Ecosystem for COllaborative Manufacturing PrOceSses – Intra- and Interfactory Integration and AutomaTION
(Grant Agreement No 723145)

D7.8 Lab Scale Use Case Deployment with Lessons Learnt I

Date: 2017-12-19

Version 1.0

Published by the COMPOSITION Consortium

Dissemination Level: Public

Co-funded by the European Union's Horizon 2020 Framework Programme for Research and Innovation under Grant Agreement No 723145

Document control page

Document file: D7.8 Lab Scale Use Case Deployment with Lessons Learnt I v1.0.docx
Document version: 1.0
Document owner: TNI

Work package: WP7 – Integration of Internal and External Elements
Task: T7.4 – Components Integration
Deliverable type: R

Document status:
☒ Approved by the document owner for internal review
☒ Approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Mike Hayes, Willie Lawton, TNI Marc Jentsch, FIT	2017-10-24	(Strawman for initial discussion with CNET & ISMB)
0.2	Mathias Axling CNET, Giuseppe Pacelli ISMB	2017-10-31	Approval from CNET & ISMB to circulate, ok to use D2.4 graphics
0.3	Mathias Axling CNET, Javier Romero Negrin ATOS, Paolo Vergori, Giuseppe Pacelli ISMB, Alexandros Nizami, Dimosthenis Ioannidis CERTH, Matteo Pardi, NXW , Jose Angel Carvajal Soto FIT, Constantinos Ziazios, Ifigeneia Metaxa ATL	2017-12-01	Contributions from ATOS, CNET, ISMB, NXW & CERTH, FIT, ATL
0.4	Jesper Thestrup IN-JET, Dimitris Gkortzis ELDIA	2017-12-13	Peer review inputs from ELDIA & IN-JET. Added legal notice, exec summary, list of figures and tables. Adjusted to align with deliverable template.

Internal review history:

Reviewed by	Date	Summary of comments
Dimitris Gkortzis (ELDIA)	2017-12-04	No changes
Jesper Thestrup (IN-JET)	2017-12-13	Minor comments and several formatting errors corrected.

Contents

1 Executive Summary

This deliverable outlines the architecture defined for integration activities, a brief overview of the components developed so far as well as any initial findings and lessons learnt during the initial integration activities in a lab scale environment. It also includes an integration plan for the inter-factory and intra-factory use cases and a set of recommendations, conclusions and next steps. An ISMB docker server has been set up that acts as our lab scale environment where each COMPOSITION partner dockerizes and deploys their component.

A reference architecture has been successfully developed that is now being used in individual use cases. To date the following components have been developed based on this architecture, where possible leveraging from commercially available software modules and open source widely used protocols. A detailed description of the functionality of each of these components can be found in section 5 and a tabulated summary found in section 10.

- Authentication
- Big Data Analytics
- BlockChain Connector
- Commissioning System
- Deep Learning Toolkit
- Intrafactory Interoperability Layer
- Manufacturing Big Data Storage

- Manufacturing Decision Support System
- Market Event Broker
- MatchMaker
- Real Time Multi Protocol Event Broker
- Requester Agent
- SIEM (Security Information and Event Management)
- Simulation and forecasting tool
- Supplier Agent

To development and dockerisation state of the components have been outlined, along with action plans for use in inter and intra-factory use cases.

At a high level there are two communication mechanisms in the COMPOSITION system: message-based communication over MQTT or AMQP using the Event Broker[1], and request-response REST HTTP interfaces (D2.3 The COMPOSITION Architecture Specification I, 2017).

With a few exceptions[2], the components are loosely coupled and integrated in the system by conforming to a common communication infrastructure and data schemas. There are dependencies on data generated by other components, but for most of the data, the dependency is not direct as this data is distributed through the event broker, so the direct connection is to this component.

2 Introduction

2.1 Purpose, context and scope of this deliverable

This deliverable outline technology components and efforts by COMPOSITION partners in preparing for the initial lab scale use case deployment. These deployments are critical preparation steps for the technology components and testing their inter-operability and suitability for the various industry use cases. This includes a definition of the architecture devised and an overview of the components developed so far. All of the use cases can be described as either 'inter-factory' and 'intra-factory' and an integration plan is outlined with latest status update at the time of writing. An ISMB docker server has been set up that acts as our lab scale environment where each COMPOSITION partner dockerizes and deploys their component and the current status of each technology component outlined.

It also captures with lessons learnt, conclusions and recommendations.

2.2 Content and structure of this deliverable

The architecture is described in section 2, their development status described in section3 and inter and intra-factory action plans outlined in sections 7 & 8. In section 6 an outline of findings and lessons learnt so far is compiled which includes its use of and compatibility of each component with off the shelf (OTS) software.

3 Technology Component Architecture

The diagram fig. 1 below describes the COMPOSITION system from a business architecture functional view. The Agents are the intra-factory interface to the inter-factory (Marketplace) responsible for interacting and sharing data with other actors in the Marketplace. Agents and user interfaces (HMI) depend on tools for modelling, simulation and analysis of factory data and processes. Business functionality, e.g. manufacturing analysis, is implemented by components providing generic functionality for complex event processing (CEP) and neural networks (deep learning). These in turn use the communication functionality, the intra-factory interoperability and shop-floor connectivity functional packages. The inter-factory packages provide the necessary infrastructure for the agents to find other agents, negotiate and enter agreements. The components are described in section 5.

[1] Real-time Event Broker, Message Broker
[2] E.g. Big Data Analytics and Deep Learning networks are tightly integrated.

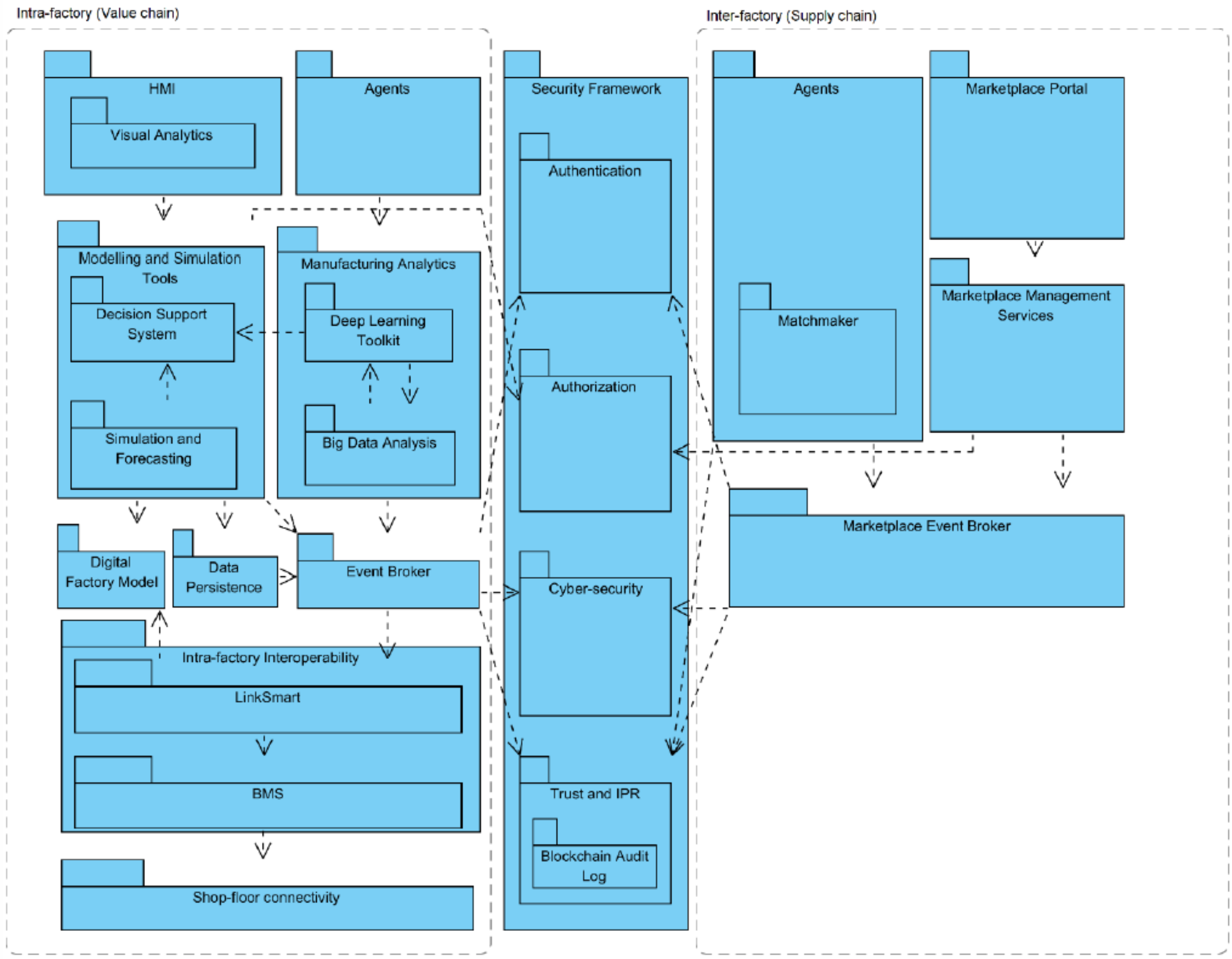
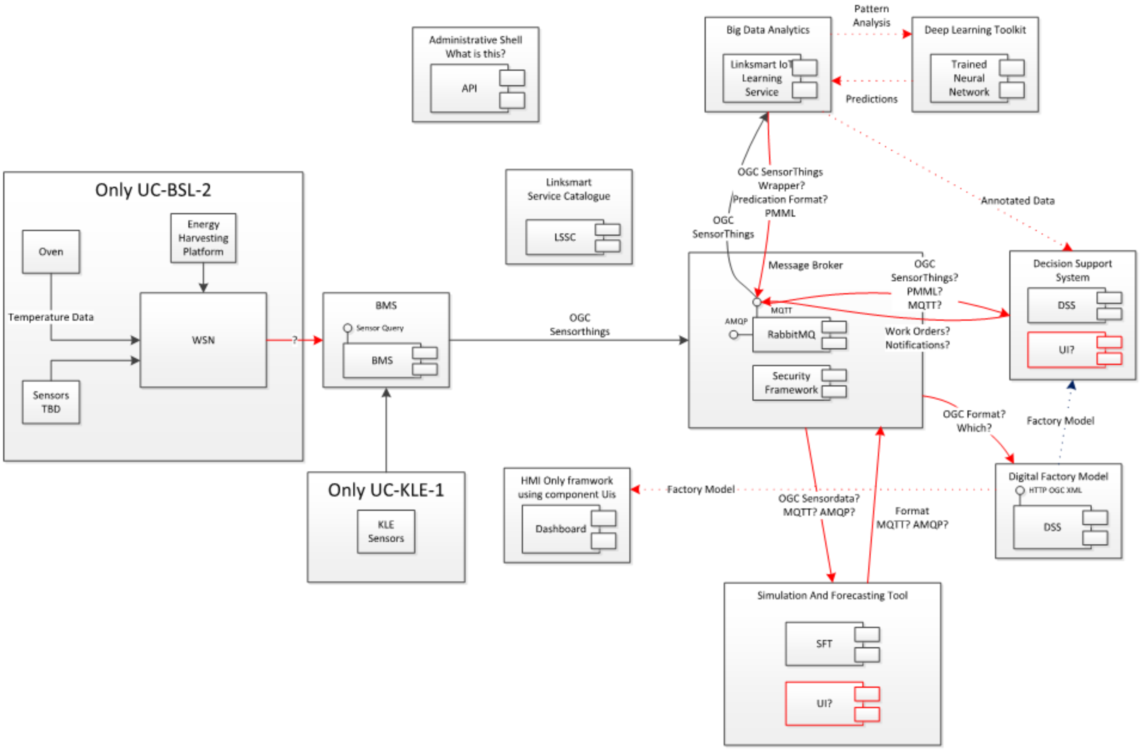


Figure 1: High-level functional view of COMPOSITION architecture.

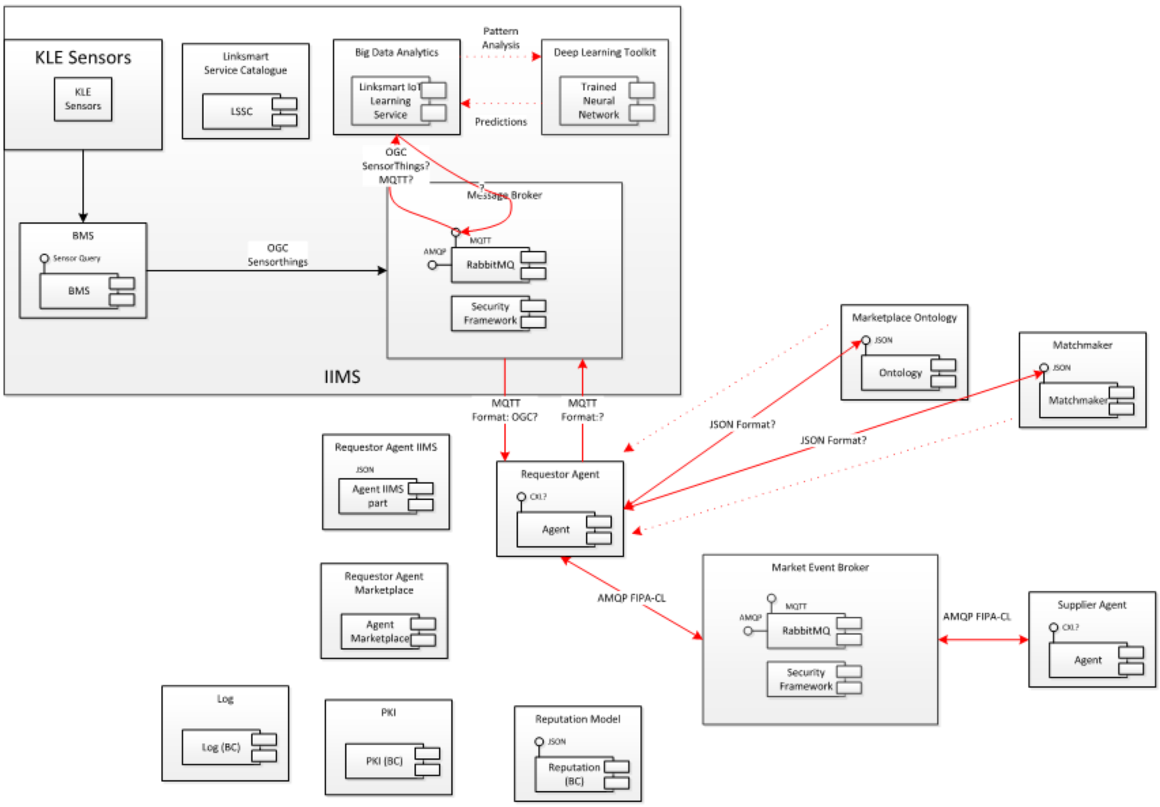
There are two communication mechanisms in the COMPOSITION system: message-based communication over MQTT or AMQP using the Event Broker[3], and request-response REST HTTP interfaces (D2.3 The COMPOSITION Architecture Specification I). With a few exceptions[4], the components are loosely coupled and integrated in the system by conforming to a common communication infrastructure and data schemas. There are dependencies on data generated by other components, but for most of the data, the dependency is not direct as this data is distributed through the event broker, so the direct connection is to this component.

This architecture is then used to draw out more detailed flow path for specific use cases. Here are 2 examples:

[3] Real-time Event Broker, Message Broker
[4] E.g. Big Data Analytics and Deep Learning networks are tightly integrated.



UC-BSL-2 & UC-KLE1



UC-KLE-4

Figure 2: Use case flow path examples

4 Summary of Technology Components in Development

The following table provides an outline of the technology components in developed in COMPOSITION and their status at the time of writing. Further details can be found in sections 5.

Technology component	Owner	Dockerised?
Access Control	ATOS	No, but underway
Agent Marketplace	ISMB	No
Authentication	ATOS	Yes
Big Data Analytics	FIT	Yes
BlockChain Connector	CNET	Yes
Commissioning System	CNET	No
Deep Learning Toolkit	ISMB	Very first draft version dockerized
Intrafactory Interoperability Layer	ISMB	Not a component itself, but the baseline of sub-components for inter-operability is dockerized
Manufacturing Big Data Storage	NXW	Planned
Manufacturing Decision Support System	ATL	Underway
Market Event Broker	CNET	Yes
MatchMaker	CERTH	Yes
Real Time Multi Protocol Event Broker	CNET	Yes
Requester Agent	ISMB	No
SIEM	ATOS	No
Simulation and Forecasting Tool	CERTH	Planned by M18
Supplier Agent	ISMB	No

Table 1: Technology components developed

5 Detailed Description of Technology Components

The following is a detailed description of the technology components developed, each marked with the lead COMPOSITION partner.

5.1 Access Control

Access Control[5] is provided by EPICA, an Authorization component based on XACML v3.0[6] that provides an attribute-based access control mechanism. It provides the means to define the security policies used to protect resources, for afterwards any request to access a protected resource will first be evaluated against these policies and the evaluation result will be enforced depending on the outcome.

The following Figure 3 presents an overview on the architecture of EPICA

[5] D4.1 Security Framework I Section 4.2
[6] https://www.oasis-open.org/committees/xacml/

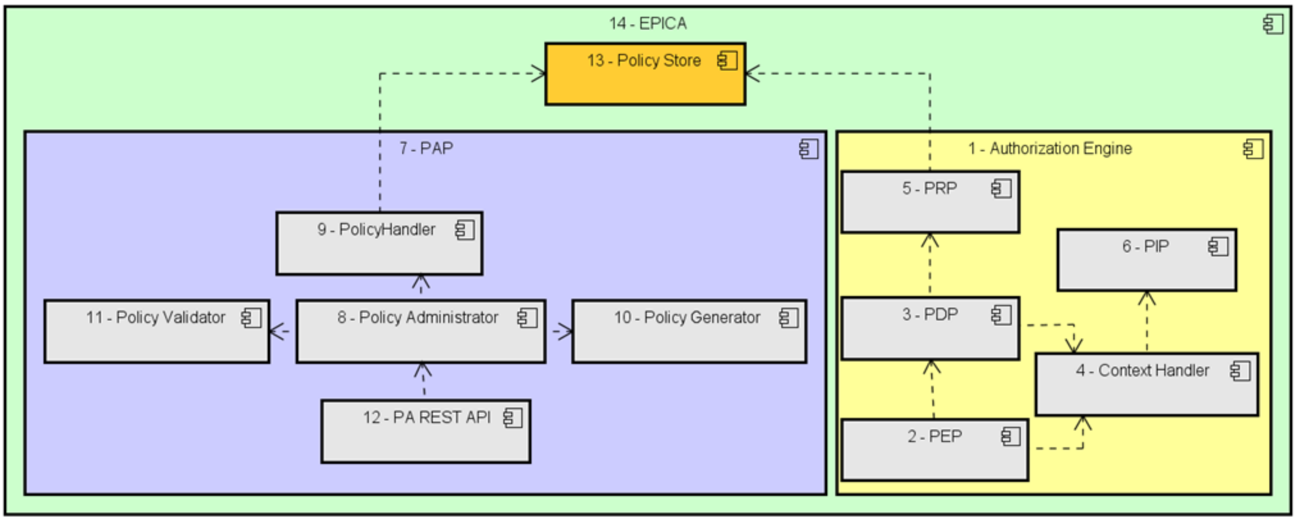


Figure 3: EPICA Architecture

To evaluate the access to a resource EPICA needs a token, with the user information, and the resource to be accessed. The token, in this case is the one obtained from COMPOSITION Authentication component when a user is authenticated.

The following Figure 4 offers an overview of the possible interactions between the Access Control component and other components that can need authorization mechanisms within COMPOSITION.

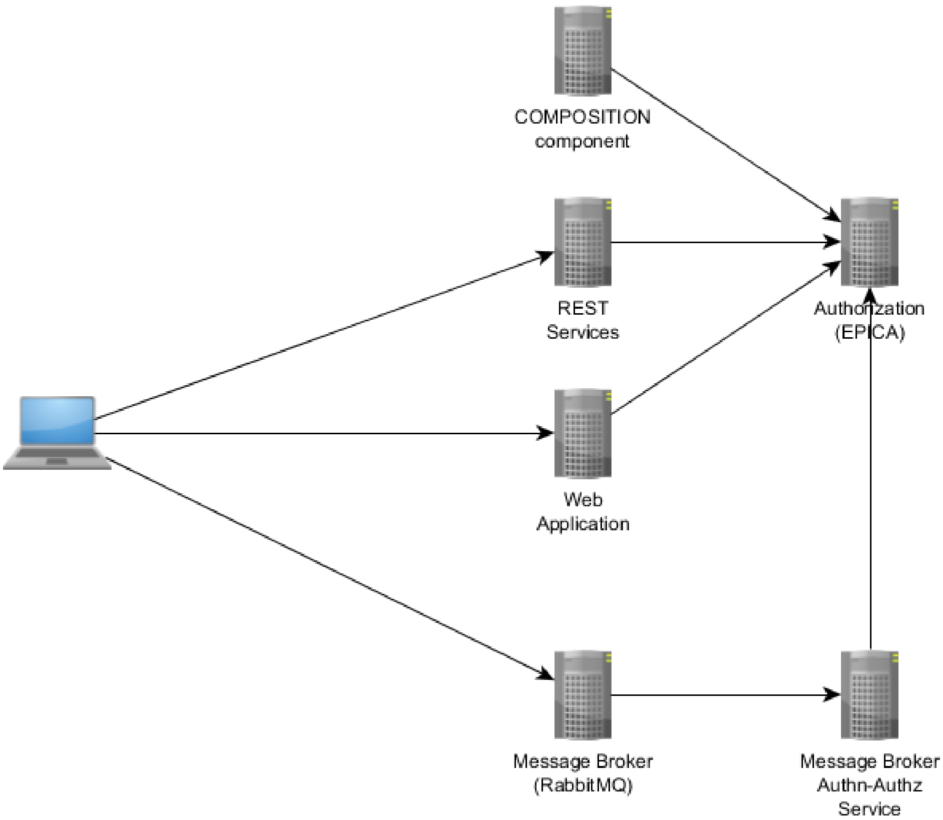


Figure 4: Interaction between Authorization and other components

More information on Authorization component can be found in D4.1 Security Framework I Section 4.2

The component is currently being dockerized. Once deployed the rules for accessing resources need to be defined depending on the needs and will be on-demand as other components are being deployed.

5.2 Agent Marketplace

The COMPOSITION Agent Marketplace is the container of the COMPOSITION agents. Agents within the Marketplace may implement market-specific services (such as the whitepages or the Matchmaker), or they can act on behalf of industry stakeholders participating in the Marketplace. Required communication infrastructure is provided by a suitable message broker (namely the Marketplace Event Broker), which provides message delivery services to all other components through a well-known, publish-subscribe, interaction paradigm. (See D2.3) It exists a well-defined set of messages, in JSON format, that define the different communications between interacting agents.

5.3 Authentication

The COMPOSITION Authentication[7] component is responsible for providing authentication mechanisms for users, applications, services and devices; and it´s based on Keycloak[8] open-source system.

From the available standard authentication protocols in Keycloak, COMPOSITION Authentication component makes use of Open ID Connect protocol (OIDC) which is based on OAuth 2.0[9] and unlike it, it´s an authentication and authorization protocol.

One of the major features of Keycloak is the possibility to customize the Authentication Service through the Service Provider Interface (SPI) framework which offers the possibility to implement custom providers or override built-in ones. This feature is used to provide authentication to COMPOSITION Message Broker components overriding built-in authentication mechanisms, with this approach there is only one centralized point for authentication and user management.

The following figure 5 offers an overview of the possible interactions between the Authentication component and other components that can need authentication mechanisms within COMPOSITION.

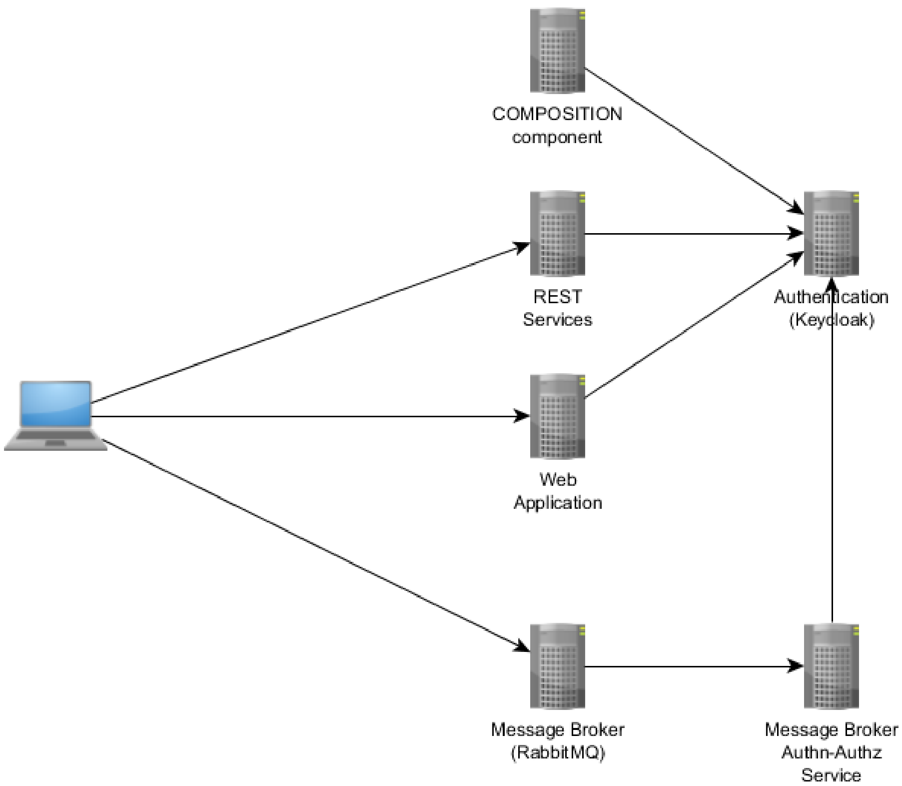


Figure 5: Possible interaction between Authentication and other components needing authentication

[7] D4.1 Security Framework I Section 4.1
[8] http://www.keycloak.org/
[9] https://oauth.net/2/

The component is currently dockerized and deployed although configuration is still needed and will be done on demand depending on the services, applications or devices to be secured, as well as the users needed.

More information on Authentication component can be found in D4.1 Security Framework I Section 4.1

5.4 Big Data Analytics

For the analysis of the incoming data, we will used a tailored version of the LinkSmart® IoT Learning Agent. The LinkSmart® IoT Learning Agent was developed for all kind of store-less data processing, from simple data annotation or aggregation to complex data machine learning techniques. The agent is ideal for intelligent on-demand data management or analysis in IoT environments, from edge computing to cloud computing.

The LinkSmart® IoT Learning Agent provides Complex-Event Processing as service and Real-time Machine Learning Orchestration as a service. The agent provides three APIs, the Stream Mining API (Statement API), the Learning API (CEML API) and the IO API. The Statement and CEML (see below) APIs are CRUD (Create, Read, Update, Delete) and JSON based, while the IO are write-only (for Input) or read-only (for Output). The APIs are implemented as HTTPs RESTful and MQTT.

The Complex-Event Machine Learning (CEML) framework is a methodology that allows to predefine the learning phases of the phenomena to learn. In this manner, the learning process can be unattended orchestrate and distributed and remotely manage in all kind of industrial environments. For more information about the LinkSmart® Learning agent can be found in the deliverable D5.1 Big Data Mining and Analytics Tools I.

5.5 BlockChain Connector

COMPOSITION will use blockchain technology to provide a log of transactions that will ensure the integrity and non-repudiation of messages, e.g. agent negotiation and contracts in the marketplace or material/shipment tracking in the factory. The blockchain connector is the component that interfaces COMPOSITION, implements the COMPOSITION specific functionality and interacts with a blockchain implementation. Multichain, an open source product implementing the Bitcoin protocol with some additions, has been selected as the blockchain implementation. Multichain is relatively easy to configure and deploy, not dependent on an underlying “currency” to function, very versatile and free to use as deployed by COMPOSITION. The blockchain connector can work off a specific exchange in the Rabbit MQ event broker using the shovel as described in (D6.1 Real-Time Event Broker I), where all events sent to this exchange are automatically forwarded to the regular exchange, while also being processed by the blockchain connector. The exact integration with the broker is still in development. The means of verification against the chain is also in development.

A blockchain adapter will also be developed to store and retrieve the public keys needed by the subscribers to verify the signature of message received on the message queue. Each participant in COMPOSITION will need to deploy a blockchain node to have access to the public key and thus be able to verify the signature of messages.

A proof-of-concept has developed and deployed on Docker test server for review. A similar test was performed at industry hackathon in another Docker environment. Docker images for the blockchain need a parameterized configuration for deployment.

5.6 Commissioning System

During technical scenario sessions, the need was identified for a component to configure sensor setup, e.g. PLC register mappings and hardware and software identifiers, when deploying the COMPOSITION IIMS, re-configuring or installing new sensors. The information about the deployed equipment will have to be distributed to the involved components from the Digital Factory Model instance for the factory. The component responsible for coordinating this was named the commissioning system. The responsibilities of this component are currently managed by manual configuration and static set-up.

5.7 Deep Learning Toolkit

The Deep Learning Toolkit is the COMPOSITION component in charge of providing predictions based on the analysed data concerning a fixed future timeframe.

The Core of the Deep Learning Toolkit dwells in its Artificial Neural Networks that are specifically designed for fitting the data that are fed to the component. Due to the nature of the machine learning and consequent deep learning data science, the Deep Learning Toolkit will not be able to have a single implementation to be reused among different scenarios. Instead, one instance of the component will be specifically designed for each use case, that necessary will envisage different data types. Hence, each and every of its Artificial Neural Networks is specifically designed to provide one and only one prevision.

Despite having different implementations, the Deep Learning Toolkit has a common background, which is composed by four different and completely separated instances. Every instance has one expected output that is inputted to the next instance in a waterfall schema, in an asynchronous manner. In the following, the four phases are described:

- The first phase is called Data Pre-processing and happens after the initial Data Harvesting phase. It is in charge of loading and formatting data coming from the shop floor and is a totally offline phase. This module is designed under the assumptions that data types are predefined and operatively consolidated. Therefore, it is not expected to receive batches from the shop floor with missing features. It is required that the number of input features and the batches size is predetermined in the design phase. What this module does, is to consolidate incoming data in order to transform human readable information in datasets that are allowed as input to Artificial Neural Networks. This submodule is usually applied to historical data that are retrieved from an existing information infrastructure at the end user's premises.
- The second phase is called Training and happens offline, too. In this step, a percentage between 60% and 80% of data are used from the output of the previous phase. This part of the dataset is fed to the Artificial Neural Network in order to train the network itself, alongside a list of hyperparameters. It is not worth to list and explain the hyperparameters and the reasoning behind them, which is a subject for D5.3 which it will be publicly available at M16.
- The third phase is called Validation. This is the last offline phase and it's a necessary step required by every Artificial Neural Network. The Validation submodule uses the remaining portion of the dataset left over from the Training phase and provides metrics for evaluating the reliability of the networks. As for before, used metrics and outcomes related to different validation constrains are going to be discussed in details in D5.3 which it will be publicly available at M16. This phase ends with the designer of the Artificial Neural Network manually evaluating the validation phase metrics. If they satisfy the required accuracy, the output is then deployed in the next phase.
- The fourth phase is the online step and it's what brings the system live in the factory. Therefore, it's an operational phase and is called Continuous Learning. Among the four instances only this is the only one that requires to be deployed, usually at the shop floor level at least in case of intra-factory activities. In cases where inter-factory interoperability is required, the component will follow the same deployment as the as the Marketplace Agent.
  This phase expects in input the same batches as for the Validation phase that were exactly identical to the one used to feed the Artificial Neural Network in the training phase. The main difference is that this time these batches are the result of an aggregation of live data, measured from the very same sensors, instead of being an historical collection, like the previous case. It is fundamental to notice that batch size might vary from the one used in the two previous steps, whether the number of features remains the same.
  The output of this phase provides the latest prevision on the data that will vary from the original trained Artificial Neural Network, deployed only when enough data will be provided to the network itself. So, it should not be expected in a predictive maintenance scenario which might have decades of sensors recollection and thousands of documented breakdowns, that few dozens of breakdowns events could affects the previsions. In this phase, live data collected in batches are acquired and processed automatically by the network that provides feedback in an asynchronous manner.

Each of the described phases is implemented in a standalone component that is self-contained and operates indistinctively and independently from the others. This choice has been done on purpose during the design phase, in order to allow to operate and leverage on different parameters without affecting directly all phases at ones. In fact, each of the phases depend on the input of its predecessor and produces the output for its successor, but the parameters involved in the process are independent and differ from one another in each of them.

There are still two unmentioned scenarios that are not documented here for different reasons, which are the retraining phase and the dispatching phase. The former is a command a process that is issued by an internal evaluation process in which the whole Training phase is done on the deployed network transforming tensors and neurons of the Artificial Neural Network. It is worth mentioning that this phase is not systemic not scheduled and depends only form the live batches provided. The latter is a common the implementation of a common ground that envisage actions such as authentication, authorization, publishing and subscribing for performing dispatching operations, that are common among COMPOSITION components and only few details are component dependent. As for before, both phases are going to be detailed in D5.3 Continuous Deep Learning Toolkit for Real Time Adaptation I which will be publicly available at M16.

5.8 Intra-factory Interoperability Layer

The COMPOSITION ecosystem requires a common ground for exchanging data at shop floor level. The intra-factory interoperability layer carries out the task providing an infrastructure using the publisher/subscriber paradigm. The infrastructure leverages on a broker based protocol for handling communications among the intra-factory components. Many broker based system has been tested in heterogeneous scenarios, making a final choice for a MQTT protocol supporting software, namely RabbitMQ.

In the intra-factory scenario there is the need to create a hierarchy that defines the topics structure for the event broker. The COMPOSITION project sets a background that will be common among all components that will be identified by:

- a topic root that will use the “COMPOSITION” tag as identifier;
- the discriminative dichotomy identifier of the intra or inter-factory scenario;
- the component name that is in charge of generating the data;
- the scope of the data produced.

Every component that needs to exchange information within the COMPOSITION intra-factory communication layer will be virtually demanded to use the event broker, registering a scope based topic. COMPOSITION intra-factory components will leverage on this interconnection scalability, capability and most important without the burdensome of securing yet another communication channel that would not benefit from the enhancements of the security framework that mediates access token renewals and credentials retaining.

A key component in the intra-factory interoperability layer is the Building Management System (BMS). It provides a model for interconnecting the COMPOSITION ecosystem within the shop floor acting as a translation layer. In fact, it provides connectivity from sensors to the COMPOSITION components. Within this component, information is pervasively collected from any connected systems in order to support the management operators in making decisions and to take direct control for automation tasks. Since the BMS enables the connection with all the major automation standards (such as BACnet, Konnex, Modbus, etc.) allowing to seamlessly inter-connect them altogether, it will act as a bridge between the cyber-physical systems (sensors, gateways, etc.) and the other IIMS components.

5.9 Manufacturing Big Data Storage

Manufacturing in assembly lines consist in a set of hundreds, thousands or millions of small discrete steps aligned in a production process. Automatized production processes (or production lines) produce for each of those steps small bits of data in form of events. The events possess valuable information, and additionally, the data in the events usually are meaningless if they are not contextualized, either by other events, sensor data or process context. When it’s possible to extract most value of the data, it must be process in real-time and on demand; in this manner, the data is processed at the moment when it is produced extracting the maximum value, reducing latency, providing reactivity, giving it context, and avoiding the need of archiving unnecessary data. At the same time, some valuable information must also be stored somewhere in order to be retrieved when necessary as an historical trace of what has been collected during the process lifetime.

This complex event processing and storage service is provided by the Building Management System (BMS). In the first place, the BMS provide a set of tools for collect, annotate, filter or aggregate (if needed) the real-time data incoming from the production facilities. This set of tools facilitate the possibility to build applications on top of real-time data. Secondly, the BMS provide a storage for information valuable to be kept during the whole machine lifetime. These raw measurements can also be enhanced by providing additional metadata to be attached to them, in case it should become necessary.

For the time being, implementations of the BMS adapters are still focused on the Interoperability Layer. Therefore Manufacturing Big Data Storage will be developed afterwards.

5.10 Manufacturing Decision Support System

The manufacturing decision support system brings support to a number of engineering practices supporting the manufacturing processes (see D3.8 Manufacturing Decision Support System I). It is integrated into the IIMS system and elaborates MQTT and HTTP to exchange information with other subsystems, in order to provide context for better decision support and process visualisation. Authentication, authorization and access control is implemented through the integration of security framework. DSS also collaborates with the Deep Learning Toolkit for maintenance prediction and simulation purposes. For visualisation purposes, it uses the models from simulation toolkit expressed in BPMN format. DSS using the microservice paradigm and it is packaged as a docker image for testing and production purposes. Moreover, it provides a mobile app for in-place uses.

5.11 Market Event Broker

The market event broker is the instance of the message broker used in the COMPOSITION Marketplace (see deliverable D6.1). It interacts with most components and is the hub through which marketplace agents communicate. However, this is through the standard AMQP protocol and needs no special configuration or development of the broker itself. The broker is tightly integrated with the security framework, which provides identity and access management for all brokers in COMPOSITION system (federated or clustered). Scalability configuration tests (cluster, federation setup in Docker) have not yet been performed. The REST tunnel is in proof of concept phase. The intra factory instance has been deployed and integrated with the security framework, see section 5.15.

5.12 Matchmaker

The Matchmaker is one of the core components of the COMPOSITION Marketplace. This component aims to match requester and supplier agents participating in the Marketplace based on different selection criteria. Furthermore, the Matchmaker component is used by agents in order to match requests and offers between the agents. The Matchmaker's functionality is exclusively depended on Collaborative Manufacturing Services Ontology. The Matchmaker infers new knowledge by applying rules on the knowledge stored in the Ontology.

The following picture presents the interactions of Matchmaker component. Internally, the Rule-based Matchmaker sub-component interacts with the Marketplace Ontology store. The Rule-based Matchmaker applies rules to Ontology and infers new knowledge in order to perform matching. Externally, the Matchmaker component interacts with the Marketplace's agents. The Matchmaker offers to agents matching services. Moreover, it offers services related to Ontology's manipulation.

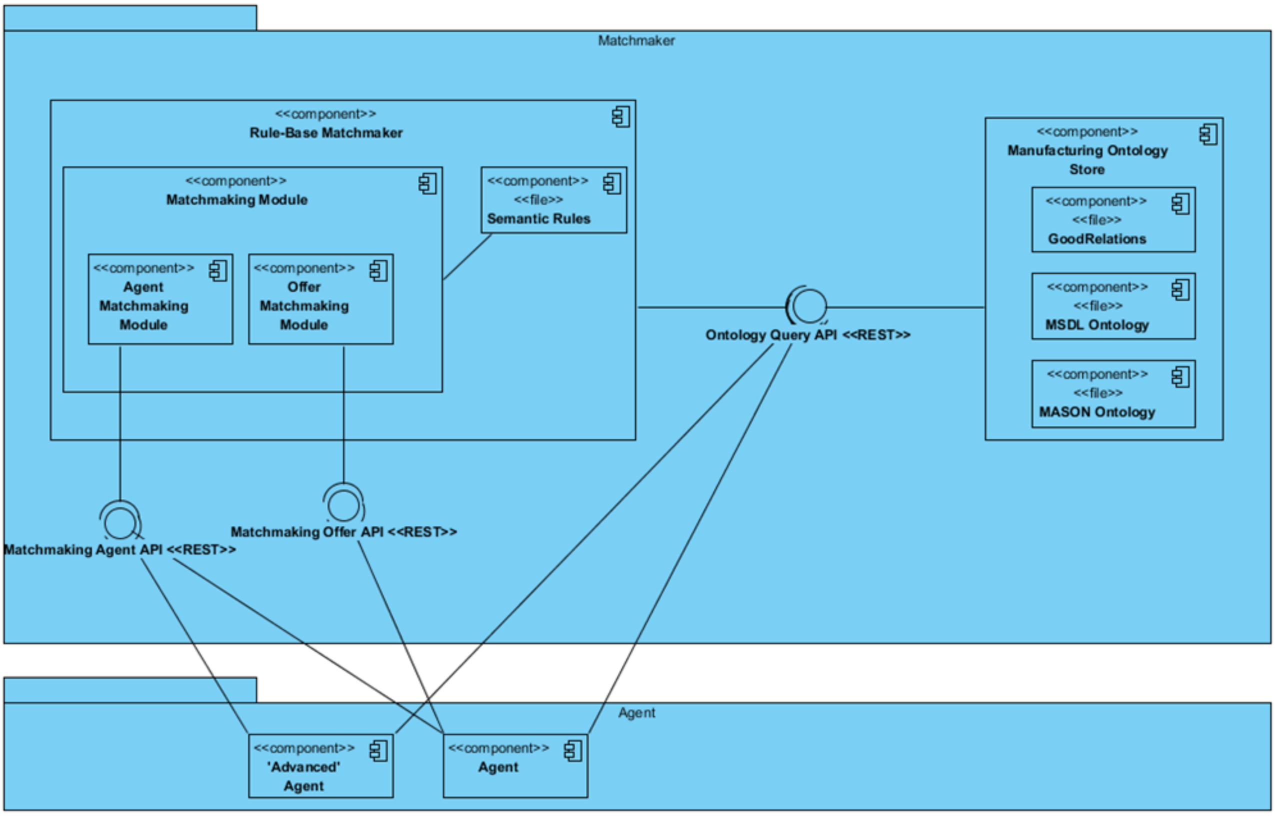


Figure 6: Matchmaker architecture

Development status

The current version of the Ruled-based Matchmaker has been developed in Java and it is offered through RESTful web services. The Matchmaker can infer new knowledge by applying semantic rules in the knowledge stored into the ontology. The set of rules (Jena rules) has been created supports matchmaking between services (requested & offered). Also supports offer and request matching based on price, quantity, delivery time and company ranking.

In next releases the set of semantic rules will be extended in order to support more complex decision criteria based on real data. Also a common format for data exchange between Matchmaker and Agents should be defined in order the component be fully integrated with Marketplace Agents.

A first version of Marketplace Ontology has been developed in OWL language and covers both manufacturing and e-commerce domains. Moreover a first version of Ontology API has been implemented. The API offers CRUD operations to Marketplace Agents.

Deployment

A Docker image has been created for the current version of the Matchmaker. The image has been deployed at Docker Hub and it is able to be pushed to COMPOSITION Inter repository.

5.13 Real-Time Multi-Protocol Event Broker

The real time multi-protocol event broker is the instance of the message broker used in the COMPOSITION factory IIMS (see D6.1). As the hub for all message-based communication, it interacts with most components in the IIMS. However, this is through the standard MQTT protocol and needs no special configuration or development. The broker is tightly integrated with the security framework, which provides identity and access management for all brokers in COMPOSITION system (federated or clustered). Scalability configuration tests (cluster, federation setup in Docker) have not yet been performed. The REST tunnel is in proof of concept phase. The main component, RabbitMQ, is deployed on the test server Docker host. The integration with security framework and blockchain connector is partly complete and deployed at the test server.

5.14 Requester Agent

The Requester Agent is the agent exploited by a factory to request the execution of an existing supply chain or to initiate a new supply chain. Due to the dynamics of exchanges pursued in COMPOSITION, there is no actual distinction between the two processes, i.e., for any supply need a new chain is formed and a new execution of the chain is triggered. The Requester agent may act according to several negotiation protocols, which can possibly be supported by only a subset of the agents active on a specific marketplace instance. (See D2.3)

5.15 XL-SIEM

XL-SIEM[10] (Cross-Layer SIEM) provides the capabilities of a Security Information and Event Management (SIEM) solution with the advantage of being able of handling large volumes of data and raise security alerts from a business perspective thanks to the analysis and event processing in a Storm cluster. The main XL-SIEM functionalities can be summarized in the following points:

- Real-time collection and analysis of security events.
- Prioritization, filtering and normalization of the data gathered from different sources.
- Consolidation and correlation of the security events to carry out a risk assessment and generation of alarms and reports.

The architecture of XL-SIEM is presented in Figure 7:

[10] D4.1 Security Framework I Section 4.4

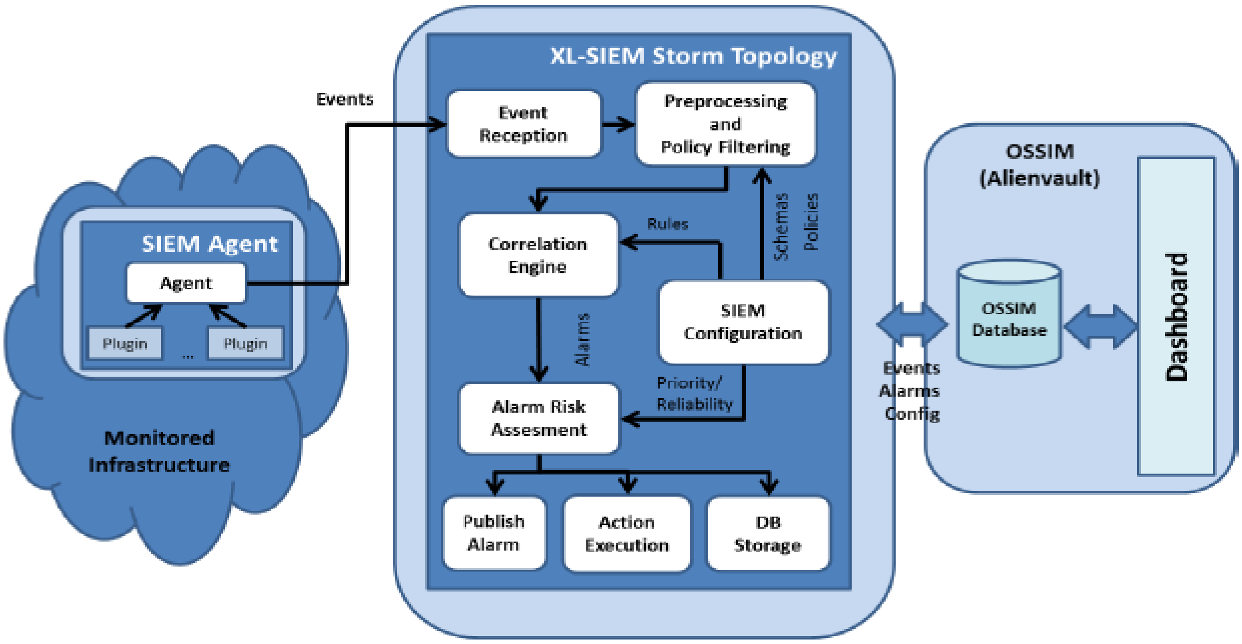


Figure 7: XL-SIEM architecture

The SIEM Agents are responsible for data collection and are deployed within the monitored infrastructure. In case of any event occurrence, they are sent to XL-SIEM core where they are processed and correlated. The OSSIM is responsible for storing gathered events and eventual alarms that were produced during the correlation process. The OSSIM has also visualisation capabilities enabling data inspection.

More information on XL-SIEM component can be found in D4.1 Security Framework I Section 4.4

The XL-SIEM is deployed on Atos premises and is under continuous improvement process.

5.16 Simulation and Forecasting Tool

The Simulation and Forecasting Tool component is part of the high-level platform of COMPOSITION, the Integrated Information Management System (IIMS). Simulation and forecasting tool's main purpose is to simulate process models and provide forecasts of events whose actuals outcomes have not yet been observed.

Its main interactions are LinkSmart middleware, Digital Factory Model, Data persistence storage, Decision Support System (DSS) and Visual Analytics tool. The first three components are the inputs for Simulation and forecasting tool. These components offer all the required historical and live data to the Simulation and forecasting tool in order to export simulations and provide forecasting. The DSS and Visual Analytics tool are the outputs. DSS will be use simulated data in order to suggest solutions and actions. Moreover, the data will be visualized and presented by the Visual Analytics component.

Development Status

A first analysis of pilot site data related to BSL, KLE and ELDIA use cases has been completed. Algorithms related to Descriptive statistics, linear regression analysis, Markov models, Genetic algorithms for optimization and Correlation heatmaps have been developed in Python programming language and applied to related data and use cases. The first results have been evaluated.

Deployment

A Docker image for the Simulation and forecasting tool is planned to be created by M18. This image will be deployed at COMPOSITION Intra repository.

5.17 Supplier Agent

The Supplier Agent is the counterpart of the Requester agent on the composition marketplace. It is usually adopted by actual suppliers to respond to supply requests coming from other stakeholders in the marketplace. Factories transforming goods typically employ at least one Requester agent, to get prime goods and one supplier agent to sell intermediate products to other factories. (See D2.3 The COMPOSITION Architecture Specification I).

6 Map to use cases

In Confluence, the COMPOSITION partners have created a mapping to the use cases, which is monitored and tracked regularly. The mapping is created based on a set of tiers/priorities as determined in consortium discussions. The entire set of entries is too lengthy for inclusion here, but examples from one of the use cases is shown in the table below

Confluence Spaces People Create

Pages /... / Task 2.3 COMPOSITION Architecture

Implementation of Use Cases through Composition Components

Created by Marc Jentsch, last modified by Alexandros Nizamis on Nov 14, 2017

Tier 1

UC-BSL-2 Predictive Maintenance

Component	Responsible Partner	Expected Input from other Components (Source/Input)	Description of the Functionality in the UC	Output	Development Status	Next Release Steps	Expected Time for First Deployment
Deep Learning Toolkit	ISMB	• NXW: oven live data connected to IIMS • TNI-UCC: data from all sensors deployed in the shop-floor • FIT: pattern analytics on all data	The component will provide the latest prevision on next possible breakdowns, based on live data analysis on a trained artificial neural network with historical data.	Report in the form of JSON data, ready to be redistributed to the designated container.	Already adapted the low-fi demo structure from KLE-6 to design an ad-hoc ANN that fits the UC. Currently assessing provided historical data, but waiting for full data availability.	Trained ANN with real historical data and synthetic live, until existing machinery will be connected to the IIMS providing live reading from sensors.	> M16
Wireless sensing network (WSN) platform	TNI-UCC	(Expecting the oven built in sensors to separately supply temperature data to DSS.)	Adding sensors on or near the blowers to gather extra info and determine if fans are operating correctly. Investigating use of power consumption, speed, magnetic, acoustic/ultrasonic and vibrational. These are feed to the DSS (deep learning toolkit?)	A to D conversion done on WSN board and data sent to gateway/edge device/LAN. Architecture to achieve this is under review.	Looking at SoA in acoustic and vibrational sensors for suitability. At system level determine what can be done to aggregate sensor readings (e.g. 1 current clamp to measure combined output of 3 fans) Also need to figure out how data is received and fed to LAN/DSS repository	Select sensors, devise WSN architecture	Aro. M15-16 in TNI-UCC labs and BSL white space
Energy harvesting (EH)platform	TNI-UCC	Battery or DC power source may also be available	Determine if we can extend battery life of the WSN platform using ambient energies (or ultimately eliminate the need for battery replacement)	Power source	It will depend on the WSN infrastructure, sensors use, duty cycle etc. These all govern power consumption	Develop simulation model of WSN load Develop prototype energy harvesting solution	Simulation model aro. M18 EH proto around M24
Big Data Analytics (LinkSmart® IoT Learning Service)	FIT	• Live Data Coming From the Shop-floor as MQTT messages using OGC SensorThings Standard Payload (NXW) • A endpoint to request predictions (ISMB)	The component can apply on-demand stream processing algorithms and redistributed to the to other REST or MQTT endpoints. Additionally, provides (near-)real-time learning orchestration and management capabilities. Also, it can be used to learn using existing implemented algorithms in it.	• Processed data (aggregated, annotated, etc.) • Pre-processed labeled data ready to be train • Real-time alert generation • Predictions using trained models	• Ready to be integrated. • Ready to be use. • 100% OGC Sensor Things 1.0: part 1 complaint • Improving: documentation and continuous integration • Waiting for any change request for any partner.	Current release version is the 1.5.1. The FIT-UCC-Data team is studding the oven data.	>M15
Human-Machine-Interfaces	FIT	• (at least) Predicted data (~ 3 weeks of likelyhood to break down, time frame might need to be adjustable) [Deep Learning Toolkit] • Live and historical (adapatable time frame) sensor data • Name of data source (machine + part)	Visualization about past and future development of parts that need to be maintained	• Visualization (no data outputs)	• First interaction models and wireframes available	Refined interface wireframes	>M16
Digital Factory Model	CERTH	• Static data about actors, assets, equipment etc from BSL (many of them are available at D2.1 or BSCW) • BPMN diagram from FIT • Dynamic sensor data from shop-floor	DFM component will keep in common format information about static and dynamic data related to UC. Other components will be able to get/post data from/to DFM by using the DFM API	• A common format for data description (DFM Schema) • Static and dynamic data in XML format based on DFM Schema (DFM API)	• DFM Schema is available. It is based in well-known standards such as B2MML, gbXML and BPMN for static data representation. Dynamic data representation is based in OGC Observations and Measurements standard • A first version of DFM API is available to consortium. Other components will be able to get or set data to a DFM 'live' instance by using HTTP requests. Data will be stored in a MongoDB in a format compatible with DFM Schema	Replacement of OGC XML format with OGC JSON format, for the description of sensors data	N/A

Table 2: Example of map on JIRA to use cases

7 Findings and Lessons Learnt so far

Based on discussions amongst partners the following is a list of lessons learnt so far:-

- The use of tools such as Confluence assisted us in defining and adjusting priorities, architectures and methodologies
- In general, it is difficult to get clear insights about market segmentation and potential since the inter and inter-factory collaboration field/ topic is relatively new and strong competition is currently forming. For example, very little information about the competitors in this space was found.
- Ever changing use-cases and use-case prioritization made it very hard to structure and focus on the most relevant aspects but it was a worthwhile process and it tested the modularity and re-configurability of the technology components
- As there was an intense focus on delivering high impact use cases, the ones selected are quite "high-class" and thereby complex. Ideally, they should be should be as simple as possible initially with a minimal viable product-thing remit. However, this is a trade-off versus showing something that has high impact and relevance to the COMPOSITION industry partners.

- It was found that when consortium partners got together much progress was made by virtue of having everyone together physically and acting in a dynamic environment. In future projects it is recommend to do more of this where possible and as early as possible.
- Interaction with the other FoF-11 project has been proven very valuable, as it boosts knowledge sharing and it brings together people that are in essence trying to address digital automation for collaborative manufacturing and logistics.

The following tables summaries the technology components in development with assigned owner and overview of its use of and compatibility with off the shelf (OTS) software.

Technology component	Owner	OTS SW used	Compatible with
Access Control	ATOS	EPICA	XACML v3.0
Agent Marketplace	ISMB	RESTful web Services RabbitMQ	AMQP protocol
Authentication	ATOS	Open ID Connect protocol (OIDC) which is based on OAuth 2.0	Keycloak OS
Big Data Analytics	FIT	LinkSmart® IoT Learning Agent	MQTT & OGC SensorThings
BlockChain Connector	CNET	Rabbit MQ broker	Bitcoin protocol
Commissioning System	CNET	RESTful Web Services	PLC register mappings, OGC SensorThings
Deep Learning Toolkit	ISMB	Supporting libraries such as Numpy. Frameworks such as TensorFlow.	MQTT, Tensorflow, Keras
Intrafactory Interoperability Layer	NXW	(NXW) BMS	BacNET, Konnex, MODBUST, etc.
Manufacturing Big Data Storage	NXW	BMS	
Manufacturing Decision Support System	ATL	Rabbit MQ MQTT Restful web services	AMQP Protocol MQTT Protocol PMML BPMN
Market Event Broker	CNET	Jena API	AMQP Protocol
MatchMaker	CERTH	RESTful web services	OWL Ontology, HTTP protocol
Real Time Multi Protocol Event Broker	CNET	REST tunnel Rabbit MQ	MQTT protocol
Requester Agent	ISMB	RESTful web Services RabbitMQ	AMQP protocol
SIEM	ATOS	XL-SIEM OSSIM database	
Simulation and Forecasting Tool	CERTH	No. Only editors such as ATOM	
Supplier Agent	ISMB	RESTful web Services RabbitMQ	AMQP protocol

Table 3: Technology component owner, SW used and compatibility

8 Intra-factory Action Plan

The following is an action plan that was devised and captured on JIRA based on consortium meetings in order to co-ordinate and monitor progress to develop and integrate the various technology components:-

Title	Description	Deadline	Responsible	Status
Docker deployment Wiki	A page/guide describing where and how to deploy the docker components, ATOS, ISMB	15.10.2017	ISMB, ATOS	Done
Security setup	Security Framework auth setup description (RabbitMQ, not REST)	9.10.2017	ATOS	Tested by ISMB
Deploy Linksmart Service Catalogue on docker lab server at ISMB	Get access to Portainer and deploy. Provide example of registration message.	15.10.2017	FIT	Already dockerized
Deploy Big Data Analytics (LS Learning Service) on Docker lab server	Provide examples on wiki.	15.10.2017	FIT	Already dockerized
Simulation and Forecasting deployment on Docker lab server	Simulation and Forecasting deployment on Docker lab server	end of M18	CERTH	To be done by M18
Visual analytics deployment on Docker lab server	Visual analytics deployment on Docker lab server	after M18	CERTH	To be done after M18
Digital Factory Model deployment on Docker lab server	Interface descriptions published for other partners (No JSON format, XML at this date). JSON conversion at later date (end November?)	15.10.2017	CERTH	In progress
Deep learning Toolkit deployment on Docker lab server	The pre-trained ANN for predictive maintenance will be deployed	end Oct M14	ISMB	Dockerised
Predictive Maintenance sensors	Install acoustic and power monitoring sensors in BSL	mid Nov M15	TNI	N/A
Asset tracking sensors	Do initial lab experiments in Tyndall and BSL whitespace with UWB WMU & GPS	mid Nov M15	TNI	N/A
BSL Sensors connected to COMPOSITION installation	Possibly connected through PRIMO (BSL maintenance and scheduling system).	TBD	TNI, NXW, BSL	Working on how to connect them with BMS
BSL visual alarms connection	The visual alarms (amber lights) at BSL should be connected to COMPOSITION & shown in dashbord or similar - how to get this data)	TBD	NXW, BSL	Working on how to connect them with BMS
Interoperability Layer	T5.5 ensure interoperability of components	mid Nov M15	CNET, ISMB	Baseline components dockerized, interconnection TBD
Fill level sensors, vibration at KLE	Vibration and acceleration sensors first prototype (custom sensors)	mid Nov M15	CERTH	In progress
Authorization service (to be deployed at docker lab)	Make changes to integrate with keycloak and create ACL:s (XACML)	mid Nov M15	ATOS	To be done

Decision Support System deployment on Docker lab server	DSS for process performance visualization .	end Oct M14	ATL	Integration (Dockerizing)
Decision Support System deployment on Docker lab server	DSS for predictive maintenance	mid Nov M15	ATL	Integration (Dockerizing)
Decision Support System mobile application	Notification service for mobile devices, and backend	mid Nov M15	ATL	Integration (Dockerizing)
BMS	First deployment with emulated sensors	end Nov M15	NXW	Dockerizing the environment
BMS	integrate with LS and output OGC ST	end Dec M16	NXW	To be done
BMS sensor integration	Actual sensors	Jan 2018 M17	NXW	To be done
Verify information flow in system and test	Components can listen to sensor data and messages	M17	CNET ISMB	To be done
Verify API calls between components	Component can call other component APIs	M17	CNET ISMB	To be done

Table 4: Intra factory action plan

9 Inter-factory Action Plan

An inter-factory plan similar to that outlined in section 8 was also developed.

Title	Description	Deadline	Responsible	Status
Semantic Matchmaking (+Ontology) deployment on Docker lab server	Swagger definition of API	end Nov M15	CERTH	Dockerised
Set up another docker host for the inter-factory	Should be on separate host. Configuration management (networks,1or2nginx, etc.)	end Oct M14	ISMB, ATOS	Done
Requestor agent I	Not double faced connectivity at this stage	end Dec M16	ISMB	In progress
Requestor agent II	Connects to both intra and inter installation	TBD	ISMB	In progress
Supplier agent		end M16	ISMB	In progress
Marketplace Broker	Broker (cloned from intra)	end Oct M14	ISMB, ATOS	In progress
Security Framework	Security framework (cloned from intra) only authentication initially	end Oct M14	ATOS	In progress
Blockchain	To be used by PKI, Log, RM, decide on how to use streams et c. "Configurable setup"	M20	CNET, ATOS	In progress
Reputation Model		M22	ATOS	To be done
PKI		M22	ATOS	To be done
Log	What transactions, messages do we log & how to configure. (Blockchain log)	M20	CNET, ATOS	Ongoing
Communication Agents-Matchmaker	Agent communication with matchmaker finalized and working in installation	end Dec 2017 M16	ISMB, CERTH	In progress
Interoperability verification	Ensure interoperability of components	M22	CNET, ATOS, ISMB	Ongoing. Continuous verification of interoperability will start when all standalone Docker images deployed will have started integration, relying on the Intra-factory Event based broker
KLE sensors installed & sending data	Linked with intra-installation task	End of Jan 2018 M17	CERTH	In progress. 1st lab prototypes ready. Waiting for PCB to create full working prototypes

Table 5: Inter-factory action plan

10 Conclusions, Recommendations and Next Steps

In COMPOSITION a reference architecture has been successfully developed that is now being used in individual use cases.

To date the following components have been developed in COMPOSITION based on this architecture, where possible leveraging from commercially available SW modules and open source widely used protocols

Technology component	Functionality
Access Control	Provides the means to define the security policies used to protect resources, for afterwards any request to access a protected resource will first be evaluated against these policies and the evaluation result will be enforced depending on the outcome.
Agent Marketplace	Container of the COMPOSITION agents. Agents within the Marketplace may implement market-specific services (such as the whitepages or the Matchmaker), or they can act on behalf of industry stakeholders participating in the Marketplace
Authentication	Provides authentication mechanisms for users, applications, services and devices
Big Data Analytics	Provides datastream processing and online learning orchestration through the CEML framework.
BlockChain Connector	Provides a log of transactions that will ensure the integrity and non-repudiation of messages, e.g. agent negotiation and contracts in the marketplace or material/shipment tracking in the factory
Commissioning System	A component to configure sensor setup, e.g. PLC register mappings and hardware and software identifiers, when deploying the COMPOSITION IIMS, re-configuring or installing new sensors
Deep Learning Toolkit	Provides predictions based on the analysed data concerning a fixed future timeframe.
Intrafactory Interoperability Layer	Provides a common ground for exchanging data at shop floor level providing an infrastructure using the publisher/subscriber paradigm.
Manufacturing Big Data Storage	[Whilst most data should where possible be handled in real time]…. some valuable information must be stored somewhere, to be retrieved when necessary, as an historical trace of what has been collected during the process lifetime
Manufacturing Decision Support System	The manufacturing decision support system brings support to a number of engineering practices. Visualise results from prediction and simulation and gives in context knowledge to the final users
Market Event Broker	This is the instance of the message broker used in the COMPOSITION Marketplace. It interacts with most components and is the hub through which marketplace agents communicate.
Matchmaker	Aims to match requester and supplier agents participating in the Marketplace based on different selection criteria. Used by agents in order to match requests and offers between the agents.
Real Time Multi Protocol Event Broker	This is the instance of the message broker used in the COMPOSITION factory IIMS. As the hub for all message-based communication, it interacts with most components in the IIMS.
Requester Agent	The agent exploited by a factory to request the execution of an existing supply chain or to initiate a new supply chain.
SIEM	Provides the capabilities of a Security Information and Event Management (SIEM) solution with the advantage of being able of handling large volumes of data and raise security alerts from a business perspective thanks to the analysis and event processing in a Storm cluster.
Simulation and forecasting tool	To simulate process models and provide forecasts of events whose actual outcomes have not yet been observed.
Supplier Agent	The counterpart of the Requester. It is usually adopted by actual suppliers to respond to supply requests coming from other stakeholders in the marketplace.

Table 6: COMPOSITION Technology Components

At a high level there are two communication mechanisms in the COMPOSITION system: message-based communication over MQTT or AMQP using the Event Broker[11], and request-response REST HTTP interfaces (D2.3 The COMPOSITION Architecture Specification I, 2017).

With a few exceptions[12], the components are loosely coupled and integrated in the system by conforming to a common communication infrastructure and data schemas. There are dependencies on data generated by other

[11] Real-time Event Broker, Message Broker
[12] E.g. Big Data Analytics and Deep Learning networks are tightly integrated.

components, but for most of the data, the dependency is not direct as this data is distributed through the event broker, so the direct connection is to this component.

Agents and user interfaces (HMI) depend on tools for modelling, simulation and analysis of factory data and processes. Business functionality, e.g. manufacturing analysis, is implemented by components providing generic functionality for complex event processing (CEP) and neural networks (deep learning). These in turn use the communication functionality, the intra-factory interoperability and shop-floor connectivity functional packages. The inter-factory packages provide the necessary infrastructure for the agents to find other agents, negotiate and enter agreements.

Next steps are to

(i) Further develop and dockerise all the components following the inter and intra-factory plans.

(ii) Integrate the components into use cases and demonstrate their viability and effectiveness

(iii) Work on the architecture as we progressively experiment with its suitability for various use cases.

11 List of Figures and Tables

11.1 Figures

11.2 Tables