



BE CPPS

Innovation Action Project

HORIZON 2020 - EU.2.1.5. - Ref. 680633

D3.3 - Digital World Components

Lead Author: ENGINEERING

With contributions from: Holonix, FINCONS, NISSATEC

Delivery Date: August 2016

List of Figures

Figure 1 - Related tasks.....	6
Figure 2 - Worlds vs. Levels	7
Figure 3 - DW Components vs. Platforms and Levels	8
Figure 4 - Engineering's FIWARELab GE catalogue management interface	12
Figure 5 - Engineering's Hadoop cluster management interface	13
Figure 6 - BEinCPPS resources on Engineering's FIWARE Lab infrastructure	14
Figure 7 - DW deployment diagram on Engineering's infrastructure	15
Figure 9 - DW deployment diagram on commercial provider's infrastructure	17
Figure 10 - CCE authentication and authorization layer	18
Figure 11 - Digital World: APIs and RW Gateways	23



Table of contents

1	Introduction	6
1.1	The Digital World landscape	7
1.2	Organization of this document	9
2	Environment	10
2.1	Engineering's environment	11
2.2	Commercial ISP's environment	15
2.3	Holonix' environment	17
2.4	Environment federation	17
3	Deployment	18
4	Integration Interfaces	22
4.1	AMQP	23
4.2	NGSI	24
4.3	AVI	24
4.4	DW API Recap	24
5	Conclusion	25



Executive summary

WP3 adopts a work breakdown structure based on the distinction between Real World (RW), Digital World (DW) and Virtual World (VW). In particular, task 3.2 is in charge of deploying DW components. From the perspective of *physical* deployment, however, BEinCPPS components can be logically grouped by Level, meaning that they typically “belong” to either the shop floor (Field Level), to a local plant/office (Factory Level), or to the global enterprise/ecosystem (Cloud Level). So, overall, we can consider the DW as a cross-cutting aspect of the Cloud and Factory Levels. In more practical terms, this means that DW components are the constituents of the FI Platform (Cloud) and of the IoT Platform (Factory). The DW is where RW entities are digitally represented, data that originate from the RW are stored and processed, RW processes are monitored and coordinated.

This document reports about the creation of a Common Cloud Environment (CCE) for the deployment of the full set of DW components. In particular, it explains the rationale for the CCE existence and how the CCE is the result of the federation of three distinct physical computing infrastructures operated by BEinCPPS partners. Moreover, it provides details about the deployment of each individual DW component and gives an overview of the public interfaces for the integration of the DW with the Field (RW Gateways) and with manufacturing applications (DW APIs). Such interfaces are exposed by specific software components, and as such they are documented in the relevant WP2 deliverables.

As a conclusion, this document gives an overview of the roadmap of the DW towards the second part of the project, when a second and final round of deployments will take place.

1 Introduction

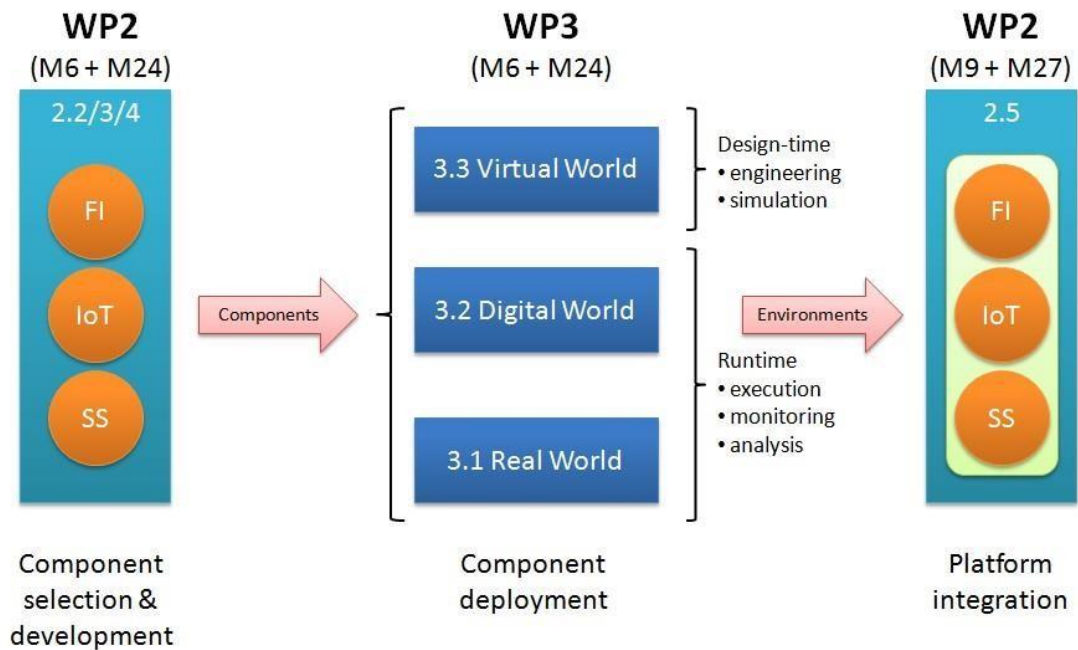
In the scope of WP2, tasks 2.2 and 2.3 are responsible for delivering two *platform designs* that target Internet of Things (IoT) and Future Internet (FI) environments, respectively. To date, such platforms have been defined by a selection of existing software components (*background*) and by the development of new or enhanced ones that address specific needs of the BEinCPPS experimentations (*foreground*). However, while both designs are concerned with component interconnection they are still abstract: the implementation of a federated/integrated platform belongs to task 2.5, the outcome of which will be available from month 9 of the project.

WP3, on the other hand, is mostly about component deployment: the instantiation in the target environments of the discrete elements identified in WP2. WP3 adopts a work breakdown structure based on the distinction between Real World (RW), Digital World (DW) and Virtual World (VW). In particular, task 3.2 is in charge of deploying DW components. The DW is where RW entities are digitally represented, data that originate from the RW are stored and processed, RW processes are monitored and coordinated. While the VW deals with *design time* concerns like the engineering and



simulation of CPSs and of their enabling hw/sw components, DW is all about *runtime interactions*.

Figure 1 - Related tasks



It is worth noting that the RW/DW/VW classification is purely functional: it has nothing to do with deployment, which in BEinCPPS is addressed by Levels.

1.1 The Digital World landscape

From the specific perspective of deployment, BEinCPPS components can be logically grouped by Level, meaning that they typically “belong” to either the shop floor (Field Level), to a local plant/office (Factory Level), or to the global enterprise/ecosystem (Cloud Level). Figure 2 below shows how Worlds and Levels overlap and create a grid that captures components in World/Level-specific *topics*. Note however that for what *software* components are concerned, Levels are more a suggestion than a constraint – e.g., a Cloud service may be successfully deployed on the Factory’s local premises as well¹.

¹ The same is not true for hardware components, which in BEinCPPS belong to the Real World are confined to the Field Level.



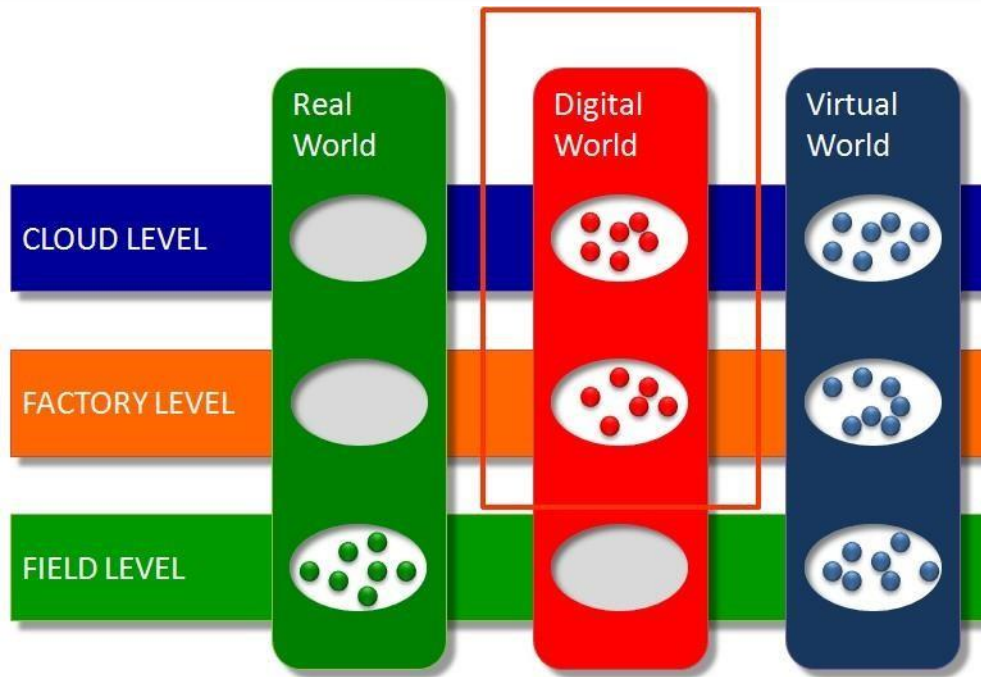


Figure 2 - Worlds vs. Levels

The Digital World is, as depicted in Figure 2, a cross-cutting aspect of the Cloud and Factory Levels. In more practical terms, this means that DW components are the constituents of the FI Platform (Cloud) and of the IoT Platform (Factory). Figure 3 shows how individual DW components are distributed with respect to Platforms and Levels. This picture is a simplified version of what was published as part of deliverables D2.5 (FI Platform) and D2.3 (IoT Platform), from where more detailed information about the platforms can be retrieved.

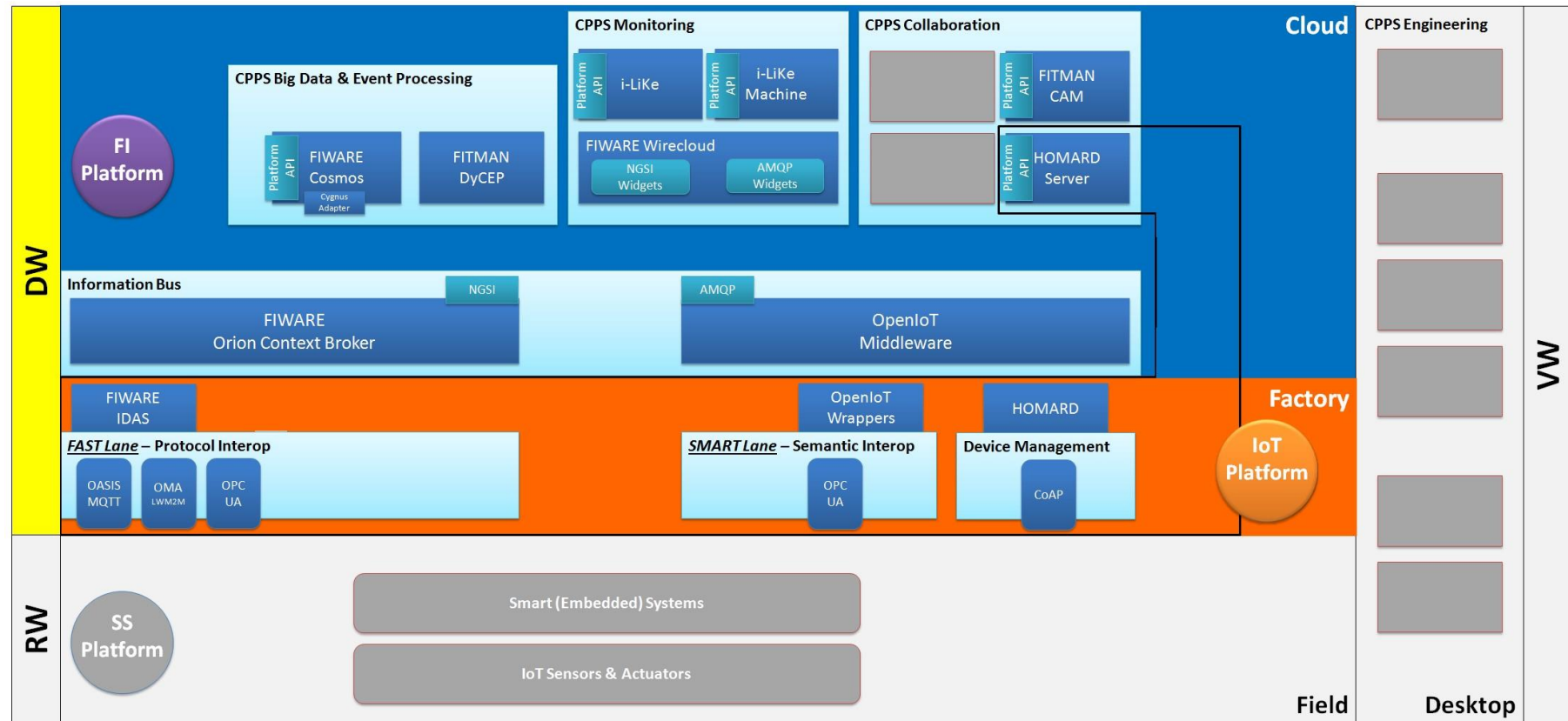


Figure 3 - DW Components vs. Platforms and Levels



Figure 3 is a full picture of the DW landscape at the time of writing, in the more general context of platform federation. As can be easily seen from the diagram, the DW is the processing infrastructure of the RW (Field Level), while VW components support both of them from the CPPS Engineering perspective². Moreover, the diagram shows how the IoT Platform has also a Cloud facet, represented by the Homard Server component.

1.2 Organization of this document

This document reports about the deployment of DW components on the computing infrastructures that were made available to this goal in the scope of the BEinCPPS project. These infrastructures, and the rationale behind them, are described in section #2 “Environment”, while individual deployments are documented in section #3 “Deployment”. The public interfaces for the integration of the Digital World with the Field (RW Gateways) and with manufacturing applications (DW APIs) are addressed in section #4 “Integration Interfaces”. Finally, section #5 “Conclusion” is an overview of the roadmap of the BEinCPPS DW towards the second part of the project, when the second and final iteration of deployments will take place.

2 Environment

When planning the deployment of DW components in the scope of WP3, it immediately became clear that the first and foremost project-level objective was the availability of a *work environment* for implementing cross-platform integration. The three Level-specific subsystems provided by WP2 (i.e., Smart System Platform for the Field Level, IoT Platform for the Factory Level, Future Internet Platform for the Cloud Level) must be made interoperable while maintaining a high degree of autonomy and self-consistency. We call this loose coupling *federation*, which is the main goal of task 2.5, together with testing and continuous support. This integration task is collaborative, as several BEinCPPS partners are involved. There is an obvious need for a common IT infrastructure where these activities can be performed. Such infrastructure must be accessible by all the stakeholders, and have a complete deployment of all the relevant software components.

To address these requirements, we have resolved to build a **Common Cloud Environment (CCE)** for the deployment of the full set of Digital World components. The CCE, as presented in this document, is at the same time a *development/testing environment* dedicated to task 2.5 (M9) and a fully-fledged *shared computing*

²Being mostly desktop software, VW components are not constrained by Level concerns.

infrastructure for the first phase of business experimentation (M9) performed by BEinCPPS' five regional champions. By their very nature, Digital World components are ideal candidates for Cloud deployment (see section #0) and having one single *global* environment supporting multiple BEinCPPS *local* sites is also an opportunity for our regional champions for putting their *Cloud roadmap*, if any, at test. The rationale is that manufacturing companies may, if allowed by their data management policies, *kick start* their first use case prototype implementations by leveraging a generic, externally-managed and easily accessible IT infrastructure. Building on this foundation, a more specialized and commercial-grade – possibly private – infrastructure may then be planned for the second iteration of BEinCPPS experiments.

Moreover, a CCE with the complete Digital World is going to be a key asset for the BEinCPPS Open Calls, where developers will provide new IT solutions based on BEinCPPS platforms and manufacturing SMEs will adopt/re-adapt BEinCPPS use cases: for anyone wishing to replicate existing deployments somewhere else, the CCE comes handy as a template environment that can be easily cloned as a starting point for further work. To this goal, most of the Digital World components have been virtualized individually using Docker³, so that a *portable* image of every deployed software is available for fast re-deployment in a different context.

While we refer to the CCE as a single logical entity, from the physical point of view this is made up of different and independent computing infrastructures which cooperate in providing a unified environment – much like the three BEinCPPS platforms with their federated approach. Such cooperation is basically along the lines of naming and security. The former is just a common DNS domain (beincpps.eu) that is managed internally by the BEinCPPS consortium and is used here to provide network address resolution for all the services deployed on CCE; the latter, much more significantly, is a common layer of authentication and authorization implemented by FIWARE Generic Enablers, which actually wrap the CCE into a security domain of its own with single-sign-on support. Note however that while all the required FIWARE components have been already deployed on the CCE, their integration with Digital World components – in particular, the authentication of interactive users of web applications like the FITMAN CAM front-end – is still a work in progress at the time of writing.

In the following subsections, all the physical computing infrastructures that comprise the CCE are briefly described.

2.1 Engineering's environment

Engineering was a key player in the FIWARE⁴ initiative, which delivered the Core Platform for Future Internet applications – i.e., a catalogue of Generic Enabler (GE) software components⁵ and some architectural blueprints⁶ for their use as building blocks for Cloud-based solutions. As part of its effort, Engineering was responsible for FIWARE project's *Test Bed* facility, and is currently operating a node of the

³ Lightweight “containers” as opposed to virtual machines – see <https://www.docker.com/>

⁴ <https://www.fiware.org/>

⁵ <http://catalogue.fiware.org/enablers>

⁶ http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Architecture ⁷

<https://lab.fiware.org/> ⁸ Software-as-a-Service




public FILab Cloud infrastructure⁷. Given this solid background and its enterprise mission as a system integrator, it was only natural for Engineering to choose the role of *FIWARE Instance Provider* as one of its exploitation channels for FIWARE results – i.e., providing a commercial-grade, managed Cloud environment for FIWARE solutions delivered in *SaaS*⁸ mode. In practical terms this meant, in the first place, creating an OpenStack-based cloud computing infrastructure⁷ and making the full FIWARE GE catalogue available on it as virtualized servers, ready for customers to instantiate using a *self-service* web interface – as shown by Figure 4. Moreover, a complete Hadoop cluster environment⁸ was also deployed to support FIWARE Cosmos/BigData usage scenario⁹, as documented in Figure 5.

⁷ <http://www.openstack.org/>

⁸ <http://hadoop.apache.org/>

⁹ <http://catalogue.fiware.org/enablers/bigdata-analysis-cosmos>





Project

current project: **FITMAN**

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots**
- Access & Security

Manage Network

- Networks
- Routers
- Network Topology

Images & Snapshots

Logged in as: fitman Settings Help Sign Out

Project (0) Shared with Me (0) Public (20) [+ Create Image](#) [Delete Images](#)

Image Name	Status	Public	Format	Actions
SO CentOS 7.2	Active	Yes	QCOW2	Launch
ubuntu 15.04	Active	Yes	QCOW2	Launch
Ubuntu 14.10	Active	Yes	QCOW2	Launch
SO Fedora 22	Active	Yes	QCOW2	Launch
winecloud-img	Active	Yes	QCOW2	Launch
cloud-rendering-3.3.3	Active	Yes	QCOW2	Launch
interface-designer-3.3.3	Active	Yes	QCOW2	Launch
augmented-reality-img	Active	Yes	AMI	Launch
VirtualCharacters	Active	Yes	QCOW2	Launch
RealVirtualInteractionGE-3.3.3	Active	Yes	QCOW2	Launch
3D-UR-XML-3D	Active	Yes	QCOW2	Launch
GIS-3.3.3	Active	Yes	QCOW2	Launch
repository-image-R3.2-2	Active	Yes	AMI	Launch
MarketPlace-2-sapn	Active	Yes	AMI	Launch
edda-ssc-img	Active	Yes	QCOW2	Launch
2d4u-3.3.3	Active	Yes	QCOW2	Launch
empty20g	Active	Yes	QCOW2	Launch
SO Ubuntu 10	Active	Yes	QCOW2	Launch
SO Ubuntu 14	Active	Yes	QCOW2	Launch
SO CentOS 6.5	Active	Yes	QCOW2	Launch
ServiceRegistry-3.3 registry-0.1.0	Active	Yes	AMI	Launch
MediaEnhancedQueryBroker-2.3	Active	Yes	AMI	Launch
JobScheduler-2.3	Active	Yes	QCOW2	Launch
CompressedDomainVideoAnalysis-2.3	Active	Yes	QCOW2	Launch
DataHandling-2	Active	Yes	QCOW2	Launch
DbAcronymizer-1.3	Active	Yes	QCOW2	Launch
SO Ubuntu 12	Active	Yes	QCOW2	Launch
SO Fedora 20	Active	Yes	QCOW2	Launch
TestVM	Active	Yes	RAW	Launch

Displaying 20 items

Instance Snapshots

[Delete Snapshots](#)

Image Name	Status	Public	Format	Actions
IoTBroker 4.3.3	Active	Yes	QCOW2	Launch
Clan 2.4	Active	Yes	QCOW2	Launch
Authorization PDP - AuthZforce 4.2.0	Active	Yes	QCOW2	Launch
Store - WStore	Active	Yes	QCOW2	Launch
Configuration Manager - IoT Discovery	Active	Yes	QCOW2	Launch
Gateway Data Handling GE - Exp4fastData	Active	Yes	QCOW2	Launch
RevenueSettlementAndSharingSystem-3.5.2	Active	Yes	QCOW2	Launch
Network Information and Control - OFNIC 3.3	Active	Yes	QCOW2	Launch
Spagetti 5.1	Active	Yes	QCOW2	Launch
Fitman_RS_20150428	Active	No	QCOW2	Launch More ...
Fitman_DH_20150428	Active	No	QCOW2	Launch More ...
Fitman_NC_20150428	Active	No	QCOW2	Launch More ...
Fitman_CB_20150428	Active	No	QCOW2	Launch More ...
2030CaptureR1	Active	Yes	QCOW2	Launch
ComplexEventProcessing-2.3	Active	Yes	QCOW2	Launch
Repository - SAP RI - 3.2.1	Active	Yes	QCOW2	Launch
Marketplace - SAP RI - 3.2	Active	Yes	QCOW2	Launch
Stream oriented 3.4.3 Kurento-4.3.4	Active	Yes	QCOW2	Launch
PoDataProvider-3.3.3	Active	Yes	QCOW2	Launch
ApplicationMashupWinecloud-3.4.2	Active	Yes	QCOW2	Launch

Displaying 20 items | [More >](#)

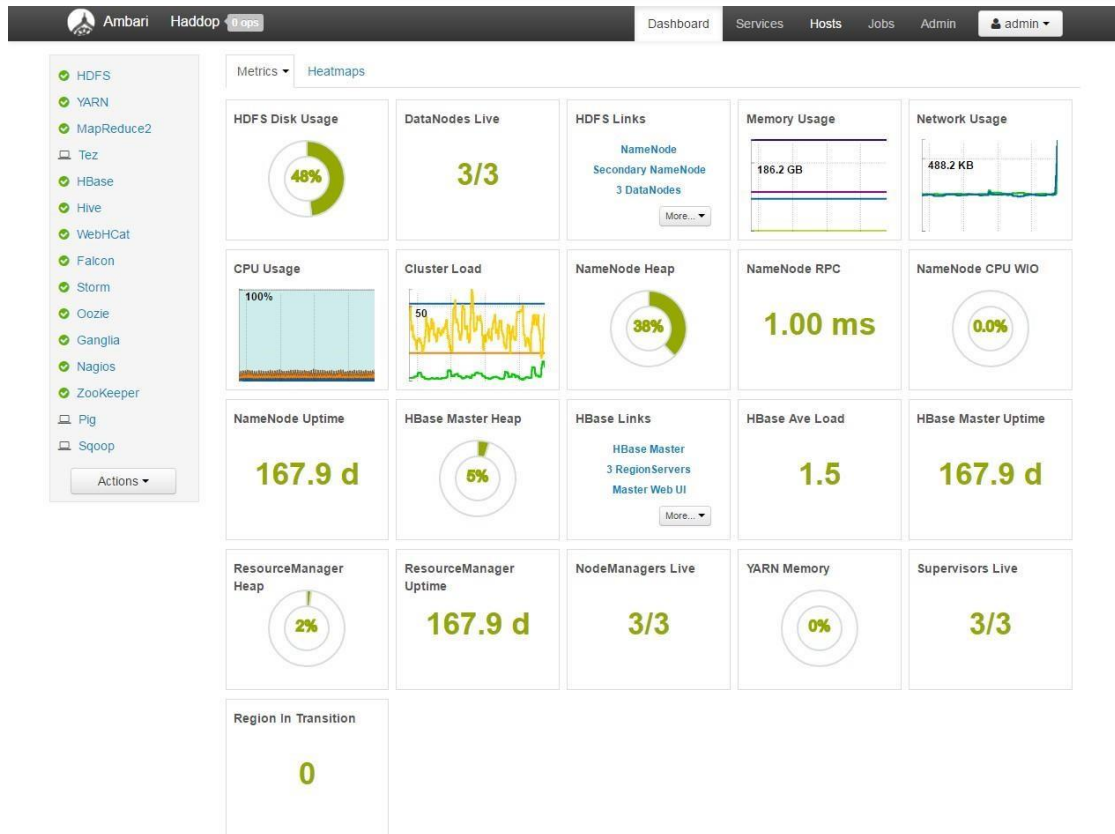
Volume Snapshots

Name	Description	Size	Status	Volume Name	Actions
No items to display.					

Displaying 0 items

Figure 4 - Engineering's FIWARELab GE catalogue management interface





Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors

Figure 5 - Engineering's Hadoop cluster management interface

Engineering's own FIWARE Lab computing facility was a key asset for the FITMAN project, where it served mainly as the cloud computing infrastructure backing the Whirlpool industrial pilot. In that context, the core software deployed was FITMAN DyCEP and FIWARE Cosmos – plus some additional pilot-specific components. In

BEinCPPS, which again sees both Engineering and Whirlpool as partners exploiting FITMAN results, we have started anew from there: the DyCEP and Cosmos components define the *CPPS Big Data & Event Processing* functional block of the BEinCPPS' FI Platform. For this BEinCPPS D3.3 deliverable, DyCEP and Cosmos were decoupled from the Whirlpool environment and made available as a general purpose Big Data and Event Processing engine. Figure 6 shows the three computing nodes of the DyCEP installation, one of them (BeInCpps1) being the service endpoint accessible from the public Internet. A shared FIWARE BigData GE virtual instance runs on Engineering's Hadoop cluster, providing the functionality that empowers DyCEP with machine learning capabilities.



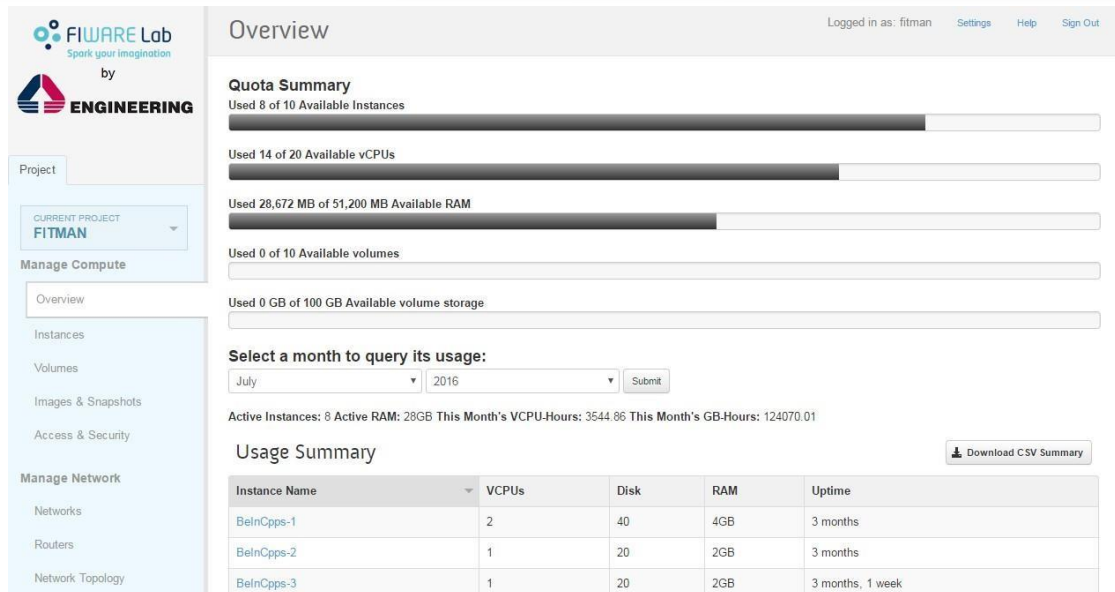


Figure 6 - BEinCPPS resources on Engineering's FIWARE Lab infrastructure

Overall, the portion¹⁰ of Engineering's FIWARE Lab infrastructures dedicated to BEinCPPS has 20 CPUs and 50GB of RAM available for running virtual machine images. Currently, only 14 CPUs and 26GB of RAM are in use, which leaves room for expansion.

Figure 7 below is an informal UML deployment diagram that gives an overview of the DW components deployed on ENG's facilities and of their service endpoints.

¹⁰ *Tenant*, in OpenStack jargon.



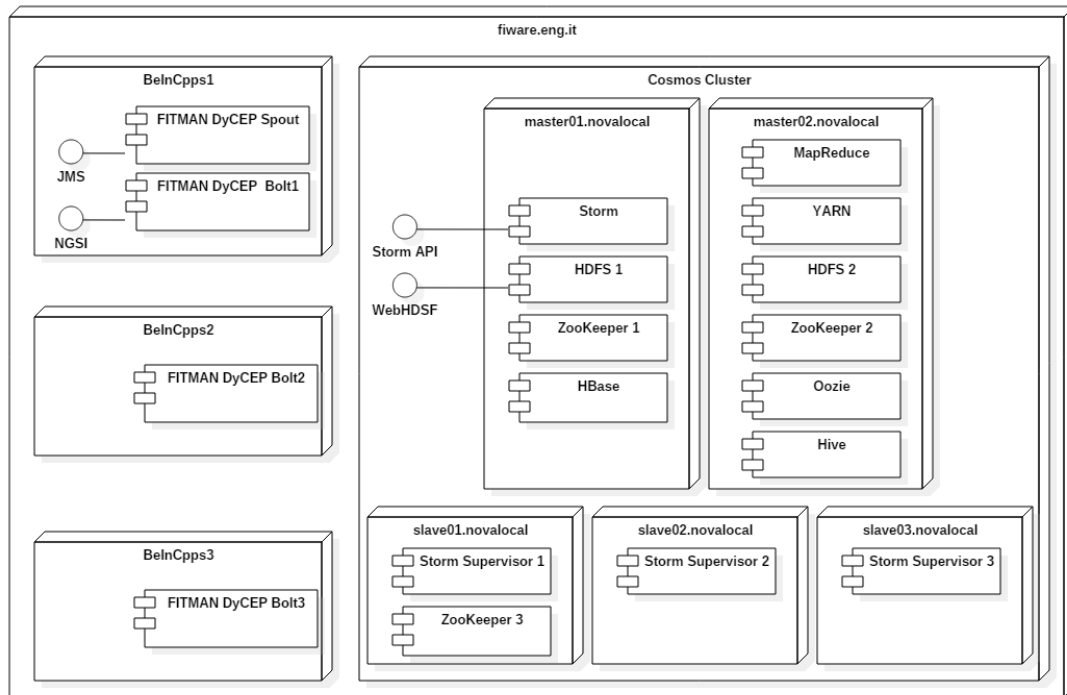


Figure 7 - DW deployment diagram on Engineering's infrastructure

This cloud environment is an integral part of Engineering's federation of corporate data centres. As such, it is not accessible from the public Internet by default. To selectively enable access, a *reverse proxy* system is in place: all server instances have a private IP address, and one single facility provides a *port mapping* service that makes specific IP-port pairs addressable through a public URL. For example, the DyCEP service endpoint is reachable here: <http://dycep-beincpps.fiware.eng.it/>. URLs like this may have additional security constraints enforced at the network level. Typically, only registered IP addresses are allowed to connect. This is not a big issue (just an administrative overhead) when such services are the Cloud back-end of some manufacturing application, as in BEinCPPS pilot scenarios. However, the whole point of having a CCE hosting DW components is making them easily available in a variety of scenarios, including the development of new applications (see opening of section #2). This requirement may be better addressed by a more open infrastructure and a different strategy for the virtualization of services – and this is the case of the solution described in the next section.

2.2 Commercial ISP's environment

To meet the challenge posed by task 2.5 and by BEinCPPS' Open Calls, the DW



needed a hosting environment that was not proprietary and, generally speaking, a more lightweight and flexible mechanism for the dynamic deployment of ad-hoc services. Regarding the former, we resorted to a high-quality but *commoditized* commercial service: a *dedicated server* with a *web-based management interface* (Figure 8), provided by [OVH](#).

[COMMENT BEINCPPS: Due to proprietary content, the figure has been removed for the public version of this deliverable]

This computing platform, physically located in the Strasbourg area, sports 32 CPU cores (Xeon E5-2630), 128GB RAM, 4TB of disk storage, 1Gbps Internet bandwidth and runs a CentoOS 7.2 operating system. It is managed remotely by Engineering. The **dw1.beincpps.eu** DNS name has been registered and points to its public IP address (149.202.205.175).

The latter point, however, was addressed by means of a de-facto standard for *containerbased virtualization*¹¹: [Docker](#). Most open source DW components have been deployed on BEinCPPS' dedicated server as Docker instances, starting from a componentspecific base image on [DockerHub](#)¹² and then applying a custom Dockerfile¹³. In practical terms this means that, with the help of the information resources provided in section #3, anyone can replicate the BEinCPPS setup on their own premises on a percomponent basis, and easily adapt its configuration to match the different environments.

Figure 9 is an informal UML deployment diagram that gives an overview of the DW components deployed on OVH's facilities and of their service endpoints – the numbers corresponding to the TCP port. **Some of these, which are highlighted in grey, are actually DW APIs endpoints** – see section #4 for details.

¹¹ In container-based virtualization lightweight virtual machines (aka containers) run application software inside as an isolated environment without the need of virtualizing the operating system as well.

¹² DockerHub is a global, public repository for the distribution of base images

¹³ A Dockerfile is a script for automated software installation and system configuration



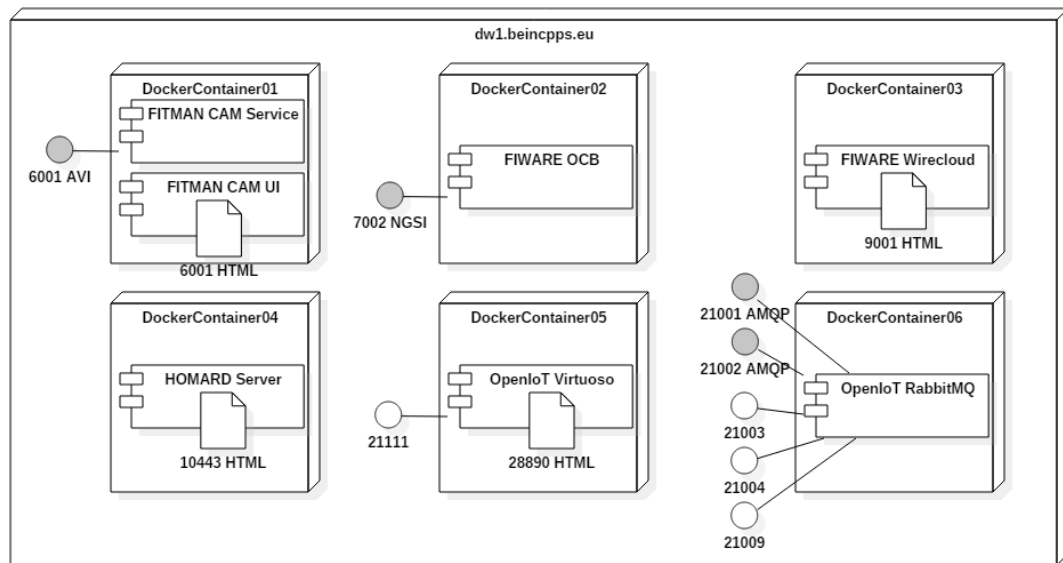


Figure 9 - DW deployment diagram on commercial provider's infrastructure

2.3 Holonix' environment

i-LiKe and i-LiKe Machine are commercial software owned by Holonix. As such, they are only available in SaaS mode in the scope of the BEinCPPS project. Holonix does not disclose details about the data centre(s) used for providing this service, as they are not relevant to the objectives of this deliverable. However, the computing resources that have been allocated to run these processes are documented in sections #3.7 and #3.9. The **dw2.beincpps.eu** DNS name has been registered and points to the public IP address of these systems (95.110.200.113).

2.4 Environment federation

The three computing infrastructures described above are the building blocks of the CCE. However, to fulfil its role as a shared computing infrastructure for the benefit of BEinCPPS pilots and Open Call winners alike, the CCE must appear from the outside as a single, homogenous virtual environment that does not require its users to know anything about its physical structure.

In our design what glues the three physical environment together, realizing a federated environment, is access control: a central authority and unified mechanism which is responsible for the secure identification of users (authentication) and access policy enforcement (authorization). The technical means to implement such functionality on top of the existing infrastructure as an add-on layer are the same used to achieve a similar goal for the FIWARE Core Platform: an Identity Management (IDM) GE and a Policy Enforcing Point (PEP) GE. The former is in charge of validating user credentials and of releasing *security tokens* that can be used to access any secured DW component across the CCE; the latter is a transparent service proxy that can be configured to perform role-based authorization checks on any access to secured DW components, without the need for them to support (or even know about) such checks directly. Both GEs have existing reference implementations



freely available as open source software: the KeyRock IDM and the Wilma PEP, both from Universidad Politécnica de Madrid (UPM).

These FIWARE components are not considered part of the DW but rather of the infrastructure, as they only address non-functional requirements. The current version of the CCE does *not* integrate this security layer yet. However, Figure 10 shows how the CCE will look like in its final release, with each physical DW environment protected by a dedicated PEP instance and one central IDM service running on ENG's premises (the *main* data centre).

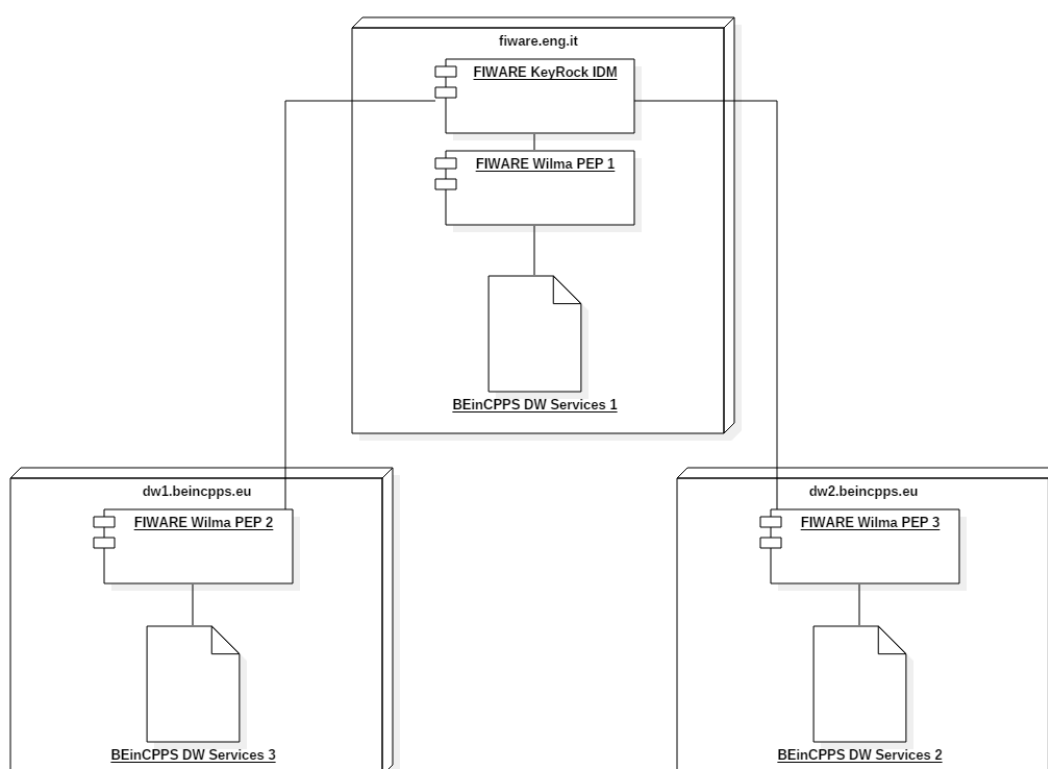


Figure 10 - CCE authentication and authorization layer

3 Deployment

[COMMENT BEINCPPS: Due to proprietary content, this chapter has been removed for the public version of this deliverable]



4 Integration Interfaces

The DW as deployed on the CCE supports integration with the Field Level through **RW Gateways** and with the external world (like legacy systems and manufacturing applications) by means of **DW APIs**. The main difference between the two concepts is that while the former are a collection of software modules, the latter are the API exposed by a specific software component. Moreover, RW Gateways are an *Edge Computing* facility, and as such their positioning in the BEinCPPS architecture is at the Factory Level: they represent the boundary between the Field and the Cloud.

RW Gateways are two distinct families of DW components: FIWARE IDAS Agents and OpenIoT Wrappers. They provide Field Level systems with direct access to the Fast Lane and the Smart Lane of the FI Platform, respectively. Both Agents and Wrappers are a generic term that identifies a range of *adapters* that are individually specific to some Field Level communication protocol – e.g., MQTT and OPC UA. Typically, such adapters are run in close proximity to the actual Field devices they are interfacing. For example, an OPC UA Agent may reside on a Factory server, providing a bidirectional communication channel between a local industrial automation Smart System on the Field side, and the Information Bus on the Cloud side. Given their *Cloud proxy* role, neither IDAS Agents nor OpenIoT Wrappers have been included in this CCE deployment.

On the Cloud side, DW APIs are exposed by OpenIoT Middleware, FIWARE Orion Context Broker and FITMAN CAM. These are platform-level endpoints, as they enable applications to interact with the DW as a whole, and indirectly with the RW as well. In the subsections that follow we describe the functional scope of such endpoints; for technical details you can refer to the documentation released and/or linked in the D2.5 “FI Platform” deliverable. Figure 11 illustrates the big picture of RW Gateways (red boxes) vs. DW APIs (orange boxes).



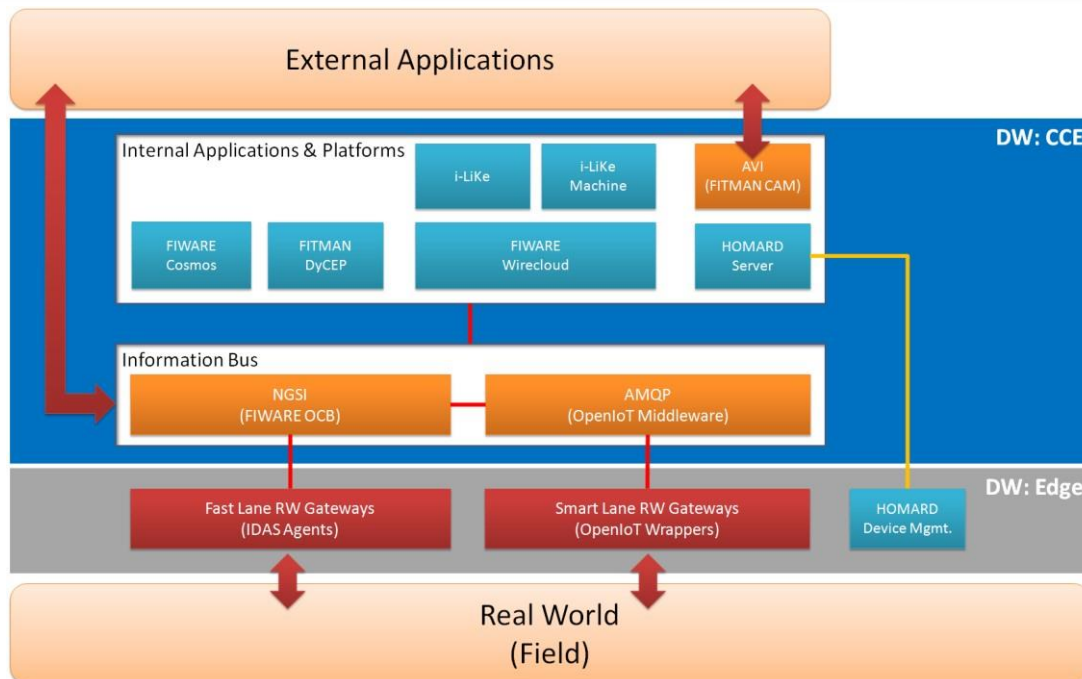


Figure 11 - Digital World: APIs and RW Gateways

3.1 AMQP

The Advanced Message Queuing Protocol (AMQP)¹⁴ is an ISO/IEC 19464 open standard for message-oriented interoperability. It is used by RabbitMQ¹⁵, the message broker adopted by OpenIoT Middleware, to provide an asynchronous, binary, wire-level protocol that any authorized client can leverage to interact with all systems connected to the FI Platform's Information Bus. This interaction happens at the lowest of levels: raw data that is exchanged between systems that agree on a common data model. An example of such systems is given by RW Gateways – more specifically by OpenIoT Wrappers, which implement the Smart Lane of the IoT Platform. *External* systems, such as domain- or use-case-specific applications, are offered the same access path: by leveraging the AMQP endpoint, these applications can then integrate themselves as first-class citizens of the DW.

¹⁴ <http://www.amqp.org/>

¹⁵ <https://www.rabbitmq.com/>



3.2 NGSI

The Next Generation Services Interface (NGSI)¹⁶ is an Open Mobile Alliance (OMA)¹⁷ open standard that targets cross-platform context information sharing. While the OMA specification is technology-agnostic, the FIWARE Orion Context Broker implementation is HTTP-based: as such, it favors openness over performance. The NGSI API plays the exact same role of the AMQP one (see previous section): providing a DW entry point point for RW Gateways (the Fast Lane of the IoT Platform) and for external applications.

3.3 AVI

The Asset Virtualization Interface (AVI) is the REST-based Open API provided by FITMAN CAM. It is based on a high-level abstraction of the RW: *manufacturing assets* – which are any item of interest for the business process – that are represented as classes, named individuals and attributes of a domain-specific, customizable OWL2 ontology¹⁸, and then exposed as *things* on the FI Platform's Information Bus (more specifically, as *named contexts* in FIWARE Orion Context Broker).

3.4 DW API Recap

From the integration point of view, NGSI and AMQP serve the same purpose: providing a direct access path to the Information Bus. In both cases, data model and semantics must be shared by all participants in the conversation. When choosing an option for integration at this level, only technology-related issues make the difference – e.g., AMQP being a message-oriented binary protocol while NGSI provides a web-based publish-subscribe service. The Smart Lane vs. Fast Lane characterization does not apply in this context: RW Gateways (i.e., OpenIoT Wrappers vs. IDAS Agents) are what makes a Lane Smart or Fast.

On the other hand, AVI offers a higher-level perspective on the Real, Digital and even Virtual Worlds. Using the AVI channel to achieve system interoperability implies the definition of a common information model inside FITMAN CAM: *things* in the RW, *entities* in the DW, *models* in the VW are all represented as virtualized assets in

CAM's repository, and shared as *pub/sub contexts* (i.e., objects with properties that can be concurrently modified by *publishers*, while notification of changes are propagated to all the *subscribers*) on the Information Bus. It is however worth noting that AVI, being implemented as a REST API over a quite complex chain of software modules, is *not* suitable for driving real-time, or even near-real-time interactions.

¹⁶ <http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/ngsi-v1-0>

¹⁷ <http://openmobilealliance.org/>

¹⁸ <https://www.w3.org/TR/owl2-overview/>



5 Conclusion

The CCE that was described in this document is, at the time of writing, actively used by some of the BEinCPPS pilots, namely the Lombardy and the Rhone-Alps regional champions – for details, see the D4.3 and D8.3 deliverables.

That said, this federated infrastructure is still in its infancy: some key aspects are missing or under-developed, in particular with respect to security. Securing all service endpoints and interactive user interfaces under one single domain is on the roadmap of the next release, which is planned for M24. As said in section #2.4, the technical means for achieving this objective have been identified, and the design of the security layer is basically done. In the coming months this design will be put at test as the integration of the FIWARE security GEs will proceed. Moreover, some administrative issues will be addressed, such as the procurement and installation of digital certificates that will protect the confidentiality of the communication channels.

It should also be noted that the boundaries of the BEinCPPS Digital World may change in the future, as new *external* platforms will be considered for inclusion. This process stems from the revision of the overall BEinCPPS architecture that is planned in the scope of WP2.1 and will be documented in the D2.2 report, to be released by M21. Our plan in WP3.3 is to keep the CCE up with the evolution of the BEinCPPS architecture, but shifting its role: an environment for the deployment of production-grade, BEinCPPS-based solutions rather than a *testbed* for the full catalogue of DW components.

