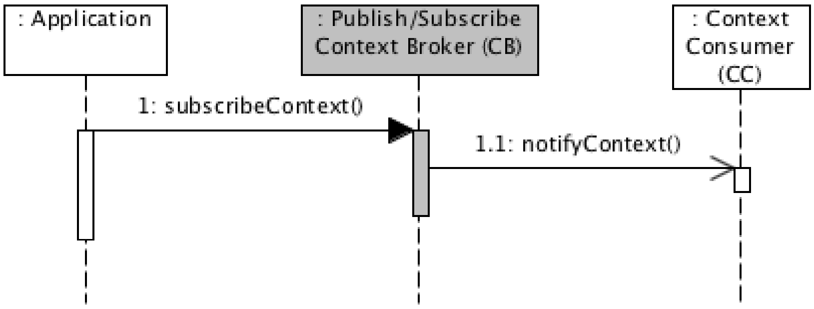


Figure 49 MOD.EN.EM. Interactions to force CCs to subscribe to specific notifications

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.

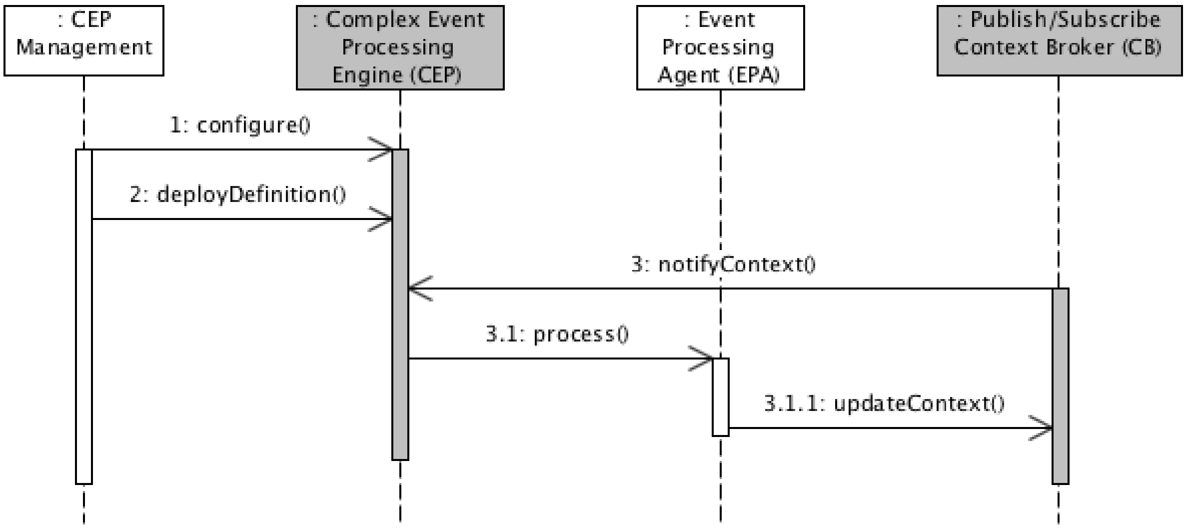


Figure 50 MOD.EN.EM. Interactions among CB and CEP

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, in 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 to create a caching system and saves bandwidth, as a web server does not need to send a full response if the content has not changed.

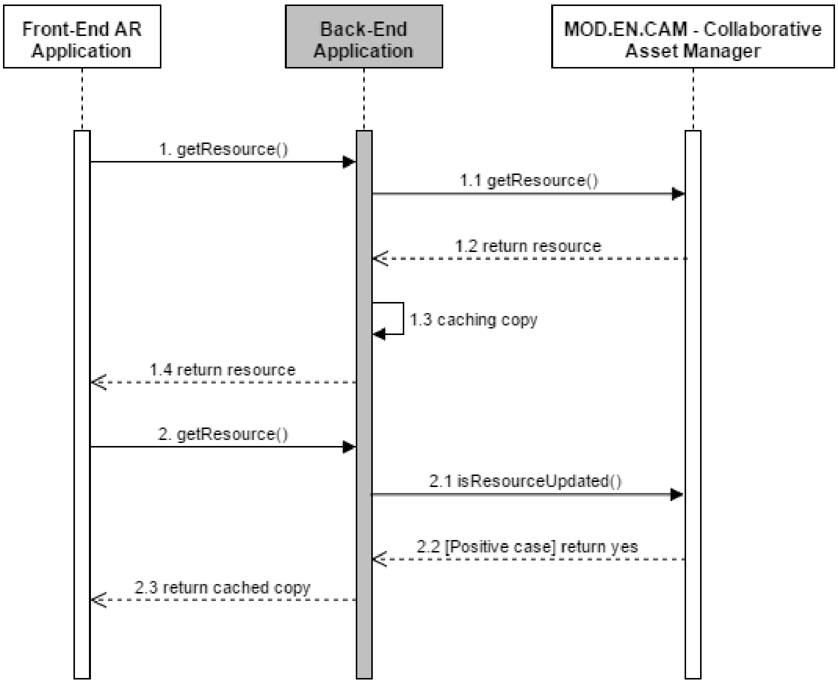


Figure 51 MOD.EN.ARG. Basic asset retrieving

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. Both M2M and H2M interactions will be enables 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 52 shows the main business processes impacting on the A4BLUE MOD.BU.KM component.

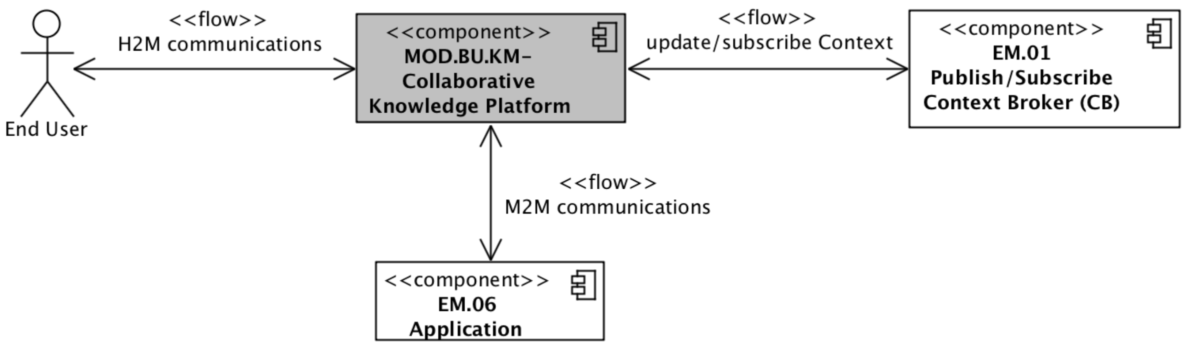


Figure 52 MOD.BU.KM. Cross-components business processes

Figure 53 and Figure 54 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.

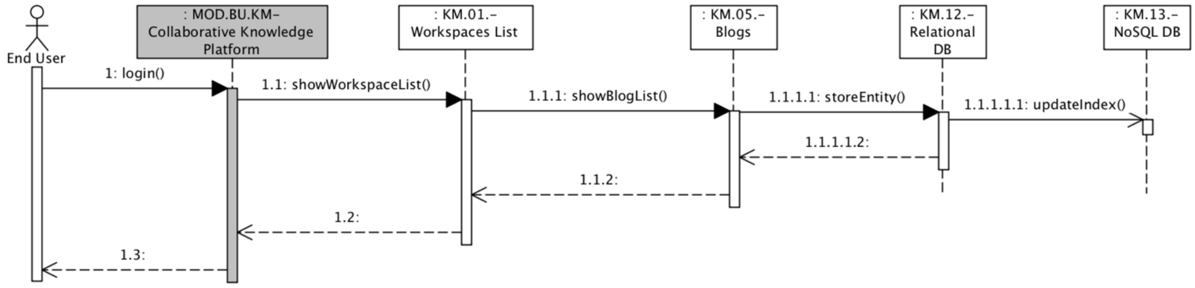


Figure 53 MOD.BU.KM. Create a new blog entry

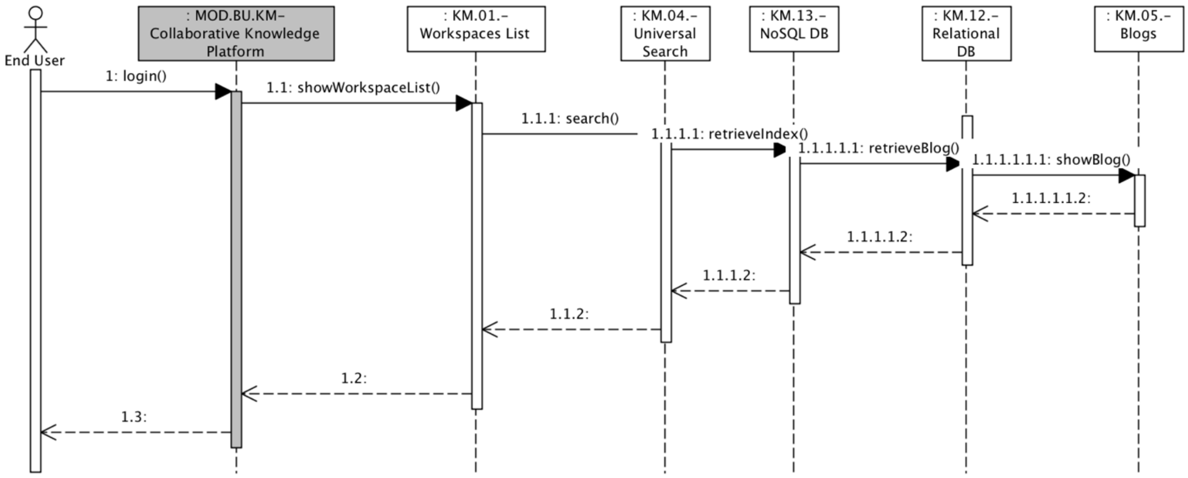


Figure 54 MOD.BU.KM. Retrieve blog entry

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

Once the "DSS.04.- DSS event protocol adapter" receives an operational data event from the "EM.01. Publish/Subscribe Context Broker (CB)" it adapts it transforming raw events related to downtimes, defects, etc, into operational data and sends it to the "DSS.02.- DSS management services" that stores them into the "DSS.03.- DSS data repository".

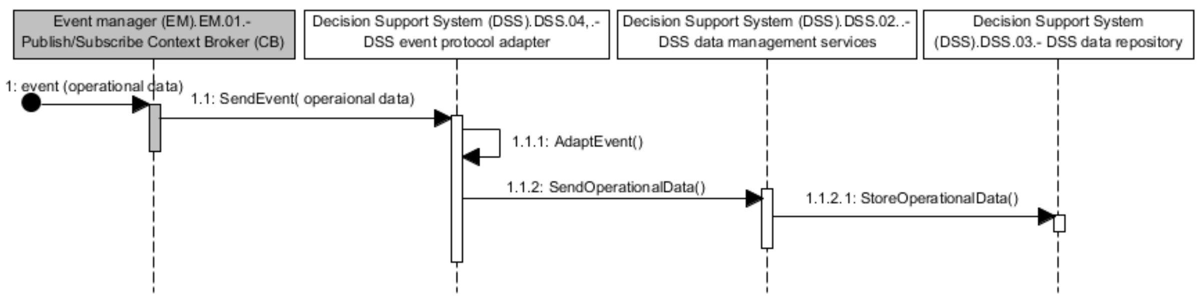


Figure 55 MOD.BU.DSS. Access decision support information – data storage

The users can access the operational info through the graphical user interface ("DSS.01.- DSS GUI" user administration graphical user interface) and the system displays the result in the appropriate format.

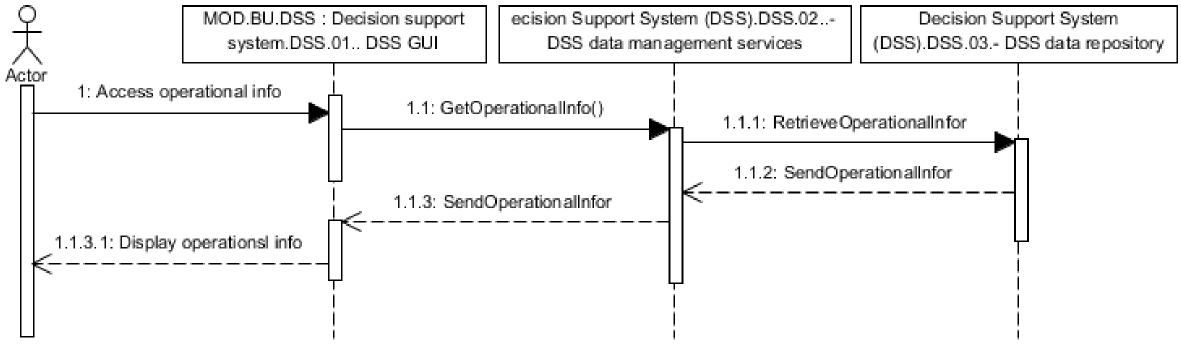


Figure 56 MOD.BU.DSS. Access decision support information – data access

7.3.3 MOD.BU.MON- MONITORING

MOD.BU.MON is expected to follow an analogous approach to the one identified in the previous section (“7.3.2 – MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)”).

Potential updates could come from the preliminary results of Task 2.5 – “Formalisation of the economic and technical assessment framework)” and should be reported in the main outcome of the related task (i.e. D2.4 due at M13) but also in the final release of this deliverable (i.e. D2.6 due at M19).

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

7.3.4.1 Interaction among internal sub-components

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 (i.e. R3.1) has been identified as an essential capability and represented in the specifications section.

CRUD users

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.

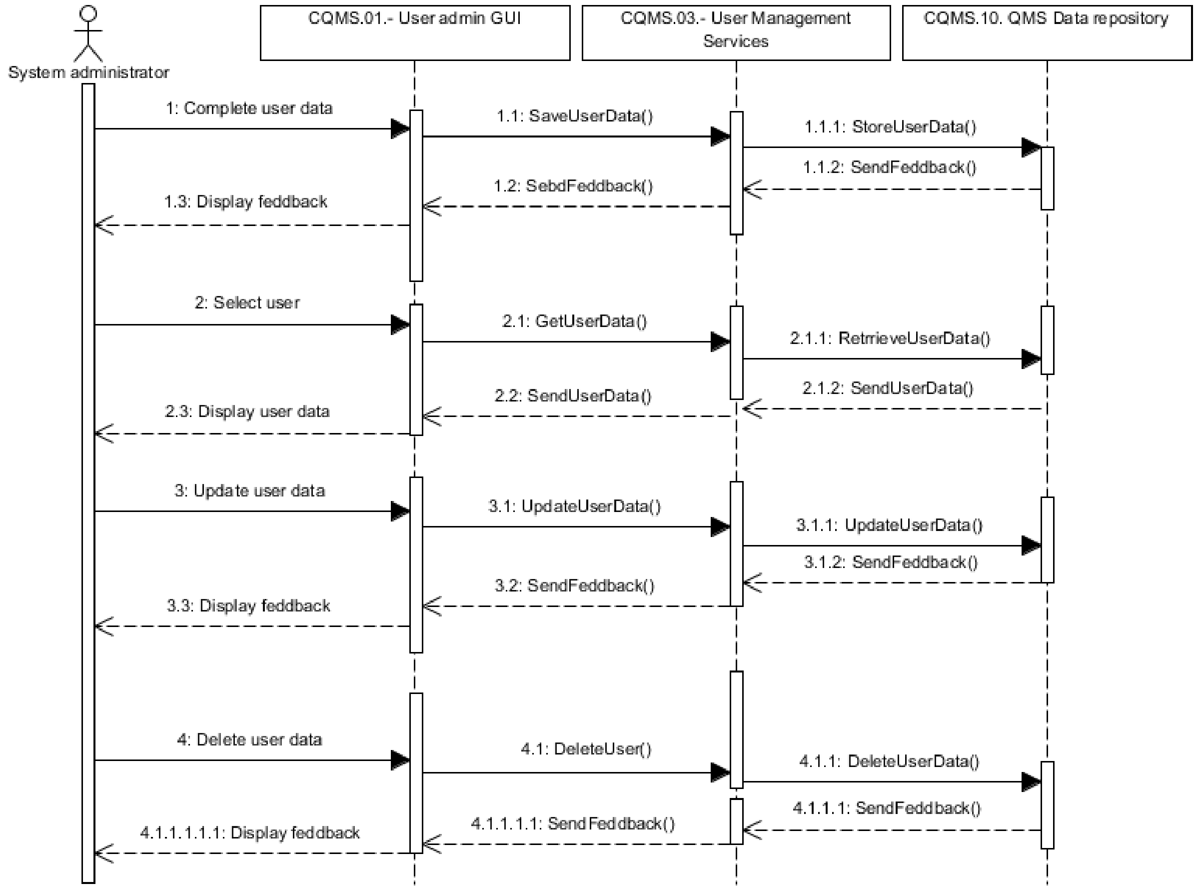


Figure 57 MOD.BU.MON. CRUD users

<u>CRUD worker satisfaction questionnaire</u>

The system administrator can 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.

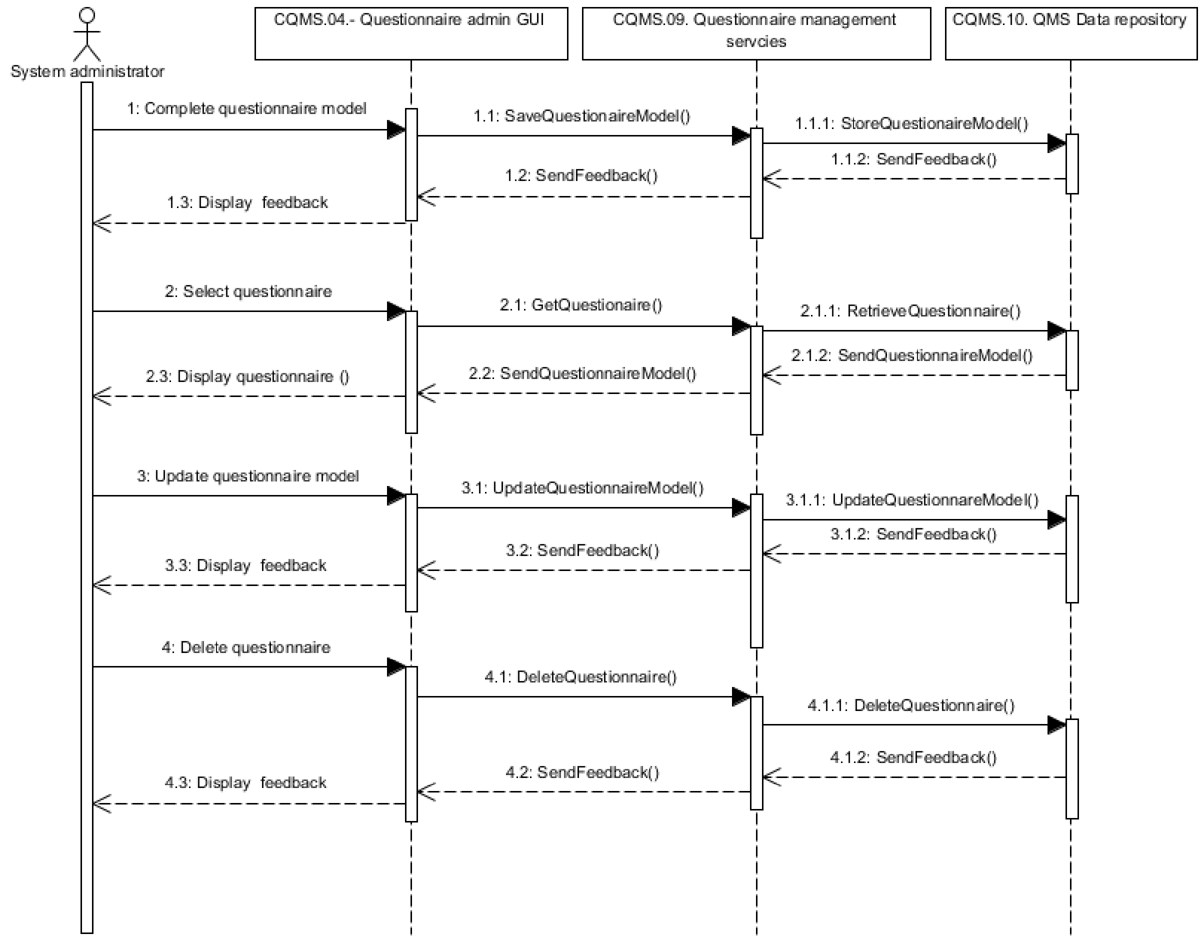


Figure 58 MOD.BU.CQM. CRUD worker satisfaction questionnaire

Complete worker satisfaction questionnaire

The shop floor operator is able to complete and save the satisfaction questionnaire. Before saving it the system checks if all the required questions have been completed. If the input values are valid the system saves the results into the database and it provides feedback on the result of the process by displaying the appropriate message. If they are not valid the system displays a message asking to review them.

In the online mode once the questionnaire is saved an event (i.e. saveQuestionnaire) is published.

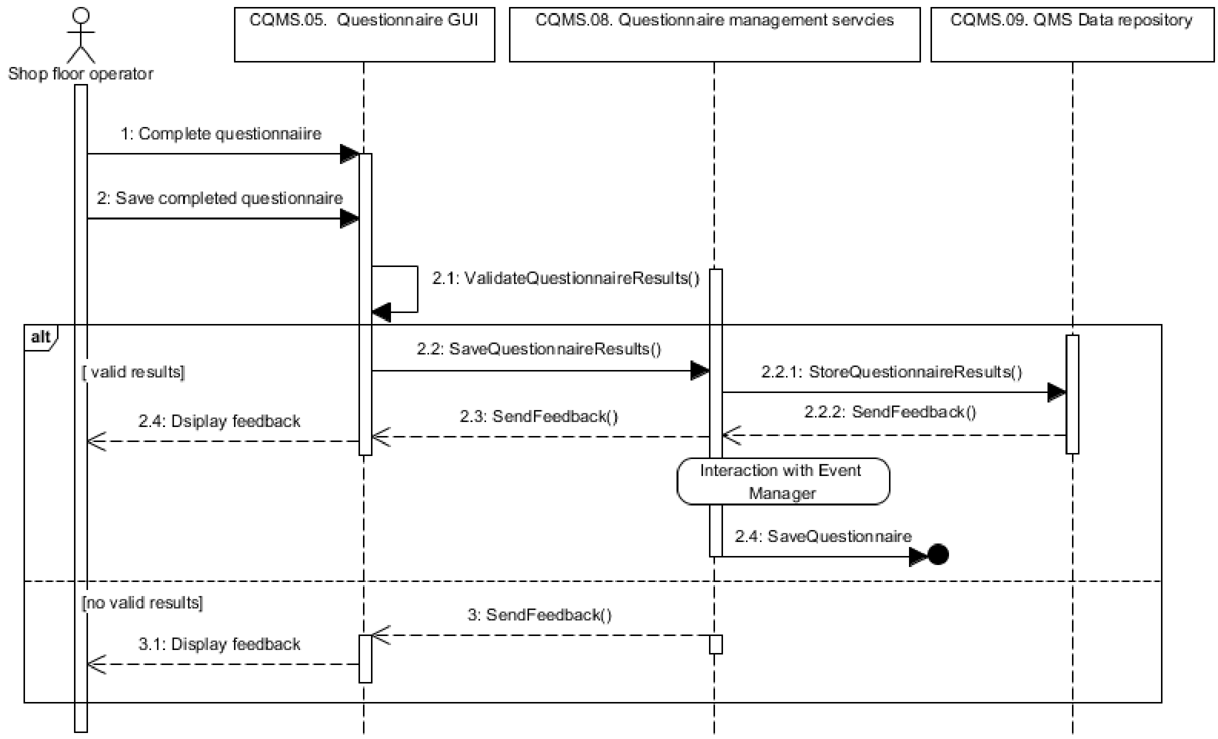


Figure 59 MOD.BU.CQM. Complete worker satisfaction questionnaire

<u>Access individual view of worker satisfaction questionnaires results</u>

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

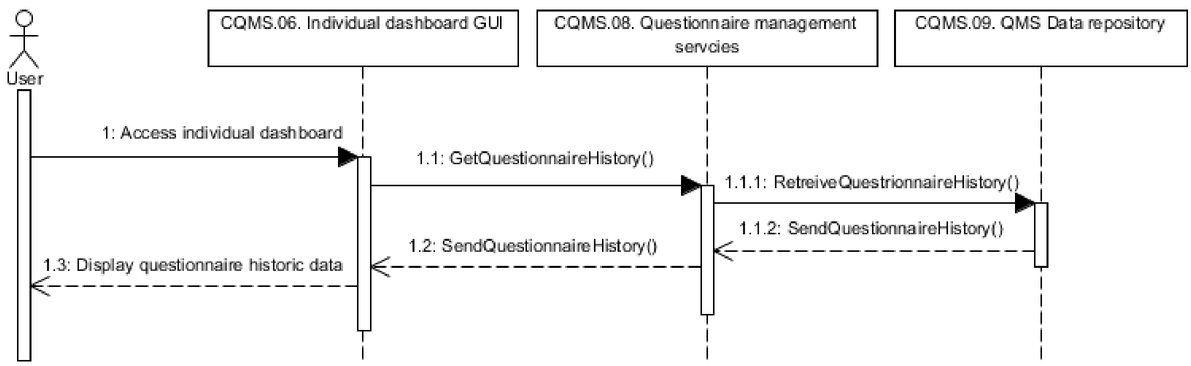


Figure 60 MOD.BU.CQM. Access individual view of worker satisfaction questionnaires results

<u>Access aggregated view of worker satisfaction questionnaires</u>

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

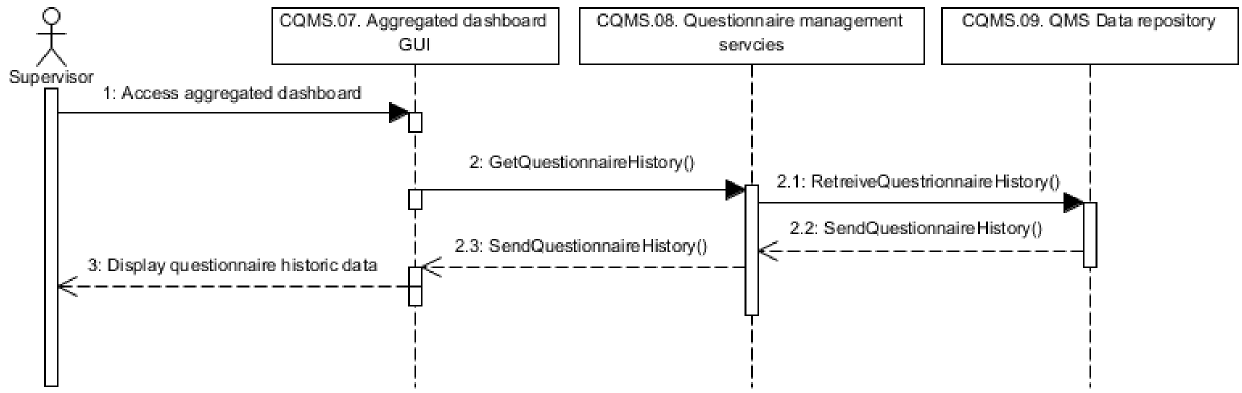


Figure 61 MOD.BU.CQM. Access aggregated view of worker satisfaction questionnaires

7.3.4.2 Interaction with other A4BLUE components

Figure 62 shows the online operation sequence diagram. In this case the event raised once a questionnaire is properly completed is adapted to meet the event protocol supported by the Event Manager and published into the "EM.01. Publish/Subscribe Context Broker (CB)". Furthermore, events raised by the Event Manager (MOD.EN.EM) and consumed by the Computer-based tool for Quantitative Measurement of Satisfaction system are processed to be able to execute the appropriate method provided by the questionnaire.

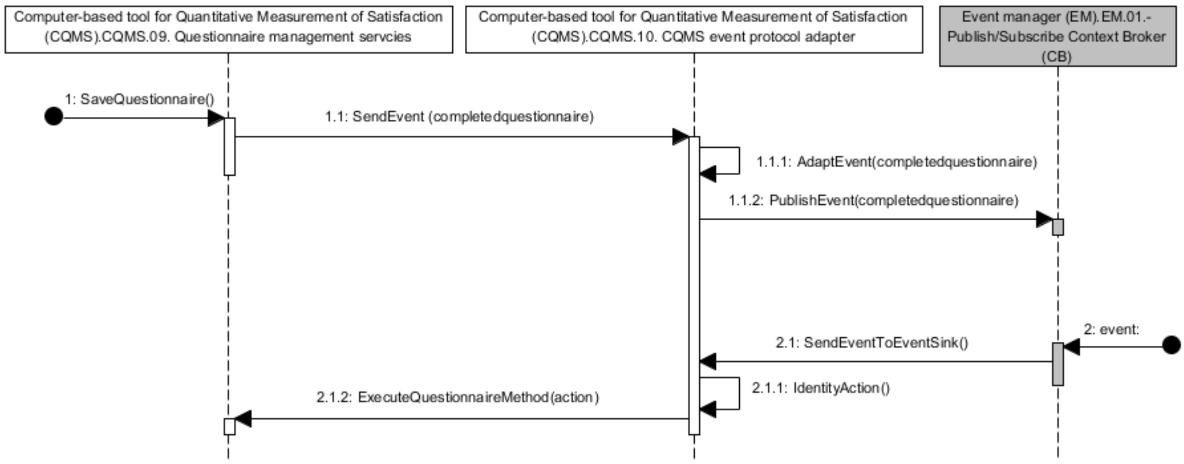


Figure 62 MOD.BU.CQM. Interaction with other A4BLUE component – online operation

8 FBB SPECIFICATION

8.1 SHOPFLOOR LAYER

8.1.1 MOD.SH.AM- AUTOMATION MECHANISMS

8.1.1.1 FBB Specification

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

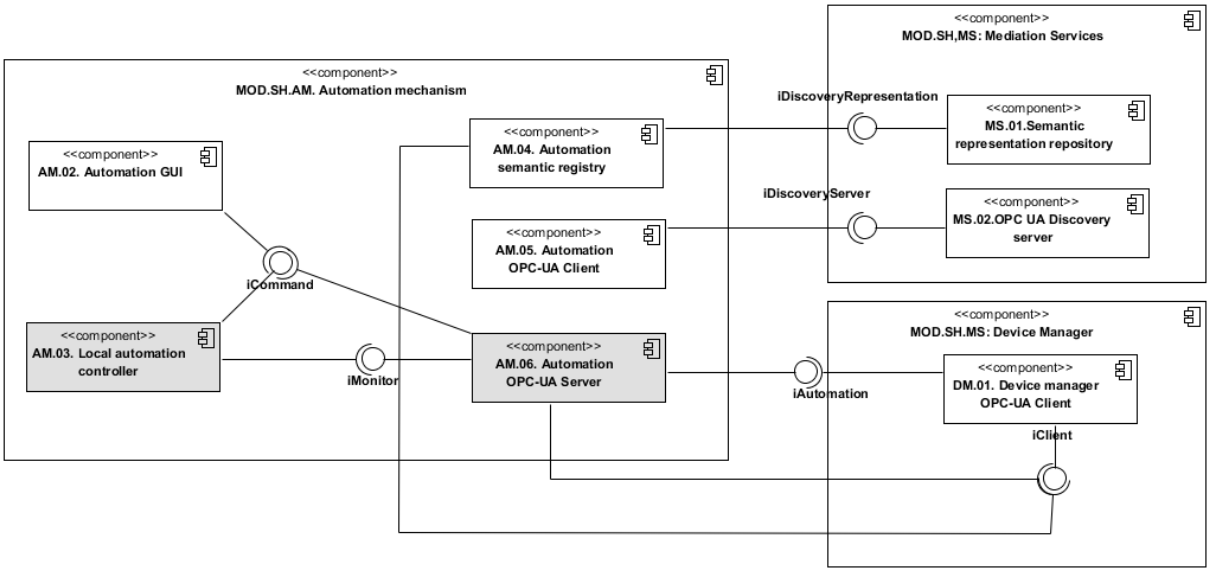


Figure 63 MOD.SH.AM overall interaction

8.1.1.1.1 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"

8.1.1.1.2 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".

8.1.1.1.3 AM.03. Local automation controller

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

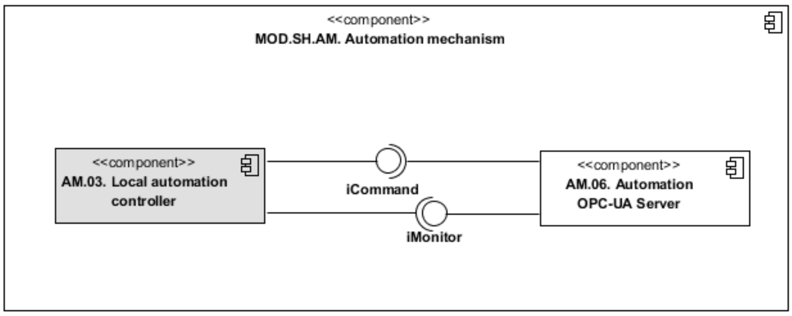


Figure 64 MOD.SH.AM.03. Local automation controller interfaces

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.

The following table shows an initial approach to the main methods exposed by each interface.

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

Table 12 MOD.SH.AM.03. Local automation controller main methods

8.1.1.1.4 AM.04. Automation semantic registrator

The "AM.04. Automation semantic registrator" registers: (1) the legacy system semantic representation in the "MS.01. Semantic representation repository"; (2) the communication information (IP and Port) in the "MS.02 OPC UA discovery server" and (3) the data to be monitored in the "DM.01. Device manager OPC UA".

The FBB consumes the *registerSemanticRepresentation* method of the **iDiscoveryRepresentation** interface exposed by the "MS.01. Semantic representation repository", 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.01. Device manager OPC UA".

8.1.1.1.5 AM.05. Automation OPC-UA Client

The "AM.05. Automation OPC-UA Client" registers communication information (IP, port) into the "MS.02 OPC UA discovery server".

The FBB consumes the *registerServer* method of the **iDiscoveryServer** interface exposed by the "MS.02 OPC UA discovery server".

8.1.1.1.6 AM.06. Automation OPC-UA Server

The "AM.06. Automation OPC-UA Server" provides updates of the monitored automation data to the subscribed components and executes automation methods.

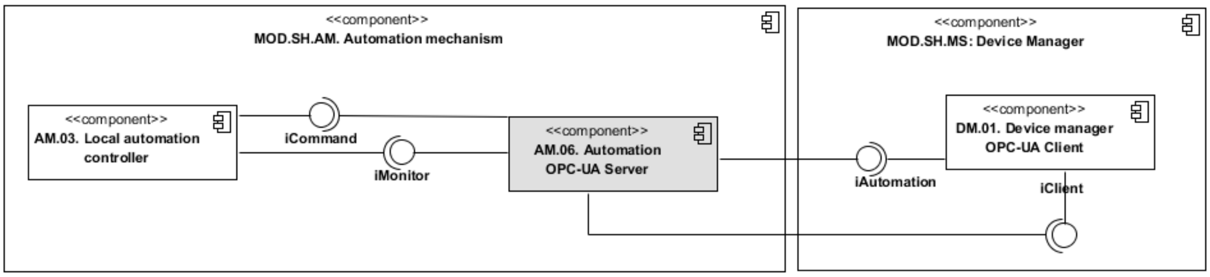


Figure 65 MOD.SH.AM.06. Automation OPC-UA Server interfaces

The FBB exposes **iAutomation** and **iMonitor** interfaces and consumes the **iClient** interface exposed by the "DM.01. Device manager OPC UA" to publish events in the appropriate format.

The following table shows an initial approach to the main methods exposed by each interface.

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

Table 13 MOD.SH.AM.06. Automation OPC-UA Server main methods

8.1.1.1.7 AM.07. Automation data repository

The "AM-07. Automation data repository" enables the storage of automation mechanism related information. It is accessed through the "AM.03.- Local automation controller".

8.1.1.2 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) and in CESA a new automation (i.e. robot for deburring process) will be introduced.

8.1.1.3 A4BLUE Enhancements

Where automation mechanisms are already available A4BLUE will include features to support integration of the automation and adaptation of the overall system. Furthermore, if not available OPC-UA client/server will be developed.

8.1.2 MOD.SH.MS- MEDIATION SERVICES

8.1.2.1 FBB Specification

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

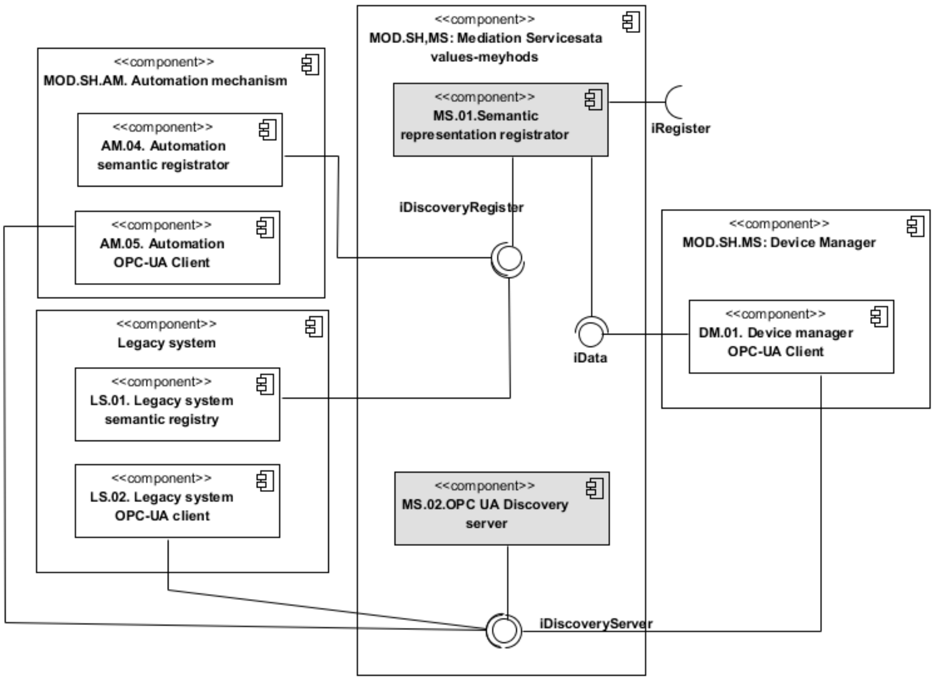


Figure 66 MOD.SH.MS overall interactions

8.1.2.1.1 MS.01. Semantic representation registrator

The "MS.01. Semantic representation registrator" stores the semantic representation provided by the automation mechanisms and legacy systems as well as the data to be managed into the "CAM.03. Asset repository" through the "CAM.03.CAM SERVICE" of the MOD.EN.CAM Collaborative Asset Manager".

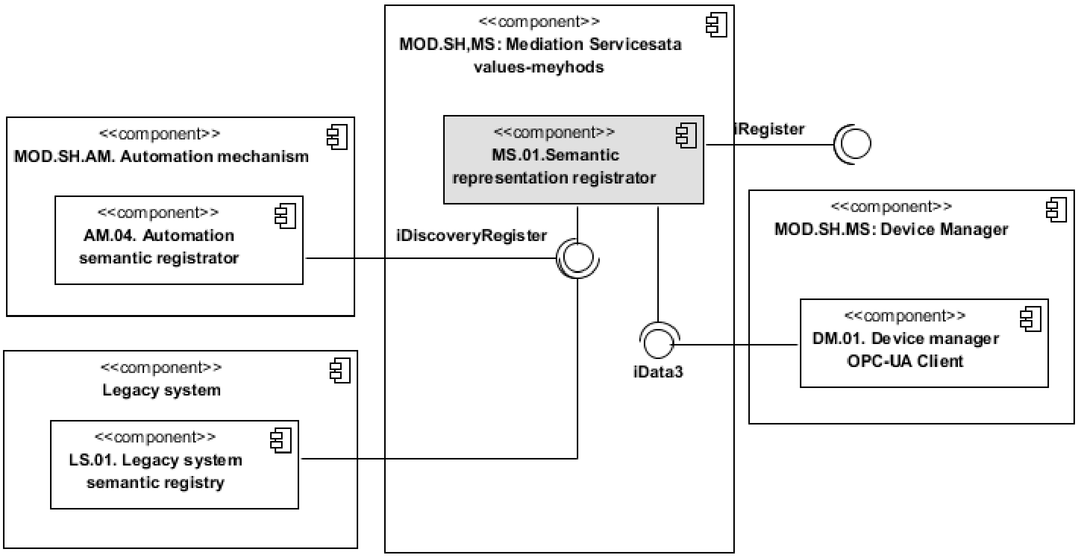


Figure 67 MS.01. Semantic representation repository interfaces

The FBB exposes **iDiscoveryRegister** interface and consumes the methods exposed by the "CAM.01. CAM-SERVICE" and the "DM.01. Device manager OPC UA client". The following table shows an initial approach to the main methods exposed by this interface.

Interface	Method	Description
iDiscoveryRepresentation	registerSemanticRepresentation	Registers the semantic representation

Table 14 MS.01. Semantic representation registrator main methods

8.1.2.1.2 MS.02 OPC UA discovery server

The MS.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].

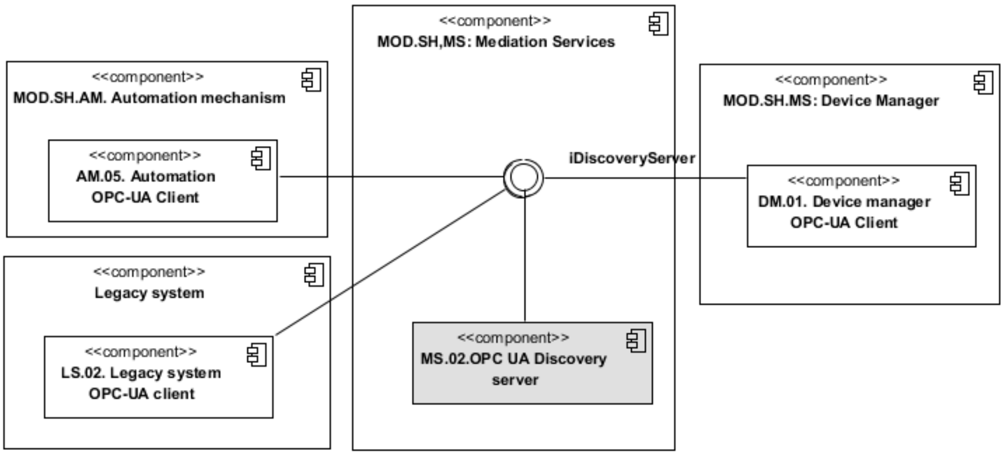


Figure 68 MS.02. OPC-UA discovery server interfaces

The FBB exposes **iDicoveryServer** interface. The following table shows an initial approach to the main methods exposed by this interface.

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

Table 15 MS.01. Semantic representation repository main methods

8.1.2.2 Background Assets

The MOD.SH.MS-component supports the discovery process. To support a standardised approach the “MS.02 OPC UA discovery server” component must follow the specification of the Local Discovery Service (LDS) provided by the OPC Foundation[11].

8.1.2.3 A4BLUE Enhancements

N/A

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

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.

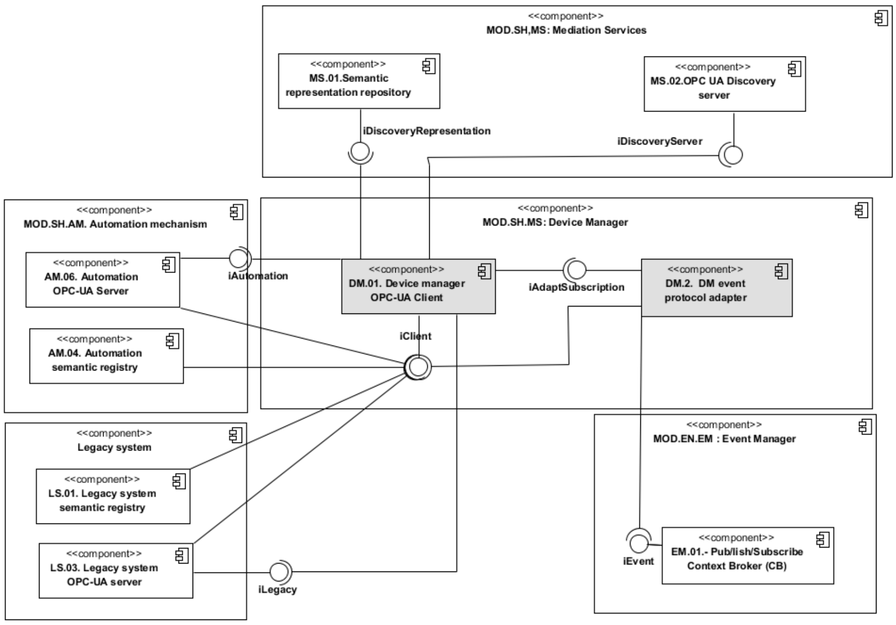


Figure 69 MOD.SH.DM. Overall interactions

8.1.3.1.1 DM.01. Device manager OPC UA client

The “DM.01. Device manager OPC UA client” supports functionalities to:

- retrieve the endpoint of the appropriate OPC-UA Server (e.g. “AM.06. Automation OPC-UA Server”, “LS.03. Legacy system OPC-UA Server”) from the “MS.02 OPC UA discovery server”;
- subscribe to the data to be monitored through the appropriate OPC-UA Server (e.g. “AM.06. Automation OPC-UA Server”, “LS.03. Legacy system OPC-UA Server”);
- publish a register/ de-register event through the “DM.02. DM event protocol adapter”;
- execute the appropriate methods when required.

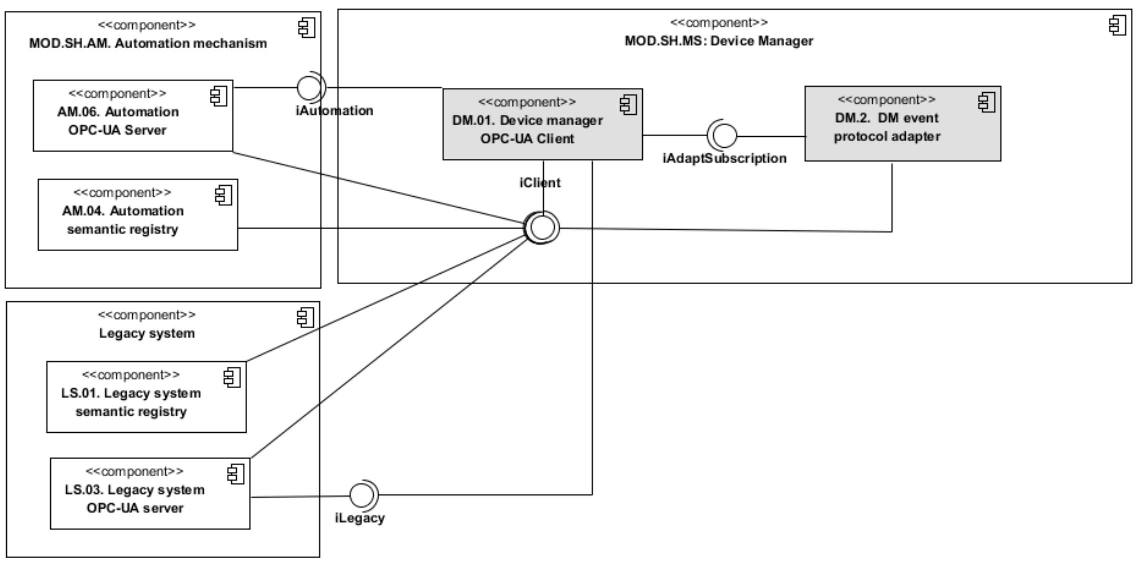


Figure 70 DM.01. Device Manager OPC-UA interfaces

The FBB exposes **iClient and iData** interfaces and consumes the *subscribeToData* and *executeMethod* methods of the **iAutomation** interface exposed by the "AM.06. Automation OPC-UA Server", the *subscribeToData* method of the **iLegacy** interface exposed by the "LS.03. Legacy system OPC-UA Server" and the *publishEvent* method of the **iAdaptSubscription** interface exposed by the "DM.02. DM event protocol adapter".

The following table shows an initial approach to the main methods exposed by the **iClient** and iData interfaces.

Interface	Method	Description
iClient	methodToExecute	Execute method.
	getDataSink	Provides an endpoint to receive the information of the monitored data.
iData	setSubscription	Subscribes to monitored data.

Table 16 DM.01. Device manager OPC-UA methods

8.1.3.1.2 DM.02. DM event protocol adapter

The "DM.02. DM event protocol adapter" supports publish and subscribe capabilities and can adapt the events produced by the" DM.01. Device manager OPC UA" to the event protocol supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM and executes the appropriate method provided by the automation mechanism.

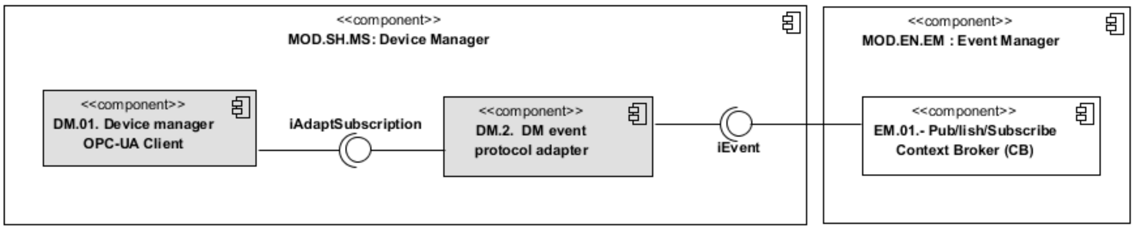


Figure 71 DM.02. DM event protocol adapter interfaces

The FBB 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 and the methodToExecute method of the **iClient** interface exposed by the "DM.01. Device manager OPC UA client".

The following table shows an initial approach to the main methods exposed by the interface.

Interface	Method	Description
iAdaptSubscription	publishEvent	Publish events that should be transformed to comply with the event protocol supported by the Event Manager.

Table 17 DM.02. DM event protocol adapter main methods

8.1.3.2 Background Assets

MOD.SH.DM is in charge of the operation process. A potential background asset candidate has been identified to support some of the identified functionalities however it should be evaluated in the scope of Task 4.1 (CPS enabling connectivity and management) considering the detail design outputs to check if it can fit the expected functionalities. Other potential background asset or approaches will be investigated if required.

IoT HUB

IoT Hub is a framework built based mainly on FIWARE Generic Enablers capable to collect information from IoT devices and distribute them through a NGSI broker.

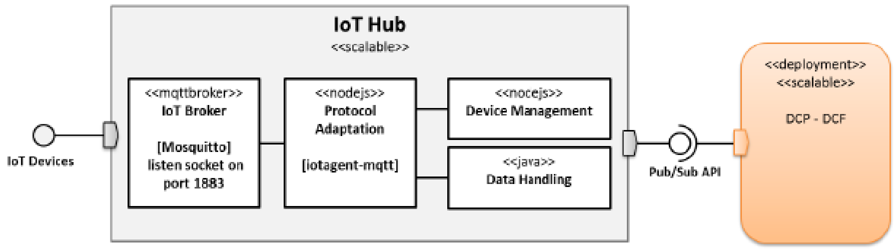


Figure 72 IOT HUB

8.1.3.3 A4BLUE Enhancements

Potential enhancements should depend on the final background asset selected.

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.

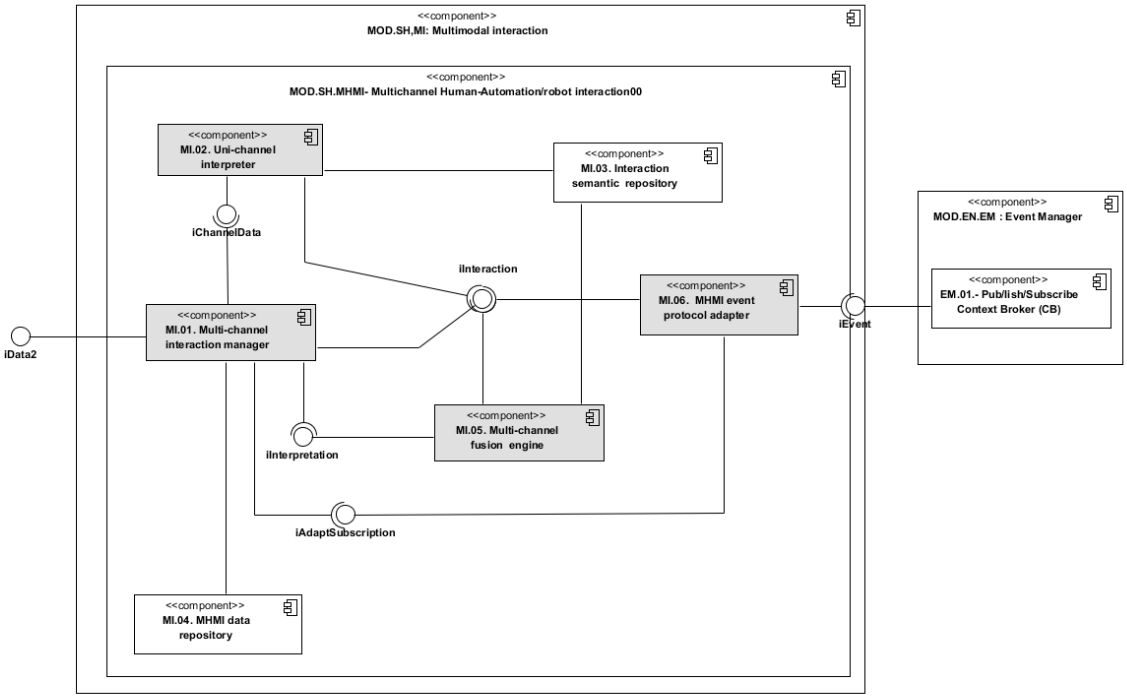


Figure 73 MOD.SH.MHMI. Overall interactions

8.1.4.1.1 MI.01. Multi-channel interaction manager

The “MI.01. Multi-channel interaction manager” orchestrates the rest of components to perform the Human to Machine (H2M) and Machine to Human (M2H) interaction. It supports functionalities to:

- get the inputs provided by the interaction sources (i.e. interaction devices, GUI);
- send collected data to the appropriate interpreter (MI.02. Uni-channel interpreter);

- sends partial interpretations to the "MI.05. Multi-channel fusion engine";
- publish and subscribe (e.g. context info, actions) events through the "MI.06. MHMI event protocol adapter";
- manage the feedback on multi-channel interpretation
- support data persistence for interpretation incremental learning readiness by interacts with the "MI.04. MHMI data repository"

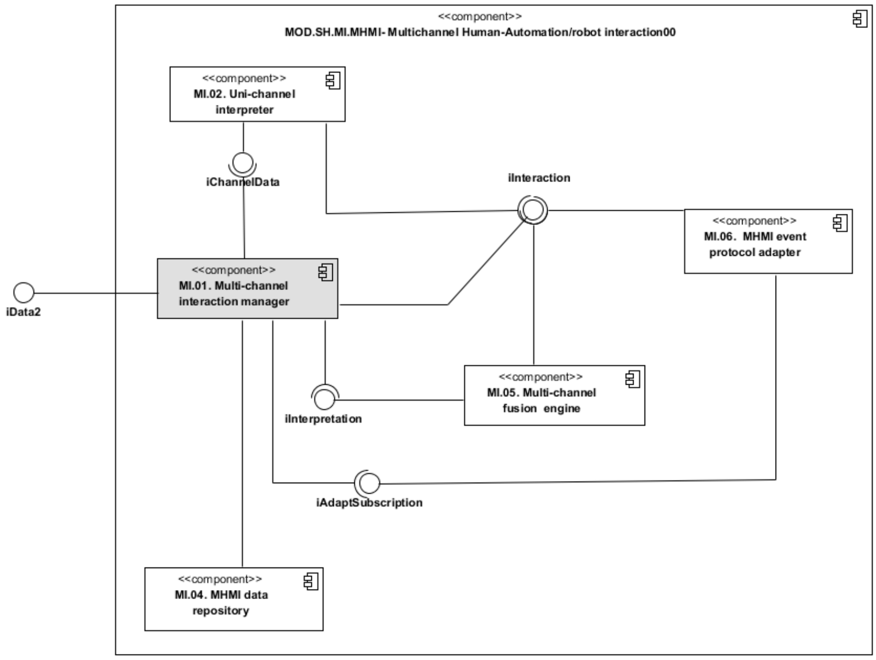


Figure 74 MI.01. Multi-channel interaction manager interfaces

The FBB 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 *publishEvent* method of the **iAdaptSubscription** interface exposed by the "MI.06. MHMI event protocol adapter".

Furthermore, it interacts with the "MI.04. MHMI data repository" to store and retrieve the required information.

The following table shows an initial approach to the main methods exposed by each interface.

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 18 MI.01. Multi-channel interaction manager main methods

8.1.4.1.2 MI.02. Uni-channel interpreter

The "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 "MI.03. Interaction semantic repository" and sends the obtained channel interpretation to the "MI.01 Multi-channel interaction manager"

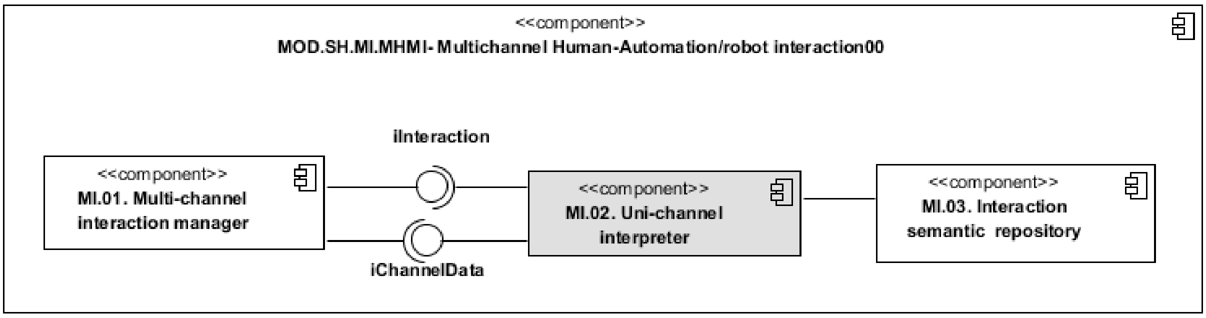


Figure 75. MI.02 Uni channel interpreter interfaces

The FBB 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 "MI.03. Interaction semantic repository" to store and retrieve the required information.

The following table shows an initial approach to the main methods exposed by the **iChannelData** interface.

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

Table 19. MI.02. Uni-channel interpreter main methods

8.1.4.1.3 MI.03. Interaction semantic repository

The "MI.03. Interaction semantic repository" supports the storage and data retrieval of the interaction semantic repository.

8.1.4.1.4 MI.04. MHMI data repository

The "MI.04. MHMI data repository" support data persistence for interpretation incremental learning readiness.

8.1.4.1.5 MI.05. Multi-channel fusion engine

The "MI.05. Multi-channel fusion engine" fuses the uni-channel interpretations (i.e. partial) considering the instruction's semantic representation provided by the "MI.03. Interaction semantic repository" 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.).

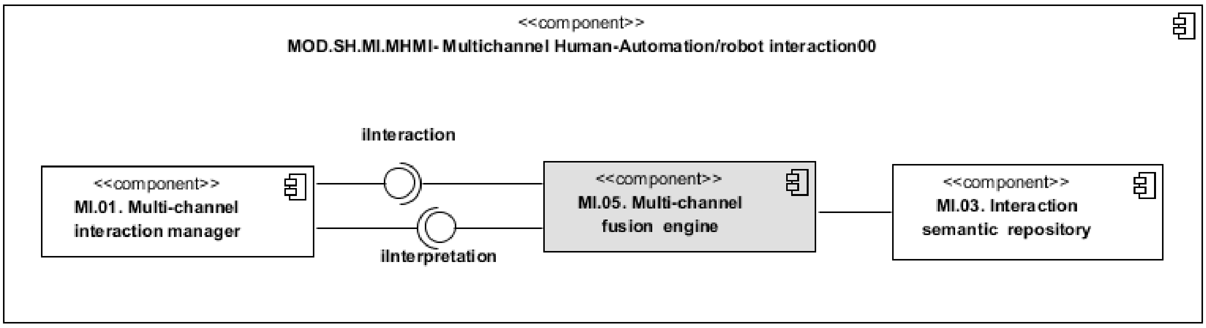


Figure 76 MI.05. Multi-channel fusion engine interfaces

The FBB exposes **iInterpretation** interface and consumes the *saveMultichanneInterpretation* method of the **iInteraction** interface exposed by the "MI.01 Multi-channel interaction manager". Furthermore, it interacts with the "MI.03. Interaction semantic repository" to store and retrieve the required information.

The following table shows an initial approach to the main methods exposed by each interface.

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

Table 20 MI.05. Multi-channel fusion engine main methods

8.1.4.1.6 MI.06. MHMI event protocol adapter

The "MI.06. MHMI event protocol adapter" supports publish and subscribe capabilities and can adapt the events produced by the" MI.01. Multi-channel interaction manager" to the event protocol supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM and executes the appropriate method exposed by the **iInteraction** interface of the "MI.01. Multi-channel interaction manager".

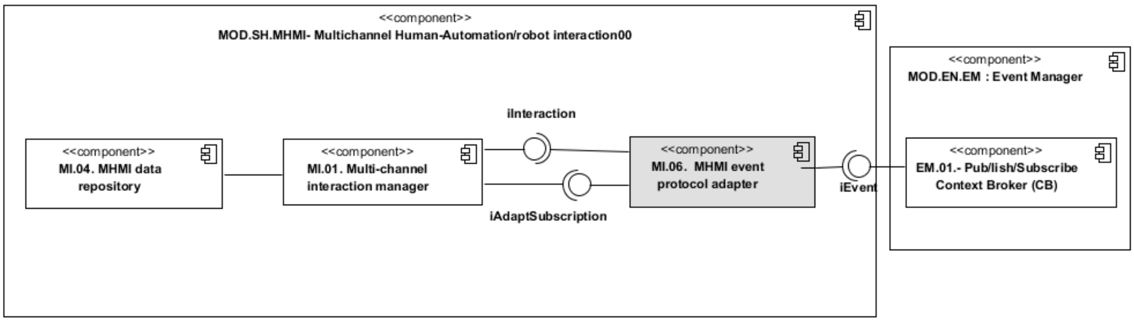


Figure 77 MI.06. MHMI event protocol adapter interfaces

The FBB 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 and executes the appropriate method exposed by the **iInteraction** interface of the "MI.01. Multi-channel interaction manager".

The following table shows an initial approach to the main methods exposed by the interface.

Interface	Method	Description
iAdaptSubscription	publishEvent	Publish events that should be transformed to comply with the event protocol supported by the Event Manager.

Table 21 MI.06. MHMI event protocol adapter main methods

8.1.4.2 Background Assets

4X3 Multimodal human-robot natural communication component developed in the scope of FourByThree project[12] 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 it is 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.

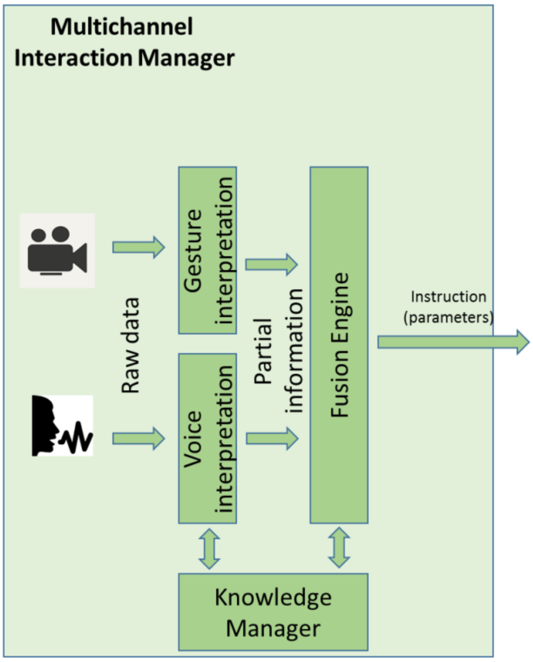


Figure 78 4X3 Multimodal human-robot natural communication component

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[13] based node.

[12] FourByThree (): http://fourbythree.eu/

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

8.1.4.3 A4BLUE Enhancements

The MOD.SH.MMHI will considerably 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. operator related info) as well as manage the feedback process and support event exchange.

8.1.5 MOD.SH.MI.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.

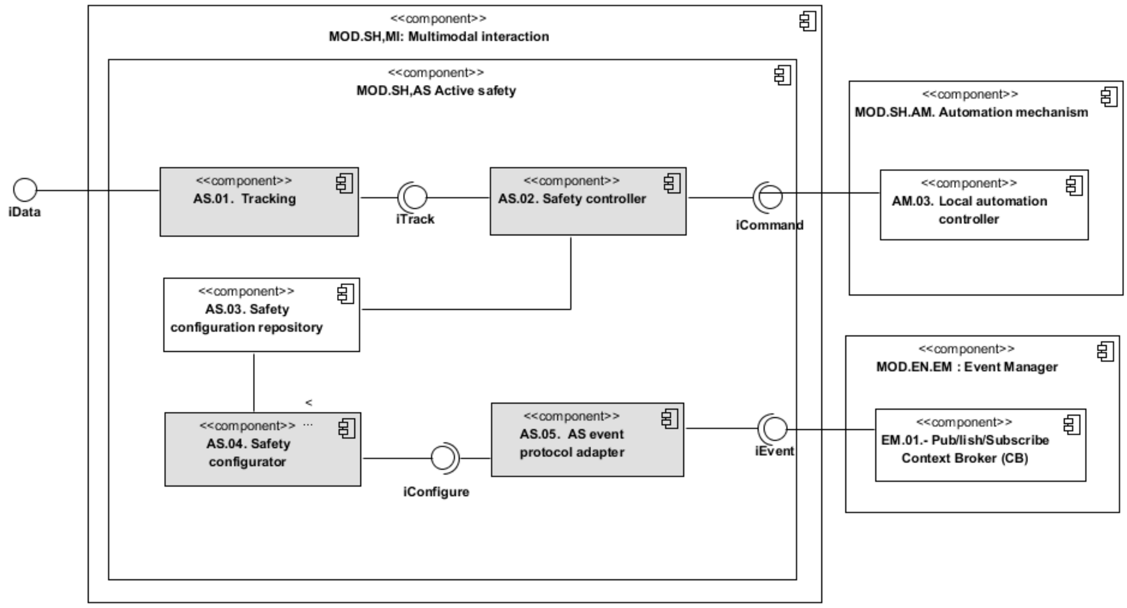


Figure 79 MOD.SH.MHMI. Overall interactions

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.

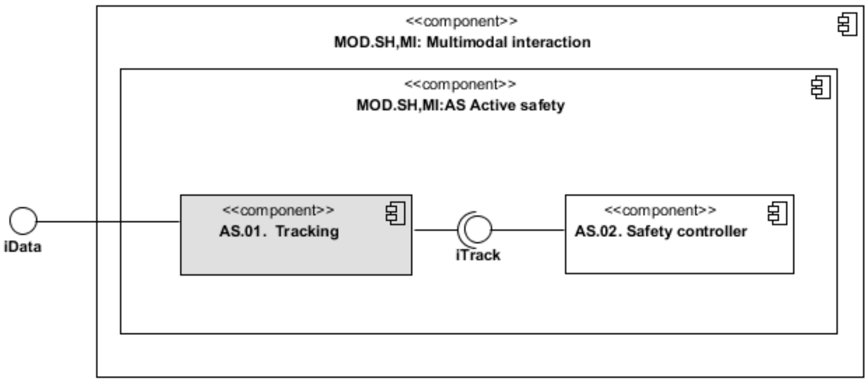


Figure 80 AS.01. Tracking interfaces

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

The following table shows an initial approach to the main methods exposed by **iData** interface.

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

Table 22 AS.01. Tracking main methods

8.1.5.1.2 AS.02. Safety controller

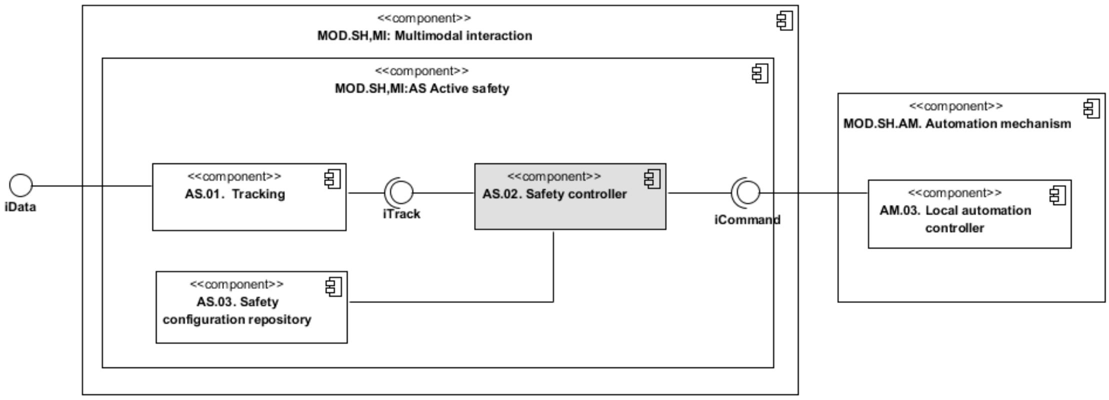


Figure 81 MI.06. MHMI event protocol adapter interfaces

The FBB 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".

The following table shows an initial approach to the main methods exposed by **iTrack** interface.

Interface	Method	Description
iData	saveTracking	Save tracking information.

Table 23 AS.02. Safety controller main methods

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.

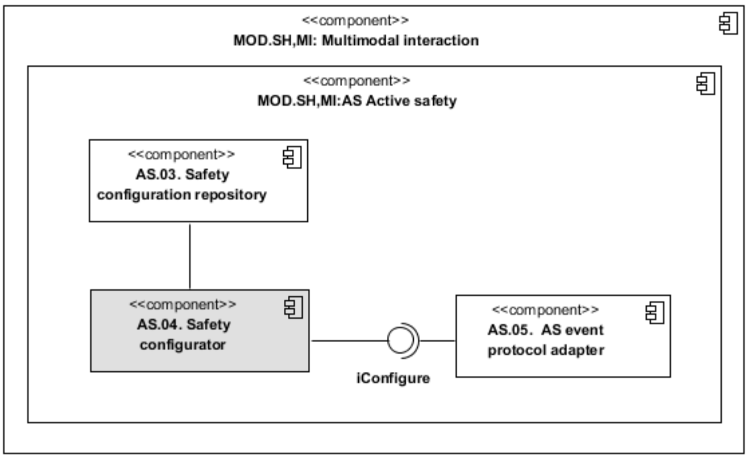


Figure 82 AS.04. Safety configurator interfaces

The FBB exposes **iConfigure** interface. The following table shows an initial approach to the main methods exposed by the interface.

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

Table 24 AS.04. Safety configurator main methods

8.1.5.1.5 AS.05. AS event protocol adapter

The "AS.05. AS event protocol adapter" supports publish and subscribe capabilities and can adapt the events produced by the" AS.02. Safety controller" to the event protocol supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM and executes the appropriate method exposed by the "AS.04. Safety configurator".

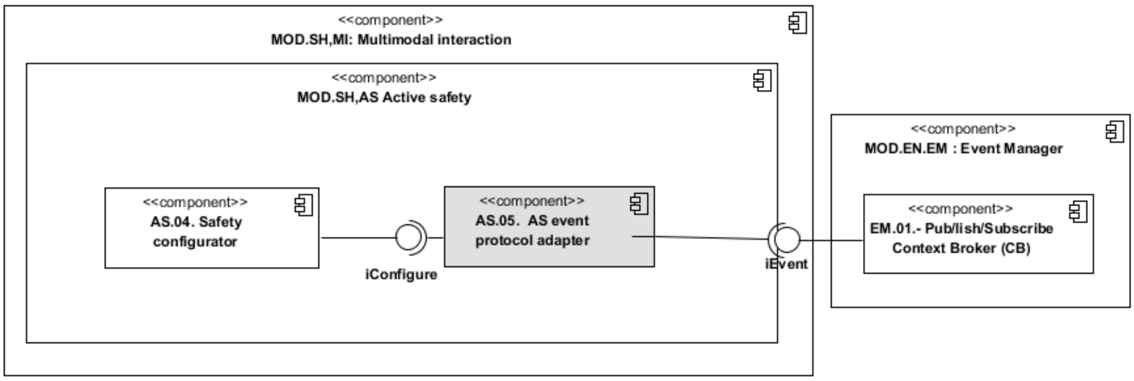


Figure 83 AS-05. AS event protocol adapter interfaces

The FBB 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 and executes the appropriate method exposed by the **IConfigure** interface of the "AS.04. Safety configurator".

The following table shows an initial approach to the main methods exposed by each interface.

Interface	Method	Description
iAdaptSubscription	publishEvent	Publish events that should be transformed to comply with the event protocol supported by the Event Manager.

Table 25 AS-05. AS event protocol adapter main methods

8.1.5.2 Background Assets

Proximity detection and Trajectory prediction modules developed by TEK in the scope of internal R&D and the EuroC project[14].

EUROC: [14] http://www.euroc-project.eu/

Proximity detection module

The proximity detection method is based on monitoring a volume around a device under control, by calculating a distance between the device under control and an obstacle. Preferably, the minimum distance between the two elements is calculated. The risk of collision between the device under control and an obstacle within that monitored volume can thus be determined.

The proximity detection method is implemented by a software module that can be configured and integrated in any application in which there is an interest in controlling a volume around a device under control, either a mobile device or a static device.

Trajectory prediction module

Trajectories can be very useful to take measures to avoid collisions between the elements in the scene even to be ready to take 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

The MOD.SH.AS will considerably rely on the above-mentioned background assets, especially since the support the target safety modes it they will be enhanced to increase reliability, consider context information (e.g. operator related info) and support event exchange.

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 as described in Section 3.3 - FITMAN), namely the FITMAN CAM SE[15] and its evolution developed and federated to the BEinCPPS Future Internet Platform in the BEinCPPS project[16].

8.2.1.1 FBB Specification

The Figure 84 here below shows the main components of the MOD.EN.CAM, describing both main Building Blocks and information flows.

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

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

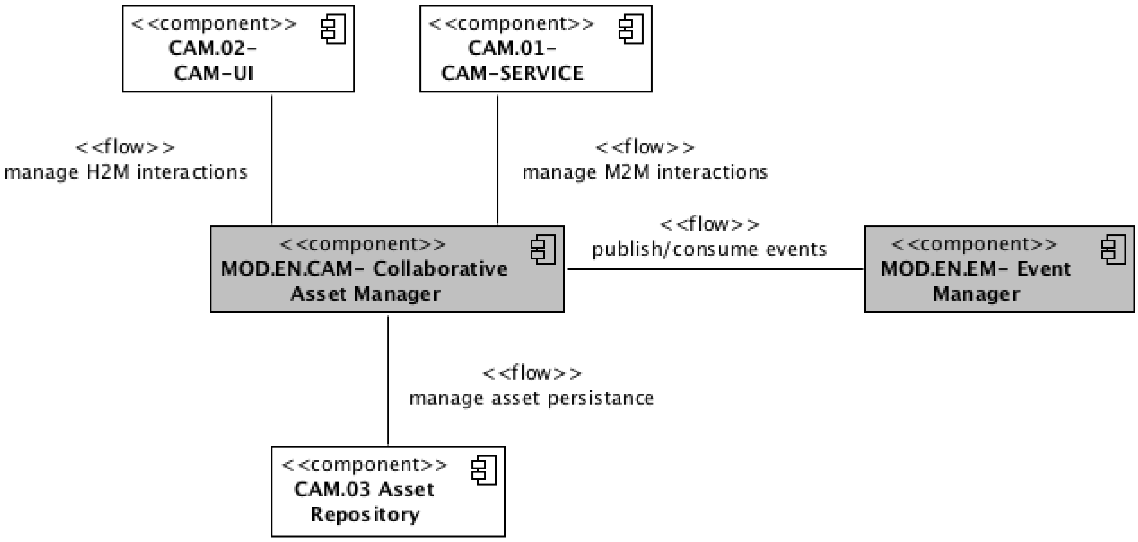


Figure 84 MOD.EN.CAM. Decomposition into Functional Building blocks

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.

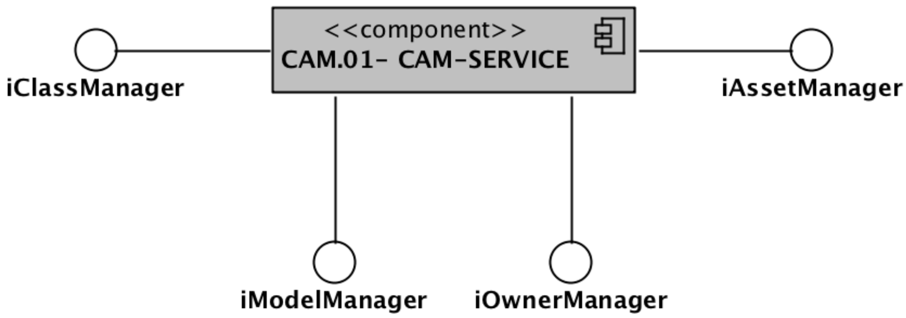


Figure 85 MOD.EN.CAM CAM Service interfaces

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, by using an ontology loaded in a RDF4J repository. Once the ontology is installed, the user can use CAM UI to extend it – i.e., adding custom domain concepts.

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 user 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 (for instance we can use the eCl@ss from AUTOMATIONML), and following the outcomes of the project Task T2.3 (i.e. the Virtual Asset Model).

Further enhancements will target the enrichments of assets representation, taking into account also 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.

8.2.1.3.1 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 D2.6 or in the outcomes of WP3.

Interface	Method	Description
iClassManager	getClasses	Get all classes.

Interface	Method	Description
	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.
	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,	Delete Relationship of assetName

Interface	Method	Description
	relationshipName)	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
	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.

Interface	Method	Description
	updateOwner(ownerName)	Update Owner named ownerName.
	deleteOwner(ownerName)	Delete Owner named ownerName.

Table 26 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 - FITMAN): FIWARE Orion Context Broker[17] and FIWARE Proactive GE[18].

8.2.2.1 FBB Specification

The Figure 86 here below shows the main components of the MOD.EN.EM, describing both main Building Blocks and information flows.

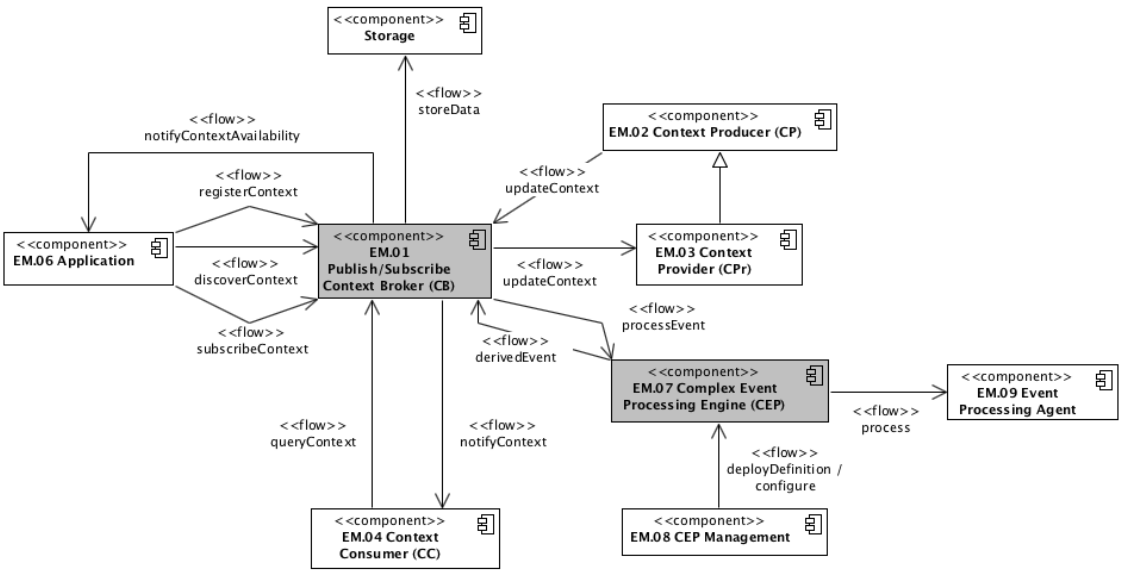


Figure 86 MOD.EN.EM. Decomposition into Functional Building blocks

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.

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

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

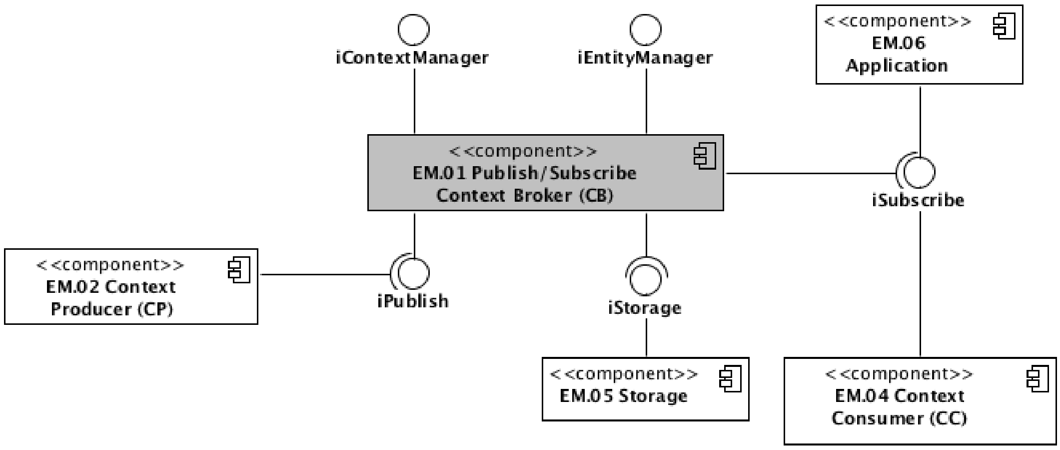


Figure 87 MOD.EN.EM. Context Broker interfaces

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 sending a request to the CB or invoking directly 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 kind of Context Consumer may expose update 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.

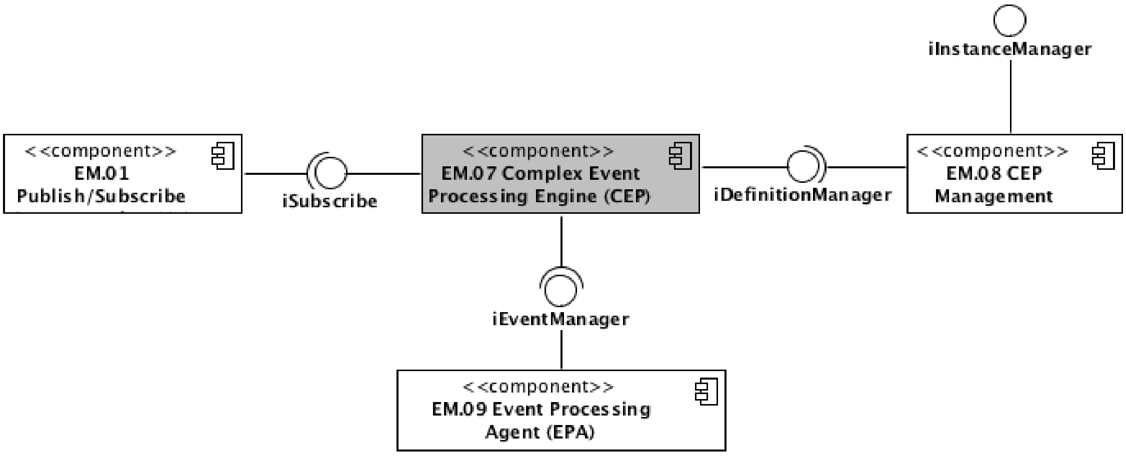


Figure 88 MOD.EN.EM. Complex Event Processing interfaces

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 to notice that Every pattern is associated with an Event processing context. Event processing context groups event instances so that they can be processed in a related way. It assigns each event instance to one or more processing context partitions. 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.

8.2.2.2 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's integrated with several other components of the A4BLUE platform as shown previously in Figure 47.

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[19]. 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.

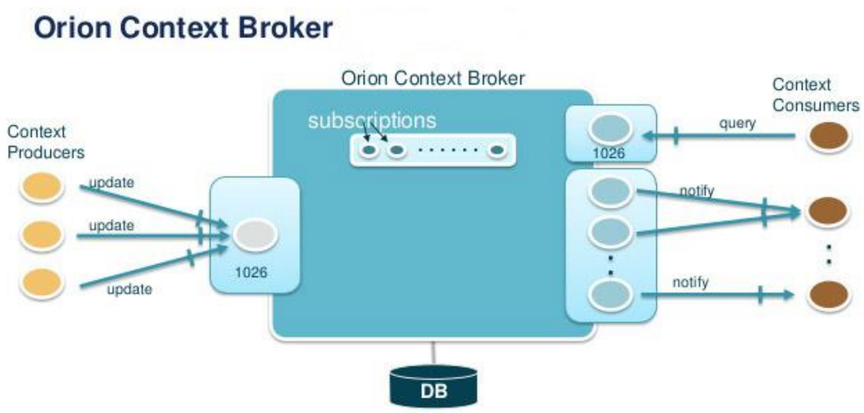


Figure 89 MOD.EN.EM. FIWARE Orion Context Broker

On the other hand, FIWARE Proactive GE is the reference implementation of FIWARE's Complex Event Processing Generic Enabler Open Specification. The CEP GE analyses event data in real-time, generates immediate insight and enables instant response to changing conditions. While standard reactive applications are based on reactions to single events, the CEP GE reacts to situations rather than to single events. A situation is a condition that 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

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

CEP engine can detect combinations of events which are meaningful, through use and detection of patterns over incoming events, and let other tools to respond to single events with some condition.

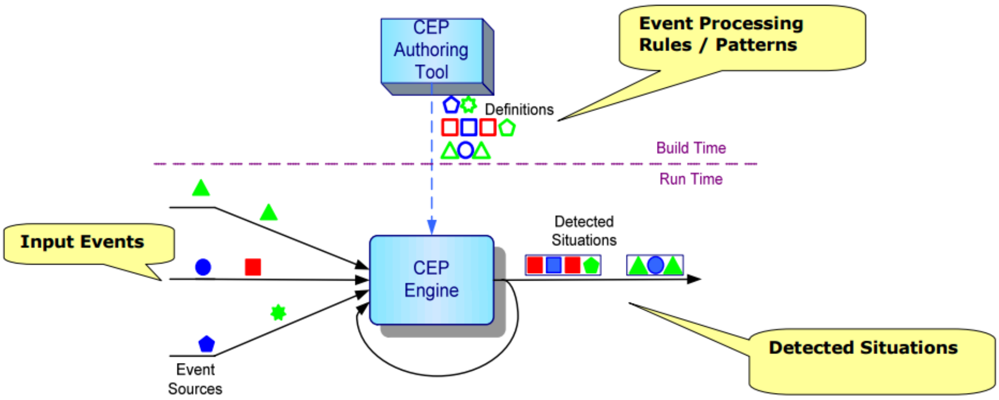


Figure 90 MOD.EN.EM FIWARE Proactive CE

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 explore also new coming solutions into the FIWARE ecosystem providing the same capabilities (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 to persist exchanged information using a lightweight storage system).

8.2.2.3.1 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 D2.6 or in the outcomes of WP5.

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.

Table 27 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 D2.6 or in the outcomes of WP5.

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.
	addDefinition	Add a new definition in the repository.
	retrieveDefinition	Retrieve the complete definition in JSON

Interface	Method	Description
		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 28 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 to notice, 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.

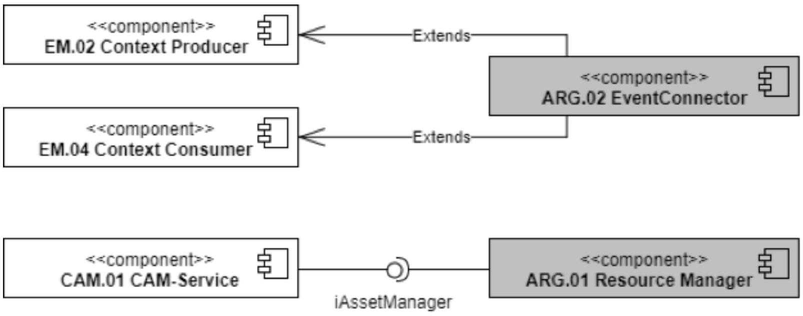


Figure 91 MOD.EN.ARG. VR/AR Training System Back-end module interactions

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 - FITMAN): FIWARE Identity Management - KeyRock [20].

8.2.4.1 FBB Specification

The Figure 86 here below shows the main components of the MOD.EN.IM, describing both main Building Blocks and information flows.

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

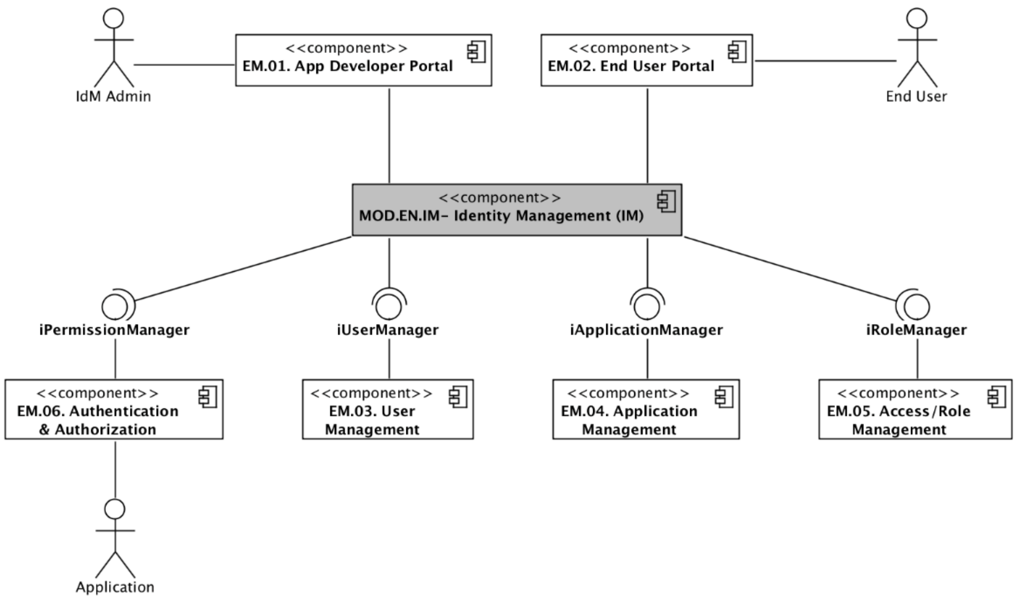


Figure 92 MOD.EN.IM. Decomposition into Functional Building blocks

8.2.4.1.1 IM.01. Application Developer Portal

The Application Developed 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[21].

[21] 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[22] and OAuth2[23].

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. A4BLUW 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[24].

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

[23] https://oauth.net/

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

8.2.4.3.1 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 D2.6 or in the outcomes of WP3.

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 29 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 93 here below shows the main components of the MOD.BU.KM, describing both main Building Blocks and information flows.

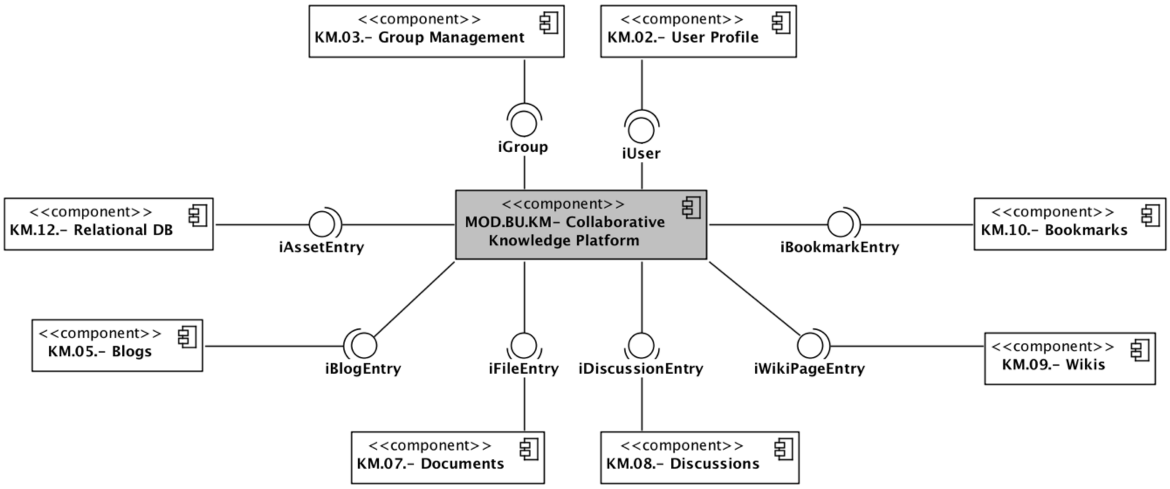


Figure 93 MOD.BU.KM. Decomposition into Functional Building blocks

8.3.1.1.1 KM.01.- Workspaces List

Workspaces List is an application providing the list of workspaces, that 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 his/her profile and starting to follow. This application shows all the teams for which she has membership and enables her 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 she quickly find what she is 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 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. Still, very used are the corporate blog for internal communications of a general nature, 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.

8.3.1.2 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, 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.

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.

8.3.1.3.1 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 D2.6 or in the outcomes of WP3.

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.

Interface	Method	Description
	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.
	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.

Interface	Method	Description
	subscribe	Subscribe to Discussion entries updates.
	updateMessage	Update a specific Discussion entry.
iFileEntry	addFileEntry	Add a new File entry.
	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.

Table 30 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.

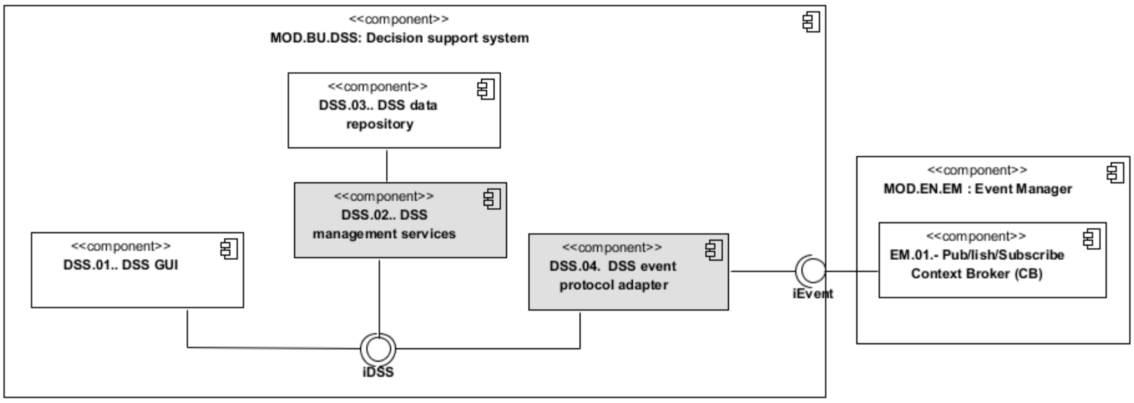


Figure 94 MOD.BU.DSS overall interactions

8.3.2.1.1 DSS.01.- DSS GUI

The "DSS.01.- DSS GUI" allows the supervisors to access an aggregated, graphical view of relevant operational results to support decision making.

The FBB consumes the *getAggredatedResults* method of the **iDSS** interface exposed by the "DSS.02.- DSS management services".

8.3.2.1.2 DSS.02.- DSS management services

The "DSS.02.- DSS management services" provides functionalities to:

- collect the operational results coming through the "DSS.04.- DSS event protocol adapter";
- aggregate and process the operational results;
- make available to the aggregated and processed operational results to the "DSS.01.- DSS GUI";
- store and retrieve data in/from the DSS.03.- DSS data repository.

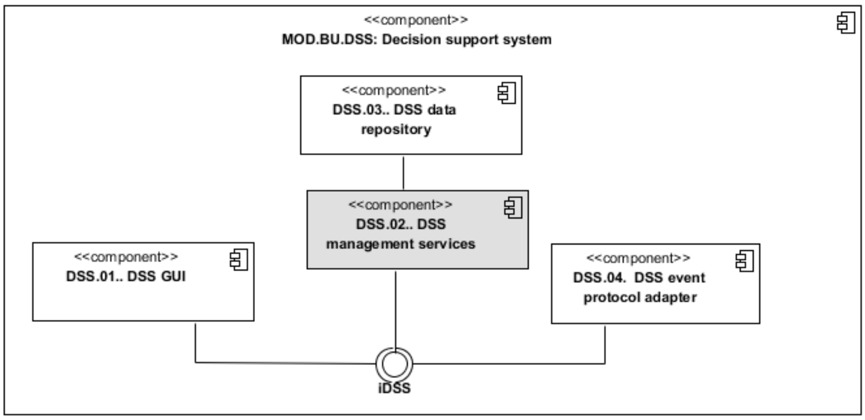


Figure 95 DSS.02. DSS management services interfaces

The FBB exposes **iDSS** interfaces. The following table shows an initial approach to the main methods exposed by the interface.

Interface	Method	Description
iDSS	setOperationalData	Save operational data
	getOperationalData	Retrieves operational data

Table 31 DSS.02. DSS management services main methods

8.3.2.1.3 DSS.03.- DSS data repository

The "DSS.03.- DSS data repository" enables the storage of operational data coming from the A4BLUE system and supporting decision making related information. It is accessed through the "DSS.02. DSS management services".

8.3.2.1.4 DSS.04. DSS event protocol adapter

The DSS.04 supports publish and subscribe capabilities and can adapt the events produced by the" CQMS.08.- Questionnaire management services" to the event protocol supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM and executes the *setOperationalData* method exposed by the **iDSS** interface of the "DSS.02.- DSS management services" component.

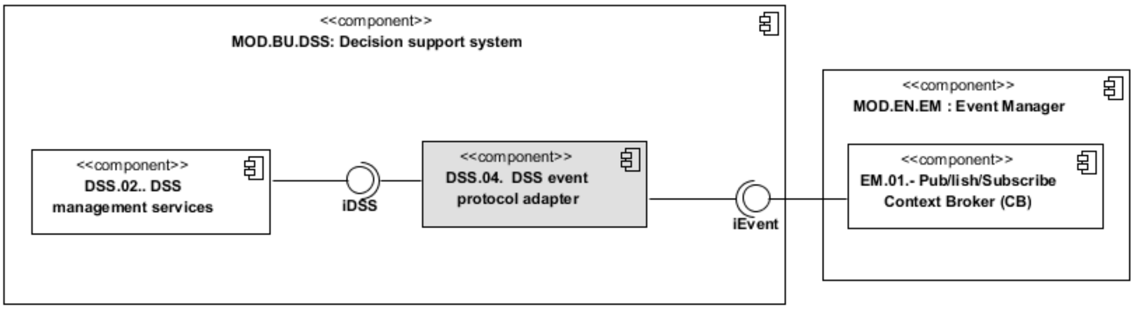


Figure 96 DSS.04. DSS event protocol adapter interfaces

The FBB 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 of the **iDSS** interface.

8.3.2.2 Background Assets

As an initial approach, several potential background asset candidates have been identified to support some of the identified functionalities however it should be evaluated in the scope of Task 5.3 (Assistance: Decision Support System) considering the detail design outputs. Other potential background asset could be investigated if required.

KNOWAGE

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 complete solution for a certain task, or combined with one another to ensure full coverage of user' 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.

Figure 97 KNOWAGE

KNOWAGE can aggregate data in custom and high-view performance dashboard, 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 several processing engines (e.g. R and Python scripts can be used to pre-process dataset before starting visualization and reporting tasks)

KNOWAGE is available on 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.

Grafana

Grafana is an open source project for visualizing time series metrics through graphs and dashboards on a web portal. It supports several time series databases as the backend datastore.

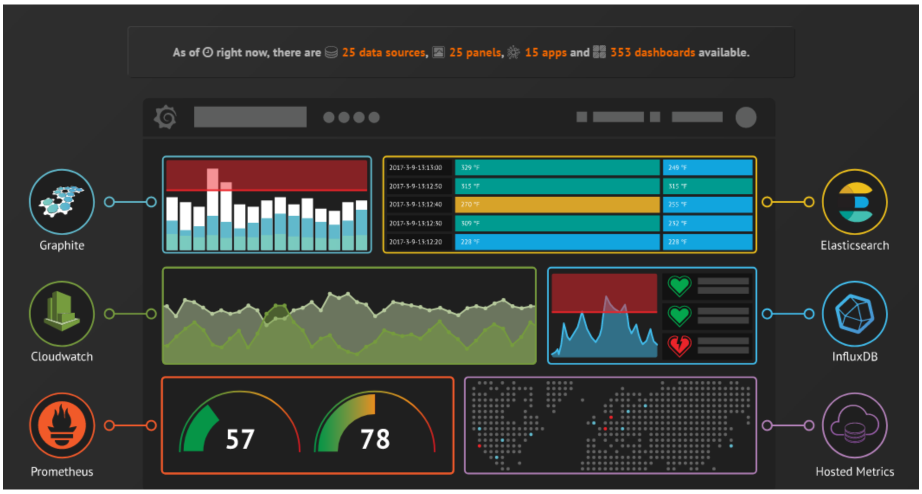


Figure 98 Grafana

Google Data Studio

Google Data Studio 360 is a web analytics dashboard report service which turns your analytics data into informative reports which are easy to read, share and customizable. The Data Studio is part of the Google Analytics 360 Suite released by Google 2015.

8.3.2.3 A4BLUE Enhancements

Potential enhancements should depend on the final background asset selected

8.3.3 MOD.BU.MON- MONITORING

MOD.BU.MON is expected to follow an analogous approach to the one identified in the previous section (“8.3.2 – MOD.BU.DSS- DECISION SUPPORT SYSTEM (DSS)”).

Potential updates could come from the preliminary results of Task 2.5 – “Formalisation of the economic and technical assessment framework)” and should be reported in the main outcome of the related task (i.e. D2.4 due at M13) but also in the final release of this deliverable (i.e. D2.6 due at M19).

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

8.3.4.1 FBB Specification

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

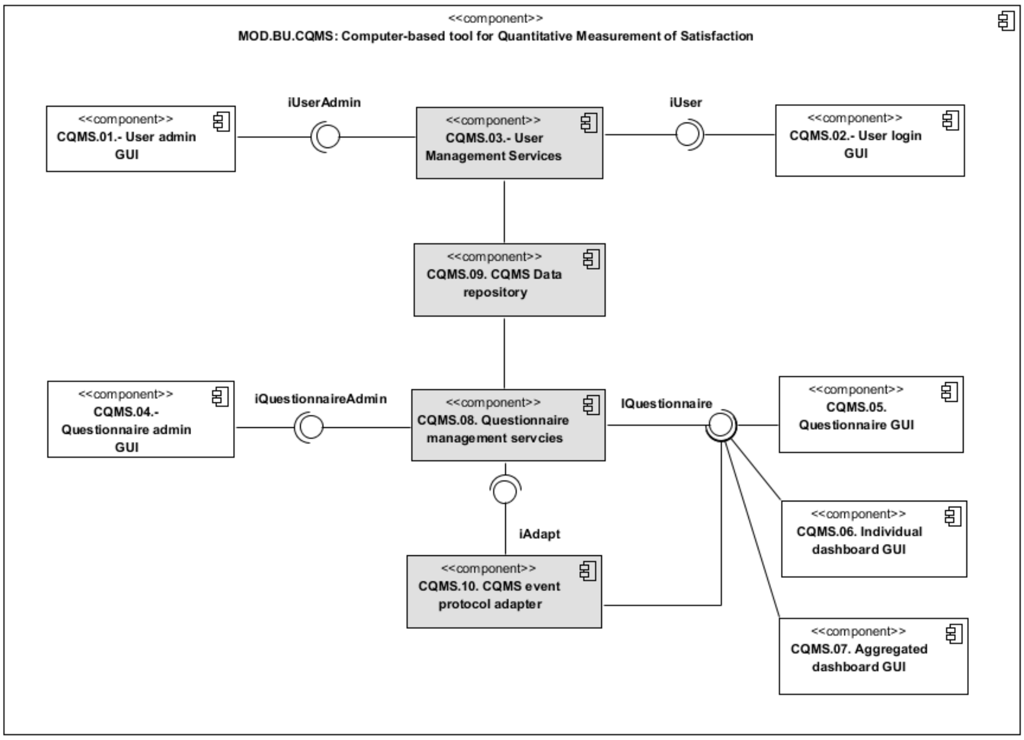


Figure 99 MOD.BU.CQMS overall interactions

8.3.4.1.1 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.

The FBB consumes the *setUser*, *updateUser*, *deleteUser* and *getUserList* methods of the **iUserAdmin** interface exposed by the CQMS.08.

8.3.4.1.2 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.

The FBB consumes the *getAuthorisation* method of the **iUser** interface exposed by the CQMS.03.

8.3.4.1.3 CQMS.03. User Management Services

The CQMS.03 provides functionalities to:

- create (i.e. including access credentials), read, update and delete questionnaires;
- generate user/token;
- check user access authorisation.

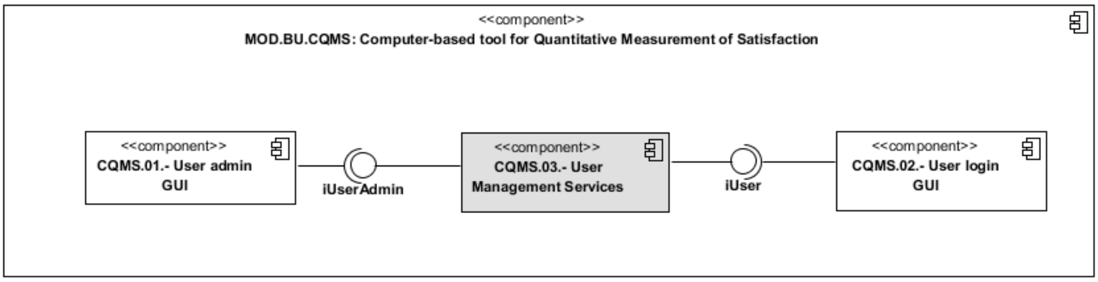


Figure 100 CQMS.03. Main interfaces

The FBB exposes **iUserAdmin** and **iUser** interfaces. The following table shows an initial approach to the main methods exposed by each interface.

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 32 CQMS.03. Main methods

8.3.4.1.4 CQMS.04. Questionnaire Admin GUI

The CQMS.04 enables the system administrator to model the appropriate questionnaires by using the CQMS.08.

The FBB consumes the *setQuestionnaire*, *updateQuestionnaire* and *deleteQuestionnaire* methods of the **iQuestionnaireAdmin** interface exposed by the CQMS.08.

8.3.4.1.5 CQMS.05. Questionnaire GUI

The 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.

The FBB consumes the *getQuestionnaireList*, *getQuestionnaire* and *saveQuestionnaireResults* methods of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.4.1.6 CQMS.06. Individual questionnaire dashboard GUI

The 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

The FBB consumes the *getIndividualResults* method of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.4.1.7 CQMS.07. Aggregated questionnaire dashboard UI

The 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.

The FBB consumes the *getAggredatedResults* method of the **iQuestionnaire** interface exposed by the CQMS.08.

8.3.4.1.8 CQMS.08. Questionnaire Management Services

The CQMS.08 provides functionalities to:

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

The FBB 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.

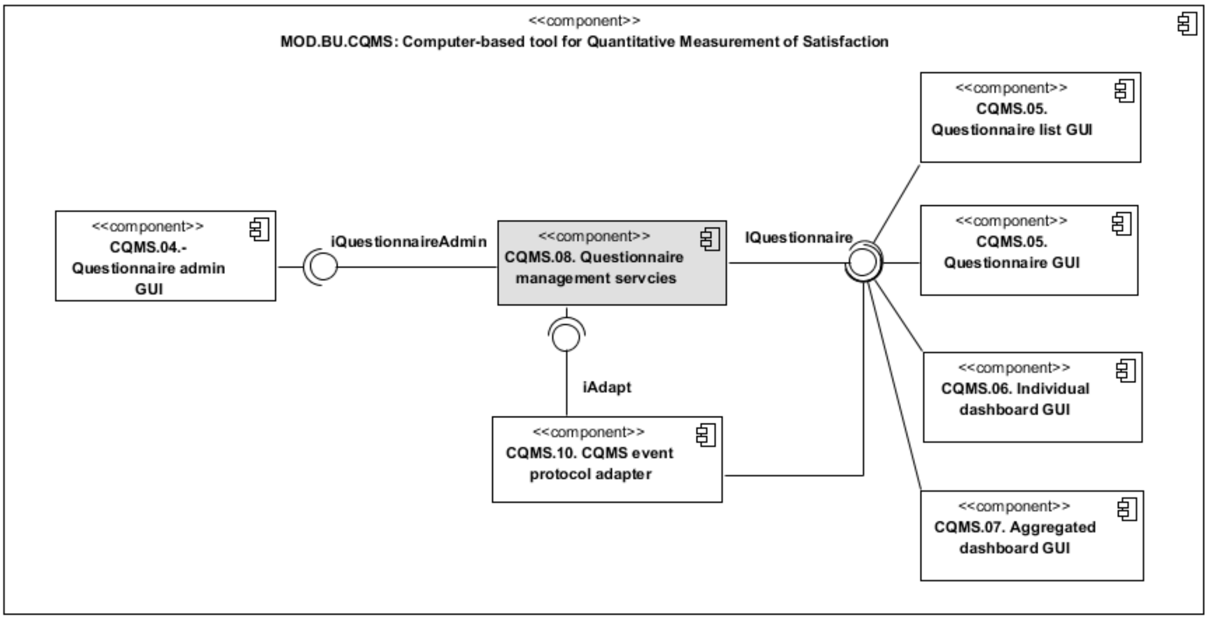


Figure 101 CQMS.08. Questionnaire management services interfaces

The following table shows an initial approach to the main methods exposed by each interface.

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 33 CQMS.08. Questionnaire management services main methods

8.3.4.1.9 CQMS.09. CQMS data repository

The “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”.

8.3.4.1.10 CQMS.10. CQMS event protocol adapter

The CQMS.10 supports publish and subscribe capabilities and can adapt the events produced by the “CQMS.08.- Questionnaire management services” to the event protocol supported by the Event Manager (MOD.EN.EM). Furthermore, it processes the event produced by the MOD.EN.EM and executes the appropriate method exposed by the **iQuestionnaire** interface of the CQMS.08 component.

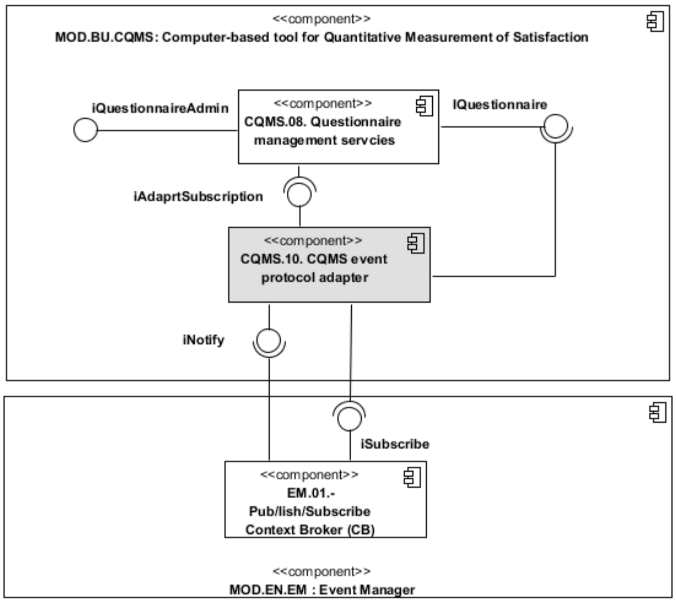


Figure 102 CQMS.10. CQMS event protocol adapter interfaces

The FBB 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 **iQuestionnaire** interface.

The following table shows an initial approach to the main methods exposed by each interface.

Interface	Method	Description
iAdaptSubscription	publishEvent	Publish questionnaire event to be transformed to comply with the event protocol supported by the Event Manager.

Table 34 CQMS.10. CQMS event protocol adapter main methods

8.3.4.2 Background Assets

FIS-TAR Health Questionnaire (HQ-SE)[25] is a specific enabler (SE) developed in the scope of the FISTAR project[26]. The HQ-SE provides a RESTFUL API that allows:

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

- to create simple assessment (on-demand) and monitoring (periodicity based) questionnaires from csv files (no graphical user interface is provided),
- to schedule monitoring questionnaires,
- to collect the results from the questionnaires,
- (optionally) to store questionnaire models and results into the repository. based on configuration options,
- to 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 and FI-STAR Event Service SE to be used by other components, based on configuration options.

8.3.4.3 A4BLUE Enhancements

The main goal of the A4BLUE Computer-based tool for Quantitative Measurement of Satisfaction is to provide 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 considerably rely on the above mentioned background asset, especially since it provides the capability to define, manage and schedule questionnaires. Furthermore, the HQ-SE will be enhanced by including graphical user interfaces to simplify the definition of the questionnaire models and enhance publish/subscribe capabilities to support online working mode as described in the section about CQMS.10.

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

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
LS.01. Legacy system semantic registrator											
LS.02. Legacy system OPC-UA Client											
LS.03. Legacy system OPC-UA Server											
AM.01. Automation hardware											
AM.02. Automation GUI											
AM.03. Local automation controller											
AM.04. Automation semantic registrator											
AM.05. Automation OPC-UA Client											
AM.06. Automation OPC-UA Server											
AM.07. Automation data repository											
MS.01. Semantic representation registrator											
MS.02 OPC UA discovery server											
DM.01. Device manager OPC UA											
DM.02. DM event protocol adapter											

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.01. Multi-channel interaction manager											
MI.02. Uni-channel interpreter											
MI.03. Interaction semantic repository											
MI.04. MHMI data repository											
MI.05. Multi-channel fusion engine											
MI.06. MHMI event protocol 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											
DSS.01. DSS GUI											
DSS.02. DSS management services											
DSS.03. DSS data repository											
DSS.04. DSS event protocol adapter											

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
MON.01. MON GUI											
MON.02. MON management services											
MON.03. MON data repository											
MON.04. MON event protocol adapter											
CQMS.01.- User admin GUI											
CQMS.02.- Login GUI											
CQMS.04.- Questionnaire admin GUI											
CQMS.05.- Questionnaire list 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 protocol adapter											

Table 35 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.

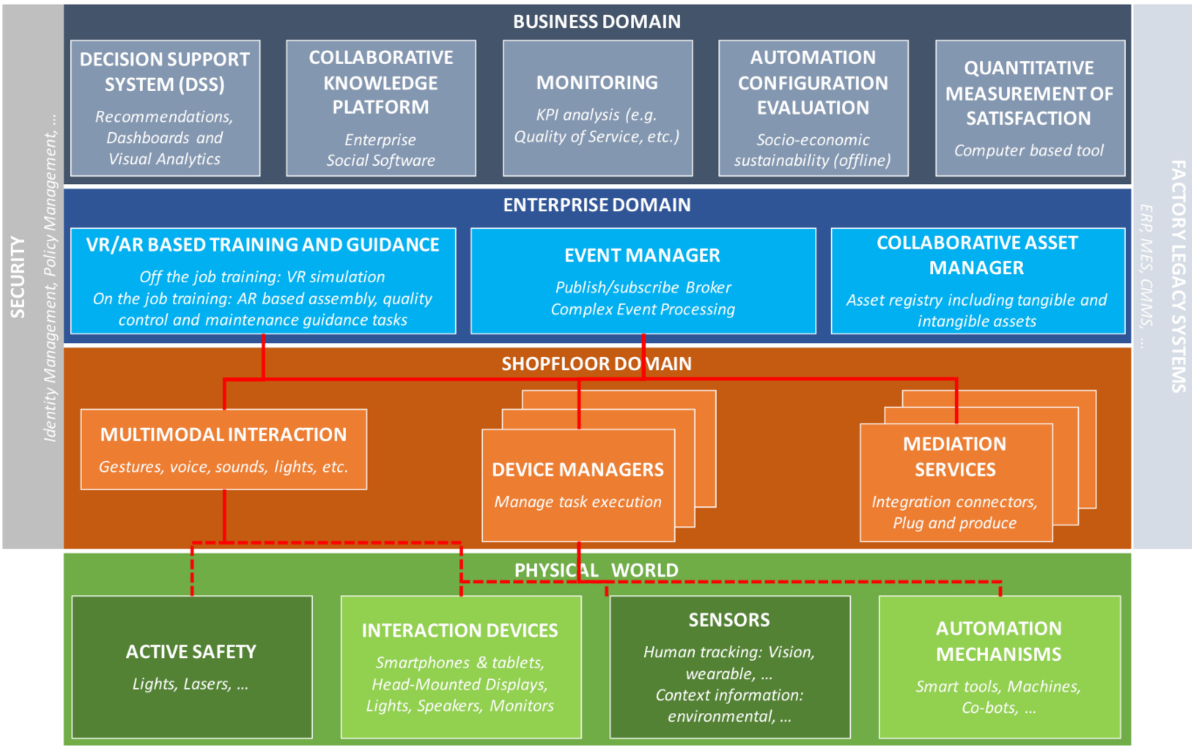


Figure 103 A4BLUE F&M Architecture

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 ad shopfloor level, but 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 the components in this domain aims 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, making use of all the information gathered around the adaptive system. Knowledge plays an important role in the life of organizations in current economy. They are 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 right persons in 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 alpha 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) before starting 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 manufacuting 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-ecomic 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, provide 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 D1.1. and D1.2 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.

Future revisions to the overall A4BLUE Platform design, if any, will be reported as part of D2.6 representing the final version of the Modular and Functional Architecture of the A4BLUE platform for adaptive assembly system.

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 D1.1)

Please find here below an extract of D1.1, 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**
1.1	Continuous data collection for analysis of system performance and optimisation needs.
1.2	On-the-job work instructions that guide the worker through assembly or support processes (i.e. inspection, routine maintenance) to reduce the need for organised off-the-job training and supervision.
1.3	The ability to easily reconfigure the workplace when introducing a new automated system or robotics (e.g. plug & produce capabilities).
1.4	Direct connection to internal control systems (e.g. Enterprise Resource Planning, Manufacturing execution systems, etc.) to adapt the assembly process.
1.5	Constant recording of tool usage data to a central system to improve maintenance activities.
1.6	Monitoring work station performance for future process improvement.
1.7	Functions that are able to change their behaviour autonomously to accommodate new products and production processes.
1.8	Continuous interaction all systems in the organisation for resource allocation.
1.9	Capabilities included in the automated system or robot that take advantage of the available workers expertise/ knowledge.
DESIRABLE	
1.10	The ability to evaluate optimal levels of automation for workers (i.e. from fully automated to fully manual through collaborative).
1.11	The ability to reconfigure themselves to increase efficiency and minimise effort and increase efficiency when changing production processes.
1.12	Self adjusting capabilities to cope with changing needs of workforces and different worker capabilities.
1.13	The ability to optimise by themselves to reduce the need for human intervention and adjustment.
1.14	Abilities for determining optimal levels of automation to meet economic requirements.
1.15	The ability to self-adjust to compensate for reduced technical capabilities (older computer programs).
1.16	Constant logging of production waste data for the purposes of future planning.
1.17	Constant recording of automation / robot usage data to a central system to manage maintenance activities.
1.18	The ability to self-adjust to compensate for lower training and experience levels.
1.19	Capabilities for evaluating workers' levels of satisfaction of and identify potential workplace improvements.
1.20	Direct connection to organisational systems for post-production product service and support.

Table 36 User Requirements: Organisational Level

2. Automation and Robotics	
ESSENTIAL	
Code	**Description**
2.1	Robots have safety capabilities that immediately stop the robot in the event of an accidental collision.
2.2	Safety mechanisms that make operators comfortable when collaborating with automation/robots during assembly.
2.3	Safety capabilities that adapt the speed of the robot according to the distance or speed of the operator.
2.4	Robots have safety capabilities that move the robot away from the worker in the event of an accidental collision.
2.5	Robots that work collaboratively and safely with an operator on shared tasks in fenceless environments.
2.6	Automation / robotics that are controllable by the operators working in the system.
2.7	Automation / robotics that can change safely by themselves to meet different physical capabilities of the involved operators, such as size differences.
2.8	Automation / robotics that can self-adapt its configuration to an operator's physical characteristics (i.e. height, arm length) to avoid potential ergonomic issues.
2.9	Safety capabilities that differentiate between people and other kinds of potential obstacles, and adapt the automation/robots behaviour to suit.
2.10	The ability to make operators aware of whether or not the safety mechanisms and devices are functioning effectively.
DESIRABLE	
2.11	Automated / robotic functions that will adapt to suit each operator's preferred working methods.
2.12	Automation / robotics that can change themselves safely to meet varying production demands.
2.13	Automation / robotics that can change safely on their own to meet different experience capabilities of the involved operators.
2.14	Automation / robotics that can change safely on their own to meet different environmental conditions like varying light and noise levels.
2.15	Functionalities to adapt the safety strategy based on the operators preferences and what is happening in the area surrounding the robot.
2.16	Automation / robots that can adapt its speed to correspond with an operator's profile (i.e. expertise, skills, capabilities, preferences, trust level).
2.17	Robots that notify management about the completion and the status of the task.
2.18	Robots should work safely alongside or near to an operator but on separate tasks.

Table 37 User Requirements: Automation and Robotics

3. Communication and Interaction Mechanisms	
ESSENTIAL	
Code	**Description**
3.1	The automation / robot / system has feedback abilities to show it has understood a command.
3.2	A workstation PC with an interactive computer system that allows the operator to interact and control the automation / robot / system.
3.3	Operators interacting non-verbally with automation / robot / system by using handheld controls, or an emergency stop button.
DESIRABLE	
3.4	Automation / robot / systems that operators interact with using natural speaking (i.e. non-predefined commands).
3.5	The automation / robot / system has both visual and auditory capabilities to present relevant feedback and notifications.
3.6	Automation / robot / systems that can be controlled with a computer system on a mobile device (e.g. tablet, smartphone).
3.7	The automation / robot / system has visual capabilities (e.g. computer systems, lights, projected messages, etc.) to display relevant feedback and notifications to operators.
3.8	Automation / robot / systems that operators interact with using pre-defined voice commands.
3.9	The automation / robot / system uses sound or voice message to provide feedback and notifications to workers.
3.10	Automation / robot / systems that operators interact with using gestures.
3.11	Automation / robot / systems that operators can choose based on their preferences or capabilities to interact verbally and/or non-verbally with the automation / robot / system.

Table 38 User Requirements: Communication and Interaction Mechanisms

4. System Feedback and Assistance	
ESSENTIAL	
Code	**Description**
4.1	Tools / equipment provided to operators at specific stages of assembly when they are needed.
4.2	All tools and equipment for assembly always available to operators.
4.3	Ergonomic assessment capabilities so that it can provide postural guidance to operators.
DESIRABLE	
4.4	Augmented reality devices (e.g. google glasses) to provide remote assistance from qualified personnel to operators.
4.5	An ability to detect when technical assistance is needed by an operator.
4.6	The ability to recognise an operators capability and provides personalised assistance.
4.7	System optimisation proposal taken from feedback.
4.8	The ability to provide personalised assistance to meet the individual needs of an operator.
4.9	Knowledge capture / capitalization systems for process improvement.
4.10	Functionality for providing the assistance that operators can request and/or select.
4.11	Automated systems that suggest how to manage emergency and/or unexpected situations.
4.12	Automatic and continuous analysis of work as it is completed and feedback mechanisms.
4.13	An off-the-job system that uses virtual reality simulation to train operators to do tasks by reproducing it in a virtual world.
4.14	Assistance and feedback that is designed to keep operators satisfied as they work.
4.15	System feedback that keeps the operator aware of their own work progress.
4.16	Process analysis and feedback that can be accessed when requested by operators, such as productivity and performance information.
4.17	Ergonomic assessment of physical capabilities of the operator to provide assistance.

Table 39 User Requirements: System Feedback and Assistance

5. System Information and Instructions	
DESIRABLE	
Code	**Description**
5.1	Mechanisms for operators to directly input their own recommendations for work instructions, information updates or working conditions.
5.2	Mechanisms for operators to directly input multimedia content (i.e. including photos, videos, and voice) into the process information and instructions.
5.3	Functions that track operators' activity and/or work performance and inform them of recommendations and remedial actions.
5.4	The ability to verify each step of the proposed procedure and display the information related to the next step.
5.5	Capabilities that allow operators to exchange best practices/ problem solving solutions with other operators in the process instructions.
5.6	Information presented on demand using a wireless augmented reality device.
5.7	Augmented reality that provides information and instructions to operators while they are working.
5.8	Virtual reality that provides off–the-job training to operators.
5.9	Capabilities to allow operators to interrogate information / instructions further.
5.10	The capability to display work procedures that show how to do tasks using multimedia capabilities (text, pictures, images, videos).

Table 40 User Requirements: System Information and Instruction

6. System Security and Data Management	
ESSENTIAL	
Code	**Description**
6.1	Comprise IT security mechanisms that will prevent attacks from external sources
6.2	Allow information technology personnel and managers to have access to system data (e.g. data on process, data on the systems performance)
DESIRABLE	
6.3	Destroy an operator's data 5 years after they have left their company of employment
6.4	Only capture specific data about the operator (e.g. the height they set the workbench to)

Table 41 User Requirements: System Security and Data Management

ANNEX B APPLICATION SCENARIOS AND CHALLENGES (FROM D1.2)

Please find here below an extract of D1.2, listing the application scenarios and their challenges as identified in D1.2, 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 42 Application scenarios (from D1.2)

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- Adjusting wing to correct gap size	Process variability/ 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 43 Challenges identified in use case scenarios (from D1.2)