Innovation Action Project
HORIZON 2020 - EU.2.1.5. - Ref. 680633

D2.1 -D 2.1a BEinCPPS Architecture and Business Processes

Lead Editor: Klaus Fischer

With contributions from: Antonio Jara, Arjan Geven, Domenico Rotondi, Eva Coscia, Klaus Fischer, Lăcrămioara Aştefănoaei, Martijn Rooker, Mauro Isaja, Oliver Botti and Sebastian Voss

Reviewer: [Sergio Gusmeroli]

Deliverable nature	R
Dissemination level	PU
Contractual delivery date	31 January 2016
Actual delivery date	18 April 2016
Version	final
Keywords	Architecture, IoT, OSMOSE, RAMI, FITMAN, BEinCPPS Reference Architecture, BEinCPPS-Platform modular architecture

Version History

Nr.	Date	Notes and Comments
0.1	07/01/2016	Initial draft
0.2	21/01(2016	Input DFKI, ENG, FINCONS, TTTech
0.3	25/01/2016	Input on BEinCPPS architecture proposal and formatting
0.4	26/01/2016	Complete draft and further editing
0.5	27/01/2016	Further editing by lead authors
0.6	29/01/2016	Input by FINCONS and fortiss
0.7	05/02/2016	Input by ITI
1.0	28/01/2016	Adoption of new template
1.1	12/02/2016	Updated version by ENG
1.2	13/02/2016	Updated version by HLX
1.4	13/02/2016	Update by DFKI, FINCONS, HLX, and ITI
1.5	18/02/2016	Improved layout by DFKI
1.6	18/02/2016	Updated version by ENG and HLX
1.7	26/02/2016	Updated version by ENG
1.8	28/02/2016	Preparation of final version by DFKI
2.0	09/04/2016	Revision according to reviewers comments

Deliverable Peer Review Summary

Id.	Addressed	Comments

List of Figures

Table of Contents

Executive Summary

BEinCPPS Innovation Action aims to integrate and experiment a CPS-oriented Future Internet-based machine-factory-cloud service platform, at first intensively in five selected Champions from the Smart Specialization Strategy Vanguard regions (Lombardia in Italy, Euskadi in Spain, Baden Württemberg in Germany, Norte in Portugal, Rhone Alpes in France), afterwards extensively in all European regions, by involving local competence centres and manufacturing SMEs. The final aim of this Innovation Action is to dramatically improve the adoption of CPPSs (CPSs for production, or Cyber Physical Production Systems) all over Europe by means of the creation, nurturing and flourishing of CPPS-driven regional innovation ecosystems, made of competence centres, manufacturing enterprises and IT SMEs.

The definition and discussion of the architectural approach for the BEinCPPS platform is an important step for the project. This document, released as deliverable D2.1, summarizes the work on the architecture definition: it is the first version of "BEinCPPS Architecture and Business Processes" and provides the basis for the implementation of federations of service architectures.

In a state of the art overview, the deliverable presents and discusses three major architecture proposals for IoT and CPS-based service provision in manufacturing: the OSMOSE Architecture, the FITMAN Industrial IoT Reference Architecture (IIOT-RA), and the Reference Architecture Model for Industry (RAMI) 4.0. OSMOSE introduces the three worlds: Real, Digital, and Virtual, and the concept of a semantically-enabled semi-permeable membrane which connects these worlds. FITMAN's IIOT-RA introduces three functional domains: Smart, Digital and Virtual Factory. RAMI defines three dimensions of enterprise system design and introduces the concept of Industrie 4.0 components. In addition to the conceptual architectures, state of the art technologies and tools for the design and implementation of systems according to this architecture proposal are presented.

The state of the art section, therefore, tries to provide a wider view as compared to usual approaches not only analysing and discussing architectural issues, but also analysing the ones related to the design, development, deployment and operation of CPSs in industrial contexts. Indeed, the new architectural approaches, like the RAMI 4.0, highlight the relevance of the development and ramp-up activities in the overall manufacturing value chain.

In the light of the state of the art assets and of the new concepts introduced by BEinCPPS, the BEinCPPS Reference Architecture (BEinCPPS-Arch) is proposed. BEinCPPS-Arch federates the most prominent Smart Systems, IoT and Future Internet platforms. It defines two distinct, orthogonal axes in its layout: the *Worlds* axis for the Real, Digital and Virtual logical domains; the *Levels* axis for the Field, Factory and Cloud physical environments. Each *World* can span the three Levels, while a dedicated module is in charge of realizing a Real-Digital-Virtual connection that is not merely a data exchange between distinct entities (as happens in traditional approaches), but rather it is implemented as an osmosis process in which atomic elements of each *World* can be transferred through semantically-enabled semi-permeable membranes into adjacent *Worlds*, where they operate as remote agents. Additionally, it is discussed how concepts of the related architectures presented in the state of the art overview can be adopted.

The deliverable introduces the BEinCPPS-Platform modular architecture as an instantiation of the BEinCPPS-Arch reference architecture. This modular platform is a composition of the most relevant state of the art software assets into a *federation of platforms*, according to the project's original concept and covering both the design and development phases, as well as the deployment and operational ones. Therefore, the BEinCPPS architecture melds and tries to harmonize assets covering the whole value chain related to the set-up and operation of manufacturing.

At the highest level, the BEinCPPS-Platform is divided into a *runtime* and a *design-time* sub-system – the former consisting of components used to integrate CPPSs and applications, the latter being about CPP-based system engineering environments and tools. Within the

runtime sub-system, components are further classified by scope: from bottom to top, these are Shopfloor (Fast Lane, Smart Lane, and Device Management), Interoperability Information Bus (FIWARE Orion Context Broker and OpenIoT Middleware), and Cloud Level (Big Data & Event Storage, Processing, Monitoring and Collaboration). Two lanes are defined to access the CPPS devices at the Shopfloor layer. The Smart Lane exploits OpenIoT[1] assets, applying semantic annotation on data streams, using a domain-specific Reference Ontology. The Fast Lane is based on FIWARE technology, as in the original IIOT-RA: it trades the ease of integration and the advanced capabilities of the Smart Lane for a simpler and lighter middleware, that requires a tighter coupling of applications with their target environment but delivers a synchronous, low-overhead communication channel. Both Lanes are equally supported by a Device Management layer based on the HOMARD platform.

The design-time sub-system provides tools supporting CPS-system design at different levels of abstraction, adopting the BSM/TIM/TDM approach from the MSEE project.

At BSM (Business Services models) level are tools for the design of the business processes to be supported by the CPPS-based solutions; at TIM (Technology Independent Models) level are those tools that adopt well known standards (UML, SysML, BPMN, etc.) to represent system properties and behaviours and to design and run (co-) simulation. At TSM (Technology Specific Models) are those tools that are used to model technology-specific components, such as PLCs.

The BEinCPPS-Platform modular architecture provides the blueprint for the integration tasks. The initial design presented in this deliverable is going to be refined in a second release of this deliverable, following the first deployment of the BEinCPPS-Platform to the five Champions' sites and taking into account lessons learned during the first run of experimentations. In particular, the final selection of state of the art assets that will compose the final platform federation is going to come from hands-on experience, and may differ from what is currently presented here.

[1] Linked Sensor Middleware platform - http://open-platforms.eu/library/deri-lsm/

1. Introduction

1.1. Objective of the Deliverable

D2.1 is the starting point for the design of the federated platform architecture of BEinCPPS. It describes several architectures such as OSMOSE, the FITMAN Industrial IoT Reference Architecture and the Reference Architecture Model for Industry (RAMI) 4.0. Furthermore, a link to the CPS and CPPS layer is discussed.

The BEinCPPS action is currently in its initial phase and the BEinCPPS champions' scenarios are still under investigation to be able to design an effective architecture. This deliverable, therefore, focuses on analysing approaches and discussing ideas for a future BEinCPPS architecture.

1.2. Structure of the Deliverable

Following this introduction in Section 1, Section 2 presents the state of the art proposals of the three architectures OSMOSE, FITMAN and Industrial Reference Architecture Model for Industry (RAMI) 4.0. Additionally, state of the art concepts for the CPS and CPPS layer is discussed. Section 3 presents the conceptual ideas of the BEinCPPS architectures as they are developed at this stage of the project and discusses links to the state of the art architectures. The main contributions of the deliverable the definition of the BEinCPPS Reference Architecture (BEinCPPS-Arch) and the BEinCPPS-Platform modular architecture are presented in Section 4 and 5, respecitively. Section 6 at this stage briefly describes the situation regarding business processes and Section 7 concludes the deliverable.

1.3. Applicable Documents

[DoA]: Description of Actions for the BeInCPPS project, providing the basis for the entire project and this deliverable content

2. BEinCPPS Motivation

Cyber-Physical Systems (CPS) are the next generation of engineered systems in which computing, communication, and control technologies are tightly integrated [15]. Indeed, the US NIST CPS framework document [13] defines CPSs as elements that *"... integrate computation, communication, sensing, and actuation with physical systems to fulfill time-sensitive functions with varying degrees of interaction with the environment, including human interaction".*

The "cyber" part of CPS is used to mainly focus on computing as represented, for instance, by embedded computers. However, with CPS covering an increasingly larger spatial area and unprecedented coordination among components within CPS, communication has become indispensable for CPS. The cyber characteristics of CPS are thus new from both its computing and communication subsystems. The "physical" part of CPS refers to physical processes through which CPS interacts with its surroundings. As expressed by Alf Isaksson, ABB keynote speech, at the EU CPS Conference [8] 30th October 2013, manufacturing industry is looking at Cyber Physical Production Systems (CPPS) as the next industrial revolution. Cyber-Physical Production System (CPPS) vision is related to Systems of Physical Objects and corresponding Virtual (Digital) Objects that communicate via omnipresent information networks. However, the development of competitive CPPS manufacturing processes for future factories supported by fast adoption of innovative CPPS solutions (adaptive, able to support the product-service duality) is not immediate. While large industries possess human, financial and business resources to engage themselves in both

CPPS process and product development programmes, SMEs lack resources, technical knowledge as well as business development skills to do so successfully and timely. Without the required experimentation at multiple levels of CPPS technologies (business-technical), CPPS development and adoption by SMEs will be seriously compromised and the competitiveness of European SMEs will be undermined in the CPS global race.

To strengthen demand for ICT-based advanced manufacturing technologies, the European Commission piloted 'ICT Innovation for Manufacturing SMEs' (I4MS)[2] an instrument to stimulate the take-up of advanced technologies by manufacturing SMEs. EU grants support the testing in real production conditions of existing advanced manufacturing technologies (e.g. robot solutions, high-performance cloud-based engineering simulation, intelligent sensor- and actuator-based equipment and innovative laser applications) to promote their up-take in manufacturing industry. As stated by Dr Khalil Rouhana (Director of Components and Systems, DG CNECT) and Dr Max Lemke (Head of Unit for "Complex Systems and Advanced Computing) at the very recent *Cyber-Physical Systems: Uplifting Europe's Innovation Capacity* workshop, "*CPS vision is not science fiction anymore; it is on its way to becoming a reality. What is at stake for Europe's actors is immense as the value of the CPS market is estimated at more than 850 B€ and currently Europe has a 30% share of the World Embedded Systems Market*". Being SMEs and Mid-Caps key to European competitiveness and growth, I4MS found strategic to extend with a specific initiative the current areas addressed by the I4MS initiative to avoid letting European SMEs and Midcaps lag behind in the CPS race (as solution-application development and CPS solution early adopters) [9]. It is crucial to empower and optimize the manufacturing in the local regions and inside pan-Europe, instead of outsourcing it to emerging countries in order to be competitive. Therefore, reducing the entry barriers, and optimizing the production costs thanks to the integration of CPPS will move forward to the competitiveness of the European market in general, and the European market of SMEs with production facilities and needs in particular.

In the I4MS context, there is a need to develop the right environment with the required support actions for technical and business development services that will respond to all SME demands to test and experiment CPPS technology on a larger scale, by leveraging on Large Enterprises pilots and simultaneously involving a large set of SMEs both as new applications developers and as new CPPS equipment testers.

BEinCPPS' vision is that the full adoption by EU SMEs of CPPS systems and their related service platforms and innovation business models will allow Europe to achieve the ambitious target by 2020 to have 20% of the GDP coming from Manufacturing and related services.

To approach this vision, the BEinCPPS project aims to integrate and experiment with a FI-based machine-factory-cloud service platform firstly intensively in five selected S3 Vanguard regions, afterwards extensively in all European regions, by involving local competence centers and manufacturing SMEs. The final aim of this Innovation Action is to dramatically improve the adoption of CPPSs all over Europe by means of the creation, nurturing and flourishing of CPS-driven regional innovation ecosystems, made of competence centers, manufacturing enterprises and IT SMEs.

The BEinCPPS project stems upon three distinct pillars:

- The first pillar is a **three-layered (machine-factory-cloud) architecture**, to be referred to as BEinCPPS-Arch in what follows. BEinCPPS-Arch will be implemented as a platform (BEinCPPS-Platform) derived from state-of-the-art R&I advances in the fields of Internet of Things, Future Internet and CPS / Smart Systems. The BEinCPPS

---

2 I4MS (http://i4ms.eu/) provides € 77 million funding over the period 2013-2016. About 200 SMEs are expected to take part in I4MS with more than 150 innovation experiments over the next 3 years.

Platform will be able to support the design and execution of CPPS-enabled business processes and to bi-directionally interoperate data pertaining to the machine, the factory and the cloud levels, to support via open APIs the development of additional innovative applications. Building blocks of the run-time architecture will be FIWARE technologies/Generic Enablers as well as Specific Enablers coming from the FITMAN trial project, whereas the design-time support will be provided by combining existing tools and platforms from the MSEE project and from CPSe-Labs catalogue. The BEinCPPS-Platform will be interconnected with SmartSystems technologies mainly coming from the projects ARTEMIS. BEinCPPS-Platform will also take into account standards and solutions in strictly related fields like IoT (looking at approaches and solutions developed within the IERC-IoT European Research Cluster, AIOTI-Alliance for Internet of Things Innovation), and advanced manufacturing (and specifically the German Industrie 4.0 initiative). Finally, BEinCPPS-Arch will be synchronized with the international actions from Industrial Internet Consortium, IPSO Alliance, IEEE P2413, OMA, oneM2M (including ETSI, 3GPP, and worldwide SDOs), in order to be compliant with international initiatives, while at the same time BEinCPPS-Platform will leverage all the added values and advantages that European market has in these technologies with respect to emergent international actions.

- The second pillar is a pan-European SME-oriented experimentation ecosystem. This ecosystem starts from five Champions of the Vanguard regions with competence centers linked to large manufacturers (champions) and will be progressively expanded to a community of application developers and additional local SME-driven experimentations. In a first phase of the project, the Champions will provide requirements to the platform integrators, including the required support of existing de-facto standards at the three levels of machine-factory-cloud. The second phase consists of application experiments: an Open Call for IT SMEs developers will award 10 third parties which will extend the BEinCPPS platform and test it in its five champion instantiations. In a final third phase, the extended platform will be instantiated and deployed in additional 10 third parties manufacturing SMEs, which will replicate tests and experimentations in various locations (e.g. Eastern Europe), and sectors, application domains different from those specific to the Champions. For this new set of experiments, a generic and manufacturer-independent CPPS demonstration and experimentation platform[3] will be also made available.
- The third pillar is a well-founded method and toolbox for Innovation management. The scope is to enrich an existing TRL-based methodology for KETs technology transfer by (1) a CPPS certification, education and training programme for young talents and experienced blue collar workers and (2) a well-founded three-fold (objectives-variables-indicators) method for results assessment and evaluation. The new blended SME-oriented methodology will create an evidence set of showcases for BEinCPPS experimentation credibility, in order to facilitate pan-European replicas and expansions. The final aim of this phase is to nurture and stimulate the creation of regional CPPS-oriented ecosystems of competence centers, application developers and equipment manufacturers SMEs, able to attract additional complementary funds from local authorities and this way playing a multiplying role in the industrial adoption by SMEs of CPSs and their related service platforms and business models.

In the BEinCPPS project, workpackage WP2 (Federated Platform Architecture, Integration and Testing) is meant to integrate, “open” and customize the components of the three-layered IT infrastructure of the BEinCPPS pyramid (Machine Factory Cloud) for the CPPS case. The positioning resembles that of the RW-DW-VW architecture proposed by the OSMOSE project. The three different sources of such components will be investigated and

[3] SmartFactoryKL in Kaiserslautern (Germany) kindly provided inside the Industrie 4.0 programme by DFKI, http://www.smartfactory-kl.de/

analysed: IoT components coming from the projects belonging to the IERC cluster, FI components coming from FIWARE programme and FITMAN projects in particular, Smart systems components derived from Ecsel / ARTEMIS and other initiatives in the field of embedded systems, systems of systems and CPSs. Such components will be integrated into three reference architectures (for Machine Factory and Cloud levels), opened by developing proper APIs (open standards) and easy-to-access HCI primitives and finally customized for the CPPS case, ready to be instantiated and deployed in the five regional champions. A desired consequence of the setup in WP2 is a continuous evolutionary process driven by the CPPS ecosystems created in the five regions and in all Europe which would constantly add new components, new APIs and new compliance with standards, so that at the end of the project and of the 10+10 additional application / equipment assessment experimentations, the resulting platform will be ready for commercialisation and boost CPPS innovation in many industrial sectors and application domains.

3. State of the Art

In this section we present three architecture proposals which comprise the state of the art for the regarding the objectives of the BEinCPPS project with respect to the architecture definition. Additionally, we present state of the art approaches for the design and the integration of the CPS layer.

3.1. OSMOSE

According to the FInES Research Roadmap 2025 [18], Sensing Enterprise (SE) and Liquid Enterprise are two Qualities of Being which are considered strategic for any future enterprise. The Sensing Enterprise will emerge with the evolution of the Internet of Things (IoT), when objects, equipment, and technological infrastructures will exhibit advanced networking and processing capabilities, actively cooperating to form a sort of "nervous system" within the enterprise next generation [19].

The concept represents a fundamental change in the business models and information systems that is not immediate, and should be supported by methods and tools capable of supporting the evolution of the traditional organizations towards the tremendous possibilities offered by the IoT-enabled worlds. Indeed, latest research in the area is making quite clear that the take-up of the Sensing Enterprise concept will enable very advanced and promising new business models and applications thanks to the adoption of FI technologies. Research initiatives, such as the OSMOSE or Proasense European projects (www.osmose-project.eu, www.proasense.eu), as well as dedicated scientific sessions and working groups such as the one held in the last IFAC World congress[4] promoted a wide debate on the need for a convincing unifying holistic model for a Digital Sensing Enterprise as well as a common reference architecture for next generation Enterprise Applications based on the IoT and other FI technologies and generic enablers.

The Liquid Enterprise can be considered as an enterprise having fuzzy boundaries, in terms of human resources, markets, products and processes. Its strategies and operational models will make it difficult to distinguish the 'inside' and the 'outside' of the company [18]. This concept can be better explained if a metaphor from physics is adopted. Let us imagine that the Liquid-Sensing Enterprise is, in fact, a pot internally subdivided into three sectors by means of three membranes and forming the Real-Digital-Virtual sector [6, 5]. As Figure 1

[4] http://tc.ifac-control.org/5/3/events/incom2015/enterprise-reference-maturity-and-assessment-models-for-the-future-internet-based-enterprise-erma4fie-codes-7a932-or-83d3x/at_download/file

sketches, a blue liquid is poured into the bottom section (Real World population), a red liquid into the middle section (Digital World population) and a green liquid into the third sector (Virtual World population).

If the membranes are totally impermeable (left part of Figure 1), the three liquids (worlds) will never mix together and, if they want to communicate, they need to send blind messages across the membranes. This meets the classical definition of interoperability, which is defined as the ability of disparate and diverse organizations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between them [1]. However, in those interoperability scenarios, the two or more actors are totally independent entities (e.g. two or more enterprises, organizations, or even people and objects).

On the other hand, if the membranes are semi-permeable, by following the rules of osmosis, each of the world's population could pass through the membrane and influence the neighbouring world, so that in reality in the blue Real World is possible to find a red-green shadow ambassadors of the Digital/Virtual World and similarly for the other Worlds (see right part of Figure 1).

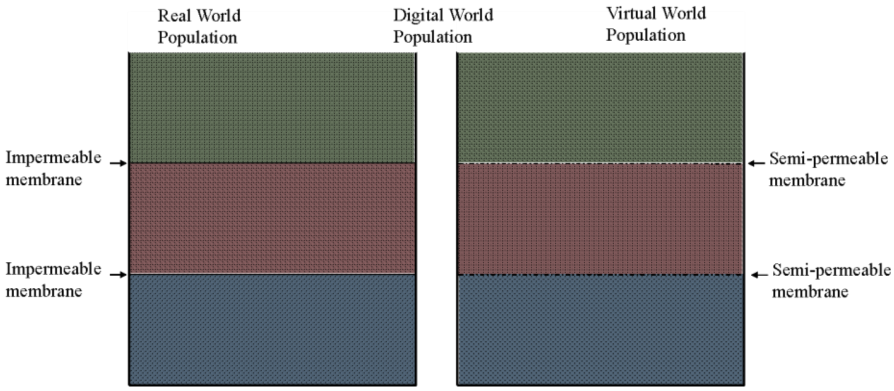


Figure 1: Liquid-Sensing Enterprise Physics Metaphor

Figure 2 presents the overall OSMOSE architecture as a starting point for the implementation of metaphor of Figure 1 in a software system. The Real World (R – blue circle), Virtual World (V – red circle) and Digital World (D – green circle) are separated from each other in order to enable security and privacy. Communication between the three worlds has to cross the membranes of the world, which are incorporated into the OSMOSE Middleware (light-blue triangle).

The OSMOSE worlds are characterized as follows:

Real World: The physical worlds as we use to know it but extended with IoT devices which allow them to be integrated in the architecture.

Digital World: Any digital information which is available with respect to the real world. Business processes are executed in the digital world to manipulate the state of the real world towards the business goals of an enterprise.

Virtual World: Hypothetical worlds to run what-if scenarios to produce further insight about real or digital world processes.

The OSMOSE Middleware is responsible for intelligent communication delegation which is supported by osmosis and context management (purple/grey and orange trapezia). Additionally, a Data Access Gateway (brown triangle) is provided allowing to access data and models of the three worlds seamlessly.

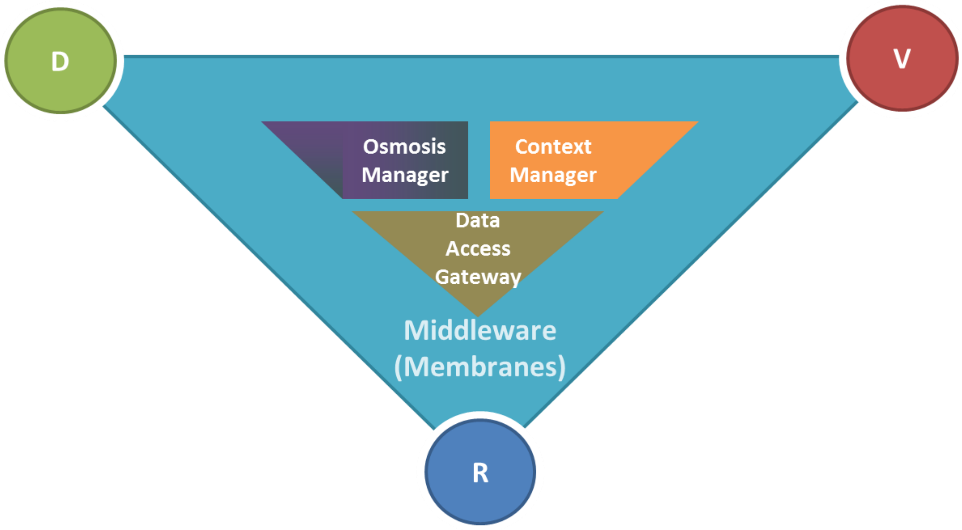


Figure 2: Overall OSMOSE Architecture

The architecture combines the Service-Oriented Architecture (SOA) [10] and Event-Driven Architecture (EDA) [17] paradigms using the Enterprise Service Bus (ESB) architecture model[5] inside the OSMOSE Worlds and the OSMOSE Middleware. E. Chapell defines the ESB in the following way: "An Enterprise Service Bus is a standard-based integration platform that combines messaging, Web services, data transformation and intelligent routing in a highly distributed, event-driven Service Oriented Architecture" [7], p. 1]. Thus, the architecture is furthermore highly distributable and the OSMOSE Worlds as well as the OSMOSE Middleware can be distributed over many locations. For passing events among worlds, asynchronous communication patterns like publish/subscribe are used. With this it is possible to establish links between entities in the different worlds which provides the means to synchronize them across the borders of the worlds. Additional logic can be integrated in the messaging layer which decides on whether a published event will stay within the world or will cross the world boundary which implements the metaphor of the OSMOSE membrane. The worlds can provide services to the ESB which might then be used by the other worlds to access information which they need in the processes which react to the events which are passed to them.

By utilizing complex-event-processing (CEP) [16] and Semantic Web technologies, intelligent and controlled communication is enabled. Knowledge bases structured with ontologies inside the Real, Digital and Virtual World as well as the OSMOSE Middleware allow semantic reasoning and knowledge depositing for events, entities, services and processes. Figure 3 illustrates the knowledge base architecture. The common knowledge base that is located in the OSMOSE Middleware structures the common knowledge about events, entities, services and processes, whereas the knowledge base extensions that are located in the three worlds are built on the common knowledge base.

This modular ontology approach [14] that substitutes a complex and huge common ontology into domain-specific, i.e. world specific, ontologies reduces complexity in ontology evolution, maintenance and reusability. Additionally, reasoning performance inside the worlds or inside the middleware for classification of particular classification problems, e.g. for event delegation inside one of the worlds, is significantly improved. Another advantage of the modularization approach is the good applicability for distributed systems. The ontology

[5] http://www.ibm.com/developerworks/library/ws-soa-eda-esb/

modules can be distributed over different locations just as the components of the architecture. With this the knowledge base extensions inherit the concepts of the common knowledge base (not vice versa) and may be used as if they were defined in the knowledge base extension itself. Changes made to the common knowledge base are transparent to the extensions that inherit the concepts.

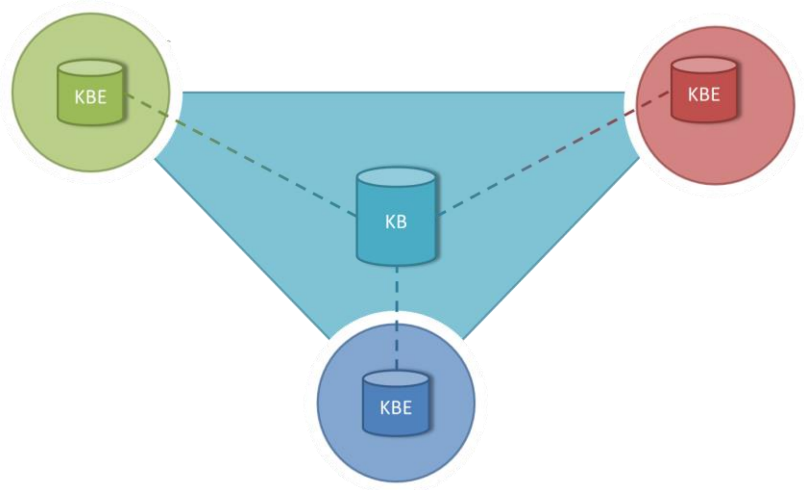


Figure 3: OSMOSE Knowledge Base Architecture

While the concepts of real and digital world in OSMOSE are quite clear, OSMOSE's view to the virtual world is limited. In work outside of OSMOSE one can find the following different interpretations of the concept of a virtual world:

Virtual Reality: Aims at creating virtual environments which are as close as possible to the real world. This interpretation is in line with the OSMOSE point of view because virtual realities are still hypothetical worlds even if they are very close to reality.

Augmented Reality: Real world entities are extended by virtual artefacts coming from virtual worlds. This is also included in the OSMOSE picture because OSMOSE foresees the process of augmentation.

Social Networks are a virtual environment in which human users can present themselves and contact other users.

Virtual enterprises: Loosely coupled sets of enterprises doing business together. They might very well take advantage of social networks to find promising business partners.

Clouds: There are two aspects in cloudification. The first includes virtualization of whole operating systems. The machine on which the application is running is no longer a physical machine but a virtual machine. In the second case the main idea is to host applications or services in the network. With Facebook, Youtube, Google Docs, etc. there is a large number of such services and applications available and in everyday use. However, shop floors in manufacturing today use dedicated hardware and control systems which are installed on site.

3.2. FITMAN

FIWARE[6] is an open initiative in the scope of the Future Internet PPP (FI PPP) program[7], 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.

6 https://www.fiware.org/

7 https://www.fi-ppp.eu/

In the same FI PPP scope, **FITMAN**[8] (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[9] 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 4 below, borrowed from the public FW4I site[10], illustrates IIOT-RA's components and their mutual relationships.

[8] http://www.fitman-fi.eu/

[9] http://www.effra.eu/

[10] http://www.fiwareforindustry.eu/

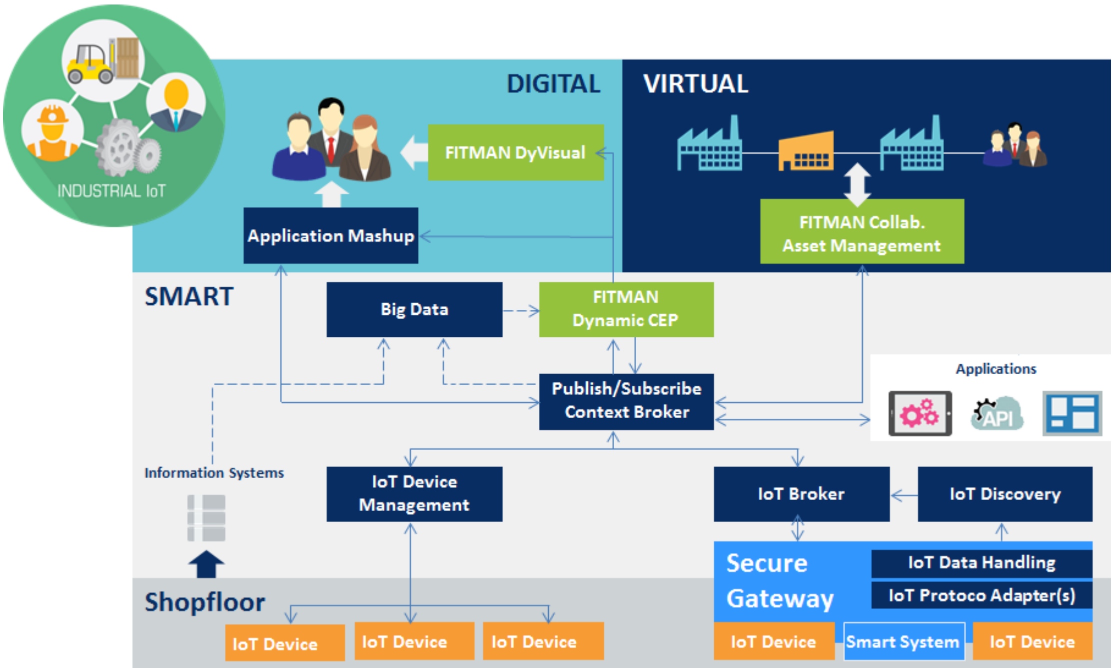


Figure 4: FIWARE for Industry IIOT Reference Architecture

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[11] (IoT DM) GE from FIWARE. This is a lightweight middleware that adapts common IoT protocols like OASIS MQTT[12] and OMA LWM2M[13] to the *FIWARE Open API for IoT cloud enablement* – i.e., OMA NGSI[14] with a REST-over-HTTP [12], Ch. 5] binding. The FIWARE Generic Enabler Catalogue offers an open source implementation of IoT DM which is based on the concept of *IoT Agents*: small protocol-specific modules (typically developed using the C++ language to maximize runtime performance) that do a straightforward *protocol adaptation* job without interfering with the data payload in any way. This very modular software architecture allows for an easy integration of new protocols on need. The Fast & Wide Lane of IIOT-RA is best suited for very large (thousands of devices) automation and monitoring scenarios with very tight time constraints (near-real-time) but 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,

[11] http://catalogue.fiware.org/enablers/backend-device-management-idas

[12] http://mqtt.org/

[13] http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry

[14] http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/ngsi-v1-0

and in Cloud territory, a FIWARE IoT Broker[15] 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[16] (PSCB) module. 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[17] and the FITMAN Dynamic CEP[18] (DyCEP) SE. These work 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. Four different components provide web-based interfaces for users to *interact* with the Shopfloor. The FITMAN **DyVisual**[19] SE is for dynamic rendering of 3D content described using the XML3D[20] 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**[21] GE allows user-specific *cockpits* to be built by assembling *widgets* on a web canvas. Widgets are modular UI components, selected from a library or developed for ad-hoc purposes, that leverage a common framework

[15] http://catalogue.fiware.org/enablers/iot-broker

[16] http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[17] http://catalogue.fiware.org/enablers/bigdata-analysis-cosmos

[18] http://www.fiwareforindustry.eu/ : see Lab/FIWARE Enablers for Smart Factories/DyCEP

[19] http://www.fiwareforindustry.eu/ : see Lab/FIWARE Enablers for Smart Factories/DyVisual

[20] http://xml3d.org/

21 http://catalogue.fiware.org/enablers/application-mashup-wirecloud

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[22] (**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.

Beyond what included in the IIOT-RA and in its Specific Enabler Catalogue, the FITMAN project delivered two additional components that, while not released under an open source license, provide key functionalities in a IoT / CPPS industrial environment.

The **iLike** component collects and organises data about the in-factory lifecycle of products that are gathered from the CPPS. These data are visualised through advanced interfaces and ad-hoc services for advanced analytics and monitoring of production.

The iLike component has been extended after the FITMAN project, with a mobile app (called the **iLike Machine app**) that offers an intuitive visualisation of the configuration and functioning of manufacturing machine, collecting and analysing data from the PLCs.

The **Virtual Obeya** (vObeya) is a time-less and space-less virtual collaboration environment that offers virtual rooms (obeyas) to teams of people that can visualise and interact with visual tools to create what-if scenarios and engineer and design together new products and systems

3.3. RAMI 4.0

The *RAMI4.0* (Reference Architecture Model for Industry 4.0)[23] specification was 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 *RAMI4.0* 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.

The *RAMI4.0* **3D model** (see Figure 5) summarizes its objectives and different perspectives and provides relations between individual components. The model adopts the basic ideas of the *Smart Grid Architecture Model*[24] (SGAM) which was defined by the European *Smart Grid Coordination Group* (SG-CG) and is worldwide accepted. The SGAM model was adapted and modified according to the Industrie 4.0 requirements.

---

22 http://www.fiwareforindustry.eu/ : see Lab/FIWARE Enablers for Virtual Factories/CAM

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

24 ftp://ftp.cencenelec.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/Reference_Architecture_final.pdf

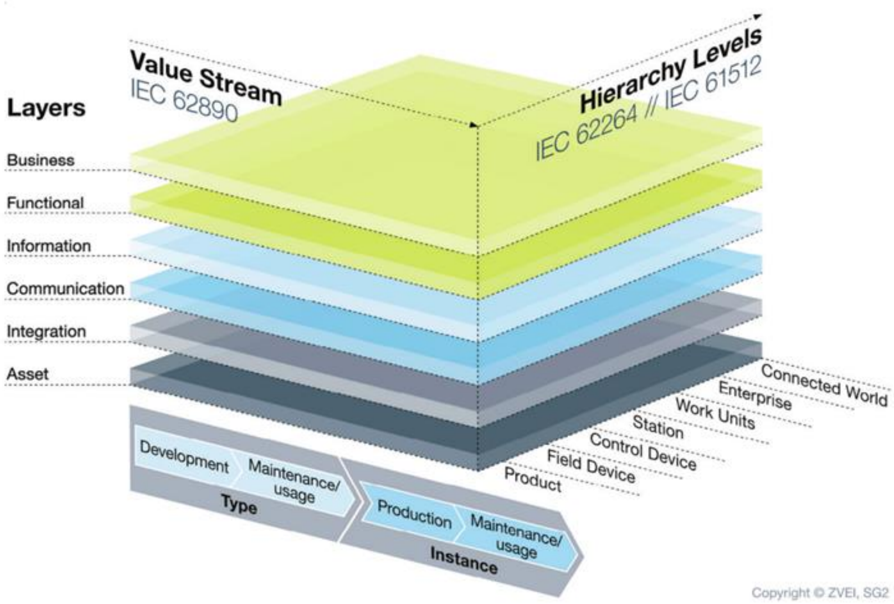


Figure 5: RAMI4.0 3D Model

The *RAMI4.0 3D Model* aims at supporting a common view among different industrial branches like automation, engineering and process engineering. The *3D Model* combines:

- **Hierarchy Levels** (*Y Axis*): this axis collects the hierarchy levels envisaged by the IEC 62264 international standards on the integration of company computing and control systems;
- **Cycle & Value Stream** (*X Axis*): the second axis represents the life cycle of facilities and products. The *RAMI4.0* takes the IEC 62890 standard for life cycle management as a reference point to structure the life cycle. This axis focuses on features able to provide a consistent data model during the whole life cycle of an entity;
- **Layers** (*Z Axis*): the vertical axis, finally, represents the various perspectives from the assets up to the business processes.

The combination of the elements on these three axes is quite innovative, especially the elements on the *X Axis*. Indeed, the *RAMI4.0* is the only reference architecture to explicitly analyse and take into account entities' life cycles as it will be further analysed in the following.

While the *Y Axis* reports the traditional manufacturing levels, the other two axes includes innovative aspects that it is worthwhile to highlight:

- on the *Z Axis*:
    - the *Asset Layer* takes care of the functionalities related to the management of *entities*; in the *RAMI 4.0* specifications an *entity* can be a physical component, a document, or even humans,
    - the *Integration Layer* takes responsibility of interfacing the "real world" with its "IT representation" as well as of HMI features,
    - the *Communication Layer* takes care of standard communication features based on a uniform data format,
    - the *Information Layer* focuses on providing features for events' (pre-) processing (including events' processing rules formalization and storage), data quality and integrity, data persistence,
    - the *Functional Layer* provides basic services for the business processes as well as support rules and decision making logic,

    - the *Business Layer* implements the business processes orchestrating the services and resources provided by the *Functional Layer*;
- on the *X Axis*:
    - the *Type* represents the conceptual aspects of a product or of a service (e.g., a specific model of a car, smartphone, etc.), where, as already stated, a product can be goods, machineries, factories or any other things that can be produced;
    - the *Instance*, instead, is an instantiation of a *Type* and, therefore, represents a specific good, machine, factory, etc.;
    - both *Types* and *Instances* have their lifecycle and IDs. An ID is a set of one or more attributes (e.g., a S/N) that can univocally identify a specific *Type* or a specific *Instance*.

Usually, a *Type* is created from some initial ideas or hints (e.g., market analysis), and then it is designed and validated. Once the validation activity has been completed, the *Type* is released for series production (therefore starting generating *Instances*) and its *maintenance* activity starts (e.g., evolution of a car model).

One of the main objective of *RAMI4.0* 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.

As stated, *objects* in the *RAMI4.0* can potentially be any element (e.g., models, types, instances, production lines, factories, etc.) that has a *life cycle* and, therefore, has data associated with it. Elements "active" within the *RAMI4.0* layers are called *Industrie 4.0 component* (*I4.0 component*) to distinguish them from *objects*.

I4.0 components on the one hand represent *objects* within the framework, but, on the other hand, have the ability to interact with other elements. 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 systems 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 *RAMI4.0* 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 6.

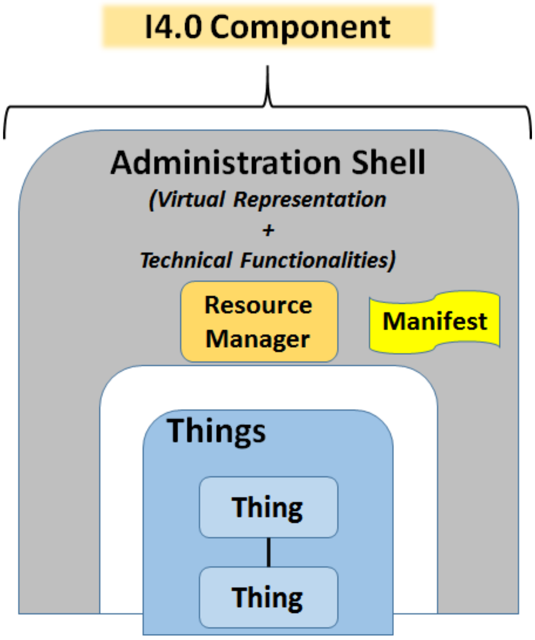


Figure 6: I4.0 Component

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*** (named after the JAR files *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. The *Manifest*, therefore, contains public information, based on a standardised semantics, required to effectively interact with the related *I4.0 component*.

Neither it is necessary that the *Administration Shell* and its *object(s)* have to be physically co-located, nor an *I4.0 component* necessarily represents a physical entity as is depicted in Figure 7. Indeed, in this figure there are *I4.0 components* co-located with their *objects* (i.e., the "robot" and the "device"), while other *objects* could have their *Administration Shells* located somewhere else, for example in a "repository". The *Administrative Shell* in some way must be considered as the "entity" it represents from a *RAMI4.0* point of view. If the *Administrative Shell* is co-located with its object (i.e., it is embedded, or it embeds, the object) then the previous statement is obvious; if, instead, the *Administrative Shell* is remote and the object is physical, then the *Administrative Shell* has to take care on communicating with the object using means and protocols the object supports to acquire data or control the physical entity. As indicated in Figure 7, there could be *I4.0 components* related to physical objects (i.e., the "robot", the "device, the "machine" and the "product"), while other *I4.0 components* represent immaterial objects (e.g., a software solution, and a technical drawing).

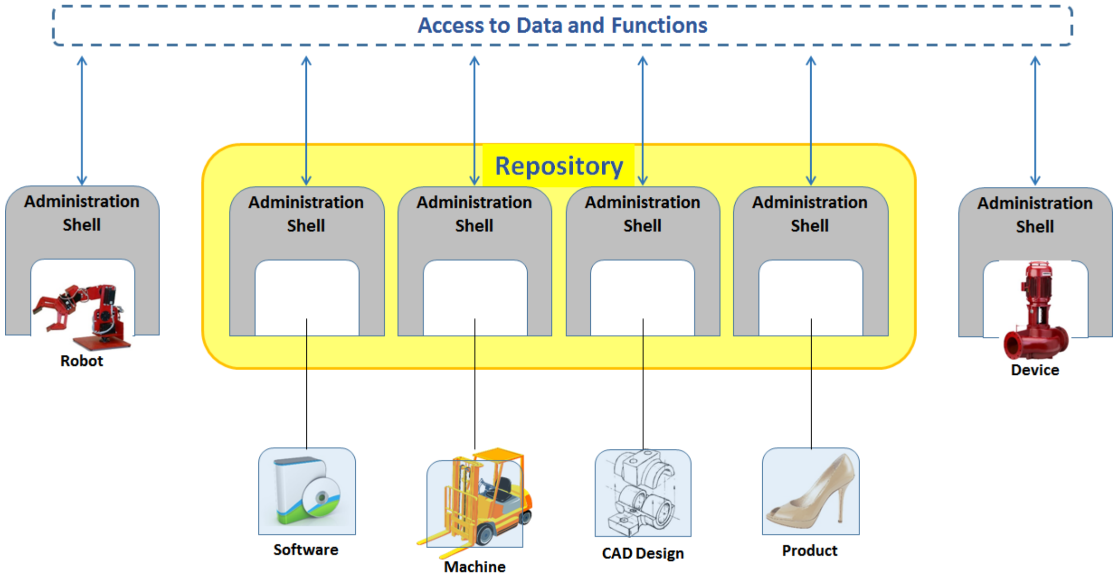


Figure 7: I4.0 Component Examples

Currently *RAMI4.0* 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*[25] (EDD), and the *Field Device Tool*[26] (FDT) specifications;
- the *Functional and Information Layer* the *Field Device Integration* [2] (FDI) specification as integration technology.

The *FDI* is a new specification that aims at overcoming incompatibilities among some manufacturing devices specifications. Essentially the *FDI* specification defines the format and content of the so-called *FDI Package* as a collection of files providing:

- the device *Electronic Device Description* (EDD) (formalised using the IEC 61804-3 *Electronic Device Description Language*), which includes the device characteristics specification, the business logic and information defining the user interface elements (UID - User Interface Description);
- the optional User Interface Plugin (UIP) that defines programmable components, based on the Windows Presentation Foundation specifications, to be used for developing UI able to effectively interact with the device;
- possible optional elements (called *attachments*) useful to configure, deploy and use the device (e.g., manual, protocol specific files, etc.).

An *FDI Package* is therefore an effective mean through which a device manufacturer defines which data, functions and user interface elements are available in/for the device.

The *RAMI4.0* specification currently indicates, for end-to-end engineering, the *AutomationML* [3] and the ProSTEP iViP[27] specifications.

Anyway, the *RAMI4.0* 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.

---

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

[26] http://www.fdtgroup.org/technical-documents

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

3.4. OpenIoT

OpenIoT was one of the EU FP7 projects within the IoT European Research Cluster (IERC). The project focused on developing a framework for integrating a set of sensors and for developing applications that can integrate data acquired from sensors in a seamless and homogeneous way. To this end, the OpenIoT platform is heavily based on semantically enhanced information both to characterize all elements accessible through and within the platform (e.g., sensors, services, etc.), as well as acquired data. Therefore, the systematic provision of meta-information eases the identification and selection of elements of interest for the end-users, as well as provides a consistent interface to use and integrate them.

The OpenIoT platform can be roughly split into three layers (see Figure 8):

- the **Utility-App Plane**: that enables the specification of service requests, the presentation of service results, and the configuration of functionalities over the sensors and the services that are deployed within the OpenIoT platform;
- the **Virtualized Plane**: that provides the means for discovering, accessing and processing IoT data in a semantically interoperable way;
- the **Physical Plane:** that deals with the acquisition of observation from the physical world, through either physical or virtual sensors.

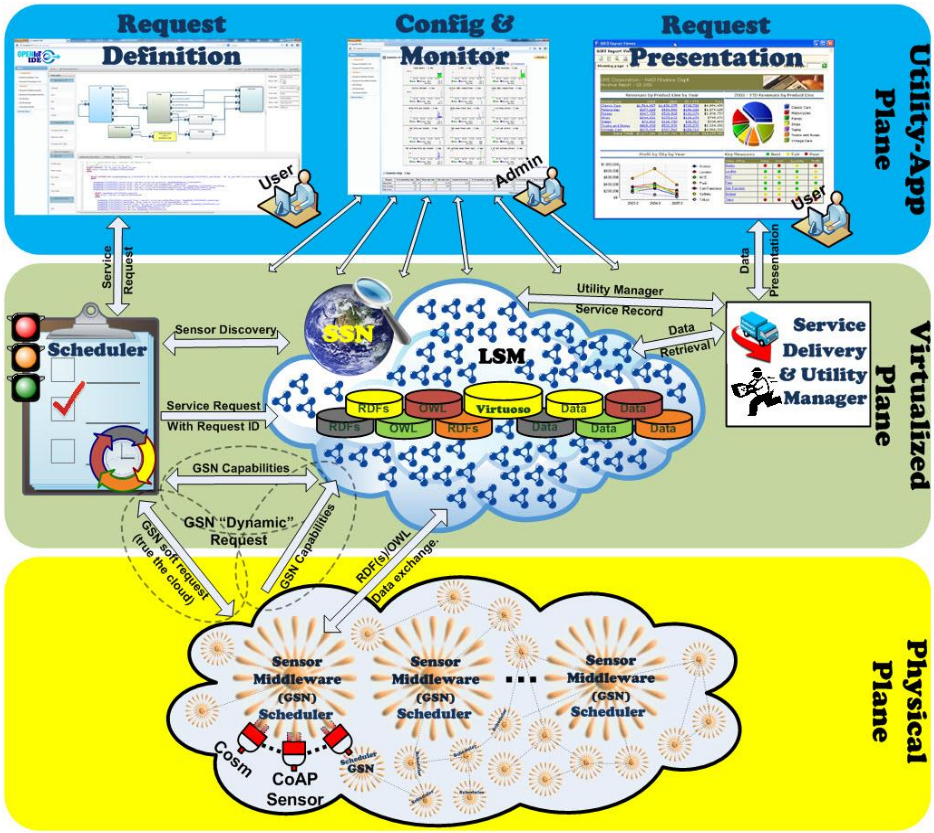


Figure 8: OpenIoT LSM Internal Architecture

The *Physical Plane* can potentially use any communication technologies to integrate sensors, singularly or through gateways, using *wrappers*. These take care not only of mapping the communication protocol from the ones supported by the sensor(s) to the ones supported by the OpenIoT platform, but, more specifically, to semantically enrich the sensor's interface and data so to provide to the intermediate layer the information in a uniform, consistent and semantically enriched form.

The OpenIoT platform has been deployed essentially in "open contexts" (like smart cities or domotics) characterized by a wide variety (in terms of types of sensors) of sensors and sensors' owners (e.g., end-users providing access to sensors they have deployed at home or in their environment), and of applications' needs (e.g., applications that collect data from a set of sensors and display them graphically).

As stated, all the elements managed within the OpenIoT intermediate layer are semantically annotated and, to this end, OpenIoT makes use of the RDF standard to encode metadata and other linking information as envisaged by the Semantic Web approach[28]. Therefore, the OpenIoT managed data constitute a "web of data" that can be distributed among different computing systems without affecting their relations and usability.

The OpenIoT platform is using ontologies and taxonomies like: WGS84[29] (geospatial ontology), LinkedGeoData[30] (spatial areas and points of interest ontology), W3C SSN[31] (Semantic Sensor Network ontology), LSM[32] (Linked Sensor Middleware ontology), as well as DUL (DOLCE+DnS Ultralite) an OWL upper ontology that describes very general concepts common to all/many knowledge domains.

The most relevant component of the OpenIoT intermediate layer (the *Virtualized Plane*) is the **Linked Sensor Middleware** (**LSM**) that actually provides all the functionalities related to the data sources' integration, data elements dispatching and storing.

As represented in Figure 9 the OpenIoT *LSM* communicates with different kinds of *wrappers* in the southbound interface. As clearly indicated in the figure, there are:

- *Physical Wrappers* that connect physical devices to the *LSM* and that can simply take care of semantically enhance the sensors, as well as act as protocol-mappers;
- *Mediate Wrappers* that act as gateways among sensors networks and the *LSM*;
- *Linked Data Wrappers* that allow to integrate data in relational databases within the overall OpenIoT framework.

The *wrappers* normally expose a RESTful API.

[28] http://www.w3c.org/standards/semanticweb/

[29] http://www.geonames.org/ontology/documentation.html

[30] http://linkedgeodata.org

[31] http://www.w3c.org/2005/Incubator/ssn/

[32] http://open-platform.eu/library/deri-lsm/

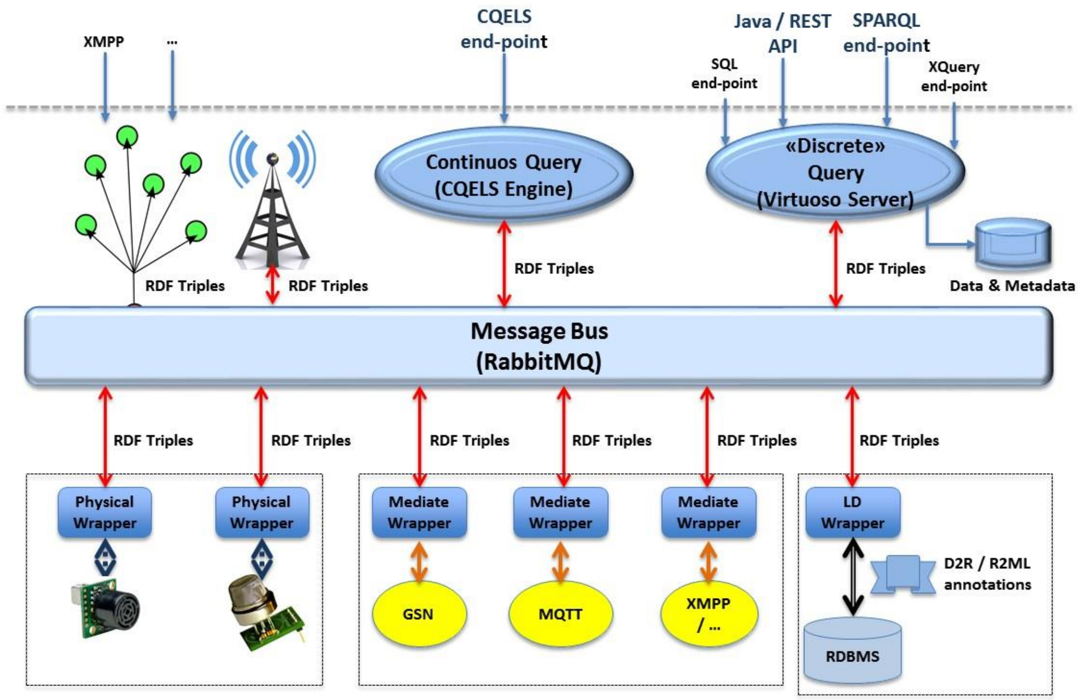


Figure 9: OpenIoT LSM Internal Architecture

As evident from the figure, the *LSM* essentially envisages a *Message Bus* in charge of collecting all events as provided by the southbound interface and dispatch them to a set of *LSM* components that span from output channels (e.g., the *XMPP* one of the left side of the figure) directly providing events dispatched via the *Message Bus*, to storage facilities and query engines end points.

The *LSM* takes care of storing all information (and meta-information) related to sensors, services, and data. To this end it uses the OpenLink Virtuoso data server that is able to provide the functionalities of a traditional RDBMS, as well as the ones of an object-oriented RDBMS, virtual database, RDF, XML, free-text, web application server and file server in a single system. Due to its hybrid approach, Virtuoso also support various query languages including SQL for relational data, XQuery, XPath, XSLT for XML data, and SPARQL for RDF data. Virtuoso recognizes most RDFS and OWL predicates for reasoning. Virtuoso is also in charge of providing the OpenIoT SPARQL 1.1 endpoint.

Additionally, OpenIoT provides facilities to execute queries over Linked Stream Data both in pull-based or push-based fashion.

Push-based continuous queries are managed via the CQELS engine using the CQELS formalism[33]. Continuous queries are first registered in the system, and continuously executed when new data arrives, with new results being output as soon as they are produced. The CQELS engine provides the *LSM CQELS* endpoint.

The *LSM* northbound interface is therefore a combination of different endpoints as depicted in the previous figure.

Even if the contexts addressed by OpenIoT do not completely overlap with the BEinCPPS ones, the OpenIoT *LSM* approach is interesting and effective for the BEinCPPS.

[33] CQELS is an extension of SPARQL 1.1 standard to support *Continuous Query Evaluation over Linked Stream* (http://code.google.com/archive/p/cqels/)

3.5. CPPS Engineering approaches

To make the adoption of CPPS a real breakthrough in manufacturing and to enable new business processes, design methodologies and tools are necessary and must be offered to several types of stakeholders, each one having not only different perspectives and operating in different domains (e.g.: software engineers and hardware engineers) but also different levels of understanding of the technologies to be adopted on the field level.

Therefore, the BeInCPPS platform will adopt a layered approach to the CPPS engineering, where tools and methodologies offers different levels of abstraction from the technical detail of the final implementation.

3.5.1. MSEE Toolbox

The approach based on different levels of abstractions is based on the MDSEA approach integrated with a suite of tools for model creation, transformation and execution developed in the MSEE project (www.msee-ip.eu), that provided a methodology (based on GRAI) and toolbox to support the model refinement and that results, in the end, in the complete specification of a running system.

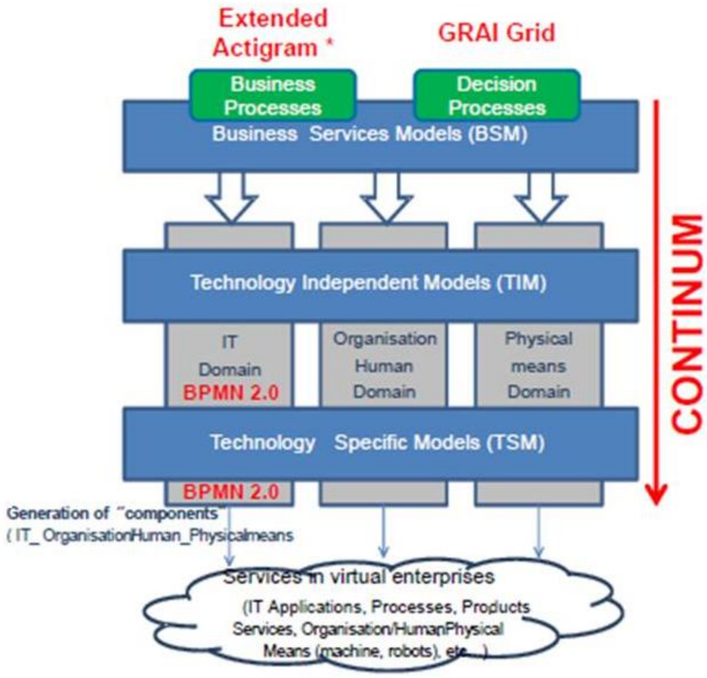


Figure 10: The MSEE Model Driven Service Engineering Architecture

Besides the fact that the MSEE toolbox is originally conceived for the design of service systems, BeInCPPS sees the possibility to re-target it to the CPPS design (more on Sect 5.7), as Service Engineering and CPPS Engineering share different commonalities, among which: as said above, the need of supporting different perspectives and stakeholders, of transforming models from more abstract levels up to implementation, of defining executable business processes by orchestrating different services and supporting interoperability of exchanged data.

At the BSM (Business Service Model) level, Extended Actigram Star (EA*) artefacts are used within a graphical environment to model business processes, whereas the GRAI methodology (GRAI Nets) is adopted to support design of decisional points. UML diagrams can capture the domain specific data.

EA* diagrams can be transformed into BPMN2.0 executable models of business processes.

In the TIM (technology Independent Model) level, the business processes are refined and detailed, whereas IT artefacts (UML specific diagrams) are introduced to guide the specification of IT elements and their interactions.

The usage of the MSEE Toolbox supports not only the creation of the BSM models, but also the transformation from BSM to TIM models, thus transforming EA* diagram into BPMN2.0 models that can be executed within the Toolbox suite.

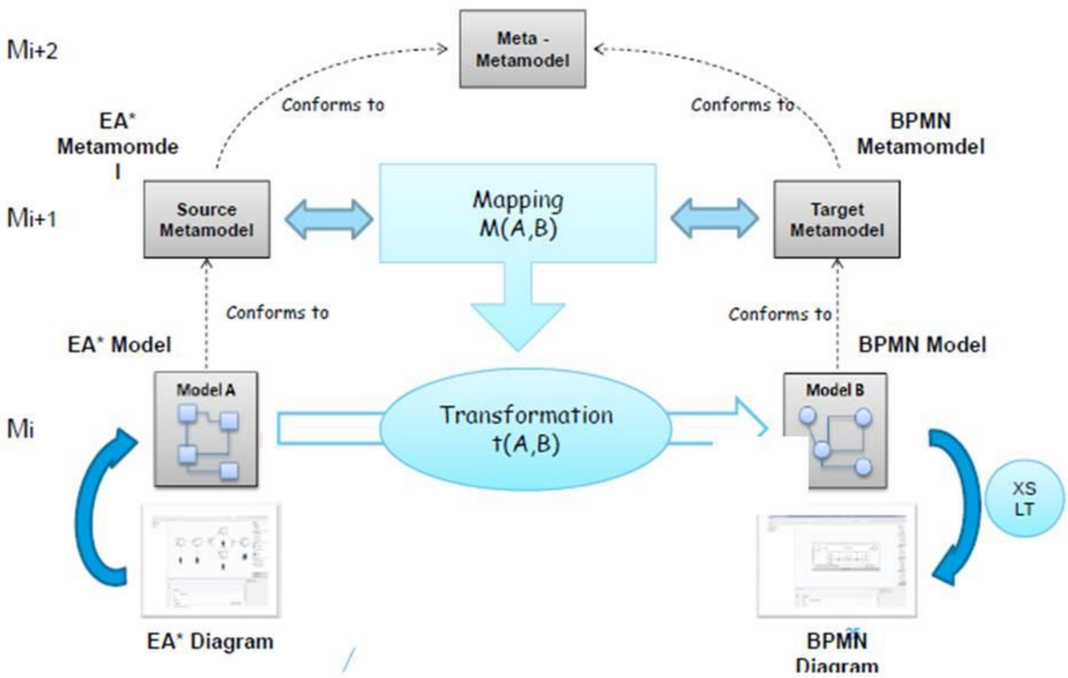


Figure 11: Transformation from EA+ diagrams to BPMN processes

3.5.2. CPSe-Labs Platforms for CPS Engineering

The CPSe-Labs project, offering funding and technological support for experiments (based on the adoption of CPS) to business operating in various engineering and technological domains, provides access to a catalogue of technology platforms, some of which suitable for the engineering and simulation of the CPSs.

4DIAC (see more in the next section) offers both an IDE and a run-time environment (called FORTE) for programming PLC.

CRESCENDO[34] is an Eclipse-based IDE for the collaborative modelling and simulation (*co-simulation*) of CPS, using the standard VDM formal method[35] and its languages (VDM-SL, VDM++, VDM-RT) for modelling the software and the 20-Sim[36] for the hardware

[34] CRESCENDO (http://crescendotool.org/) was originally developed by the FP7 DESTECS project as an extension of OVERTURE (http://overturetool.org/)

[35] The Vienna Development Method was originally defined by IBM in the 1970s – http://overturetool.org/method/

[36] http://www.controllab.nl/en/products/20-sim.html

modelling. It brings together the software and hardware engineers, thus allowing the collaborative modelling of the CPS.

SYMPHONY[37] is similar in both technology and objectives, but while CRESCENDO is focused on individual CPS, SYMPHONY targets SoS (system of systems). This IDE uses a combination of CML[38] and SysML[39] as its modelling language. It is worth noting that both platforms are of a conceptual nature, in the sense that their output is mainly *knowledge and documentation*.

AIDE platform offers support to simplify the integration of different CPS engineering tools (in particular the CRYSTAL toolset, see section #3.6.1) through innovative approaches for data integration based on open standards (OSLC[40]) and open source software for code generation. The OSLC Lyo Code Generator supports the development, integration and testing of adapters based on the OSLC standard. The Ecore is a meta-model of an OSLC domain specification, being at the core of the code-generator.

Other modelling tools, specifically oriented to the design and simulation of embedded systems (CRYSTAL) and PLC (4DIAC), and thus technology dependent, are described in the following section

3.6. CPS-ization

The ongoing transformation of the field level in the factory towards more intelligent, active devices is referred to as the *CPS-ization* of the factory. To this end, a number of trends and architectural building blocks have been identified and are described in this chapter. These refer both to the field level technology itself as well as to the methods and tools that are used for the development of such systems.

In particular, we look into:

- CPS System Engineering Environments Platform
- Model-based Engineering for Systems Design
- Computation architectures for Field level
- Communication architectures for Field level

3.6.1. CPS System Engineering Environments (CRYSTAL)

BEinCPPS will consider the CRYSTAL (CRitical sYSTem engineering AcceLeration) platform as a basis. CRYSTAL has taken up the challenge to establish and push forward an Interoperability Specification (IOS) and a Reference Technology Platform (RTP) as a European standard for safety-critical systems development. The CRYSTAL RTP is a generic model-based tool integration platform composed of a set of interoperable tools, methods and processes designed to improve the development of safety critical embedded systems.

After designing the different tool chains, an adequate systems engineering environment (SEE) is instantiated according to a platform building process which relies on the repository of tools and methods (Figure 12). In this context, the RTP provides a complete SEE with a set of engineering methods and processes, as well as engineering tools.

---

[37] SYMPHONY (http://symphonytool.org/) was originally developed by the FP7 COMPASS project (http://www.compass-research.eu/), and is based on the same OVERTURE tool that CRESCENDO was derived from

[38] The COMPASS Modelling Language – http://www.compass-research.eu/approach.html

[39] The OMG SysML modelling language: *www.omg**sysml**.org/*

[40] Open Services for LifeCycle Integration: http://open-services.net/

The integration and interconnection of tools is needed in order to support collaboration between all relevant business processes of an enterprise for example interactions with customers, suppliers, and partners. This generic integration platform hosts tools from different vendors (COTS, open source) and is able to host additional technologies from end users.

The IOS provides interoperability and collaboration between tools across the entire engineering lifecycle. The IOS standard allows loosely coupled tools used in the product development process to share and interlink their data based on standardized and open Web technologies that enables common interoperability among various life cycle domains. In such way, the RTP should be seen as the main solution provided by CRYSTAL; it encompasses: the IOS standard, methodologies about how to use the IOS in the product development process, development tools that support IOS and the Platform Builder tool that will support the basic configuration of a system engineering environment making use of IOS and the tools supporting it.

The IOS consists of a specification for achieving common tool and data interoperability in heterogeneous systems engineering development environments. In particular, it encompasses the specification of three main aspects:

- The specification of communication paradigms and protocols to be used for exchanging information between integrated tools and data repositories,
- The specification of data exchange formats (or syntax, referring to the formats used for serializing data as strings, e.g., RDF/XML, XMI/XML, JSON, etc.), and
- The specification of the semantics of the information to be interpreted and exchanged across these tools and data repositories (or abstract syntax, referring to the definition of sets of concepts for lifecycle integration, defined with their properties and relationships).

The key idea pushed forward in the IOS consists in relying on standardized integration interfaces for supporting interoperability in the system engineering lifecycle, with the goal to overcome redundant integration problems (e.g. related to point-to-point and ad-hoc integration architectures, isolated tool silos, users locked-in, reusability & maintenance of integration assets) across the boundaries of engineering disciplines, application domains and tool providers. Such standardized integration interfaces have to define lightweight and generic concepts as a common denominator for all the artefacts used holistically throughout the development cycle. In the context of lifecycle interoperability, the main focus is put on the semantics of the links and dependencies between the artefacts crossing the boundaries between the engineering disciplines (i.e. related to requirement engineering, design & implementation, and V&V related activities).

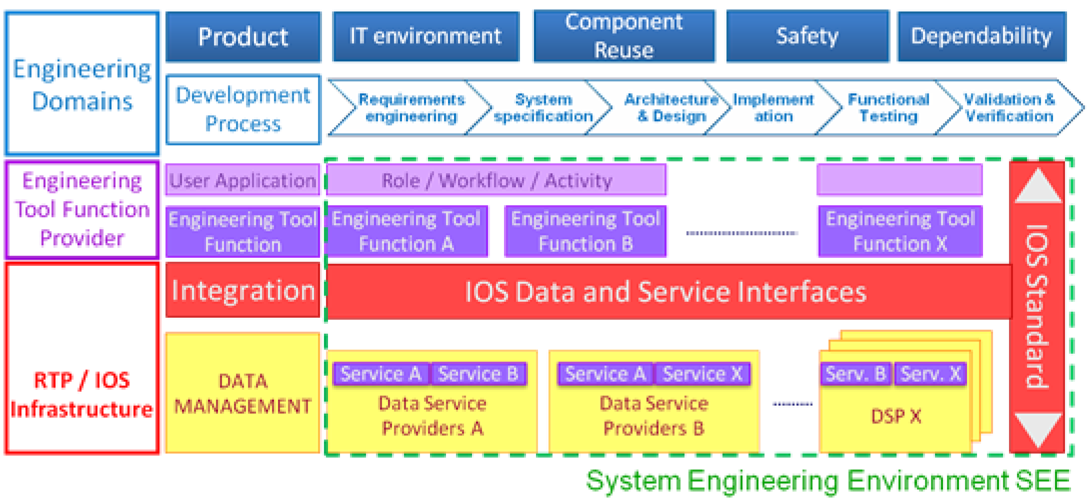


Figure 12: Technical approach for standardized tool integration

Figure 13 presents the IOS layered architecture, which has been enhanced in CRYSTAL from the CESAR and the MBAT IOS. The top part of the figure encompasses the tool and domain-specific syntax and semantics, possibly based on proprietary and island solutions, which is basically out-of-scope of the IOS. On the contrary, the bottom part sketches the scope of the IOS, (a) specifying a common way for handling Lifecycle Interoperability (with respect to communication protocols, syntax, services and semantics used as a common ground for exchanging lifecycle artefacts and control flows between integrated engineering tools in a standardized way), and (b), the set of other Engineering/Interoperability Standards, supporting in depth Systems Engineering activities, and to be interfaced with Lifecycle Interoperability concepts.

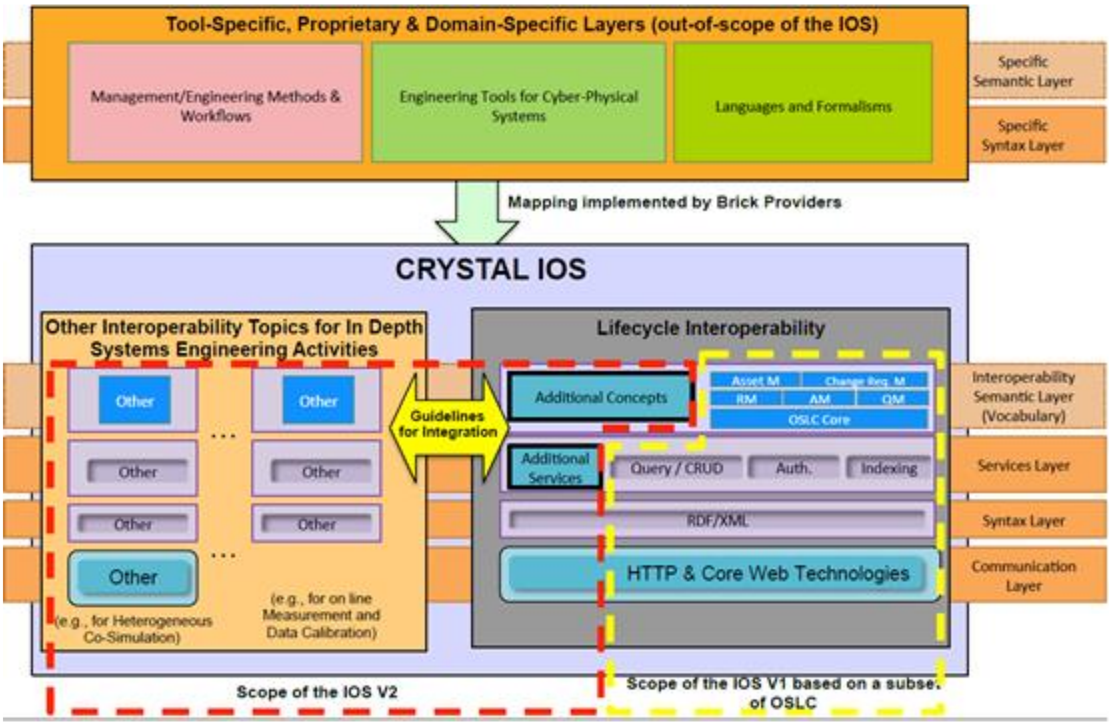


Figure 13: The CRYSTAL IOS layered architecture

A follow-up IOS standardization activity to enhance the further uptake of the Interoperability Specification in various engineering activities has been started in the project CP-SETIS. Their approach is depicted in Figure 14.

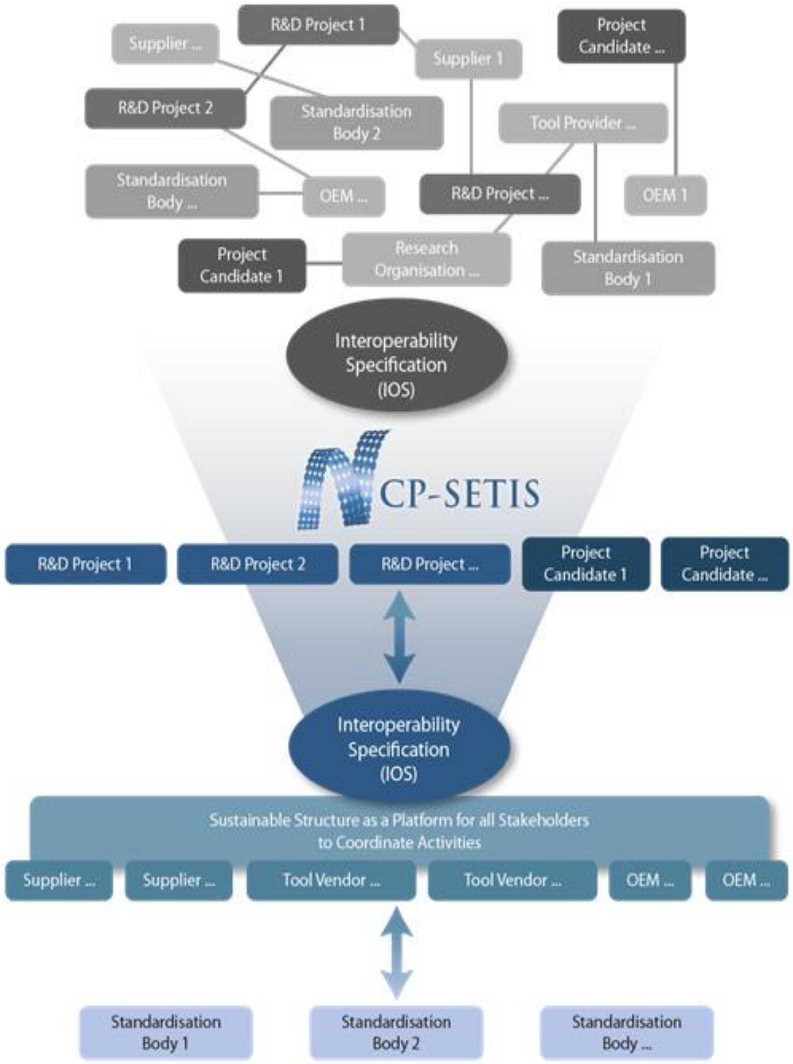
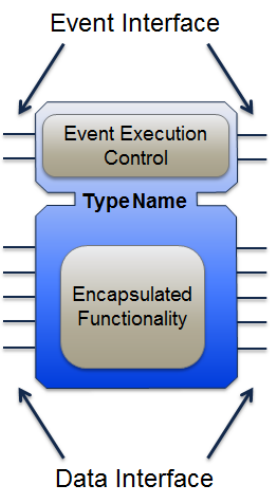


Figure 14: The CP-SETIS IOS Standardization approach

3.6.2. Model-based Engineering for Systems Design

In embedded systems the market demands a more and increasingly complex number of functions. In automotive systems these increasing number of functions could be energy braking or adaptive cruise control. As such highly integrated systems are based on a combination of mechanical, electric/electronic, and software parts to implement these functions, their complexity — specifically increased by their interactions and the integration of HW- and SW-related aspects — pose a substantial challenge to the development of such software. To effectively manage this complexity, development processes in general, and model-based approaches in particular, support the development assuming an idealized (component- based) model of computation, abstracting away from implementation issues like interference aspects of the execution platform resulting from shared computation or memory resources. However, as requested by the standards (e.g. ISO 26262), those simplifying abstractions must be met by development steps that ensure that the assumptions behind these abstractions are not violated by the properties of the implementation platform. For example, during SW-/HW-integration platform mechanisms must avoid that task deadlines are not met if executed on a single ECU (time-separation), or signal changes of one task do not unintentionally change those of another task (space-separation). The use of cyber-physical systems in production systems results in the "smart factory". Its products, resources and processes are characterized by cyber-physical systems; through its specific properties, it offers advantages with regard to quality, time and costs in comparison with classic production systems. For the engineering and implementation of cyber-physical systems, the integrative, interdisciplinary development of product and production systems needs to be promoted. This includes the modularization of production systems into production units using model-driven development.

3.6.3. Computation architectures for Field level



Within the real time context and especially for the field of manufacturing on its way to be digitized, a main aspect to be tackled is how to make the main platform architecture flexible and adaptable. Commonly, the lower level of manufacturing systems is represented by the control applications. Control applications are built from tiny devices referred as programmable logic controllers (PLCs). PLCs are mostly proprietary to classic suppliers such as Siemens. This is a barrier to innovation which can be overcome through the adoption and the integration of open source platforms. The standard behind proprietary PLCs is IEC 61313. Since 2005, this standard has been extended by IEC 61499. The most important features of IEC 61499 is its support to the development, the deployment and the execution of distributed controllers. IEC 61499 goes beyond the concept of function blocks in three directions with respect to (1) events and data, (2) event-driven state machines, and (3) encapsulation and reuse. With these concepts, the development of distributed applications is simplified by having a clear separation between applications, systems, devices, resources. Furthermore, IEC 61499 defines the data exchange format and provides support for reconfiguration by means of fine granular application access and (re)configuration interfaces.

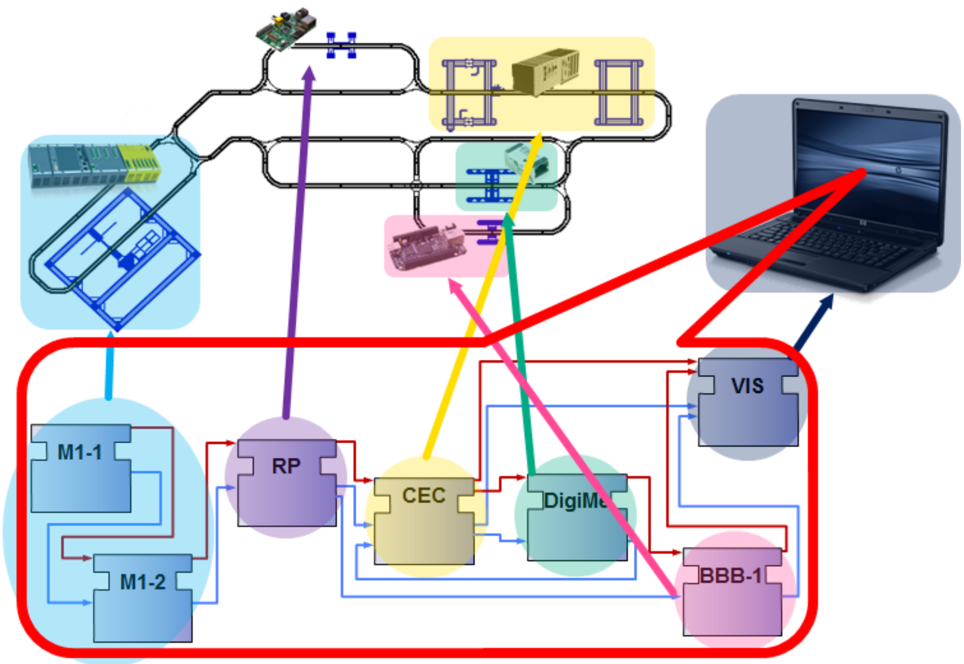


Figure 15: Applying IEC-61499

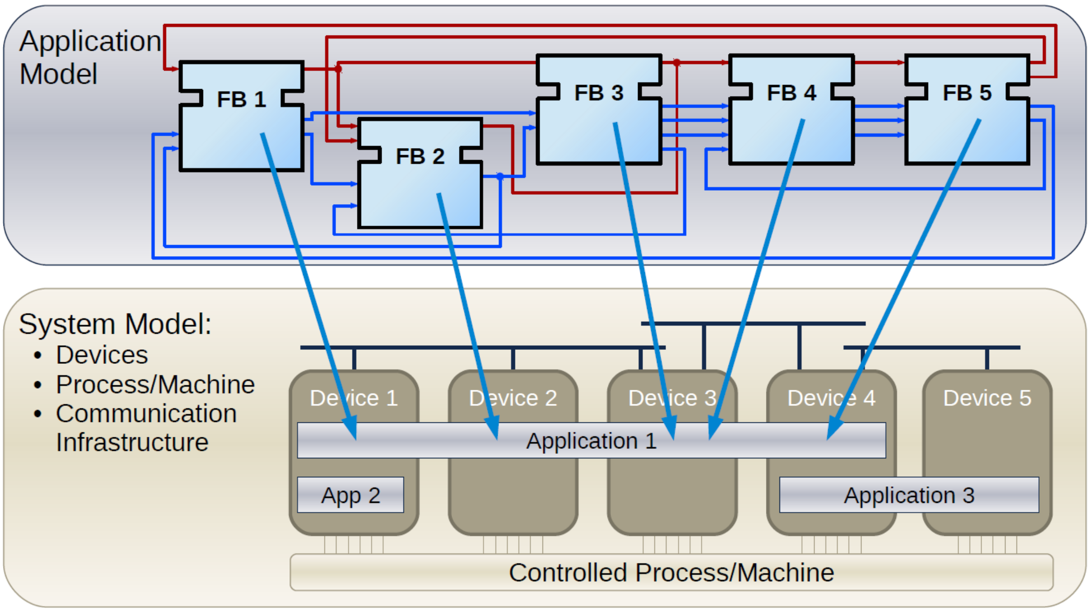


Figure 16: System and Distribution Model According to IEC-61499

As an exponent of the engineering methods implementing the technology behind IEC 61499 is 4DIAC[41], a Framework for Distributed Industrial Automation and Control. 4DIAC is an open source initiative founded in 2007 by PROFACTOR GmbH and the Automation and Control Institute (ACIN) of the Vienna University of Technology. Currently, 4DIAC is built upon the runtime environment FORTE[42] and the engineering environment 4DIAC-IDE. FORTE is a small portable C++ implementation of an IEC 61499 runtime environment targeting small embedded control devices. 4DIAC-IDE provides an extensible engineering environment for modelling distributed control applications compliant with IEC 61499.

[41] https://eclipse.org/4diac/

[42] https://eclipse.org/4diac/en_rte.php

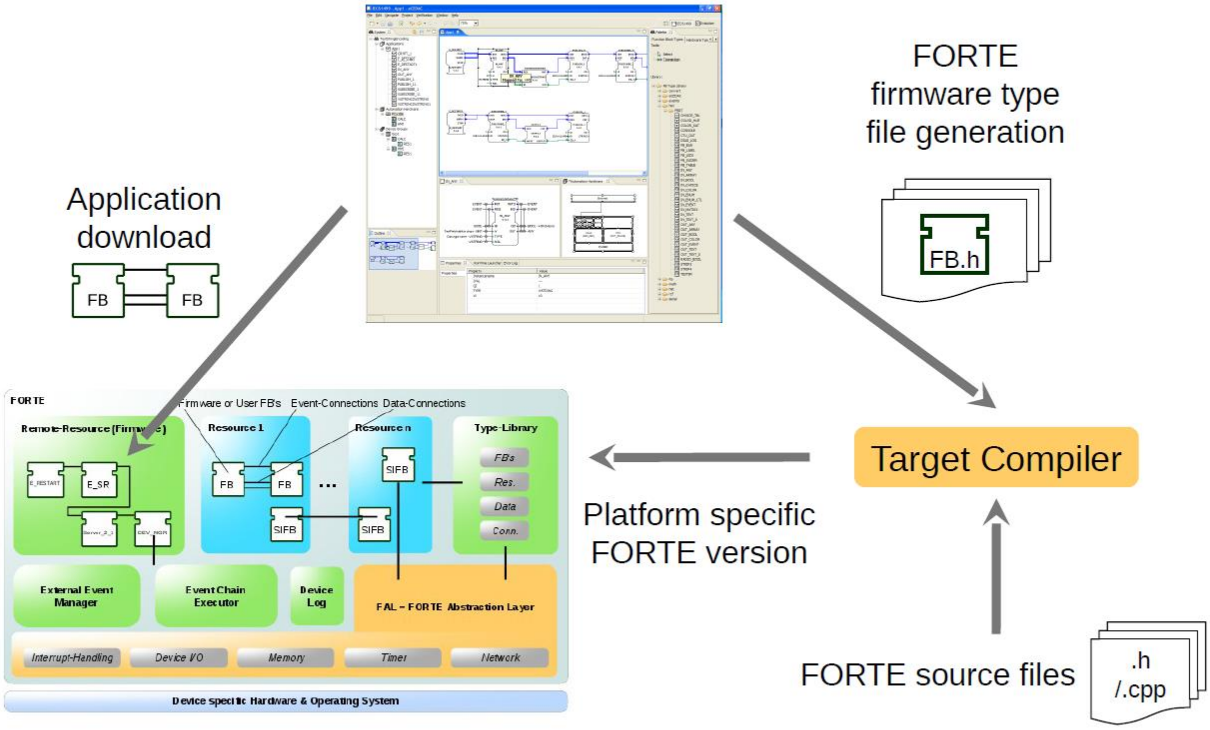


Figure 17: 4DIAC Toolchain

4DIAC has been already adopted by both academic and industrial partners. Most projects running within the European project CPSE-labs[43] are investigating the integration of an OPC-UA type of communication within 4DIAC.

3.6.4. Communication Framework between Field Level Devices / Mixed Criticality – Deterministic Ethernet

In industrial automation and in energy production, for example, improved connectivity of robots, wind turbines or substations can lead to big increases in production efficiency, reduced system downtime and human-machine collaboration. To take advantage of the improvements in efficiency, uptime and functionality that Internet of Things (IoT) can deliver, the underlying networks must provide reliable and deterministic M2M and machine-to-cloud connectivity, at prices that only open standards can ensure.

Deterministic Ethernet is an IEEE 802.1 compliant extension to Quality of Service (QoS), using scheduled mechanisms based on a global time instead of dynamic priority schemes.

Deterministic Ethernet operates using a global sense of time and a schedule which is shared between network components. Deterministic Ethernet is built upon open standards and is based on standard Ethernet (IEEE 802.3, 802.1D, 802.1Q), IEEE 802.1 AVB standard, the IEEE 802.1 TSN pre-standard; IEEE 1588 v2 or SAE AS6802 can be used as clock synchronization standards.

[43] cpse-labs.eu/germany_s.php

Figure 18: Deterministic Ethernet

In a Deterministic Ethernet network, regular unconstrained best-effort Ethernet traffic can co-exist with real-time critical traffic flows without altering the guaranteed and deterministic strict delivery timeliness of scheduled traffic flows. Mixed-criticality requirements from classic industrial deployments (e.g. factory floor) can be fulfilled and guaranteed with the support of deterministic scheduled networks and a carefully build distributed communication scheme.

To make it possible that customers converge real-time controls traffic with regular best effort traffic on one Ethernet network, in Deterministic Ethernet, time scheduled traffic is partitioned from all other network traffic. It is therefore immune to disturbance: critical functions can send messages at scheduled points in time with a guarantee of available bandwidth and message delivery. This immunity for time-critical traffic allows the convergence of many different functions such as control, data-analytics, operations and enterprise on one physical network. This means that in a Deterministic Ethernet network, latency of critical scheduled communication is guaranteed. This is called Guarantee of Service. Thus one feature of Deterministic Ethernet is that, especially in the context of IoT, it is able to provide Guarantee of Service.

By connecting real-time systems over open, standard Deterministic Ethernet, the concepts of IoT are extended to the edge of the industrial control network. This is referred to as the IoT Edge. In such an architecture, critical real-time control communication can be integrated with non-critical traffic and data acquisition, all over one converged network. Guarantee of Service from Deterministic Ethernet enables enhanced security and partitioning for a more unified information model.

Deterministic Ethernet is used in a wide range of applications where guaranteed latency is vital, either for reasons of operational efficiency of functional safety. These include autonomous driving, machine-to-machine communication and aerospace flight control.

3.6.5. Communication Framework between Field Level Devices / OPC-UA

As an orthogonal approach to Deterministic Ethernet, service-oriented communication platforms such as OPC-UA[44] or DDS[45] have recently shown to be used from the IoT edge. BEinCPPS is interested in particular in the adoption of OPC-UA. This is mostly because there are numerous[46] [47] [48] recent success stories with companies praising OPC-UA in various domains from automotive to industrial automation. OPC UA is an independent distributed

---

[44] https://opcfoundation.org/about/opc-technologies/opc-ua/

[45] http://portals.omg.org/dds/

[46] https://www.unified-automation.com/references/case-studies.html

[47] http://www.automationworld.com/communication-protocols-standards/opc-ua-opens-world-applications

[48] http://opcfoundation.org/opc-connect/2014/12/opc-ua-enabled-smart-devices-drive-intelligent-water-management/

platform for data exchange in industrial control systems. It is meant to facilitate the integration between products from different manufacturers and across operating systems. OPC-UA has been built as an improvement of OPC which has behind 20 years of development as a platform for industrial communication between SCADA systems, MAS systems, field applications and shop floor applications. OPC-UA is an open standard, and, consequently, industry, application, and most importantly, vendor independent. Thanks to this aspect, the eco-system built around OPC-UA grows at a fast pace. More technically, as explained in a 2015 guideline [4], OPC-UA successfully addresses a broad set of requirements from Industrie 4.0:

- scalability: OPC-UA scales from tiny (15 kB) footprint (Fraunhofer Lemgo) through to single- and multi-core hardware with a wide range of CPU architectures (Intel, ARM, PPC, etc.) OPC-UA is used in embedded field devices and in controllers and SCADA/ HMI products as well as MES/ERP systems;
- security: OPC-UA provides signed and encrypted transfer and X.509 certificates, Kerberos or user/password for authentication;
- transport: two protocol bindings are available, TCP-based and HTTP/HTTPS web service; Publish/Subscribe communication model can be integrated; consistent data transport is guaranteed; live and historical data are standardized; alarm and eventing via token based mechanism (late polling) are possible;
- communication: OPC-UA defines “discovery” mechanisms for identification and notification of OPC-UA devices and their functions within a network; OPC-UA participants can be located local (on the same host), in a subnet or global (within enterprise); aggregation and configuration-less procedures (e.g. Zeroconf) are used to identify and address network participants;
- standardization: OPC-UA is already an IEC standard (IEC 62541), and tools and test laboratories for testing and certifying conformity are available.

As it is, OPC-UA does not offer any real-time capabilities mostly because, in its nature, it is user driven. However, a combination between Deterministic Ethernet and OPC-UA might be of interest in order to benefit from both worlds. This is, in principle, possible, as OPC-UA is an independent platform, consequently, it is not bound to a specific network protocol. A related approach based on Time Sensitive Networking (TSN) is considered in the BEinCPPS project.

3.6.6. Communication Framework between Field Level Devices / Wireless Sensors

Regarding the wireless communication industrial use cases proposed in BEinCPPS can benefit from, which uses developments and advances in the state of the art of other European projects, wireless solutions for field devices based on DEWI’s European project outcomes could be relevant.

The ARTEMIS project DEWI (“Dependable Embedded Wireless Infrastructure”) provides key solutions for wireless seamless connectivity and interoperability in smart cities and infrastructures, by considering everyday physical environments like buildings, cars, trains and airplanes, which will significantly contribute to the emerging smart home, smart cities and even smart factories. WSNs are expected to be deployed in many more applications that will allow the tracking, control, and measurement of many aspects of the environment, industry, and body health. WSNs will be the basis for the future deployment of the IoT, cyber-physical systems (CPSs), and machine-to-machine (M2M) communications.

To make this possible DEWI introduces the concept of a locally adaptable wireless “sensor & communication bubble” (DEWI Bubble), featuring: locally confined wireless internal and external access; secure and dependable wireless communication and safe operation; fast, easy and robust access to smart environments; flexible self-organisation, reconfiguration, resilience and adaptability; open solutions and standards for cross-domain reusability and interoperability.

The DEWI Bubble consists of 3 main elements:

- Sensor and actuator nodes

- Gateways, serving as interfaces between different bubbles or to the external world
- Users (internal and external), human or machine.

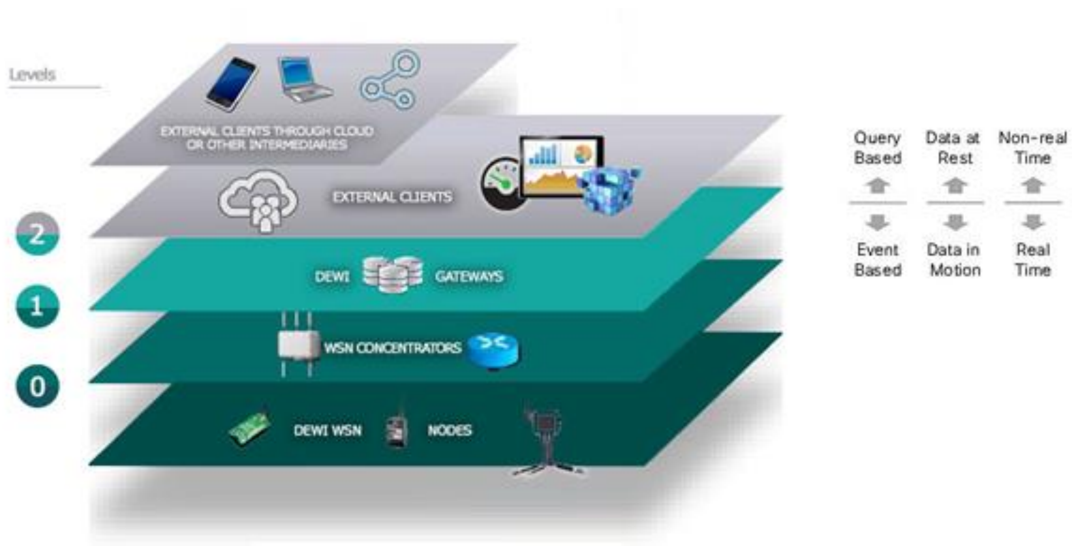
In addition, the bubble consists of appropriate extensions that provide functions for the bubble such as flexible data acquisition, aggregation and fusion, smart architecture, HW/SW co-design, security/data protection/authorization, re/auto/self-configuration, intelligent energy management and energy generation, reliability/robustness/safety, wireless standards, wireless sensor/device detection & localization.

Several different wireless communication technologies can be used within such a bubble. DEWI has a clear focus on short range technologies and corresponding standards, mainly based on IEEE 802.15 group of standards, and 6LoWPAN/ IPv6 (PFC 4919). Of special interests for industrial applications is the IEEE 802.15.4e, which provides robust mechanisms (channel hopping, time-slotted with multi frequency accurate communications, etc.) for enhancing wireless communication in demanding scenarios. This draft version of the standard is the most suited solution for providing wireless communication for field sensor devices in the scope of BEinCPPS.

In this context, not all nodes necessarily need to be wirelessly connected to each other; in this case, other nodes can act as relays. DEWI bubbles can also have different topological layouts and be organized as distributed (ad-hoc) or centralized networks. Incidentally, the DEWI Bubble is principally autonomous and should not be regarded as solely an extension to the Internet or as a first/last mile solution.

DEWI bubbles can be seen as an interesting block for providing wireless connection to the physical/field layer. The High Level Architecture (HLA) proposed within the project allows configuring and instantiating the different block to fit many different use cases.

From the communications point of view, the DEWI HLA is composed by a 3 levels structure as schemed in the figures below:



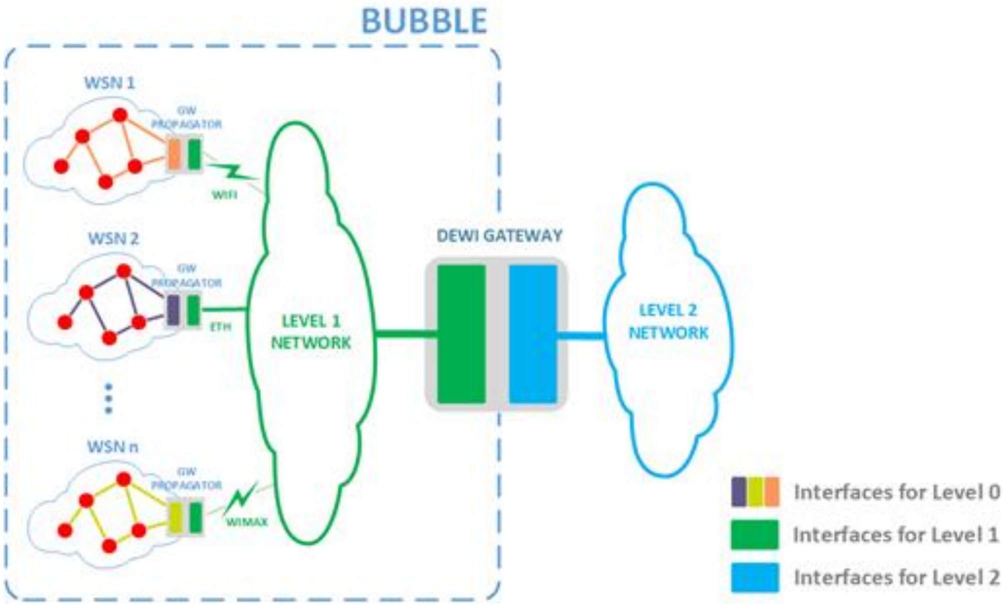


Figure 19: Scheme of the different communication levels within DEWI HLA

- **Level 0** is the communication technology/architecture inside the WSN. This technology is selected and implemented by the WSN developers. The selection is not limited and does not affect the interoperability between WSNs as it is exposed in level 1.
- **Level 1** is the communication technology/architecture inside the Bubble. It provides a communication flow between Propagators and the Bubble GW. End devices can be propagators if they use Level 1 technology to communicate directly with DEWI Gateway. WSN Concentrators are propagators that act like a gateway and enable the communication between end devices and the rest of the elements of the DEWI Bubble (rest of WSN, DEWI Gateway). For this purpose, propagators must implement a common communication protocol. The recommendation is to use an IP based network in order to enable different communication technologies (WiMAX, WiFi, Ethernet). The intra-bubble communication is a private network which has no direct access to Internet. After an analysis of the requirements, and regarding safety/security, two technologies have been chosen as the most suitable for this level:
    - The publish/subscribe protocol MQTT (most recommended option)
    - REST based solution.
- **Level 2** is the communication technology/architecture outside the Bubble. The communication is enabled through an external interface provided by the Bubble GW. This technology will be a standard for all DEWI, so clients (humans and machines) can gain access to any kind of Bubble to use their services.

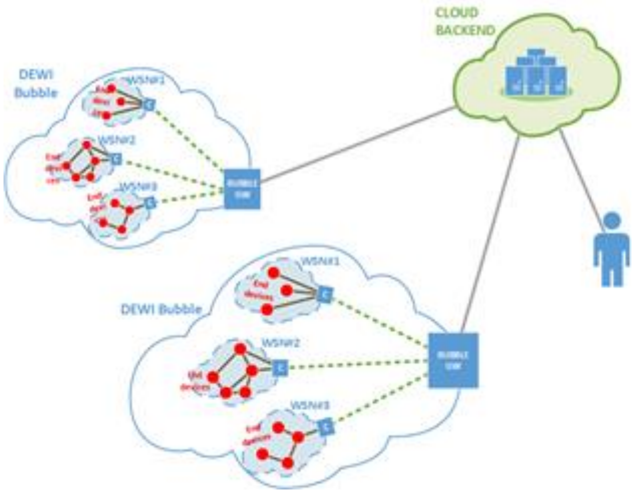


Figure 20: Possible Scenario. Communication centralized by a common backend.

From the point of view of the application of this architecture and developments inside the scope of BEinCPPS, level 0 and 1 are of special interest to allow wireless devices to connect the real world with the digital world. For instance, a WSN with a Level 1 MQTT implementation could interact with a FITMAN context broker or the OpenIoT LSM.

4. BEinCPPS Reference Architecture

BEinCPPS-Arch is the definition of a three-layered implementation, which federates the most prominent Smart Systems, IoT and Future Internet platforms.

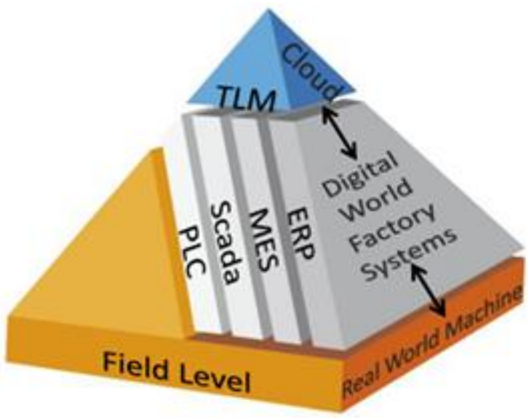


Figure 21: BEinCPPS' perspective of PATHFINDER's Automation Pyramid

As outlined by the PATHFINDER CSA, the advent of CPPSs has radically transformed the traditional hierarchical view of the Automation Pyramid (IEC 62264, ISA-88.01, ISA-95) into a more flexible one, where factory systems (MES, ERP, PDM) are more entangled with the physical world, to the extent of having some of their modules embedded in shopfloor machinery – see the left face of the pyramid depicted above. The adoption of such a paradigm in BEinCPPS is visually represented by the lower arrow on the pyramid's right face: thanks to the introduction of CPPS, traditional factory systems (Digital World at the Factory Level) and their embedded counterparts (Real World at the Field Level) realize a Real-Digital World bi-directional connection. In the first place, BEinCPPS extends this paradigm with the addition of an upper Cloud Level, where the Digital World gains a much broader scope: global business processes involving the enterprise as a whole and its manufacturing ecosystem at large (e.g., product lifecycle management and supply chain management, respectively) – see the upper arrow on the right face.

However, innovation with respect to the Automation Pyramid is even more radical in BEinCPPS. BEinCPPS-Arch defines two distinct, orthogonal axes in its layout: the Worlds axis for the Real, Digital and Virtual logical domains, the Levels axis for the Field, Factory and Cloud physical environments. These concepts intersect each other in a way that is best represented by Figure 22.

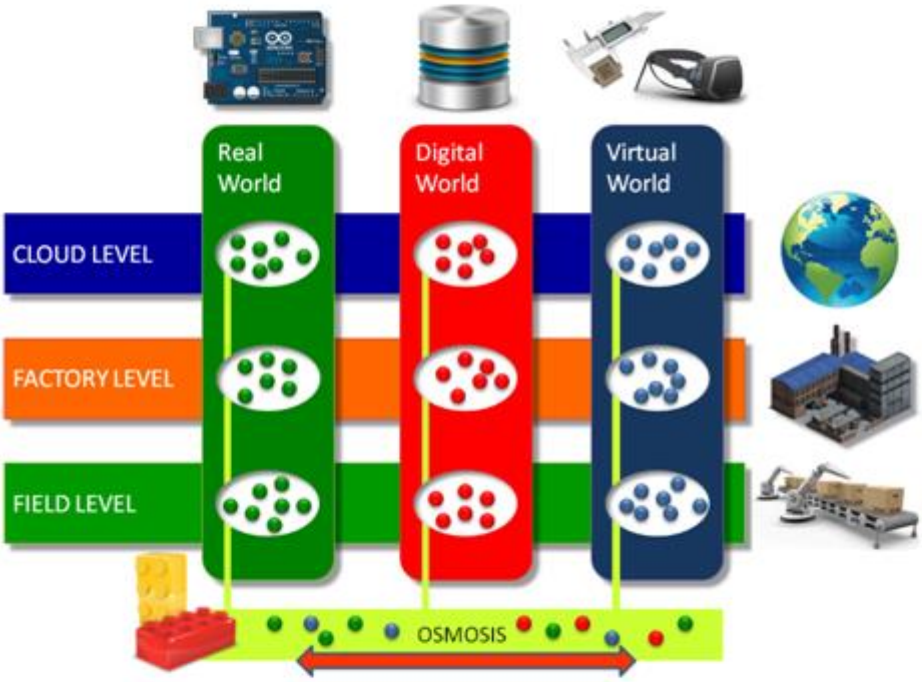


Figure 22: BEinCPPS-Arch Worlds vs. Levels

As this high-level view clearly shows, each World can span the three Levels, and a dedicated module (in yellow) is in charge of realizing a Real-Digital-Virtual connection that is not merely a data exchange between distinct entities (as happens in traditional approaches), but rather is implemented as an osmosis process: atomic elements of each World can be transferred through semantically-enabled semi-permeable membranes into adjacent Worlds, where they operate as remote agents. According to BEinCPPS-Arch, the three Worlds play roles very similar to those defined in the FInES Liquid-Sensing Enterprise context (see Section 3.1 about the OSMOSE architecture). Generally speaking, the Real dimension is about physical objects, machines, hardware and embedded systems, the Digital dimension is about software systems and data, the Virtual dimension is about abstractions like analysis, engineering and simulation.

Levels in BEinCPPS-Arch are not only a deployment concern, as their name might suggest, but also define a scope. For the Field Level, the scope is the shopfloor and its physical processes, for the Factory Level it is the production plant and its local business processes, while the Cloud Level is the broadest, encompassing the global business of the enterprise and beyond – with the possible involvement of external actors.

The intersection of Worlds (logical domains) with Levels (physical environments) creates a matrix of logical environments that are populated by different entities. The table below summarizes some of the relevant relationships – i.e., a list of the most prominent inhabitants of such environments. Items in bold are ICT asset categories that are going to be addressed by BEinCPPS-Platform's implementation; on the other hand, items in italics do not qualify as ICT assets, but rather define the runtime characterization of their environment.

	Real World	Digital World	Virtual World
Cloud Level **Scope: Ecosystem**	• Public / Virtual Private Networks (Internet) • Cloud Computing Infrastructures	• Enterprise / Ecosystem IT Systems • **Business Process Coordination & Monitoring Platforms** • **Data Interoperability & Monitoring Platforms** • **Big Data & Event Processing Platforms** • Global Data & Knowledge	• Collaborative Simulation Platforms • **Collaborative Engineering and Development Platforms** • **Advanced Visualization Platforms (3D / VR)** • **Ontologies and Models**
Factory Level **Scope: Plant**	• Private Networks (LAN / WiFi) • Local Computing Infrastructures • Cloud Proxies (Edge Computing Nodes)	• Factory / Enterprise IT Systems • **IoT / SS Interoperability & Management Platforms** • Local Data & Knowledge	• **Simulation Workstations** • **Engineering and Development Workstations** • Advanced Visualization Workstations (3D / VR)
Field Level **Scope: Shopfloor**	• **Wireless Sensor Networks** • **RT Industrial Networks** • **Embedded / Smart Systems** • IoT Devices (Sensors / Actuators) • Mobile Devices (Fog Computing Nodes) • Things & People		• Augmented Reality Systems

Table 1 - BEinCPPS-Arch Logical Environments

4.1. Alignment with OSMOSE Architecture

As already explained above we can assume that we have the corresponding three layered architectures for the digital and the virtual world. However, only in the real world we can assume that we have physical components which involve the IoT platform and with this the IoT layer. The field level in the digital world is a digital image of the field level in the physical world which applies to the factory level in an analogous manner. In the same sense the field and factory level in the virtual world maintain shadow images of artefacts in the related levels of the other two worlds to represent virtual images or setting for these artefacts and to run what-if scenarios. The cloud level in each of the three worlds provides services and software tools to support the processes in the respective worlds. We will come back to this later.

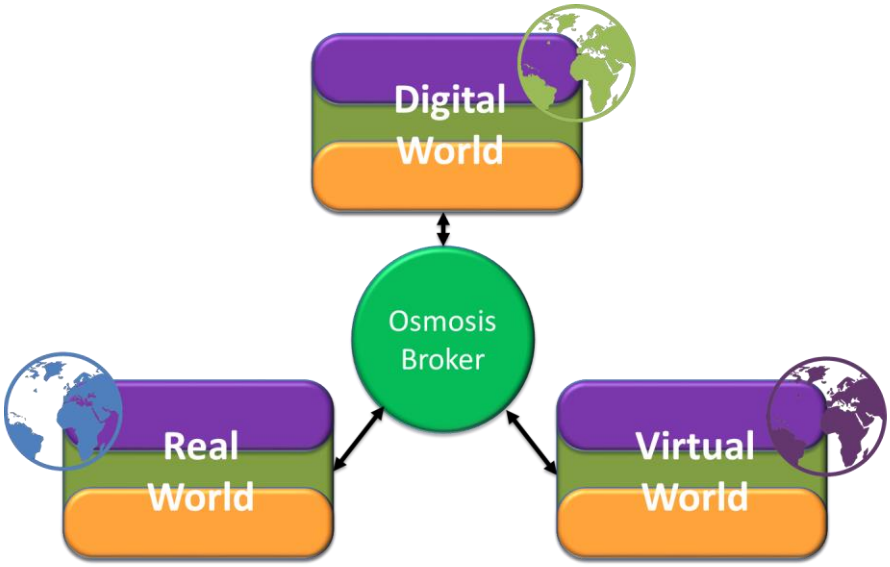


Figure 23: Simplified OSMOSE Architecture with BEinCPPS Integration

Figure 23 presents a simplified view to the OSMOSE architecture. The osmosis broker is the glue between the worlds and allows them to exchange and access relevant information. We assume that events pass between the worlds and that the osmosis broker allows the components living in any of the three worlds to access information which is necessary to access the information which is needed for the correct interpretation of the events and to deduce appropriate actions in response to the events. With this we already established links between the worlds. We already explained above that each world is structured according to a three-layered architecture and in each of the worlds the cloud layer provides services and software tools to support the processes in the respective world. The infrastructure on which the clouds are running does not need to be exclusive for any of the respective worlds. The separation of the cloud into the three worlds is conceptual.

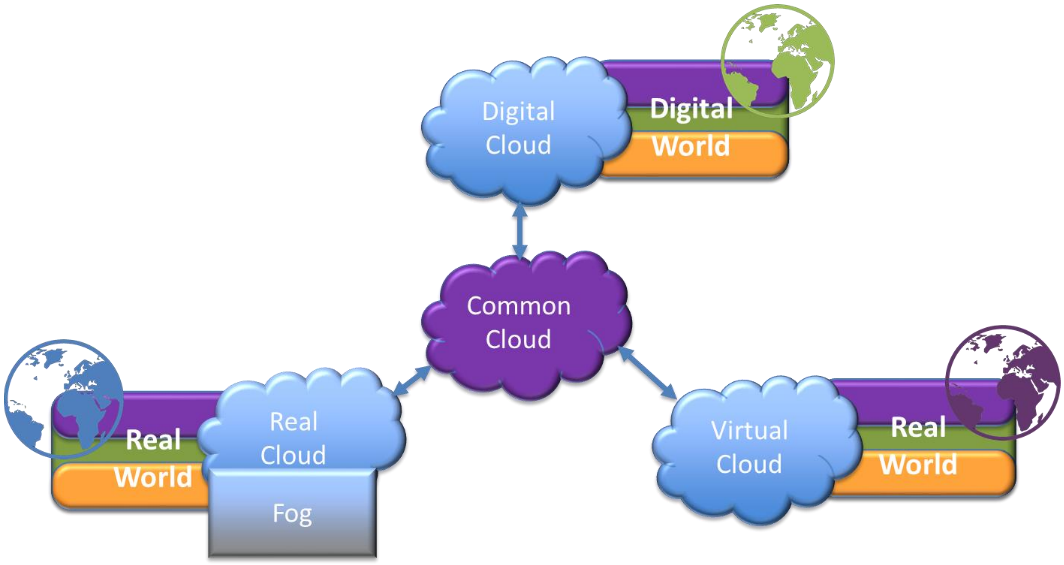


Figure 24: Cloudification of OSMOSE Architecture

Figure 24 displays this idea where we can have for each of the worlds a separate cloud. However, the separation of the clouds does not necessarily mean that the clouds of the different worlds use dedicated infrastructure. The architecture should not put any limits to the use of infrastructure or prescribe how it needs to be used. The separation of the cloud to the different worlds has the idea that we allow the possibility that there are cloud services which are dedicated to a specific world or that data stored in the cloud is kept separate for the different worlds, possibly because of privacy concerns. Of course, we allow, on the other hand, that cloud services and data mingle and with this cross the borders of the different worlds, like the OSMOSE metaphor suggests. Additionally, we can introduce a common cloud which provides generic services and tools which are helpful as provided in all three worlds. For data stored in the common cloud it is not necessary to track from which of the worlds the data was originally introduced. Google Docs, Calendar Services, Virtual Storage, Social Networks etc. are examples of generic services which are widely used. In the real world the cloud is extended with the concept of fog where fog computing is a keyword coined by Cisco[49] for the idea that physical devices can offer services to their environment in an analogous manner as a cloud infrastructure does for its clients.

4.2. Alignment with FITMAN IIOT-RA

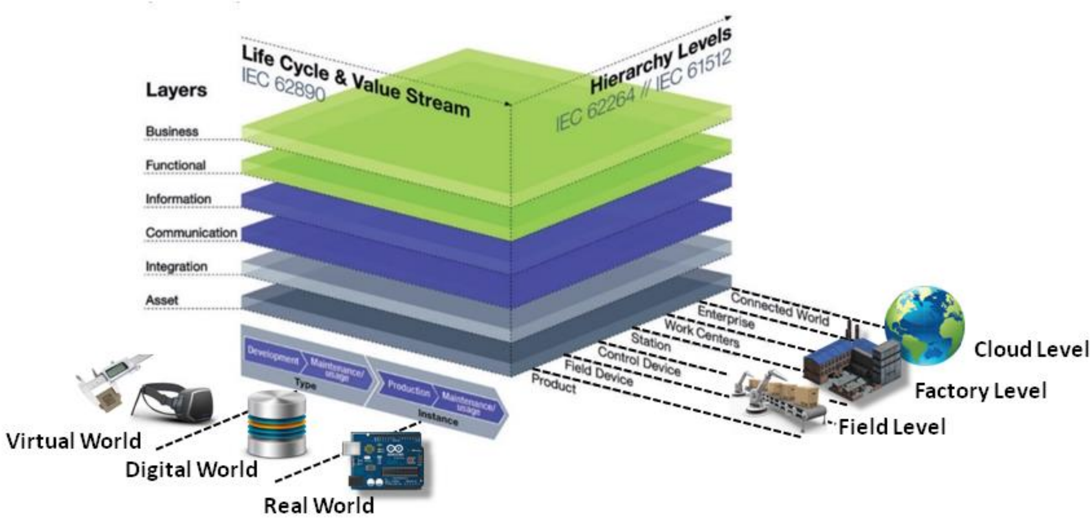
The FITMAN IIOT-RA fits well with BEinCPPS-Arch and with the OSMOSE approach. IIOT-RA is concerned with the integration of the real and the digital worlds, which happens with the mediation of IoT. IIOT-RA’s Smart and Digital domains roughly correspond to BEinCPPS-Arch’s Real World and Digital World, respectively. However, the Virtual domain in IIOT-RA has a completely different scope with respect to BEinCPPS-Arch’s Virtual World: 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 virtual things (i.e., things that do not actually exist in the real world) for the purpose of engineering, simulation and analysis (e.g., what-if scenarios). From BEinCPPS-Arch’s perspective, IIOT-RA’s Virtual domain is just a specialization of the Digital World. 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

[49] http://www.cisco.com/c/en/us/solutions/internet-of-things/iot-fog-computing.html

Osmosis process that is proposed by BEinCPPS-Arch. For these reasons, IIOT-RA was the primary inspiration for the design of BEinCPPS-Platform's modular architecture, that is described in Section 5.

4.3. Alignment with RAMI 4.0 and AIOTI HLA

RAMI 4.0 is an interesting architecture proposal. A visual rendering of how BEinCPPS' Levels and Worlds 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 to BEinCPPS' Levels, while Life Cycle & Value Stream phases have a more blurred correspondence to BEinCPPS' Worlds.



RAMI also has the notion of *I4.0 Components* (see Section 3.3), which can be mapped to different kinds of *BEinCPPS Entities* in the Real, Digital and Virtual World – more on this below.

Another conceptual abstraction that is relevant for BEinCPPS is the High-Level Architecture (HLA) proposed by the standardization working group of the Alliance of Internet of Things Innovation (AIOTI WG03)[50]. It's simple Domain Model (see Figure 25 below), which was originally developed in the scope of the IoT-A project[51], defines how *Things* can be addressed by users of ICT systems through the mediation of devices and services.

[50] http://www.aioti.eu/

[51] http://www.iot-a.eu/public

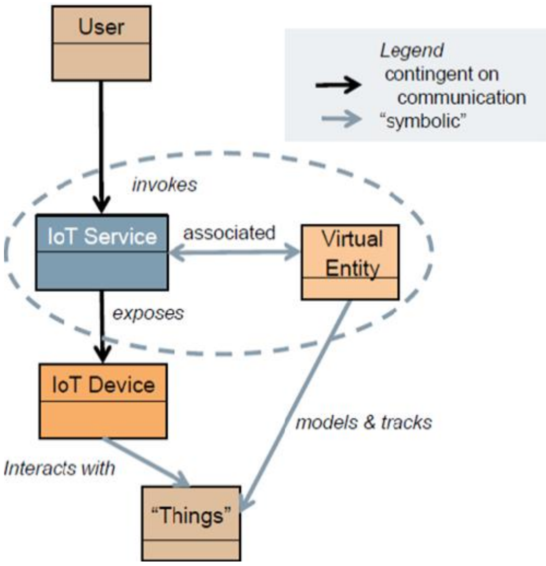


Figure 25 - AIOTI HLA Domain Model[52]

It is quite interesting to see that RAMI's I4.0 Components, AIOTI HLA's Virtual Entities and BEinCPPS-Arch Real World Artefacts 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. This convergence is shown in Figure 26 below. Notably, besides the obvious differences in naming the same concepts (Object vs. Thing vs. Real World Entity), the technical means by which this integration is achieved are different in the three perspectives, AIOTI being entirely IoT-oriented (an so requiring the use of IoT devices as mediators) while the other having a more generic scope (mediators can be any combination of hardware and software). Overall, BEinCPPS-Arch's approach is more flexible but still fully compatible with these frameworks.

[52] Source: public AIOTI WG03 workshop, Brussels 04/11/15

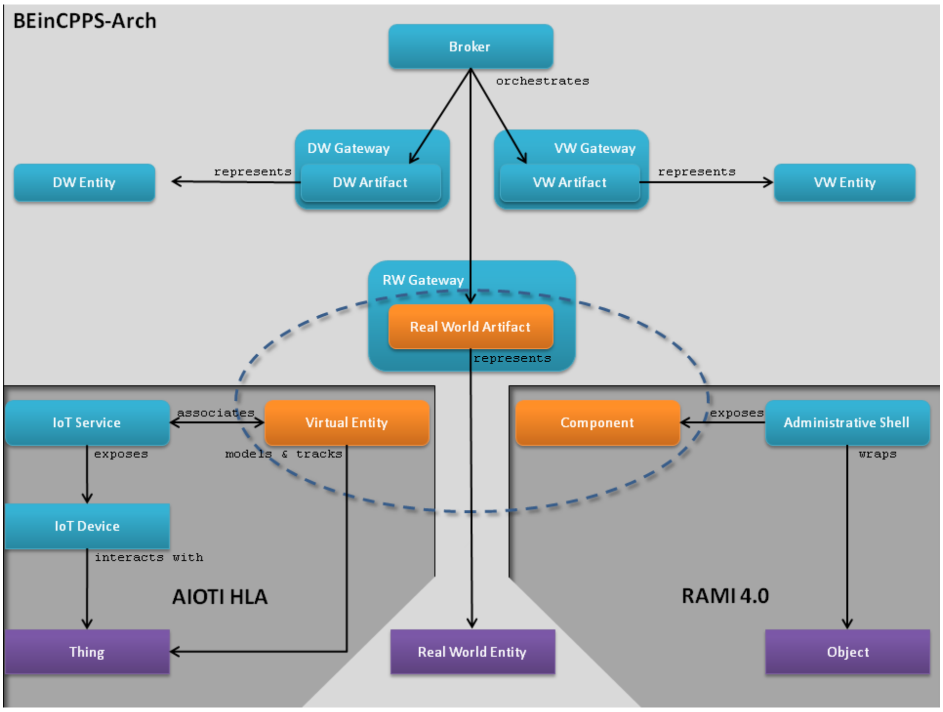


Figure 26 - RAMI vs. AIOTI vs. BEinCPPS approach to Real World digitalization

4.4. Alignment within existing layers

Development within domains such as avionics and automotive is guided by standards as provided by architectures such as (Distributed) Integrated Modular Avionics [11] (IMA) or consortia such as AUTomotive Open System ARchitecture [53] (AUTOSAR). Recently [54], AUTOSAR has been applied to the industrial automation domain as well. In a nutshell, "the AUTOSAR partnership is an alliance of OEM manufacturers and Tier 1 automotive suppliers working together to develop and establish a de-facto open industry standard for automotive E/E architecture which will serve as a basic infrastructure for the management of functions within both future applications and standard software modules."[55] BEinCPPS will consider at what extent can such an architecture be of interest to experiments running within the project.

5. BEinCPPS Modular Architecture

In this section we introduce the first draft of the BEinCPPS-Platform modular architecture. This design is the instantiation of the BEinCPPS-Arch reference architecture (presented in Section 4) as a composition of the most relevant state of the art software assets (described in Section 3) into a *federation of platforms*, according to the project's original concept. This draft provides the blueprint for the internal integration tasks that will be performed within WP2. Some additional software components are also identified here as the *missing links* for achieving full internal integration, and may be developed in the scope of WP2. This initial design is going to be refined in a second release of this deliverable (due by M21), following the first deployment of BEinCPPS-Platform to the five Champions' sites and taking into account lessons learned during the first run of experimentations. In particular, the final selection of state of the art assets that will compose the final platform federation is going to

[53] http://www.autosar.org/

[54] http://www.es.mdh.se/projects/305-AUTOSAR_for_Multi_Core_in_Automotive_and_Automation_Industries

[55] http://www.autosar.org/about/basics/background/

come from hands-on experience, and may differ from what is currently presented here. Figure 27 below is an overview of such architecture.

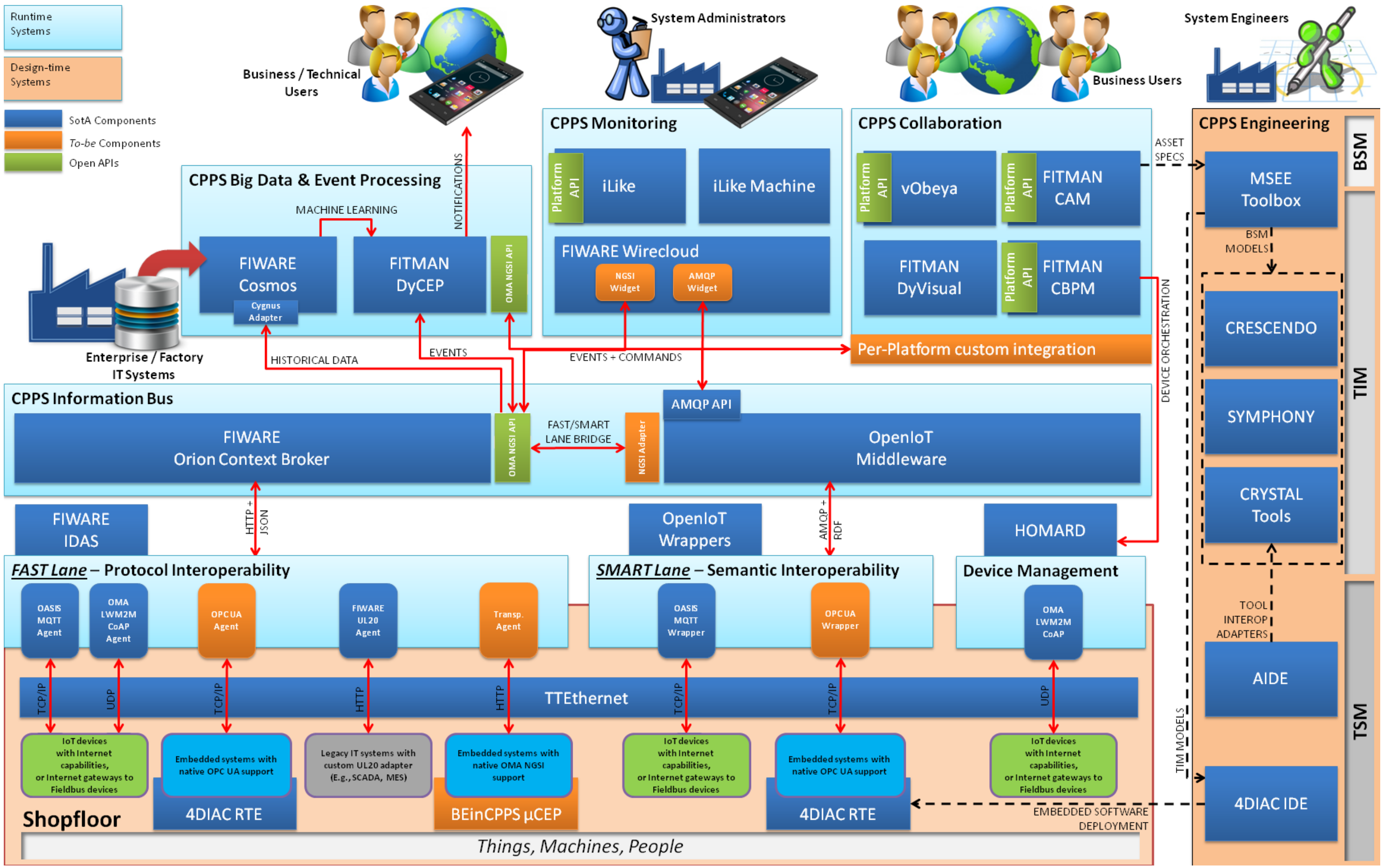


Figure 27 - BEinCPPS-Platform overview

In the above figure, components are *not* grouped according to their World / Level membership: focus is on functional relationship and data flow. For this reason, grouping has been used here to identify common *topics* that are covered by BEinCPPS-Platform. At the highest level, the overall system is divided into a *runtime* sub-system and a *design-time* one – the former consisting of components used to integrate CPPSs and applications, the latter being about system engineering environments and tools. This distinction is stressed in the diagram by different background colours of the grouping boxes (see the legend in the upper-left corner). Within the runtime sub-system, components are further classified by scope: from bottom to top, these are Shopfloor, Interoperability, Information Bus, Big Data & Event Processing, Monitoring and Collaboration.

5.1. Runtime Sub-System: the Shopfloor

The Shopfloor scope is the lowest-level approach in BEinCPPS, encompassing both hardware and software assets. Embedded Systems (field computation) and Real-Time Networks (field communication) are the two topics covered by BEinCPPS-Platform. To the former belongs BEinCPPS μCEP: a down-sized porting of the FITMAN DyCEP software component (described below in the scope of CPPS Big Data & Event Processing) that will run[56] on Linux-capable boards (e.g., RaspberryPI[57]) and network appliances. The task of μCEP is to pre-process events in close proximity to their source, avoiding network latency and thus enabling a first level of true real-time control. Thanks to its native support to the NGSI protocol, μCEP-powered boards will be able to integrate themselves directly with the upper layers of BEinCPPS-Platform (see below in Information Bus area) without the need of interoperability mediation. Another asset included in the Embedded Systems topic is the 4DIAC Runtime Environment (4DIAC RTE)[58]: a portable implementation of the IEC 61499 environment for embedded control devices (see Section 3.6.3). Finally, field communications are addressed by Time-Triggered Ethernet (TTEthernet)[59], a deterministic Ethernet[60] implementation (see Section 3.6.4)

5.2. Runtime Sub-System: Shopfloor Interoperability/Management and CPPS Information Bus

Right above the Shopfloor, the next two groups of components are those responsible for representing Things, Machines and People (i.e., *CPPS actors*) as Real World Entities in the BEinCPPS-Platform: the Interoperability/Management and Information Bus layers. Together they create two simple access paths to the heterogeneous environment of the Shopfloor: Interoperability/Management adapters supporting some of the most popular communication protocols for IoT and industrial automation – namely OASIS MQTT, OMA LwM2M/CoAP and OPC UA – integrate devices and embedded systems with a two-sided Information Bus. These layers adopt a *two-lane* approach, leveraging and extending an original concept from the FW4I IIOT-RA (see Section 3.2).

The Smart Lane exploits OpenIoT[61] assets. It applies semantic annotation on data streams, using a domain-specific Reference Ontology. Enriched data can then be easily exchanged

---

[56] As opposed to FITMAN DyCEP, BEinCPPS μCEP is not a state of the art asset: its development is planned in the scope of WP2.

[57] https://www.raspberrypi.org/

[58] The 4DIAC Runtime Environment is also known as FORTE - see http://www.eclipse.org/4diac/en_rte.php

[59] https://www.tttech.com/technologies/deterministic-ethernet/time-triggered-ethernet/

[60] See http://www.ieee802.org/1/pages/tsn.html and http://www.ieee802.org/802_tutorials/2012-11/8021-tutorial-final-v4.pdf

[61] Linked Sensor Middleware platform - http://open-platforms.eu/library/deri-lsm/

across loosely-coupled consumers – i.e., applications and services that were not specifically designed for a given ICT environment. Such *data + metadata information packets* travel on the network as the RDF[62] payload of AMQP[63] messages, which are queued and eventually dispatched to subscribers by a message broker: AMQP is, to all effects and purposes, the northbound API exposed by the Smart side of the Information Bus. It is worth noting that, while the advantages of the semantic annotation approach are obvious with respect to integration, this pattern is inherently asynchronous and not always suitable for near-real-time process control – even when the message broker implementation is extremely efficient and scalable[64].

In order to accommodate critical processes having very strict timing requirements, the Smart Lane is flanked by a Fast Lane. The Fast Lane is based on FIWARE technology[65], as in the original IIOT-RA: it trades the ease of integration and the advanced capabilities of the Smart Lane for a simpler and lighter middleware, that requires a tighter coupling of applications with their target environment but delivers a synchronous, low-overhead communication channel. *Raw data* is packaged in compact JSON[66] format, and travels through the network as HTTP request/response interactions. On the Fast side of the Information Bus, both the southbound and the northbound APIs conform to the same OMA NGSI standard with a REST-over-HTTP binding.

Both Lanes are equally supported by a Device Management layer based on the HOMARD platform[67], which also provides *sensor orchestration* capabilities by means of an IoT-specific extension of the FITMAN CBPM component - see the CPPS Collaboration section below for additional information.

Whatever the path followed by Shopfloor data, the Information Bus is the pivotal point between business logic (applications and services) and physical processes. While serving different purposes, the two Lanes are however *bridged* at the Information Bus level: the Smart Lane and the Fast Lane can mutually exchange data, thanks to an NGSI adapter on the Smart side. Beyond the Information Bus, BEinCPPS-Platform defines three distinct areas in which to provide support to applications: Big Data & Event Processing, Monitoring and Collaboration.

5.3. Runtime Sub-System: CPPS Big Data & Event Processing

The CPPS Big Data & Event Processing area is served by a couple of closely cooperating components: FIWARE Cosmos[68] and FITMAN DyCEP[69]. The latter is the central processing

---

[62] The Resource Description Framework standard from W3C – https://www.w3.org/RDF/

[63] The Advanced Message Queuing Protocol standard – https://www.amqp.org/

[64] RabbitMQ is the open source message-oriented middleware adopted by OpenIoT - https://www.rabbitmq.com/; see also this Blog article reporting on record performance achieved by RabbitMQ on Google's Cloud: https://blog.pivotal.io/pivotal/products/rabbitmq-hits-one-million-messages-per-second-on-google-compute-engine

[65] FIWARE IDAS Device Manager (http://catalogue.fiware.org/enablers/backend-device-management-idas) and the FIWARE Orion Context Broker (http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker) Generic Enablers

[66] The JavaScript Object Notation standard - http://www.json.org/

[67] The OMA Resources Dashboard - https://homard.hopu.eu/

[68] The FIWARE Cosmos Generic Enabler (http://catalogue.fiware.org/enablers/bigdata-analysis-cosmos) is a Hadoop-as-a-Service engine with a dedicated connector to Orion Context Broker (see previous paragraph on Information Bus) and a custom administration web UI.

[69] FITMAN Dynamic Complex Event Processing Specific Enabler - http://www.fiwareforindustry.eu/: navigate to Lab / FIWARE Enablers for Smart Factories / DyCEP

hub for all machine-to-machine data streams, thanks to its advanced *complex event online processing* capabilities. DyCEP also has built-in *mobile push notification* functionalities, allowing for an easy integration of end users' mobile devices in "situation awareness" scenarios. External application may integrate with DyCEP by leveraging its northbound NGSI-style API, the same way as they do with the Fast Lane Information Bus – so that DyCEP can be effectively considered as alternate, higher-level interface to CPPS that is mediated by application- or environment-specific business logic. On the other hand, Cosmos is an offline data processor: it will analyze huge volumes of *historical data* originating from enterprise information systems and/or coming directly from the Shopfloor, identify new *patterns of interest* and update DyCEP's processing tasks accordingly. This pattern is basically a *machine learning* loop, as the online processing logic can be dynamically and autonomously adapted to changing environments and emerging situations.

5.4. Runtime Sub-System: CPPS Monitoring

The CPPS Monitoring area, as the name implies, is all about low-level monitoring and control of sensors and embedded systems on the Shopfloor. FIWARE Wirecloud[70] is an *application mashup* facility that enables the rapid composition of custom, web-based UIs by assembling multiple *widgets* on a common canvas. Widgets are visual software modules that talk directly with the underlying Information Bus and between each other, each one implementing a related set of monitoring and control functions. Task-specific widgets may be developed in the scope of the BEinCPPS project, according to specific needs of pilot applications. iLike[71] is a *thing lifecycle management* platform, with built-in IoT monitoring capabilities. As opposed to Wirecloud, which is an application-agnostic runtime container for generic modules, iLike is an environment for custom-developed applications targeting specific scenarios. A specific, CPPS-oriented, app created for the monitoring of production machines is the iLike Machine app, that offers an intuitive and customisable visualisation of data collected through sensors and PLCs,

5.5. Runtime Sub-System: CPPS Collaboration

To complete the exploration of BEinCPPS-Platform's runtime sub-system, the CPPS Collaboration area includes four collaborative platforms providing the final users with some generic interfaces for interacting with the whole system at various levels. Virtual Obeya[72] is a web-based meeting place for development and management teams that is conceptually similar to Wirecloud, in that it enables the mashup of different functionality blocks into a single shared environment that is connected to the "live" CPPS world. FITMAN DyVisual[73] is an online collaborative navigator of 3D models that are kept in-synch with modifications happening elsewhere (e.g., live data from the Shopfloor that is represented in XML3D format). FITMAN CAM[74] is the management front-end of a knowledge base of virtualized assets - i.e., OWL2[75] descriptions of any item of interest in the Real World that needs to be digitally represented within the system. CAM also links to the design-time sub-system of BEinCPPS-Platform: its API can be used by engineering tools – in particular by the MSEE

[70] FIWARE Wirecloud Generic Enabler - http://catalogue.fiware.org/enablers/application-mashup-wirecloud

[71] iLike - http://www.holonix.it/en/index.php

[72] Virtual Obeya - http://www.virtualobeya.com/

[73] FITMAN Dynamic Visualizer Specific Enabler - http://www.fiwareforindustry.eu/: navigate to Lab / FIWARE Enablers for Smart Factories / DyVisual

[74] FITMAN Collaborative Asset Management Specific Enabler - http://www.fiwareforindustry.eu/: navigate to Lab / FIWARE Enablers for Virtual Factories / CAM

[75] The Web Ontology Language standard from W3C - https://www.w3.org/TR/owl2-overview/

Toolbox modelling environment (see below) – to access a common online repository of specifications of devices and systems. Finally, FITMAN CBPM[76] is both a development environment (IDE) and an execution engine (RTE) for workflows that are defined in BPMN 2.0 notation[77]. A custom extension to its internal engine[78], provided by the HOMARD project, allows for *orchestration processes* that directly interact with IoT devices on the Shopfloor.

5.6. Runtime Sub-System APIs

As shown in Figure 28 below, the runtime sub-system of BEinCPPS Platform is exposed to Factory- and Cloud-level applications by means of a basic set of Open APIs (represented as green boxes the diagram) that are actually the entry points to some of the platform's components. At the Field level, external systems can link directly to CPPSs by leveraging IoT / automation protocols (light blue boxes). These APIs will be the starting point for the development of the standard API for the Real, Digital and Virtual World of BEinCPPS, which will take place in WP2.

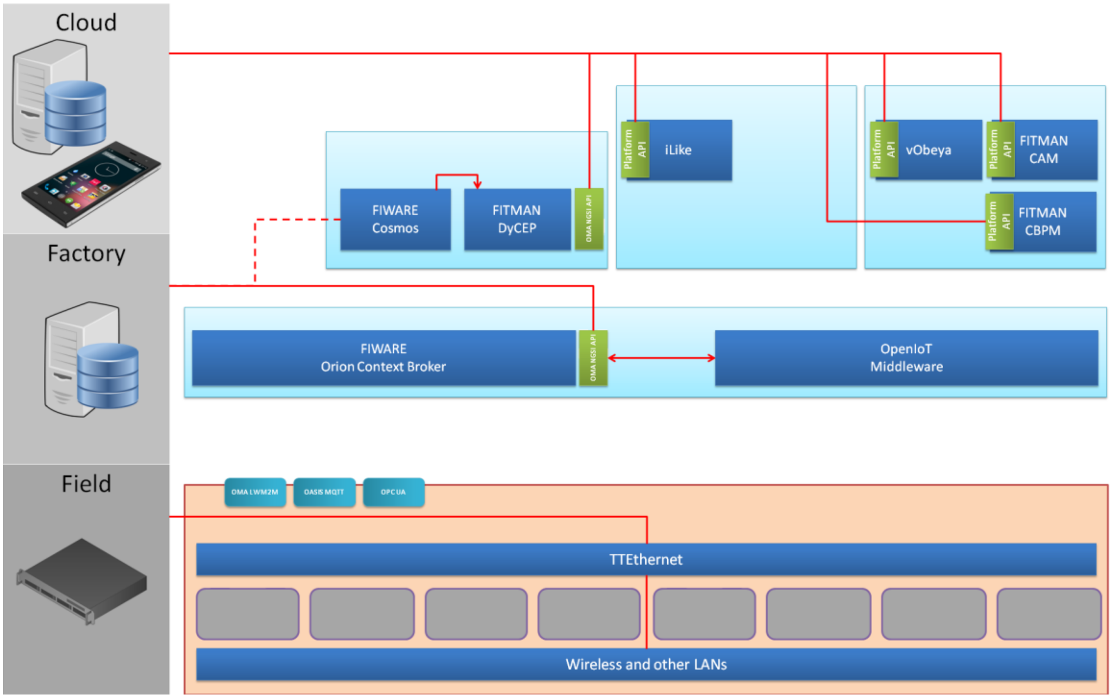


Figure 28 - BEinCPPS-Platform APIs at the Field, Factory and Cloud levels

5.7. Design-time Sub-System: CPPS Engineering

The entire design-time sub-system of BEinCPPS-Platform goes under the label of CPPS Engineering. This is an integrated collection of modules that effectively support system engineering teams in their work of modelling, simulating and developing CPPS at multiple levels: from the individual hardware or software component to enterprise-wide "systems of systems". Besides this common goal, however, different tools have different scopes in BEinCPPS-Platform. Scopes are classified according to a *level of abstraction*, following a three-layered approach that was introduced by the MDSEA methodology[79] for enterprise

[76] FITMAN Collaborative Business Process Management Specific Enabler - http://www.fiwareforindustry.eu/: navigate to Lab / FIWARE Enablers for Smart Factories / CBPM

[77] The Business Process Model and Notation standard from the Object Management Group – http://www.bpmn.org/

[78] FITMAN CBPM is based on the open source Activiti BPM Platform –http://activiti.org/

[79] Model-Driven Service Engineering Architecture, see [20].

interoperability: Business Service Modelling (BSM), Technology-Independent Modelling (TIM) and Technology-Specific Modelling (TSM). Intuitively enough, BSM deals with the structure of business organizations and the goal of business processes, while TIM and TSM are about the design and the implementation of physical processes, respectively. A visual representation of this information flow is depicted in the figure below.

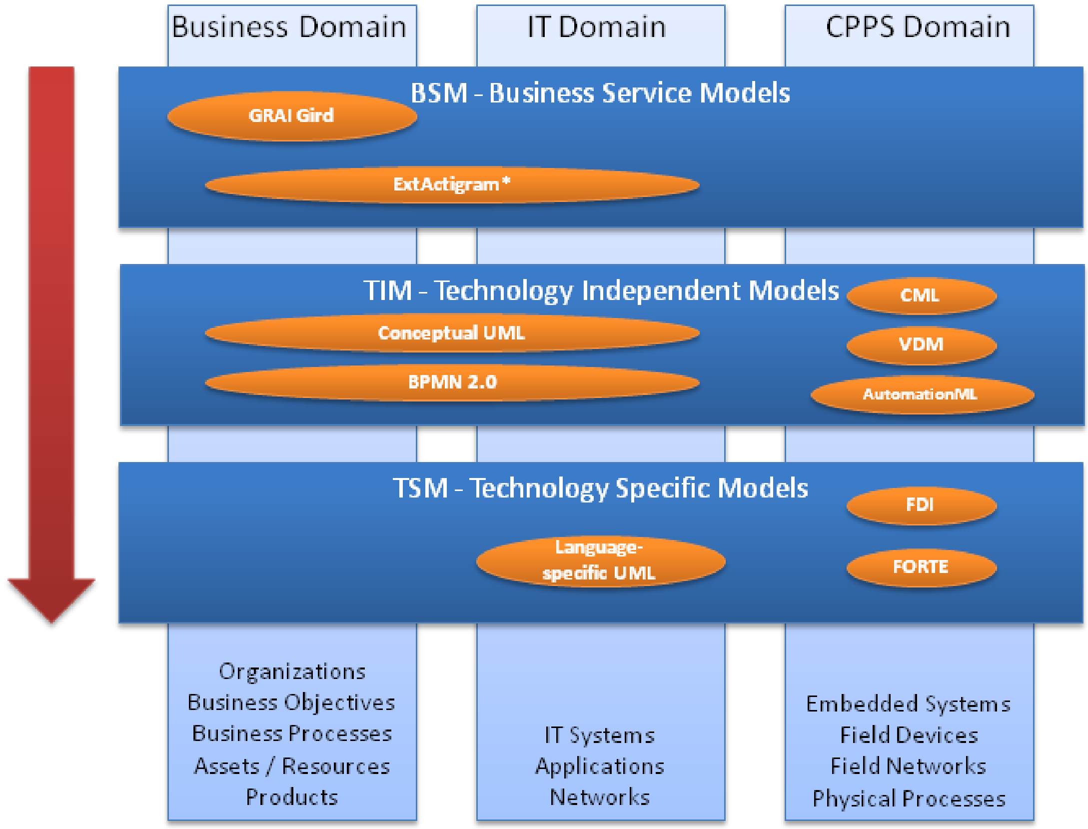


Figure 29 - BEinCPPS Modelling Domains

BEinCPPS-Platform covers all the three scopes with dedicated environments. The MSEE Toolbox is an Eclipse-based[80] IDE for BSM/TIM modelling of *abstract systems* that derives from results of the FP7 MSEE research project[81]. In BeInCPPS it will be extended and specialised to support the design of CPPS. In this tool, the perspective on CPPS is that of business processes, which are modelled in a top-down fashion starting from objectives, assets, actors and decision grids and ending with an *abstract specification* of workflows and of applications/services. The outcome are BSM and TIM artefacts that may drive further steps of the CPPS engineering phase. In particular, the business-level framework defined by BSM artefacts (Extended Actigram and GRAI Grid models) is useful for engineering teams that use TIM-scoped CPPS design and simulation tools, while TIM artefacts (BPMN 2.0 and UML Class diagrams) may be consumed by software developers working at the TSM level – more on this in Section 3.

The TIM scope is covered in BEinCPPS-Platform by two families of tools presented in Sect 3.5.2: the CRESCENDO/SYMPHONY platforms for the CPS and SoS (co)-simulation and the CRYSTAL IOS + RTP ecosystem.

[80] Built upon the Eclipse Platform – http://wiki.eclipse.org/Platform

[81] Manufacturing Service Ecosystem – http://www.msee-ip.eu/project-overview

Whereas CRESCENDO and SYMPHONY (see section #3.5.2) are platforms conceived for modelling and simulation, CRYSTAL (see section #3.6.1) is an entire ecosystem of interoperable tools (Reference Technology Platform, or RTP) provided by various vendors, all supporting a common Interoperability Specification (IOS)[82] and having, on average, a high technology readiness level. At the time of writing (Jan 2016), the actual composition of the CRYSTAL RTP is not entirely defined, but it is expected that the tools will cover mostly modelling, simulation and testing. The AIDE platform (see section #3.5.2), however, will enable interoperability between CRYSTAL tools by means of auto-generated software adapters.

TSM-scoped tools are those that deal with a specific technology, like a programming language, a family of microcontroller boards, or both. In BEinCPPS-Platform this role is played by 4DIAC (see section #3.6.3), which provides a dedicated IDE[83] for the development of IEC 61499-compliant embedded applications. Such applications are deployed to hardware systems that are enabled by the 4DIAC RTE (aka FORTE). Based on the Eclipse platform, the IDE is a complete engineering environment for distributed control applications. The hardware capability definition allows to model the control hardware and its interconnections through networks.

5.8. World, Levels and Modules

Having described in detail how the BEinCPPS-Platform modules fit together, a clarification about their mapping to Worlds and Level might also be useful. As explained in the Reference Architecture discussion (see Section 4), the intersections of Worlds and Levels can be considered as *logical environments* were specific BEinCPPS assets are hosted (see Table 1). The pictures below show how the generic classification of BEinCPPS-Arch is materialized in BEinCPPS-Platform. Assets are also classified by type (hardware, software, other) and by functional scope (e.g., engineering, simulation, interoperability, etc.) using a shape and colour scheme that is explained in the legend. Regarding the Level-wise positioning of assets, please notice that some of them have been represented here as spanning two environments; this is because either the asset is cross-level by nature (e.g., TTEthernet can be both a Field *and* a Factory infrastructure), or its actual position may depend on deployment choices (e.g., components of the Information Bus can run on the Cloud *or* on Factory premises).

[82] http://www.crystal-artemis.eu/fileadmin/user_upload/Deliverables/CRYSTAL_D_601_022_v1.0.pdf

[83] http://www.eclipse.org/4diac/en_ide.php

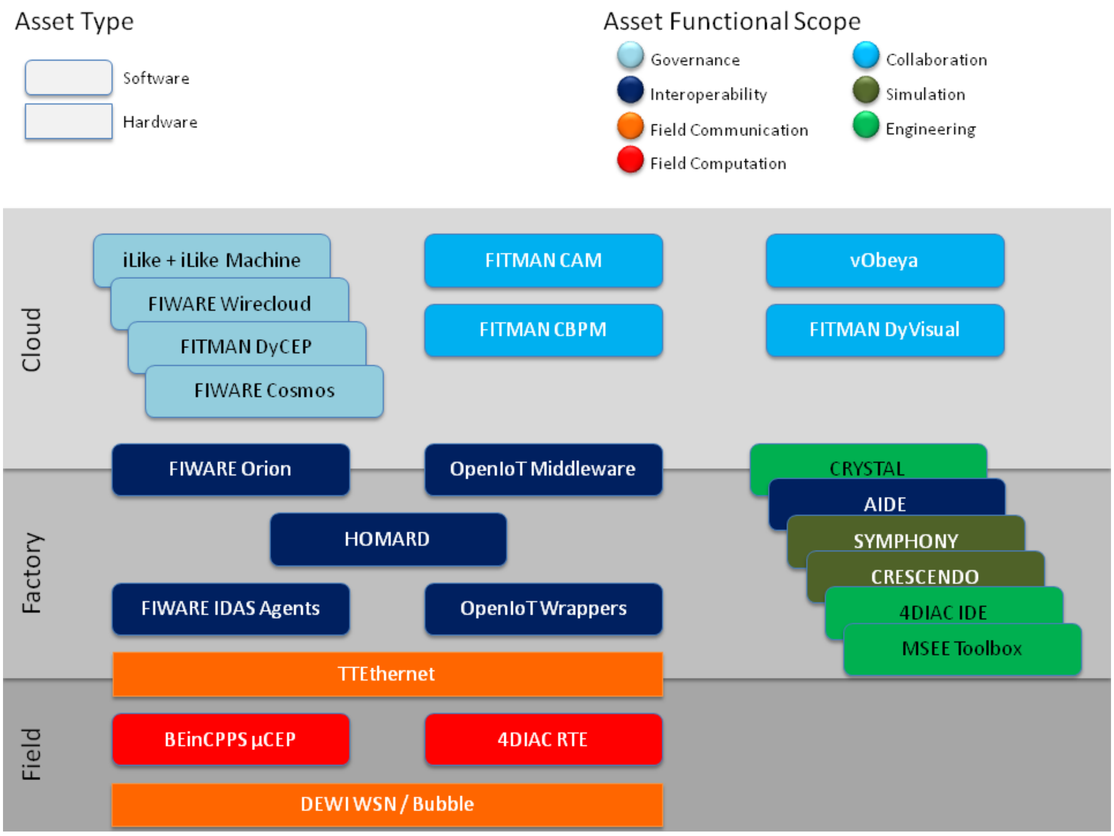


Figure 30 - BEinCPPS-Platform modules classification

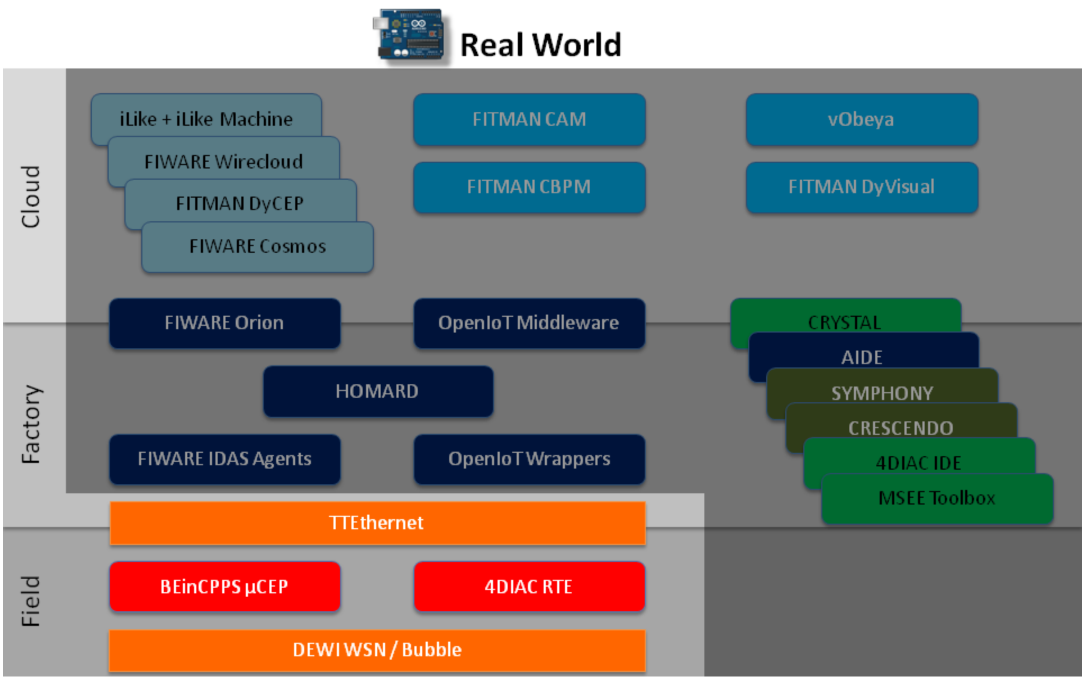


Figure 31 - BEinCPPS-Platform modules in the Real World

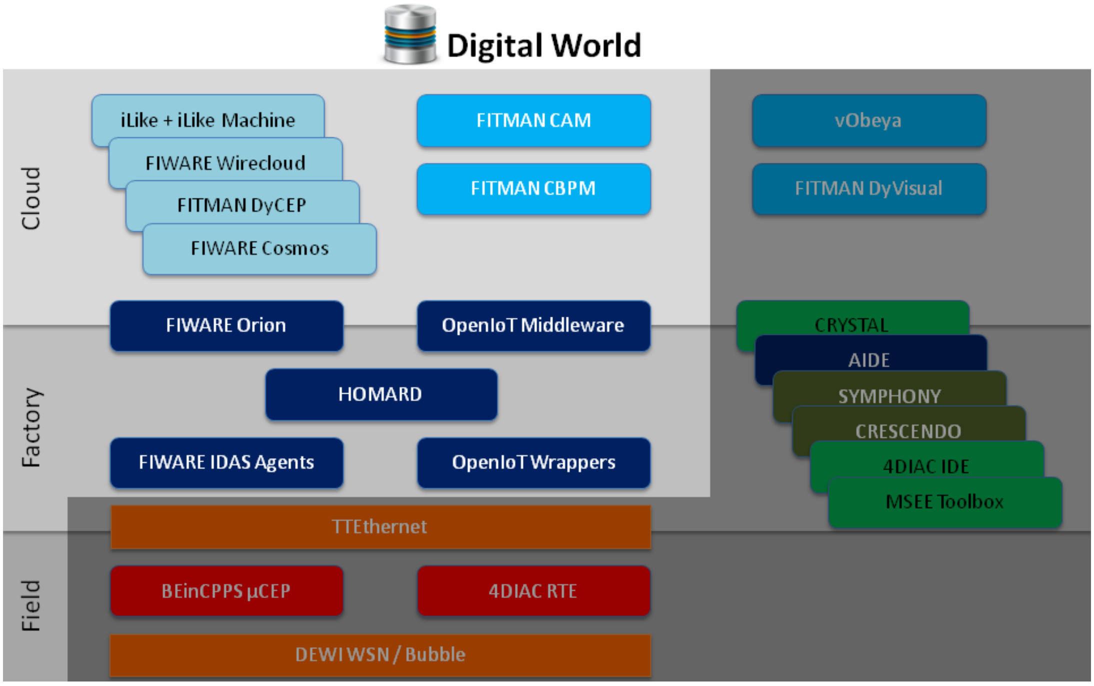


Figure 32 - BEinCPPS-Platform modules in the Digital World

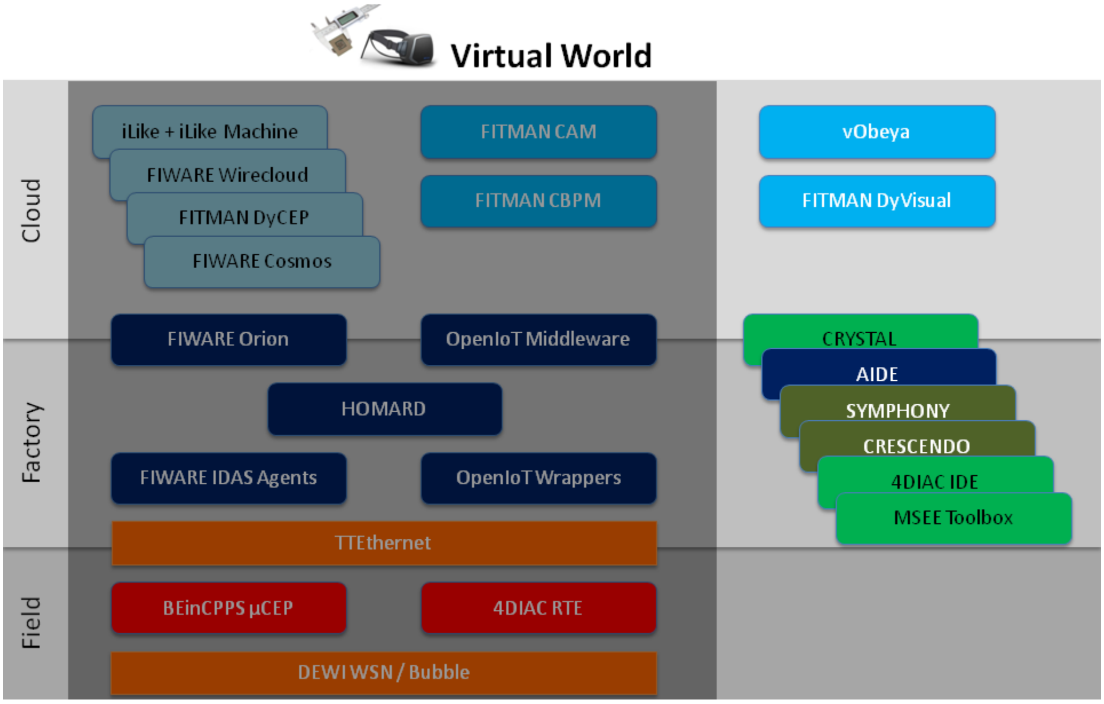


Figure 33 - BEinCPPS-Platform modules in the Virtual World

6. Business processes

At this early stage of the project it is not possible to present details of business processes which will be investigated in the course of the BEinCPPS project because these process are still to be identified. As far as business processes are concerned it is safe to assume that such processes can be described using BPMN and a BPMN business process execution engine can be used to execute such processes. At lower layers (i.e. the factory and the smart systems layer) it is likely that more fine-grained and complex processes are running for which most likely specialized representations are used.

7. Conclusions

In this deliverable we summarized the work on the architecture definition done so far. This D2.1 deliverable is the first version of “BEinCPPS Architecture and Business Processes” and provides the basis for implementation. In a state of the art overview we presented and discussed three major architecture proposals: the OSMOSE Architecture, the FITMAN Industrial IoT Reference Architecture (IIOT-RA), and the Reference Architecture Model for Industry (RAMI) 4.0, as well as concepts for the integration of the CPS layer. In addition to these, state of the art technologies and tools are presented.

The main contribution of D2.1 is the definition of the BEinCPPS Reference Architecture (BEinCPPS-Arch). BEinCPPS-Arch federates the most prominent Smart Systems, IoT and Future Internet platforms. It defines two distinct, orthogonal axes in its layout: the Worlds axis for the Real, Digital and Virtual logical domains, the Levels axis for the Field, Factory and Cloud physical environments. Each World can span the three Levels, and a dedicated module is in charge of realizing a Real-Digital-Virtual connection that is not merely a data exchange between distinct entities (as happens in traditional approaches), but rather is implemented as an osmosis process in which atomic elements of each World can be transferred through semantically-enabled semi-permeable membranes into adjacent Worlds, where they operate as remote agents.

In addition to BEinCPPS-Arch, D2.1 presents the BEinCPPS-Platform modular architecture. This design is the instantiation of the BEinCPPS-Arch reference architecture as a composition of the most relevant state of the art software assets into a *federation of platforms*, according to the project’s original concept. At the highest level, the BEinCPPS Platform is divided into a *runtime* and a *design-time* sub-system – the former consisting of components used to integrate CPPSs and applications, the latter being about system engineering environments and tools. Within the runtime sub-system, components are further classified by scope: from bottom to top, these are Shopfloor, Interoperability, Information Bus, Big Data & Event Processing, Monitoring and Collaboration.

It is clear that the content of this deliverable can only be the starting point for extensive discussions on how the BEinCPPS architecture should evolve. The BEinCPPS-Platform modular architecture as described in this deliverable provides the blueprint for the integration tasks of WP2. The initial design is going to be refined in a second release of this deliverable, following the first deployment of BEinCPPS-Platform to the five Champions’ sites and taking into account lessons learned during the first run of experimentations. In particular, the final selection of state of the art assets that will compose the final platform federation is going to come from hands-on experience, and may differ from what is currently presented here. In the course of this discussion additional architecture proposals might be put on the table and analysed. If applicable, new concepts will be integrated into the BEinCPPS architecture when they are considered helpful. Additional input for the further development of the BEinCPPS architecture will come from the experiments in the five Champions. The second version of “BEinCPPS Architecture and Business Processes” will be released at M21 of the BEinCPPS project as deliverable D2.2 and will present the final BEinCPPS architecture.

References

[1] European interoperability framework (eif) for european public services, December 2010.

[2] Fdi - field device integration technology, 2011.

[3] Whitepaper automationml: Part 1 - architecture and general requirements, October 2014.

[4] Opc unified architecture: Interoperability for industrie 4.0 and the internet of things, 2015.

[5] Carlos Agostinho and Ricardo Jardim-Goncalves. Sustaining interoperability of networked liquid-sensing enterprises: A complex systems perspective. *Annual Reviews in Control*, 39:128–143, 2015.

[6] Carlos Agostinho, Michele Sesana, Ricardo Jardim-Gonçalves, and Sergio Gusmeroli. Model-driven service engineering towards the manufacturing liquid-sensing enterprise. In *MODELSWARD 2015 - Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development*, pages 608–617, ESEO, Angers, Loire Valley, France, 9-11 February 2015. ESEO, Angers, Loire Valley, France, 9-11 February, 2015.

[7] David Chappell. *Enterprise Service Bus*. O'Reilly Media, Inc., 2004.

[8] Unit A3-DG CONNECT. Report from the workshop on cyber-physical systems: Uplifting europe's innovation capacity, December 2013.

[9] Christian Ehler. European forum for manufacturing – european parliament, September 2014.

[10] Thomas Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.

[11] René L.C. Eveleens. Integrated modular avionics development guidance and certification considerations. Technical report, NATO Science and Technology Organization, 2006.

[12] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[13] Cyber Physical Systems Public Working Group. Draft - framework for cyber-physical systems - release 0.8, September 2015.

[14] Joana Hois. *Modular Ontologies for Spatial Information*. Logos Verlag Berlin GmbH, 2013.

[15] Kyoung-Dae Kim and P.R. Kumar. Cyber-physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, May 2012.

[16] David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[17] Brenda M. Michelson. Event-driven architecture: Overview - event-driven soa is just part of the eda story, February 2006.

[18] Michele Missikoff, Yannis Charabilidis, Ricardo Gongcalves, and Keith Popplewell. Future internet enterprise systems (fines) research roadmap 2025, June 2012.

[19] Ovidiu Vermesan and Peter Friess, editors. *Internet of Things – From Research and Innovation to Market Deployment*. River Publishers, 2014.

[20] Martin Zelm, Marten J. Van Sinderen, Luiis Ferraira Pires, and Guy Doumeingts, editors. *Enterprise Interoperability: Research and Applications in Service-oriented Ecosystem (Proceedings of the 5th International IFIP Working Conference IWIE 2013)*. John Wiley & Sons, Ltd., November 2013.