BEinCPPS

Innovation Action Project

HORIZON 2020 - EU.2.1.5.-Ref. 680633

D2.2 - BEinCPPS Architecture & Business Processes

Lead Author: Mauro Isaja (ENG)

With contributions from: Klaus Fischer (DFKI), Domenico Rotondi (FINCONS), Eva Coscia (HLX), Martijn Rooker (TTTECH)

Reviewer: Giacomo Inches (FINCONS)

Deliverable characteristics

Deliverable nature	R
Dissemination level	PU
Contractual delivery date	31 July 2017
Actual delivery date	31 July 2017
Version	1.0
Keywords	

Version history

Nr.	Date	Notes and comments
0.1	05/06/2017	ToC
0.2	26/06/2017	First Input Section 5 (DFKI)
0.3	24/07/2017	Input by ENG, FINCONS, HLX, TTTech, DFKI
0.9	24/07/2017	First consolidated version for peer review
1.0	31/07/2017	Final version

Deliverable Peer review summary

Id.	Addressed	Comments

Table of contents

List of Figures

Executive summary

This document reports about the second and final version of BEinCPPS Architecture and Business Processes. While the first release (D2.1) came very early in the project schedule (M3) and aimed at setting the technical guidelines for IT systems that were going to support CPPS experiments, this new version (M21) has a higher ambition: defining a BEinCPPS-specific approach to system design, which is a key element of the overall BEinCPPS value proposition. To this goal, two main aspects are taken into account: changes in the technology landscape occurred after the first release and lessons learned during the implementation and execution of the CPPS experiments.

Regarding the technology landscape, two relevant technical papers were released that had an impact on BEinCPPS: *What Criteria do Industrie 4.0 Products Need to Fulfil?* from the Platform Industrie 4.0 initiative (ZVEI, 2017) and *The Industrial Connectivity Framework* from the Industrial Internet Consortium (IIC-CTG, 2017). From both sources, BEinCPPS derives the commitment at supporting OPC UA as the main connectivity technology for shopfloor equipment; the second also paves the way to the introduction of MQTT as the BEinCPPS standard for a data-agnostic transport protocol in the implementation of a Layered Databus architecture pattern.

Lessons learned stemmed mainly from hands-on experience in the development and integration of solutions for BEinCPPS Champions. To summarize, the first version of the BEinCPPS architecture was perceived as a rather *artificial* scheme. In particular, the Virtual World abstraction proved itself to be difficult to grasp and the Osmosis pattern was never actually exploited. The first design of the BEinCPPS platform also was overly complicated and sometimes confusing. To address these concerns, in this second release we adopt a revolutionary approach: one single vision built as progression of perspectives – i.e., Structural, Functional, Technical and Implementation – which start from a bare bone high-level design and then incrementally add up to the full picture. Moreover, some concrete examples of business processes implemented by BEinCPPS pilot sites are provided, using a convenient notation (MDSEA) that highlights the use of specific BEinCPPS components (from the Implementation Perspective) to support individual process tasks.

Finally, this document concludes by explaining the next steps in the roadmap towards the implementation of the BEinCPPS integrated platform and clarifies the relationship of such platform with external communities in the scope of Future Internet, IoT and Smart Systems.

1 Introduction

This document reports about the second and final version of BEinCPPS Architecture and Business Processes. While the first release (D2.1) came very early in the project schedule (M3) and aimed at setting the technical guidelines for IT systems that were going to support CPPS experiments, this new version (M21) has a higher ambition: defining a BEinCPPS-specific approach to system design, which is a key element of the overall BEinCPPS value proposition.

The two main drivers of this work, which lasted eighteen months, where A) changes in the state-of-the-art of relevant technologies and B) lessons learned during the implementation and execution of the CPPS experiments. The former are reported in §2, the latter in §3.

The main part of this deliverable is however §4: the BEinCPPS Architecture. As explained at the beginning of that section, it's a substantial evolution of the BEinCPPS-Arch described in the first version.

Section §5 provides some real-world examples of the Model-Driven Service Engineering Architecture approach and of the use of BEinCPPS assets in the context of concrete use cases.

Finally, §6 explains the next steps in the roadmap towards the implementation of the BEinCPPS integrated platform and clarifies the relationship of such platform with external communities in the scope of Future Internet, IoT and Smart Systems.

1.1 Scope of the deliverable

This deliverable is the final outcome of task WP2.1, which is in charge of customizing the three-layered Machine / Factory / Cloud vision of BEinCPPS for use by the five Champions. The task is led by DFKI, with the active involvement of all technical partners and of the five competence centres.

1.2 Contributions to other WPs

This D2.2 deliverable is going to affect activities in the scope of the five Champion-related WPs: WP4 Lombardy, WP5 Basque Country, WP6 Baden-Württenberg, WP7 Norte and WP8 Rhône-Alpes.

1.3 Contributions to other deliverables

All prototype deliverables in the scope of WP2 and WP3 are directly affected by the BEinCPPS Architecture specification. In particular, D2.2 will contribute to D2.4 IoT Platform Federation, D2.6 FI Platform Federation, D2.8 Smart Systems Platform Federation, D3.2 Real World BEinCPPS Components, D3.4 Digital World BEinCPPS Components, D3.6 Virtual World BEinCPPS Components (all released by M24) and finally to D2.10 Integrated BEinCPPS Platform Federation (M27).

2 State of the Art

This section briefly reports about news and changes in the technology landscape that is relevant for BEinCPPS. The baseline is the SotA analysis as reported in the first version of this deliverable. At the time, we explored several sources as candidate *influencers*. Among them, research projects and initiatives (OSMOSE, FITMAN, OpenIoT, CRYSTAL, ARTEMIS, CPSE Labs), standards (OPC UA, IEC-61499), technologies (TSN, WSN), approaches to system design (MDSEA), reference architectures (RAMI 4.0, AIOTI HLA, FITMAN IIoT-RA, Osmosis). Some of these sources actually provided assets for the building of BEinCPPS systems, in particular FITMAN, OpenIoT, OPC UA, IEC-61499, TSN, WSN, MDSEA, RAMI 4.0.

During the last eighteen months, the only significant news came from two sources: the Industrie 4.0 initiative and the Industrial Internet Consortium. In the following sections we provide details on both.

2.1 Industrie 4.0

In the Industrie 4.0 landscape, RAMI 4.0[1] was the most notable feature at the time when the first version of this report was delivered (January 2016). While RAMI 4.0 was a useful conceptual framework for presenting BEinCPPS features to stakeholdes (see §4.5), it didn't provide much in terms of concrete technology guidelines for the implementation of Industrie 4.0-compliant systems. The need for such guidelines, however, was a clear message sent from the industry. In response to this, starting from November 2016, new documents were made publicly available (ZVEI, 2017) that define the minimum criteria "products" – i.e., industrial equipment – should meet in order to be "Industrie 4.0 Ready". This was the second step of a 5+ years roadmap towards a complete standardization of Industrie 4.0 systems, as shown in Figure 1.

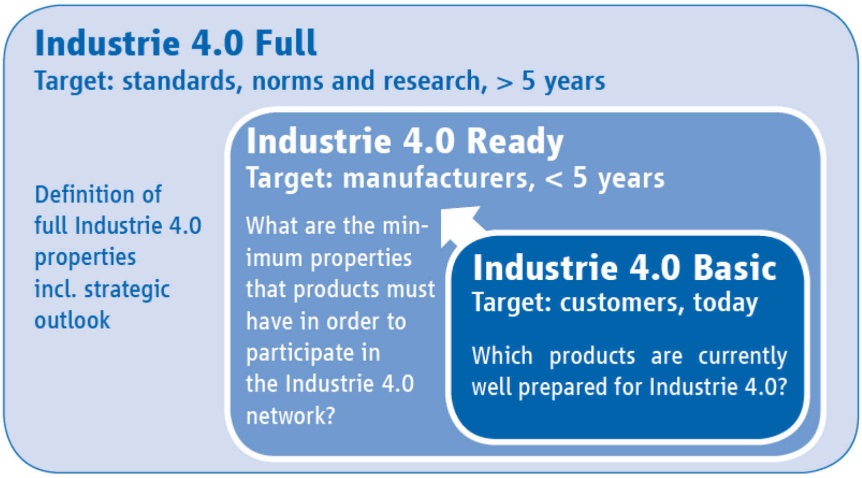

Figure 1 - Industrie 4.0 standardization roadmap (source: ZVEI)

[1] https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf

Previously, the concept of Industrie 4.0 Ready was blurred. Now, after the latest updates of the documentation (April 2017), the properties of a compliant product are those described in the table from Figure 2.

	Criteria	Requirements	L	C	Product properties 2017
1.	Identification	Cross-manufacturer identification with unique identifier (ID) attached to the product, electronically readable. Identification in: 1) Development 2) Goods transport (logistics), production 3) Sales, service, marketing 4) Network	T	M	For 1) material number[1] (electronic) in accordance with ISO 29002-5[2] or URI
			I	M	For 2) serial number or unique ID For 3) manufacturer + serial number or unique ID With 2) and 3) electronically readable, physical products via 2D code or RFID For 4) participant identification via IP network
2.	Industrie 4.0 communication	Transfer of product data and data files for interpretation or simulation, for example; product data in standardised form The product can be addressed via the network, supplies and accepts data, Plug & Produce via Industrie 4.0-compliant services	T	M	Manufacturer makes data that is relevant for the customer available/accessible online with the aid of identification, e.g. PDF via http(s)
			I	M	Product addressable online via TCP/UDP&IP with at least the information model from OPC-UA
3.	Industrie 4.0 semantics	Standardised data with manufacturer-independent unique identification in the form of characteristics with a syntax for, e.g.: 1) Commercial data 2) Catalogue data 3) Technical data: mechanics, electronics, functionality, location, performance 4) Dynamic data 5) Data regarding the lifecycle of the product instance	T	M	Catalogue data can be accessed online
			I	M	Catalogue data and data regarding the lifecycle of the product instance can be accessed online
4.	Virtual description	Virtual representation in Industrie 4.0-compliant semantics Virtual representation across the entire lifecycle. Characteristic attributes of the actual component, information regarding relationships between the attributes, production and production process-relevant relationships between Industrie 4.0 components, formal description of relevant functions of the actual component and its processes	T	M	Relevant information for customers can be accessed digitally based on the type identification (product description, catalogue, image, technical features, data sheet, security properties, etc.)
			I	M	Digital contact to service and information for product support incl. spare part information possible from in the field
5.	Industrie 4.0 services and conditions	Definition still open (service system) General interface for loadable services and messages regarding statuses Essential basic services that an Industrie 4.0 product must support and provide	T	O	Description of the device interface available digitally
			I	O	Information such as statuses, error messages, warnings, etc. available via OPC-UA information model in accordance with an industry standard
6.	Standard functions	Basic standardised functions that run on various products regardless of manufacturer and provide the same data in the same functions. These serve as the foundation for the functionality, on which all manufacturers can build their own enhancements.	T	N	Not defined
			I	N	Not defined
7.	Security	Minimum requirements to guarantee the security functionality.	T	M	A threat analysis has been performed. Appropriate security capabilities have been considered and publicly documented.
			I	M	The existing security capabilities are documented. Appropriately secure identities are available.

Figure 2 - I4.0 Product Properties (source: ZVEI)

The most important element from the BEinCPPS perspective is line #2 *Industrie 4.0 communication*: it mandates (C=M) that connected *instances* of a product (L=I) are addressable on an Internet-like network as an OPC UA server, while information on the product *type* (L=T) must be made available online by the OEM. This again is stressing the importance of the support for the OPC UA protocol and information model in BEinCPPS. However, it is also worth noting that, according to same document, an Industrie 4.0 product integrated into an Industrie 4.0 network is actually an *Industrie 4.0 Component*. Components, in the more advanced vision of "Industrie 4.0 Full", are much more than connected equipment, and the software wrapper – dubbed *Administration Shell* – will be required to integrate at the information and functional levels, as depicted in Figure 3. This however is still a work in progress, documented by working documents that are made public on a regular basis (BMWi, 2017).

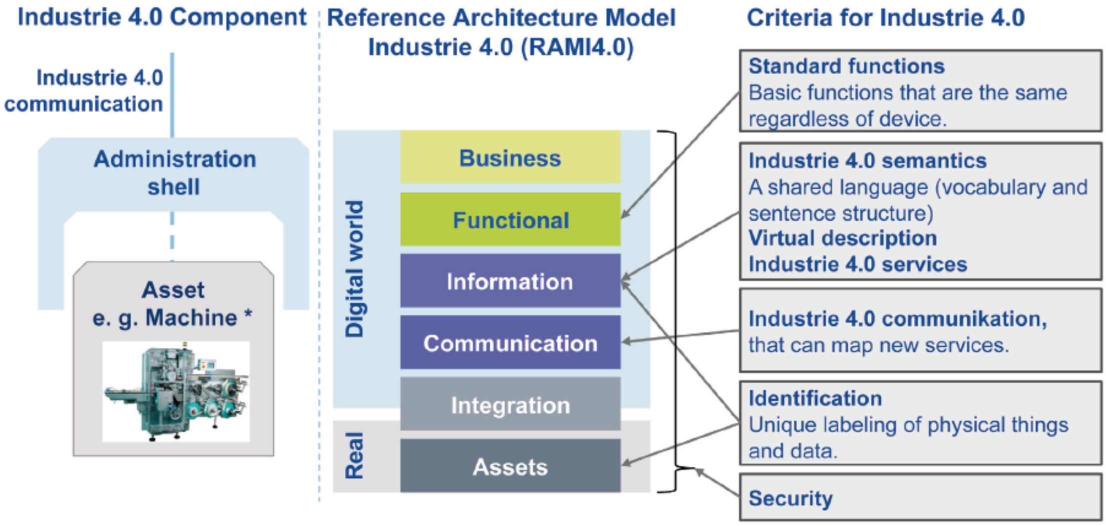

Figure 3 - Industrie 4.0 Component vs. RAMI 4.0 (source: Bosch Rexroth)

2.1.1 Industrie 4.0 and Industrial Data Space

An additional German initiative, fostered by the Fraunhofer Institute, is the *Industrial Data Space* (IDS)[2]. This initiative aims at exploiting the potential value of the data related to the industrial activities via the creation of a *virtual data space* that, thanks to existing standards, technologies and governance models, supports the secure exchange and correlation of data in a trusted business ecosystem fostering the creation of smart services and innovative business processes, while at the same time ensuring data sovereignty for the participating data owners.

IDS is an initiative that has a wider scope as compared with the Industrie 4.0 one, even if it has the potential to further expand and exploit the potential of the latter initiative. To this end a strict cooperation between these two initiatives has been activated, also fostered by the German government.

2.2 IIRA

The Industrial Internet Reference Architecture (IIRA) is a standards-based architectural template from the Industrial Internet Consortium (IIC)[3], a global initiative of business, government and research organizations. The first public release of the IIRA technical paper was in June 2015, and the IIRA itself was assessed in the first three months of the project as a candidate influencer. At the time, it was deemed as less relevant than the other candidates, and was not included in the first version of this deliverable.

[2] Industrial Data Space Association, "Reference Architecture Model for the Industrial Data Space", April 2017

[3] http://www.iiconsortium.org/

However, as part of the major overhaul of the BEinCPPS architecture that was done as a reaction to internal user feedback (see §3), some concepts from the latest version of the IIRA find their way into the final design. Of particular interest for the project was the concept of a *Layered Databus*: an architecture pattern that is part of the Implementation Viewpoint of IIRA (IIC-ATG, 2017). According to this design, an IIoT system can be partitioned into a hierarchy of logical layers, each of them defined by their *scope* – i.e., smart machines (machine scope), systems (workstation/line scope), systems of systems (plant/factory scope), Internet (global scope). Within each layer, a *databus* – i.e., a middleware system that supports loosely-coupled M2M communication adopting a commonly-agreed data model – implements horizontal connectivity, while different layers are connected by means of *adapters* that are able to route and transform data messages. This kind of approach is depicted in Figure 4 below.

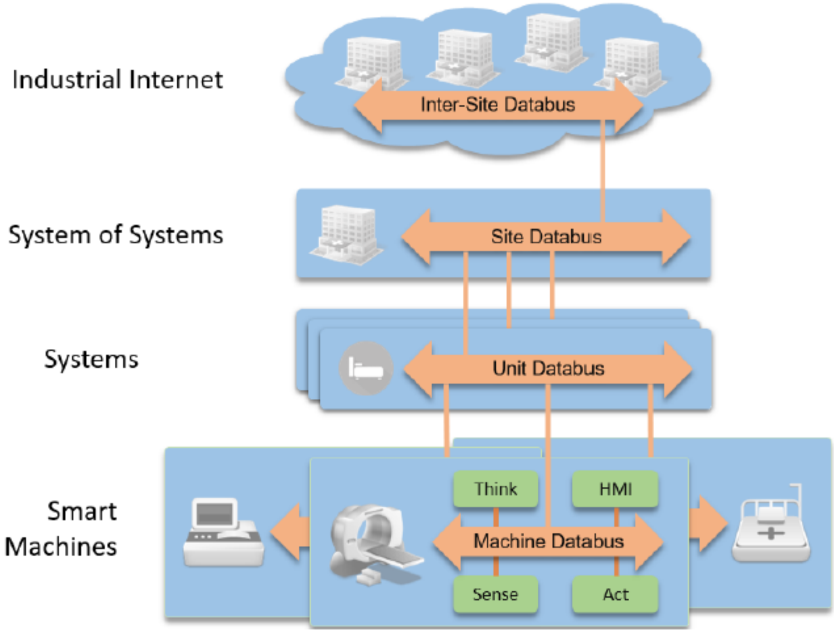

Figure 4 - IIRA Layered Databus Architecture (source: IIC)

The Layered Databus pattern has been partially adopted in BEinCPPS: the Information Bus functional block of the BEinCPPS architecture (see §4.2) plays the role of the systems of systems layer of the IIRA, in the context of a loosely-coupled horizontal communication between Cloud-based applications and services; the Field Gateway functional block does the same at the systems level, connecting shopfloor devices horizontally, and vertically (i.e., towards the Information Bus) by means of an adapter (see Figure 10).

Moreover, in February 2017 a new technical document was released by the Connectivity Task Group of the IIC: the Industrial Internet Connectivity Framework (IICF). It defines a connectivity reference architecture for data sharing, and even more importantly, identifies some *core connectivity standards* (IIC-CTG, 2017). These are DDS, HTTP/Web Services, OPC UA and oneM2M for what concerns the "frameworks", and TCP, UDP, CoAP and MQTT as "transport protocols" – all of them relying on IP as the their basic networking technology (see Figure 5). Again, OPC UA is considered the emerging communication standard in the specific domain of manufacturing. MQTT is correctly classified as a transport protocol rather than a framework, as it does not define a data type system of its own: MQTT messages have a data payload that is application-specific. For this reason, MQTT support in the BEinCPPS Field Gateway functional block is important as an enabler of the IIRA's Layered Databus pattern at the Field level.

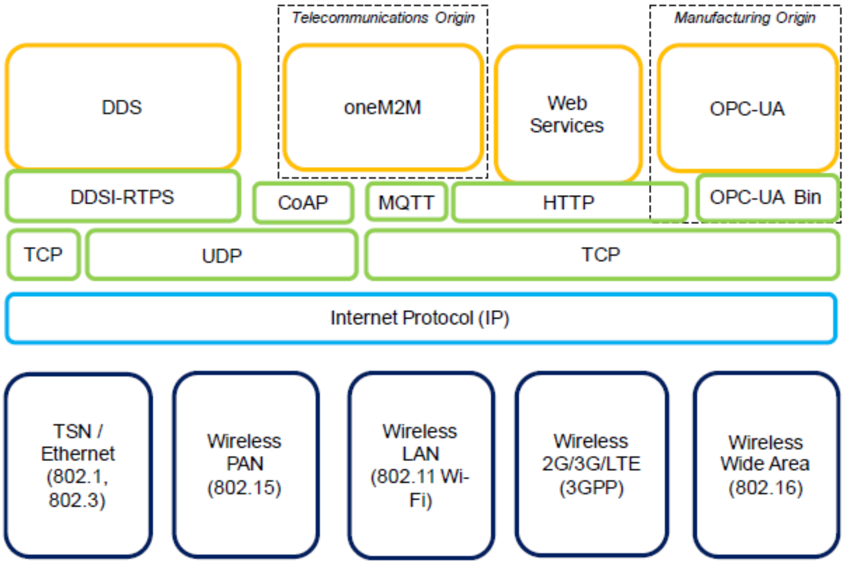

Figure 5 - IICF connectivity standards

3 Lessons Learned

To better explain the evolution of the BEinCPPS architecture during the last fifteen months, a short recap of the starting point is required.

The first version of the BEinCPPS architecture was an early effort (M3) at setting the *IT context* for CPPS experiments to be run by the five industrial Champions. It provided stakeholders with two separate viewpoints over the project's IT landscape: a simple conceptual framework, the BEinCPPS Reference Architecture (BEinCPPS-Arch), and an implementation-oriented platform design (BEinCPPS-Platform) which described a complex, although generic, system built along the guidelines of the framework.

BEinCPPS-Arch was inspired by previous work – e.g., the OSMOSE project and its *Osmosis* architectural pattern – and basically laid out a *grid* to classify the components of a complex system according to *Levels* and *Worlds*, as demonstrated by the picture below.

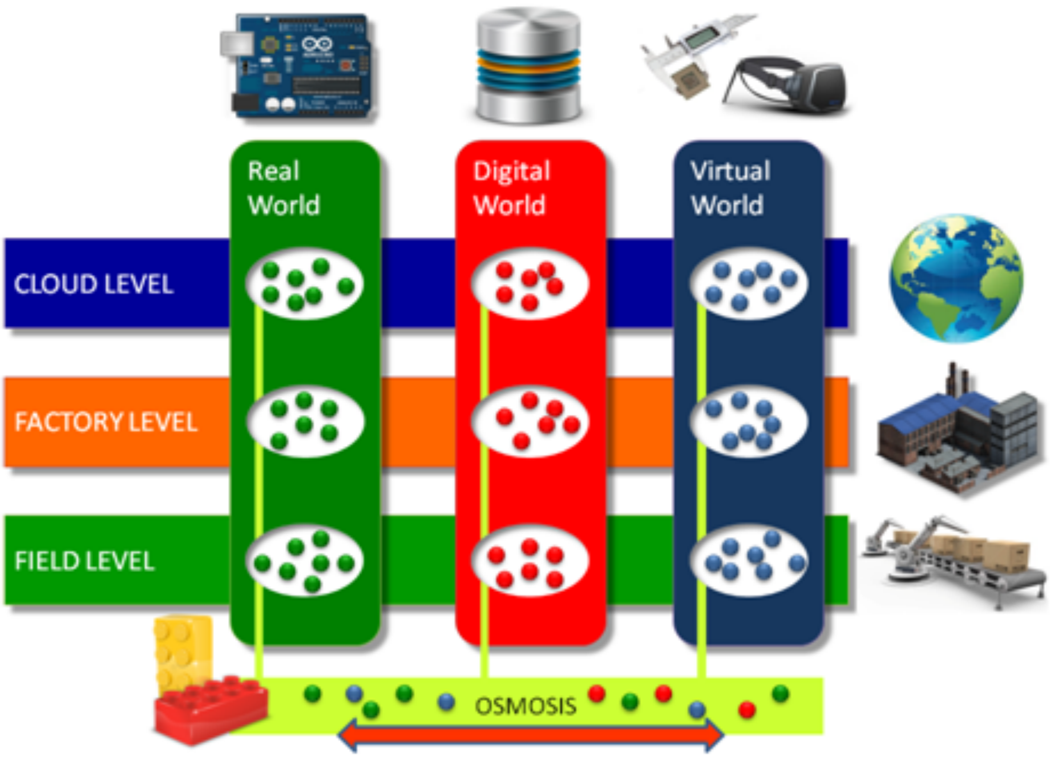

Figure 6 - BEinCPPS Reference Architecture: Levels and Worlds

The document also defined the mappings of such BEinCPPS-specific framework with the more general one from the Platform Industrie 4.0 initiative: RAMI 4.0 – see figure below.

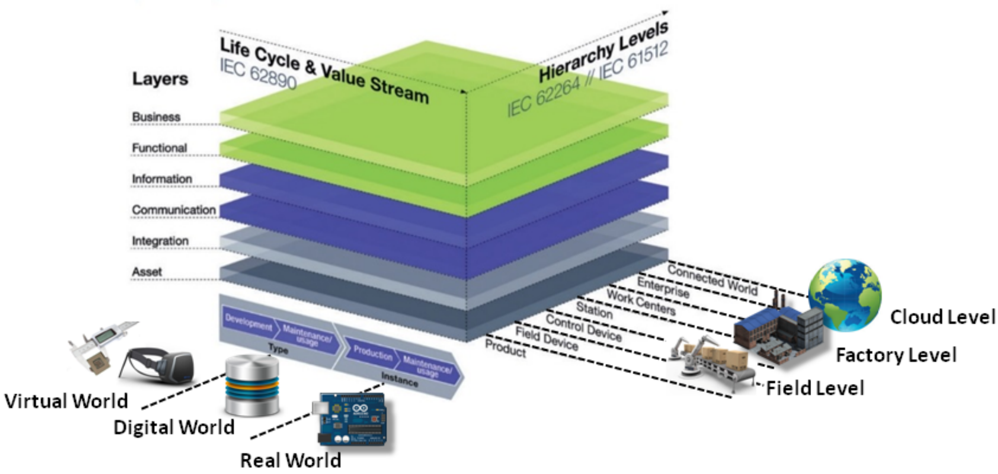

Figure 7 - Mappings of BEinCPPS-Arch with RAMI 4.0 dimensions

The BEinCPPS-Platform, on the other hand, was our first attempt at defining a *modular architecture* using (mostly) existing components from other communities like FIWARE, FIWARE for Industry, OpenIoT, etc. The resulting high-level design, depicted in the following picture, was subsequently used as the foundation for scenario-specific platforms not only by the five BEinCPPS Champions but also by the ten winners of the first BEinCPPS Open Call.

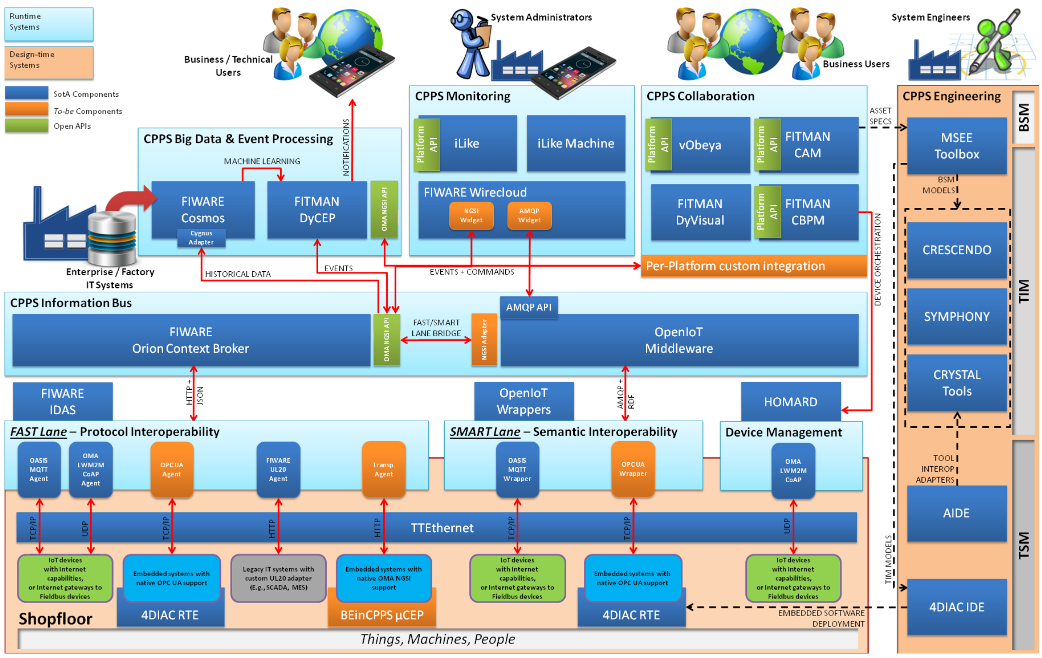

Figure 8 - BEinCPPS-Platform

The BEinCPPS deliverable D3.7 "Technical evaluation, lessons learned, recommendations", released by M18, reported in detail about the feedback received from BEinCPPS end users and IT providers, drawing some conclusions on the steps to be taken during the second iteration of the project in order to improve some *soft spots* in the general approach to CPS-ization. As part of that assessment, all IT artefacts that are part of BEinCPPS-Platform have been rated with respect to *openness*, *interoperability* and *application ease*. This for sure was a useful input for the development of the second version of the BEinCPPS architecture, which is now released three months after D3.7. However, the focus of this document is on architecture rather than on implementation, so the main feedback the authors were interested in was about BEinCPPS-Arch and BEinCPPS-Platform – the latter in its role of blueprint for concrete platform implementations, rather than as a "bag of assets". In particular, the question were: A) What was the actual value delivered to solution developers, system integrators and end users? B) How these assets can be improved?

We didn't follow a formal approach to collect this feedback. Rather, we exploited the hands-on experience that some of the authors of this document acquired firstly in their role of developers and integrators for some of the BEinCPPS Champions, and then as technical mentors of some winners of the first BEinCPPS Open Call. The conclusions that we were able to draw are quite simple, and are briefly described below.

BEinCPPS-Arch was perceived as a rather *artificial* scheme. The concept of Levels was clear enough and useful, being relevant to the physical deployment of components. Less so the concept of Worlds. In particular, the Virtual World abstraction proved itself to be difficult to grasp. Also for this reason, the Osmosis pattern – which describes how information can flow across the boundaries of Worlds – was not exploited in any of the scenario-specific solutions that were developed in the scope of the project.

The **BEinCPPS-Platform** design was perceived as overly complicated and lacking a clear rationale for the use of components having overlapping functionality. Moreover, it mixed open source and commercial assets with no clear demarcation. And finally, the choice of components was still a work in progress at the time of its release (M3), so that it didn't reflect very well the current state of things when, some months later, the work on the implementation of use cases started.

To improve over this situation, we have taken the following measures:

- A revolutionary approach: merge BEinCPPS-Arch and BEinCPPS-Platform into one single vision, built as progression of perspectives which start from a bare bone high-level design and then incrementally add up to the full picture – see §4.
- Use the work done by BEinCPPS Champions during the first project iteration as an example of how to take advantage of specific platform features – see §5.

4 BEinCPPS Architecture

The core of the BEinCPPS architecture v2.0 is described through four perspectives which can be seen as incremental layers: the Structural, Functional, Technical and Implementation Perspective, the first defining the barebones and the others adding elements relevant to their specific concerns.

4.1 Structural Perspective

The Structural Perspective defines two Domains and three Levels. It's an evolution – towards simplicity – of the BEinCPPS-Arch described in v1.0. Its goal is to lay down the basics on which to build the full picture. Its main concerns are the physical boundaries of IT systems for CPS-ization.

The two domains are Runtime Systems and Design-time Systems. The former includes IT systems supporting the *operation* of CPPS; the latter includes applications, services and tools for the *engineering* of CPPS. We can also think in terms of *online* (objective: to run CPPS) and *offline* (objective: to develop CPPS) systems.

The three Levels are only relevant to the Runtime Systems domain, and are defined in exactly the same way as their equivalents in BEinCPPS-Arch:

- Field: the scope of the shopfloor and its *physical processes*
- Factory: the scope of the production plant and its *local digital processes*
- Cloud: the scope of the enterprise and its *global digital processes*

Figure 9 shows Domains and Levels: it's the canvas on which the other Perspectives will draw.

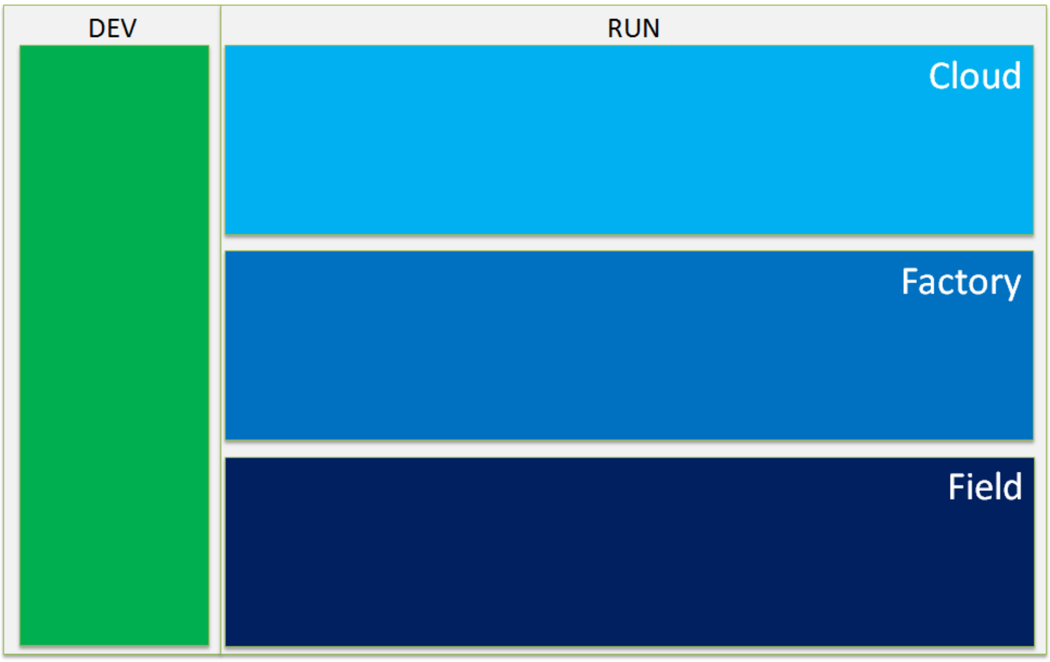

Figure 9 - The Structural Perspective

4.2 Functional Perspective

The Functional Perspective builds over the Structural Perspective: the four areas defined there are populated with Functional Blocks (FB) representing some kind of generic functionality, connected by links representing the flow of information between them. In particular, each FB belonging to Runtime Systems (RS) is positioned on the proper physical Level.

- **Field Level**
  - **Equipment & Devices**
    This is not an actual FB: it's a placeholder representing the shopfloor entities (sensors, actuators, PLCs, controller boards, smart machines, etc.) connected to RS by means of functionality provided as part of the CPPS Communication and CPPS Logic FBs
  - **CPPS Communication**
    Systems providing shopfloor-level network connectivity services to Equipment & Devices
  - **CPPS Logic**
    Systems implementing embedded logic for monitoring and control of Equipment & Devices
- **Factory Level**
  - **CPPS Middleware**
    Systems providing plant-level interoperability services; these are further classified into:
    - **Field Gateway**
      Systems providing an abstraction layer over the shopfloor, adapting device-specific communication protocols and data models to a common standard
    - **Information Bus**
      Systems implementing a logical bus over which information flows from / to the shopfloor can be exchanged without any direct coupling of producers and consumers (e.g., publish / subscribe pattern)
- **Cloud Level**
  - **Event-Data Processing**
    Systems providing the means for the centralized processing of shopfloor events and data, in scenarios that may require the use of significant computing resources (e.g., Big Data processing)
  - **Human-Computer Interaction**
    Generic end user interfaces to CPPS, performing interactive tasks that are not application-specific or that can be easily customized by means of configuration (e.g., dashboard)
  - **Ecosystem Collaboration**
    Generic tools supporting collaborative processes between humans and/or machines (e.g., workflow management)

- **Applications**
  This is not an actual FB: it's a placeholder representing the applications that implement specific CPPS solutions, connected to RS through the Information Bus

The engineering of CPPS (the Design-time Systems) is then treated as one single area.

- **Business Modelling**
  Design tools for modelling organizations and business processes that are relevant for CPPS
- **System Modelling & Simulation**
  Tools for modelling CPPS and simulate their behaviour, according to scenarios and requirements specified within the Business Modelling FB
- **System Engineering**
  Tools for creating the concrete design of CPPS, possibly from specifications developed within the System Modelling FB, and implementing embedded software

Figure 10 shows all the above mentioned Functional Blocks superimposed over the Structural Perspective.

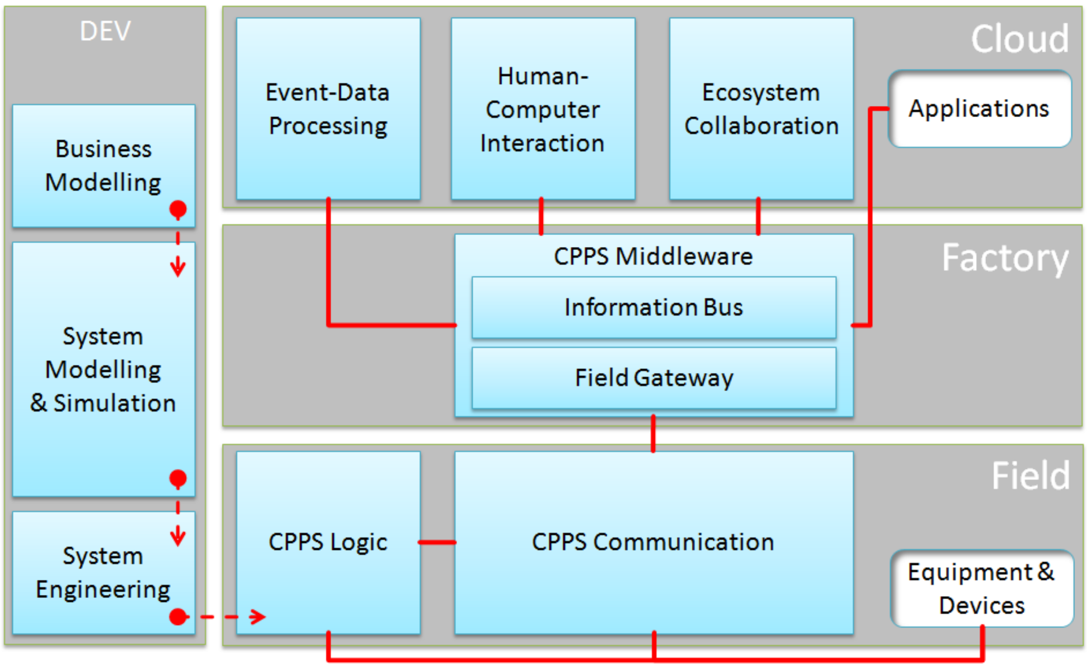

Figure 10 - The Functional Perspective

4.3 Technical Perspective

The Technical Perspective builds over the Functional Perspective: it identifies some technical standards that are mandatorily supported in key Functional Blocks.

- **OPC UA**
  Open Platform Communications Unified Architecture is a standard[4] for the interoperability of equipment in factory automation scenarios. OPC UA is governed by the OPC Foundation[5] and maintained by IEC[6]. It is the only officially endorsed protocol for the Communication layer of the RAMI 4.0 architecture: any product being advertised as "Industrie 4.0-enabled" must be OPC UA-capable (ZVEI, 2017). In BEinCPPS, where compatibility with I4.0 is of paramount importance, OPC UA-capable factory equipment is supported either directly (**CPPS Communication** FB) or through the mediation of some protocol adapter (**Field Gateway** FB). It is worth noting that OPC UA, as opposed to MQTT (see below), is not merely a communication protocol: it's a framework for defining *information models* for interoperability, in the first place (OPC, 2016). For this reason, BEinCPPS supports OPC UA also in its **System Engineering** FB, where it is used in software design tools for implementing OPC UA-capable devices.
- **MQTT**
  Message Queuing Telemetry Transport[7] is an OASIS[8] and ISO/IEC[9] standard for a *lightweight* publish/subscribe *data messaging* transport. MQTT is a TCP/IP-based binary wire protocol, the data payload being entirely application-specific. Due to its extreme flexibility, robustness, and suitability for constrained devices, it has become extremely popular in the IoT/IIoT world. In BEinCPPS, MQTT is supported in the same fashion as OPC UA: directly (**CPPS Communication**) or through the use of middleware (**Field Gateway**).
- **NGSI**
  The Next Generation Service Interfaces is an OMA standard[10] which includes functionalities for the sharing of *context information* that follow the publish/subscribe pattern. This specific subset of the NGSI specification (NGSI-9 and NGSI-10)[11], augmented with a REST-over-HTTP binding, has been adopted in FIWARE as the standard interface towards the IoT

---

[4] https://opcfoundation.org/developer-tools/specifications-unified-architecture
[5] https://opcfoundation.org/
[6] https://webstore.iec.ch/publication/25997
[7] http://mqtt.org/
[8] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf
[9] https://www.iso.org/standard/69466.html
[10] http://www.openmobilealliance.org/release/NGSI/V1_0-20120529-A/OMA-AD-NGSI-V1_0-20120529-A.pdf
[11] http://openmobilealliance.org/wp-content/uploads/2012/11/OMA-Service-API-standardization-Open-Access-to-Context-aware-Service-Enablers.pdf

world[12]. In BEinCPPS, NGSI is one of the two protocols supported in the **Information Bus** FB, the other being AMQP (see below). Support for NGSI enables *FIWARE-based applications* to be easily integrated into BEinCPPS systems.

- **AMQP**
  The Advanced Message Queuing Protocol[13] is a vendor-neutral standard[14] for a *reliable* publish/subscribe *data messaging* transport. As is the case for MQTT, AMQP is a transparent TCP/IP-based binary wire protocol that leaves applications free to agree with each other over the data model and the format of the message payload. As opposed to MQTT, it implements message queues[15] and can provide guarantees over delivery. For this reason, and due to its larger footprint, it is more suited for Enterprise applications rather than IoT (Piper, 2013). In BEinCPPS, it is supported in the **Information Bus** FB as more general-purpose integration point for applications (see also NGSI above).
- **IEC-61499**
  This is an emerging international standard[16] for the software implementation of *distributed automation* systems, which aims at replacing the legacy, ubiquitous IEC-61131[17] while maintaining backward compatibility. In BEinCPPS, it is supported both as a **System Engineering** tool (i.e., programming of automation workflows) and as a **CPPS Logic** asset (i.e., execution of automation workflows in embedded systems).
- **FMI**
  The Functional Mock-up Interface[18] is a vendor-neutral standard[19] for the interoperability of simulation tools, enabling the *exchange of CPPS models* and also *co-simulation* (i.e., cooperative, parallel simulation of distinct subsystems that contribute to a given solution). In BEinCPPS, it is supported in the **System Modelling & Simulation** FB.
- **UML**
  The Unified Modelling Language[20] is the best-known standard[21] for the *design of software systems*, which can also be applied to non-software problems and solutions. Its support in the **System Modelling & Simulation** FB of BEinCPPS is too obvious to be commented.

---

[12] https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI-10_Open_RESTful_API_Specification
[13] https://www.amqp.org/
[14] http://www.amqp.org/sites/amqp.org/files/amqp.pdf
[15] Despite its name, MQTT does *not* support message queues
[16] https://webstore.iec.ch/publication/5506
[17] https://webstore.iec.ch/publication/4550
[18] http://fmi-standard.org/
[19] https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI_for_ModelExchange_and_CoSimulation_v2.0.pdf
[20] http://www.uml.org/
[21] http://www.omg.org/spec/UML/

- **BPMN 2.0**
  The Business Process Model and Notation v2.0[22] is one of the most widely used *modelling languages for workflows*. BPMN 2.0 workflows, thanks to the support for low-level concerns in the language, can also be directly executed by BPMN-capable runtime engines. BEinCPPS supports this standard both in the **Business Modelling** FB (high-level design of business processes) and in the **System Modelling & Simulation** FB (low-level engineering of executable workflows).

Figure 11 shows the above listed technical standards superimposed over the Functional Perspective.

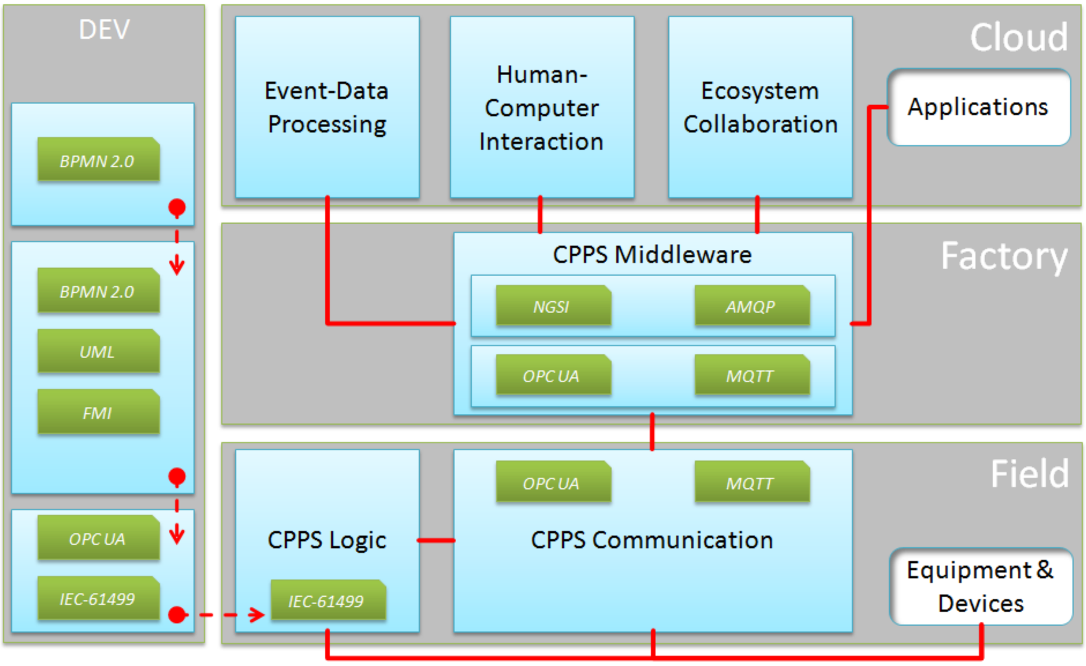

Figure 11 - The Technical Perspective

22 http://www.omg.org/spec/BPMN/

4.4 Implementation Perspective

Finally, the Implementation Perspective is the link between the BEinCPPS logical architecture, as described in the three preceding Perspectives, and its physical implementation that is the objective of development and deployment tasks of the project (WP2 and WP3). In this exercise, which is depicted in Figure 12 below, we mapped some specific components to the Functional Blocks – i.e., we identified assets that implement the functionality of FBs.

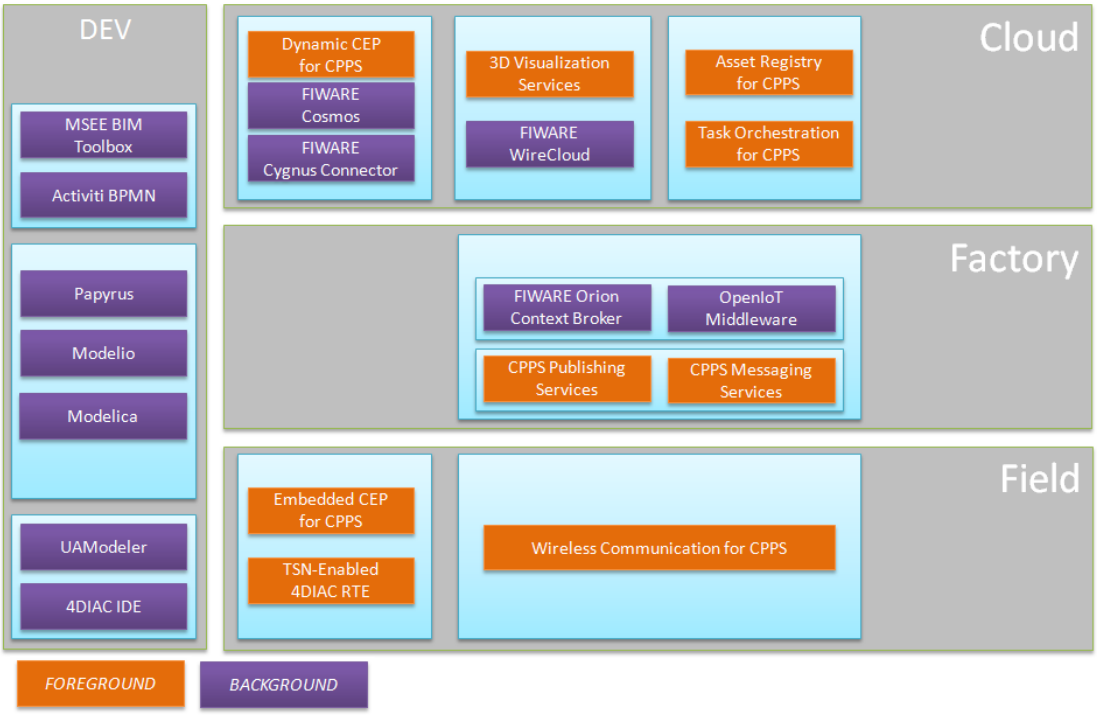

Figure 12 - The Implementation Perspective

In this context we are not going to list and describe components individually, this task belonging to upcoming deliverables that will provide not only the prototype implementation of FBs, but also the relevant documentation. However, some points are indeed worth of notice.

Firstly, for what BEinCPPS objectives are concerned, only Runtime Systems are addressed in the “BEinCPPS Platform”: FBs contributing to CPPS engineering have been populated by some specific components (mostly desktop applications) that we have identified as appropriate in this context, but no customization and/or integration activities are envisioned in the project.

On the other hand, FBs belonging to CPPS RS only contain open source software: this is in line with the project policy of contributing back IT results to the communities of developers and users. Moreover, in the simple schema of Figure 12, a clear distinction is made between background IPR (blue boxes) and foreground assets (orange ones), the latter being either a BEinCPPS-specific extension of existing software or even some radically new implementation.

4.5 BEinCPPS RA vs. RAMI 4.0

A useful tool for communication is a common conceptual framework and glossary, as those provided by the RAMI 4.0 3D model (see Figure 7). For this reason we took the additional step of mapping BEinCPPS Functional Blocks – and, indirectly, Domains and Levels – to the RAMI dimensions. The results of such exercise are graphically presented in the six figures that follow, without any further comment.

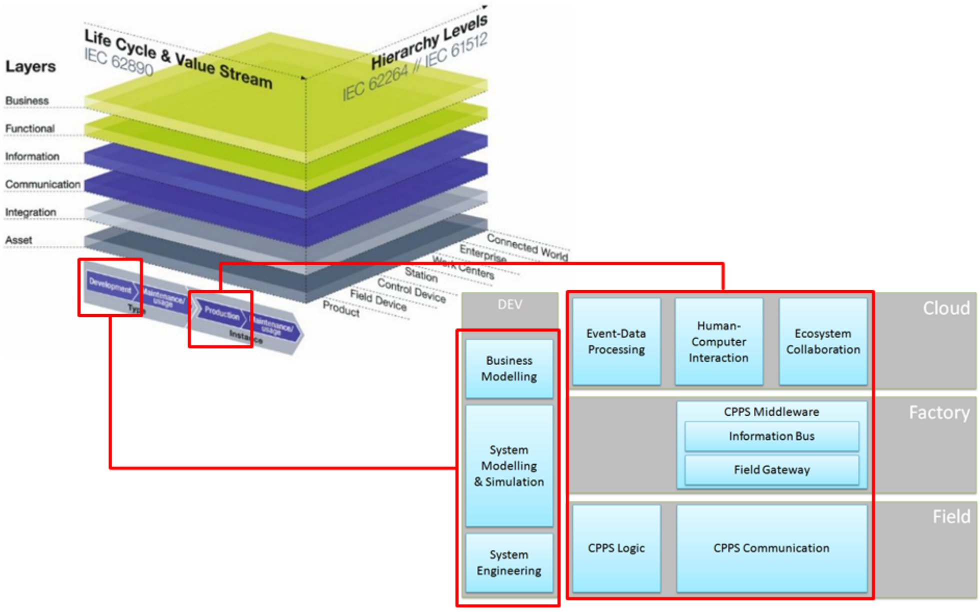

Figure 13 - RAMI 4.0 Lifecycle & Value Stream mappings

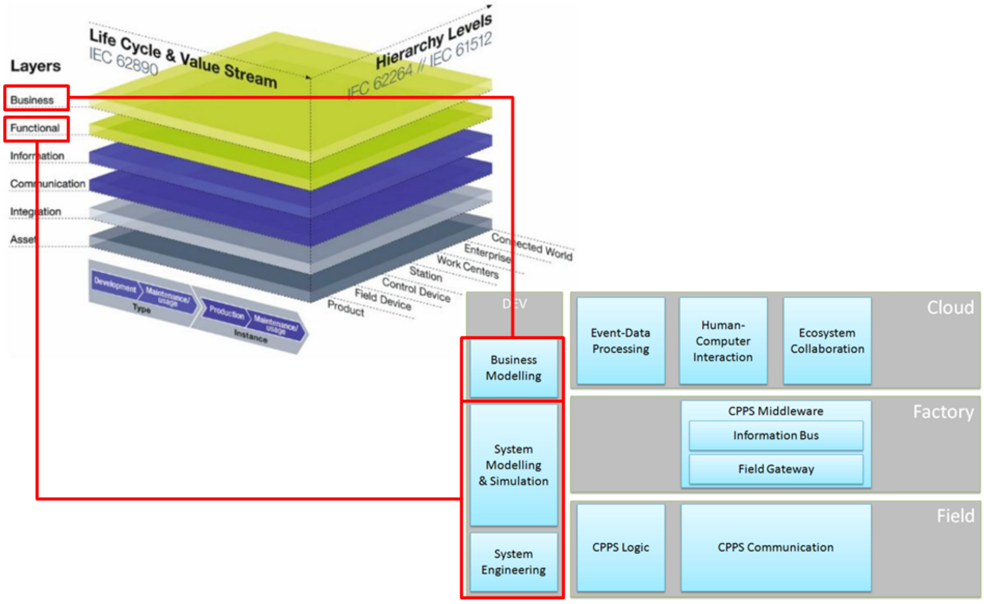

Figure 14 - RAMI 4.0 Layers mappings (part I)

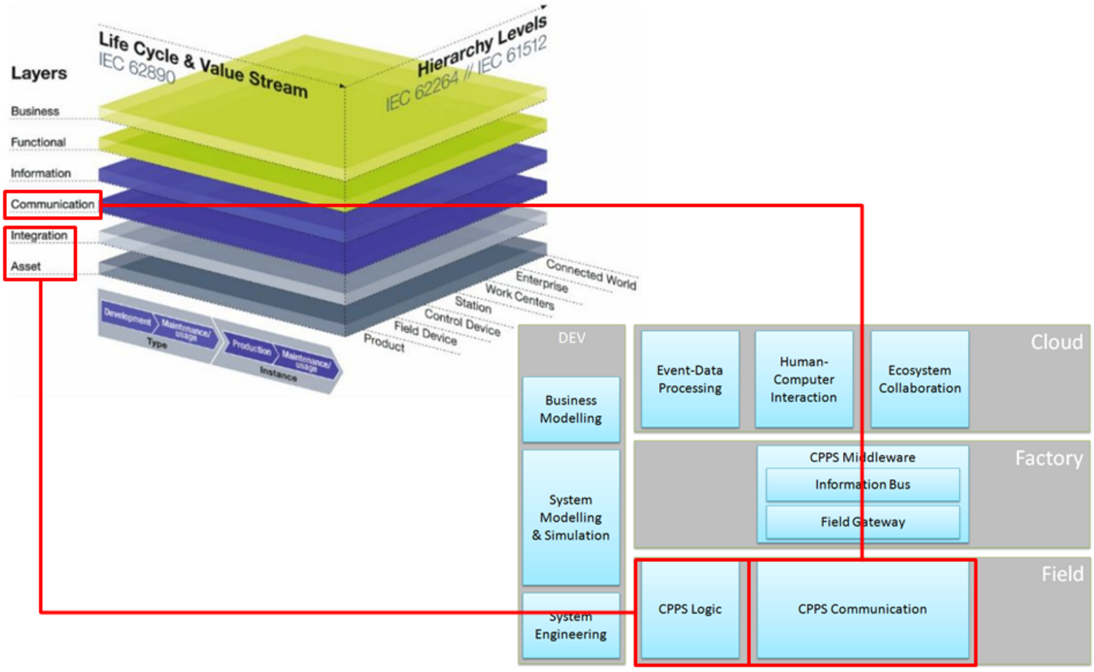

Figure 15 - RAMI 4.0 Layers mappings (part II)

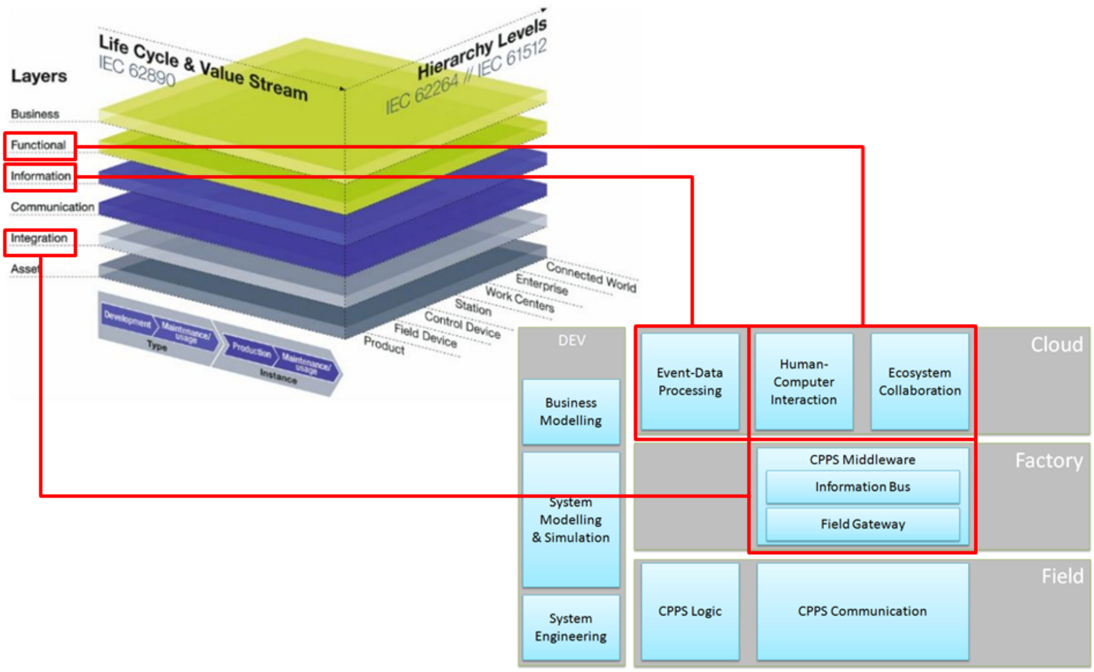

Figure 16 - RAMI 4.0 Layers mappings (part III)

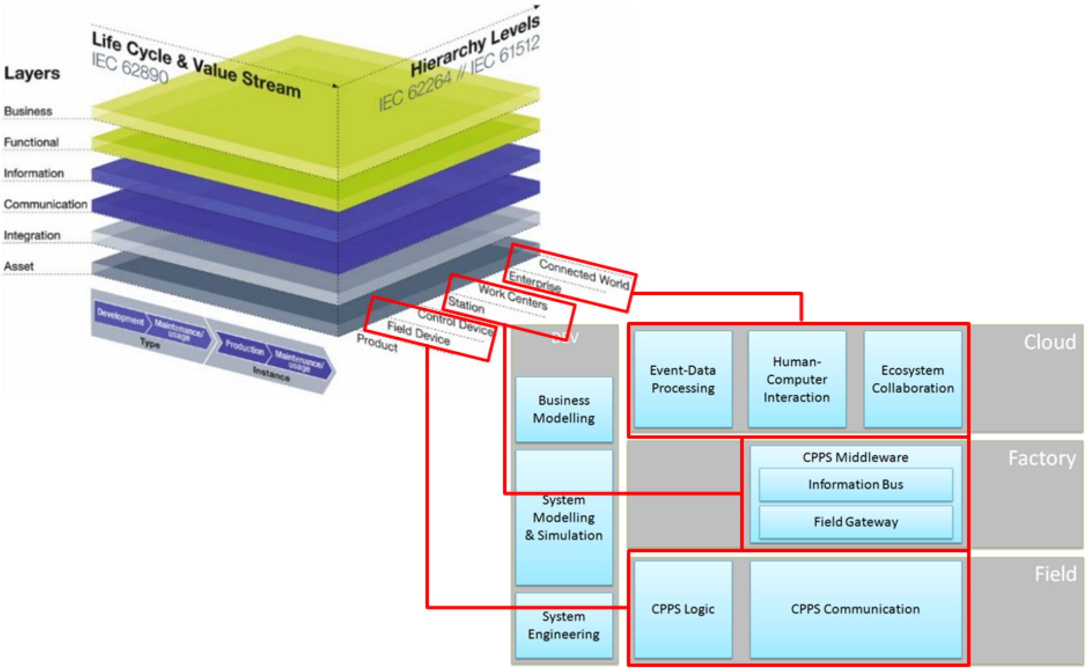

Figure 17 - RAMI 4.0 Hierarchy Levels mappings (part I)

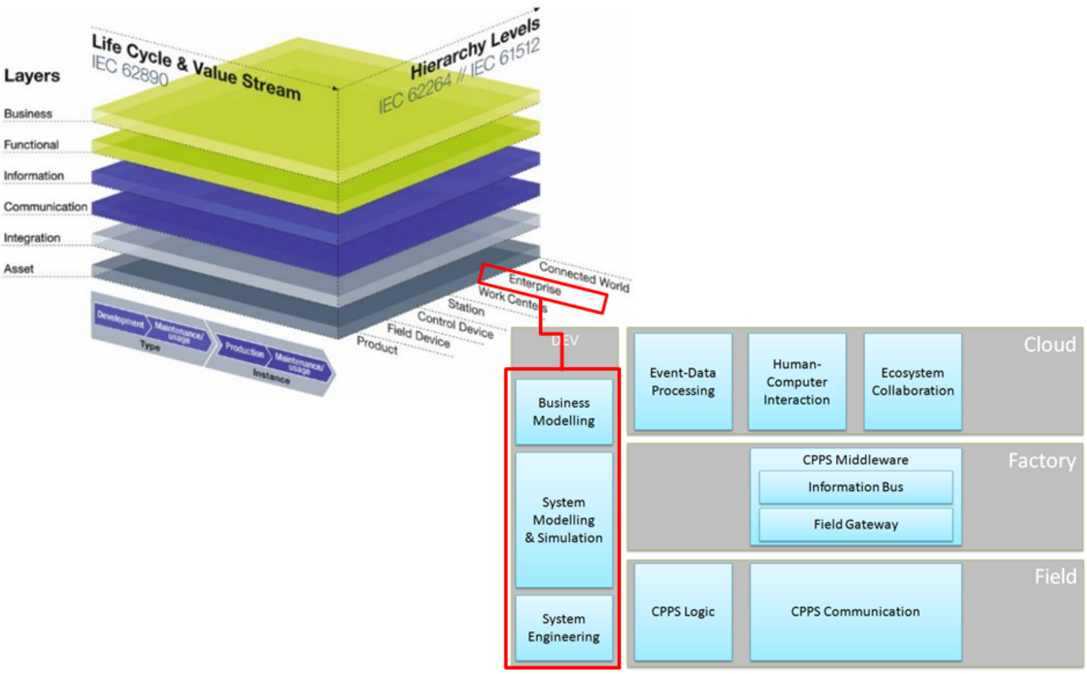

Figure 18 - RAMI 4.0 Hierarchy Levels mappings (part II)

5 Business Processes

The business process modelling topic could not be addressed by the first version of this document because too little information was available at the time (M3) regarding the business experiments and their use of BEinCPPS components. Following the definition of the BEinCPPS architecture, a specific methodology for business process modelling, derived from the MSEE project (see §5.2 below), has been adopted. MSEE extends the ideas of the Model Driven Architecture (MDA), a major topic of interest of Object Management Group (OMG)[23].

The rest of this section is structured as follows: after a basic introduction of MDA, the MSEE methodology is described; three concrete instances of models – two from BEinCPPS Champions (John Deere and Pernoud S.A.) and one from an Open Call 1 winner project (SmartPlant One) – are then given as illustrations to explain the use of the MSEE methodology in the context of BEinCPPS, and also to provide some real-world examples of the use of BEinCPPS assets.

5.1 Model Driven Architecture

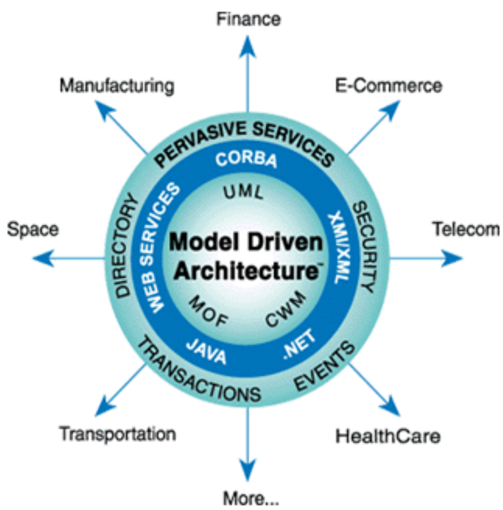

Figure 19: OMG's Model Driven Architecture[24]

The OMG promotes the use of the MDA. Based on OMG's established standards, the MDA separates business concerns and application logic from the underlying platform technology. Model driven system design allows the definition of abstraction layers in different dimensions. In the ATHENA project[25] four system aspects were identified for model abstraction: generality, system, viewpoint and modelling. Figure 20 displays these dimensions and presents the abstraction layers. An important aspect is

[23] www.omg.org
[24] Source: http://www.omg.org/mda/
[25] http://www.projectathena.eu/

model abstraction which introduces the idea that for models at any abstraction layer a hierarchy of models can be defined starting from mode instances by adding meta models. OMG introduced the Meta Object Facility (MOF) for the specification of meta models which themselves specify a set of model instances. Meta models allow to define model transformations of models of different abstraction layers for example computational independent models can be transformed into platform independent models and these again into models at the platform specific level. We will come back to this idea in the next section.

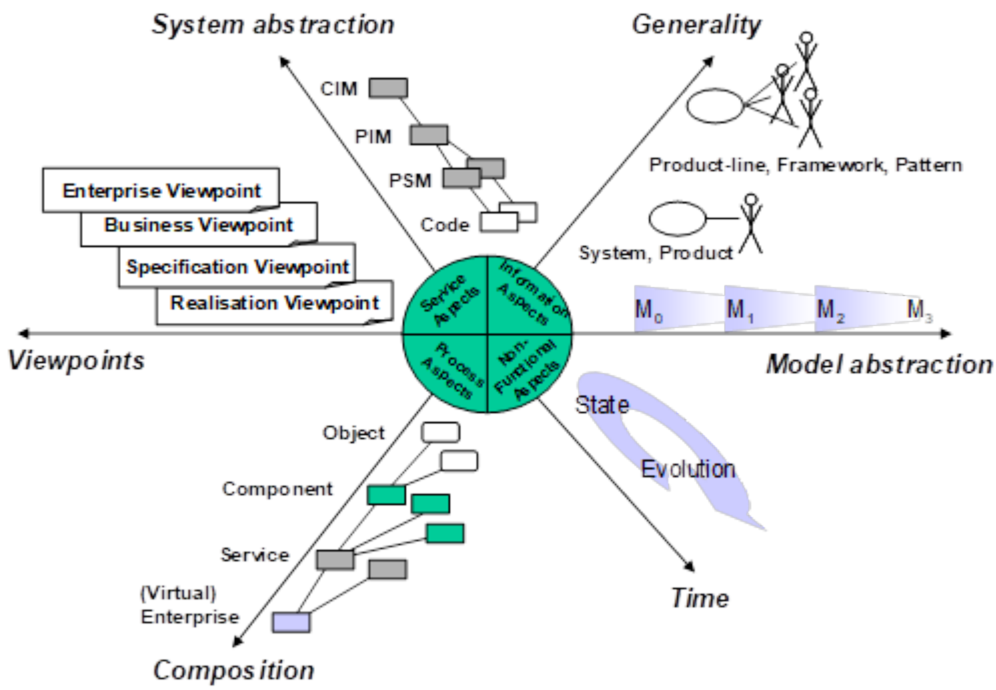

Figure 20: Integration dimensions with respect to four system aspects[26]

5.2 Manufacturing SErvices Ecosystem (MSEE)

Manufacturing SErvices Ecosystem (MSEE) was an EU FP7 project running during the 2011-2014 period. The Model-Driven Service Engineering Architecture (MDSEA) and the MSEE method and tools are the major results of the MSEE project (see Figure 21). The MSEE method distinguishes 3 modelling layers: Business Service Models (BSM), Technical/Technology Independent Models (TIM), and Platform Specific Models (PSM). At the BSM level, **Extended Actigram Star** (EA*) models are used to define the business perspective. The TIM layers adopts BPMN 2.0 and integrates it with the Universal Service Description Language (UDSL).

[26] Source: ATHENA Deliverable A6.1

The MSEE method tries to bridge the gap between CIM and PIM layer models in the original model driven architecture of OMG. EA* models capture the business perspective of processes and therefore belong to the CIM layer, while BPMN is a platform independent modelling language for processes which can be executed in business process execution environments (e.g., Activiti[27] or jBPM[28]).

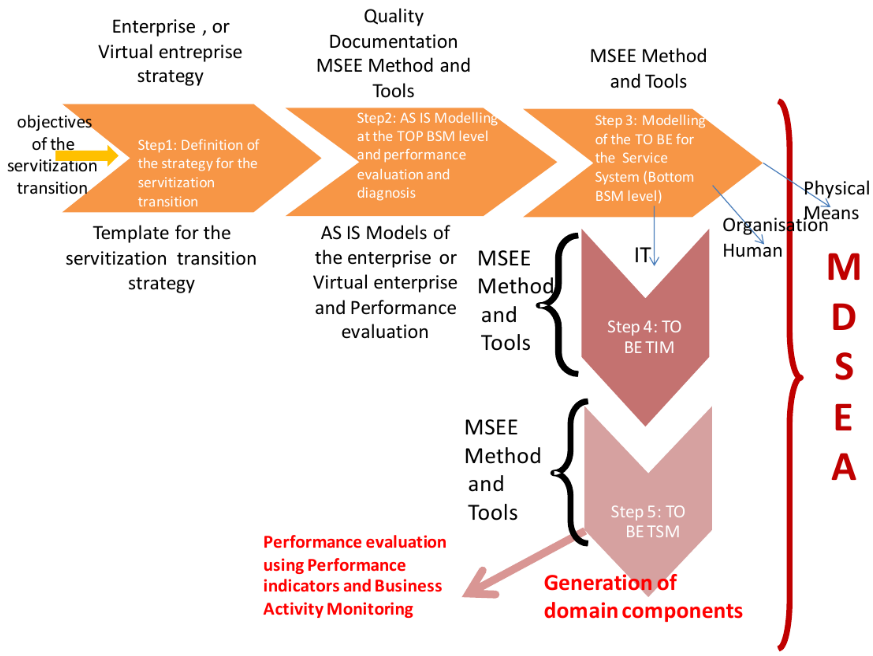

Figure 21: MSEE method using MSDEA tools and concepts[29]

A major problem in model driven system development as well as in process modelling is that available tools do not support well the integration of extensions and modifications, which were introduced at lower modelling layers, when the models coming from the higher modelling layers are modified. A way to deal with this problem is to hide as far as possible the extensions in the lower modelling layers behind services which are then connected to the models coming from the higher modelling layer. The integration of the models with the services should be as far as possible automatic, which means however that all the information necessary for the integration needs to be included in the models. This is true in the first place for the data elements that need to be passed to the different service. For the definition of data models, which are platform independent, ontologies are the most helpful tool. Figure 22 displays how the MSEE method deals with ontologies at different modelling layers.

[27] https://www.activiti.org/
[28] https://www.jbpm.org/
[29] Source: http://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-MSEED115MSEEArchitectureforServiceModelingv10.pdf

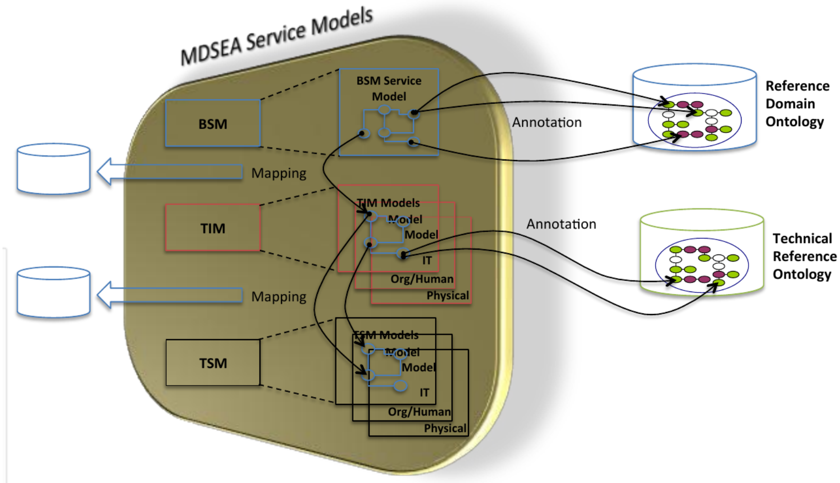

Figure 22: Ontologies in MSDEA service modelling[30]

5.3 Extended Actigrams for Process Instances

The BEinCPPS experiments target 10 application areas[31]. The following sub-sections present EA* models for three BEinCPPS experiments from the application areas "CPS-based Zero Defect Manufacturing" (John Deere), "CPS-based sensors data acquisition and management" (Pernoud S.A.) and "CPS-based Production Process Ramp-up and Commissioning" (SmartPlant One). Conceptually these EA* models are meant to be the starting point in the design process of a CPPS application.

5.3.1 Pick by Light (John Deere)

This BEinCPPS experiment is implemented at John Deere, which is the world market leader for agricultural equipment. In order to secure and increase the market share of a company like John Deere in the premium segment, it is crucial to deliver products of the highest quality, aiming at Zero Defects Production. For this reason, the focus of the BEinCPPS experimentation inside the John Deere plant in Mannheim is on the main assembly line, in particular on the support to workers during the assembly of a tractor machine in a particular station. The introduction of BEinCPPS components — and some specific additions from Open Call 1 winners — enabled quality assurance on many products variants in the same assembly station. Several new solution were introduced, including:

1. A Wireless *Pick by Light* solution avoiding the assembly of wrong, but similar looking parts.

2. A *Pick by Vision* Solution combining a smart glass and a wearable scanner.

3. A Trainable Quality Gate that does not require deep system knowledge to teach new material.

---

[30] Source: http://cordis.europa.eu/docs/projects/cnect/0/284860/080/deliverables/001-MSEED115MSEEArchitectureforServiceModelingv10.pdf
[31] http://i4ms.eu/documents/BEinCPPS_Call-1_Introduction_v3.0_20052016.pdf

Figure 23 shows the EA* model of the Pick by Light process. Subsystem which are marked with the BEinCPPS logo are links to a specific BEinCPPS component (see the Implementation Perspective, Figure 12). These BEinCPPS components are also needed to enable cases nr. 2 and nr. 3 mentioned above. This is important because the BEinCPPS solution installed can be generalized into a Pick-By-X system, where X could be any appropriate CPS system, for example a wireless display with lights and button for pick confirmation (nr.1) or a wearable system composed of smart glasses and scanner on a glow (nr.2). Finally, given the flexibility and modularity of the BEinCPPS platform, such Pick-By-X system shall not only support the workers but also reduce the integration effort of other support solutions for picking assembly parts.

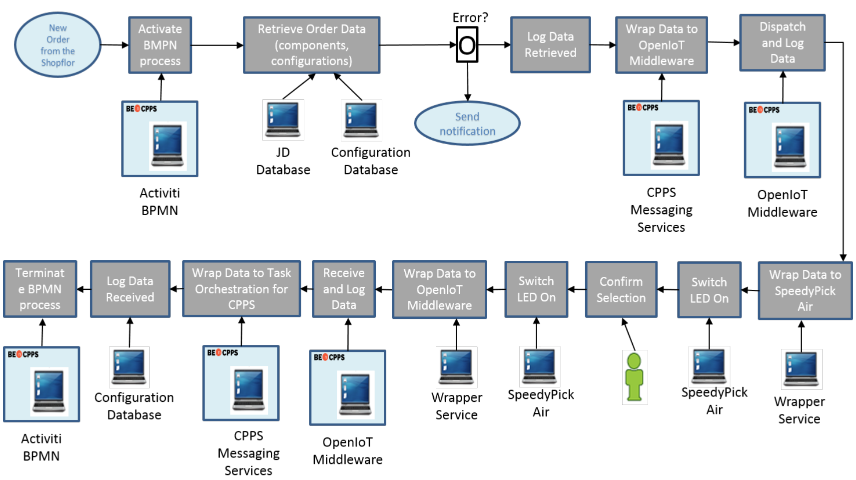

Figure 23: EA model of the Pick by Light Process*

5.3.2 Mold Maintenance and Energy Efficiency (Pernoud S.A.)

This BEinCPPS experiment is implemented at Georges Pernoud, a manufacturer of plastic injection molds which is globally renowned for its expertise. The purpose of this experiment is to add intelligence to molds (*smart mold*). The goal is to acquire a continuous information stream about the mold status and to analyze such data in order to improve the productivity, a process dubbed *E-Tooling*.

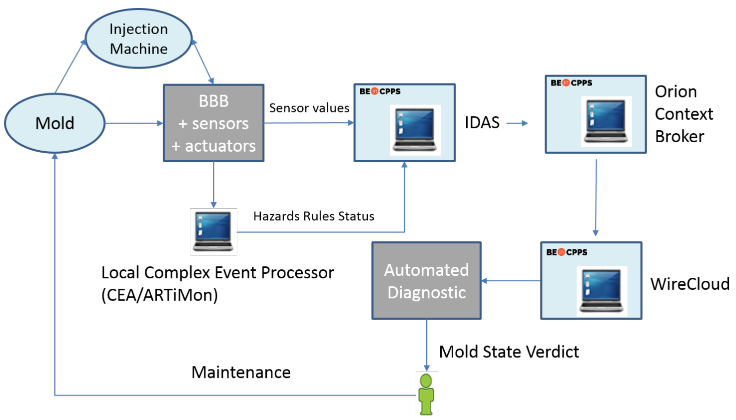

Figure 24: EA model of the E-Tooling Process*

The E-Tooling encompasses two features whose development is supported by the BEinCPPS project. The first feature consists in a machine-factory-cloud chain to monitor in real time the mold usage, having two goals: triggering some maintenance operations or invalidating some warranty claims when the mold is used outside of its working specification. The second feature consists in replacing some hydraulic parts with electric ones in order to refine their control and to lower the global energy consumption of the whole molding process.

5.3.3 Installation and Maintenance of Gateway (SmartPlant One)

SmartPlant One is a BEinCPPS Open Call 1 project conducted by 2 SMEs, which are New Generation Sensors S.r.l. and FlairBit S.r.l. SmartPlant One develops an IT solution to easily install a smart sensor gateway in the DFKI's Smart Factory and link it with a digital twin in a virtual environment. The main objective of this project is the implementation of a reliable and low-cost predictive maintenance system. Its effortless and low-cost implementation make it an affordable solution for SMEs. The solution implements a clear, precise and easy means to understand information about the early warning signals regarding machinery failure, provided to non-expert end users. To achieve this, the SmartPlant One team embed inside the system gateway sets of logics capable to statistically process data gathered from the sensor network and to deliver synthetic information from sampling data such as peak to peak, accumulated square areas of profiles. With its open-source components, BEinCPPS contributes fast implementation in order to let SME apply their IT products in highly advanced factory systems.

The EA* model in Figure 25 shows the whole procedure from installing a new gateway to the maintenance work on conveyor motors of production machines. The left hand side branch of the process describes the setup of a digital twin in the virtual environment mirroring the situation in the production line. The right hand side branch of the process describes the animation of the digital twin triggered by sensor input from the production line which is processed by the BEinCPPS complex event processing engine.

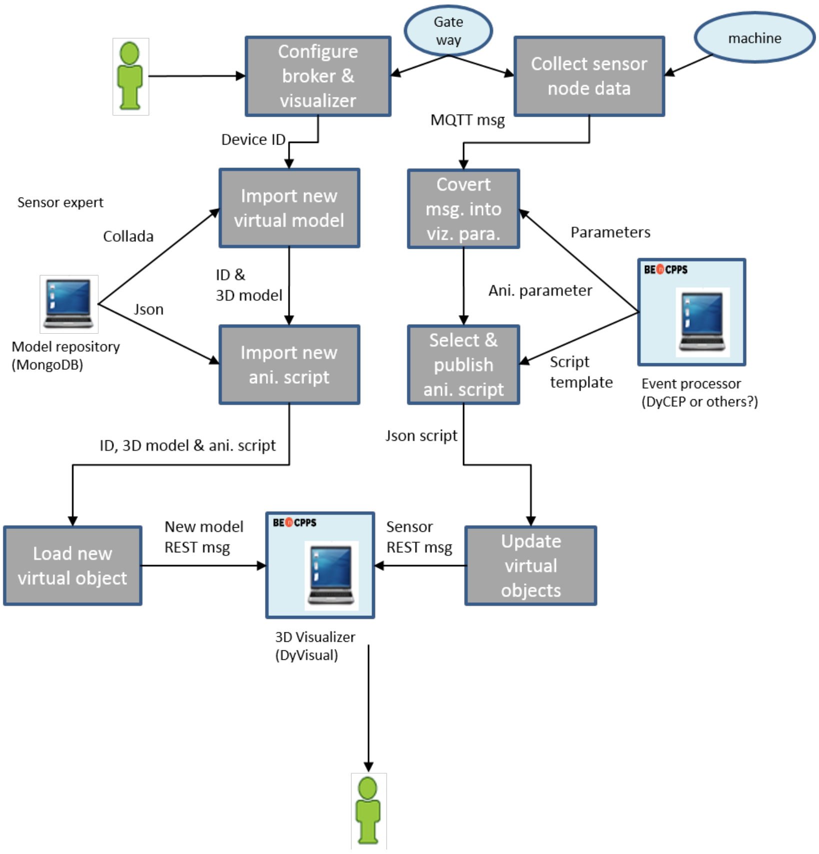

Figure 25: EA model of the process for installing and maintaining of a new gateway*

6 Conclusions

This document reports the outcome of the second and final iteration of task WP2.1, which was in charge of defining a three-layered (i.e., machine/factory/cloud) architecture for CPPS. This architecture, significantly cleaned up and simplified with respect to the first version released eighteen months ago (see §3) has been described in §4, and examples of its use have been introduced in §5.

Next steps in WP2 will be the final release of the full set of BEinCPPS foreground components as identified in the Implementation Perspective (the orange boxes in Figure 12), which will be completed by M24, and their integration into the BEinCPPS federated platform by M27. We expect all these assets to provide a solid foundation for the implementation of CPPS scenarios. To provide some guidance to the users, such scenarios have been organized in categories, which are listed in the **Solutions** column in Figure 26 below.

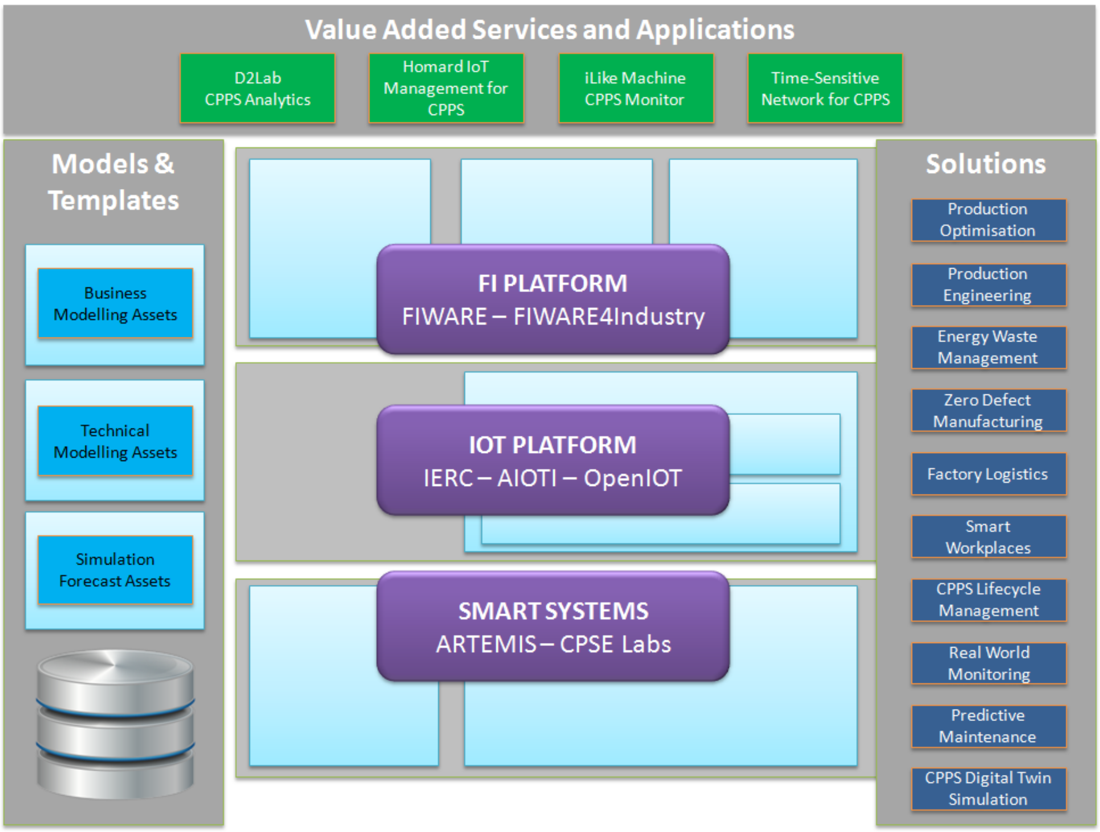

Figure 26 - Assets related to the BEinCPPS Platform

Figure 26 also shows additional BEinCPPS assets that are not part of the BEinCPPS platform but are expected to play a major role in the exploitation of project's results: **Value Added Services & Applications** (top row) and **Models & Templates** (leftmost column). The former are foreground components that, while being open and standard-compliant, are not open source software. The latter are artefacts resulting from the use of Design-time tools for the development of CPPS solutions for the five BEinCPPS Champions, and that are suitable for reuse in similar scenarios. Some examples of artefacts of this kind (i.e., business process models) have been presented in §5.

Finally, Figure 26 includes links to external communities that have contributed knowledge and background assets to the BEinCPPS project, and that in turn will benefit from BEinCPPS results. These are grouped into three topics:

- **Future Internet Platform**
  The reference communities are **FIWARE**[32] and its industry-oriented offspring **FIWARE for Industry**[33].
- **Internet of Things Platform**
  Links to the **IERC** research cluster[34], the **AIOTI** standardization initiative[35] and the **OpenIoT** project[36].
- **Smart Systems**
  Includes the **Artemis** industry association (Embedded & CPS focus area)[37] and the **CPSE Labs** consortium[38].

For all these connections, the BEinCPPS projects is either actively engaged in bilateral discussions or plans to initiate them during the second half of the program.

[32] https://www.fiware.org/
[33] http://www.fiwareforindustry.eu/
[34] http://www.internet-of-things-research.eu/
[35] https://aioti-space.org/
[36] http://www.openiot.eu/
[37] https://artemis-ia.eu/
[38] http://www.cpse-labs.eu/index.php

Bibliography

BMWi. (2017, April). *Structure of the Administration Shell.* Retrieved July 2017, from Platform Industrie 4.0: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.pdf

IIC-ATG. (2017, January). *The Industrial Internet of Things - Volume G1: Reference Architecture v1.8.* Retrieved July 2017, from The Industrial Internet Consortium: http://www.iiconsortium.org/IIRA.htm

IIC-CTG. (2017, February). *The Industrial Internet of Things - Volume G5: Connectivity Framework.* Retrieved July 2017, from The Industrial Internet Consortium: http://www.iiconsortium.org/IICF.htm

OPC, F. (2016, May). *Interoperability for Industrie 4.0 and the Internet of Things.* Retrieved July 2017, from opcfoundation.org: https://opcfoundation.org/wp-content/uploads/2016/05/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN-v5.pdf

Piper, A. (2013, February 19). *Choosing Your Messaging Protocol: AMQP, MQTT, or STOMP.* Retrieved July 2017, from VMware vFabric Blog: https://blogs.vmware.com/vfabric/2013/02/choosing-your-messaging-protocol-amqp-mqtt-or-stomp.html

ZVEI. (2017, April). *What Criteria do Industrie 4.0 Products Need to Fulfil?* Retrieved July 2017, from www.zvei.org: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/April/Kriterien_I4.0/ZVEI_BR_LF_Kriterien_I.4.0_ENGLISCH_Download_03.04.17.pdf