Production harmonizEd Reconfiguration of Flexible Robots and Machinery

Horizon 2020 – Factories of the Future, Project ID: 680435

Deliverable 6.3

Report on Self-Adaptive Large Scale Demonstrator Design and Set-Up

Lead Author: MTC

Version: 1.0
Date: 28/03/2017
Status: Submission

Version history

Version	Date	notes and comments
0.1	11/10/2016	Deliverable structure, Table of Contents with effort allocation
0.2	14/10/2016	Revision and agreement with external partners on the structure and effort allocation
0.3	06/01/2017	Contribution to Chapters 1, 2 and 7 (MTC)
0.4	10/03/2017	Contributions of partners added to the document (MTC)
0.5	15/03/2017	Contribution to Section 7.3 (SmartFactory)
0.6	22/03/2017	Internal review and revision (MTC)
0.7	23/03/2017	Contribution to Section 7.2 (IPB)
1.0	28/03/2017	Document finalised

Author List:

Evangelia Dimanidou (MTC)
Nandini Chakravorti (MTC)
Andre Hennecke (SmartFactory)
Jose Barbosa (IPB)
Filippo Boschi (Polimi)
Jeffrey Wermann (HSEL)
Frederik Gosewehr (HSEL)
Ricardo Peres (Uninova)
Giacomo Angione (Loccioni)
Johan Vallhagen (GKN)
Pierluigi Petrali (WHR)
Gregorio Iuzzolino (IEFVS)

Abstract

PERFoRM WP6 "*Validation and demonstration of reconfigurable and self-adaptive systems*" is responsible for testing and de-risking the technologies developed within the project, before these are applied to the production environment. In particular, Task 6.3 "*Self-Adaptive and Reconfigurable Large Scale Systems*" is responsible for demonstrating the self-adaptive and reconfigurable characteristics at a systems level for a large scale system in a relevant industrial environment. This deliverable presents the test scenarios for all the industrial use cases. Further deliverable (D6.6: "*Self-Adaptive Large Scale Demonstrator Documentation and Results*") will report the results of the test scenarios.

List of Figures

List of Tables

1 Introduction (MTC)

The PERFoRM project aims to deliver next generation agile manufacturing systems that are dynamically configurable, self-organised and adapt according to the current market demands of customised small lot sizes and shorter lead times. This concept is envisaged to be delivered via the implementation of a common reference architecture and standard interfaces that will support the principles of plug-and-produce components.

The project consortium has a diverse range of industrial end-users (aerospace, home appliances, micro-electric vehicles and compressors) and thus varying requirements and challenges. The structure of the project has been presented in D6.1: "Self-Adaptive Machines Demonstrator Design and Set-up" (MTC, 2016). WP6 in particular is responsible for the validation of the technologies developed within the project. As per the structure (MTC, 2016) of the project, WP6 "*Validation and demonstration of reconfigurable and self-adaptive systems*" aims to: (1) verify technologies developed within the different WPs (specifically WP1-4), (2) validate the compliance with user requirements, and (3) de-risk the technology developed before deployment in actual industrial environment.

The scope of the three tasks in WP6 (Task 6.1, Task 6.2 and Task 6.3) can be explained using the PERFoRM project framework (see Figure 1). This framework has been proposed in Deliverable 1.1: "*Report on decentralised control & Distributed Manufacturing Operation Systems for Flexible and Reconfigurable production environments*" (Siemens, 2016) and consists of three different views: *asset view*, *architectural view* and the *process view*. An explanation of the three views can be seen below:

- *Asset view*: includes shop floor equipment of a flexible manufacturing system along with devices and human resources which operate them. This view involves self-adaptive and reconfigurable machines and robots. This task involves the testing of adaptors used to interface legacy devices (e.g. robots, sensors) with the novel architecture developed within the PERFoRM project.
- *Architectural view*: includes all the software and IT systems of the architecture related to the flexible manufacturing system. There are several information and communication technologies (ICT) such as Enterprise Resource Planning (ERP), manufacturing execution systems (MES) within manufacturing operations. Additionally, modules such as (1) Data Analytics to enable functionalities such as predictive maintenance, (2) Simulation, to simulate the material flow through the workstations regarding logistics parameters e.g. lead time and (3) Storage to store the production data have been included within the architecture (see Figure 2). The integration of the legacy systems to the PERFoRM architecture is done via the use of technological adaptors and standard interfaces.
- *Process view*: considers a global vision of the architecture, testing flexibility and reconfigurability of the production line. The process view considers typical processes within the production area. Further details on the processes for each use cases will be presented in Sections 3, 4, 5 and 6. In contrast to the previous tasks, this task is responsible for validation of the vertical integration within the PERFoRM architecture (see Figure 2).

Additionally, Figure 1 also illustrates the scope of the three tasks as explained below:

- *Task 6.1*: The scope of Task 6.1 is limited to the validation of the adaptors, standard interfaces implemented for the operation level assets such as robots, machining tools and

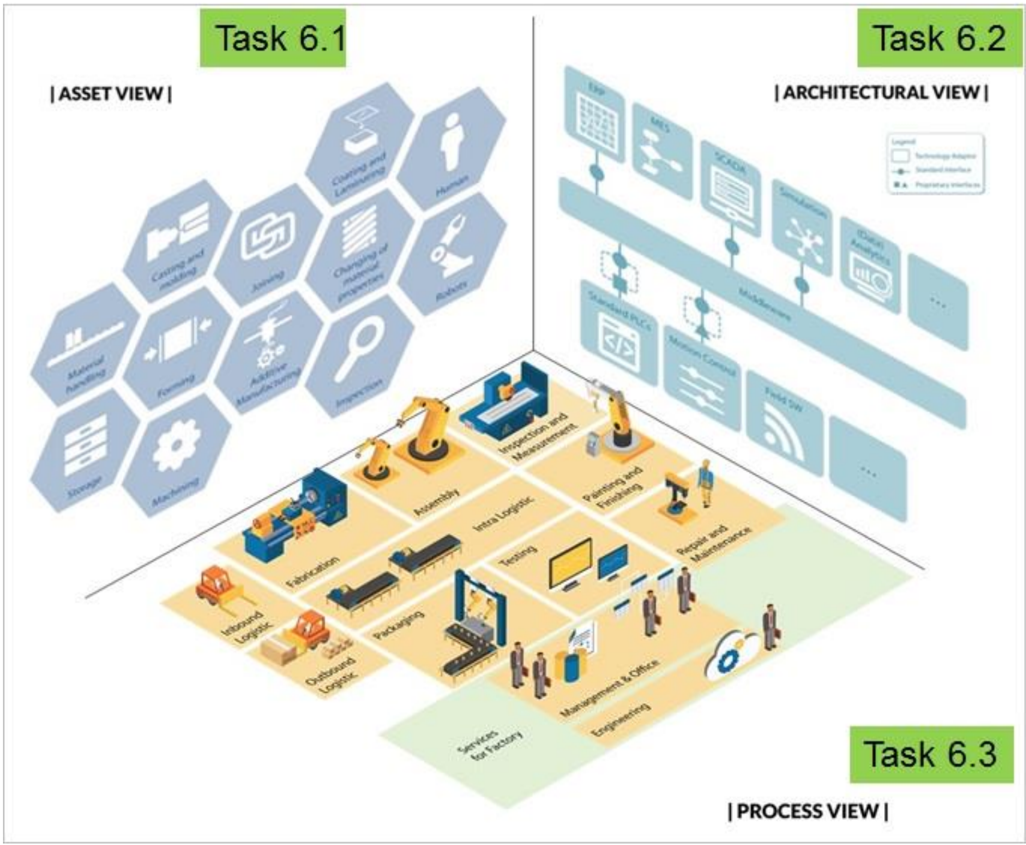

Figure 1. Scope of WP6 tasks (MTC, 2016)

  HMIs. Consequently the scope of this task falls within the realm of the *asset view* (see Figure 1) within the PERFoRM project framework. The scope of Task 6.1 in terms of the PERFoRM architecture can be seen in Figure 2. A detailed description of the elements of the architecture can be found in D6.1: “Self-Adaptive Machines Demonstrator Design and Set-up” (MTC, 2016).
- *Task 6.2*: The validation of integration of the IT assets within the PERFoRM architectural principles will be conducted as part of Task 6.2: “*Self-Adaptive and Reconfigurable Highly Modular and Flexible Assembly Systems*”. Consequently the scope of this task falls within the realm of the *architectural view* (see Figure 1) within the PERFoRM project framework. The scope of Task 6.2 in terms of the PERFoRM architecture can be seen in Figure 2.
- *Task 6.3*: The validation of the self-adaptive and reconfigurable characteristics within the relevant industrial environment, thus involving typical processes in the production floor (e.g. assembly, painting and finishing in the case of the Whirlpool use case). Consequently the scope of this task falls within the realm of the *process view* (see Figure 1) within the PERFoRM project framework. The scope of Task 6.2 in terms of the PERFoRM architecture can be seen in Figure 2.

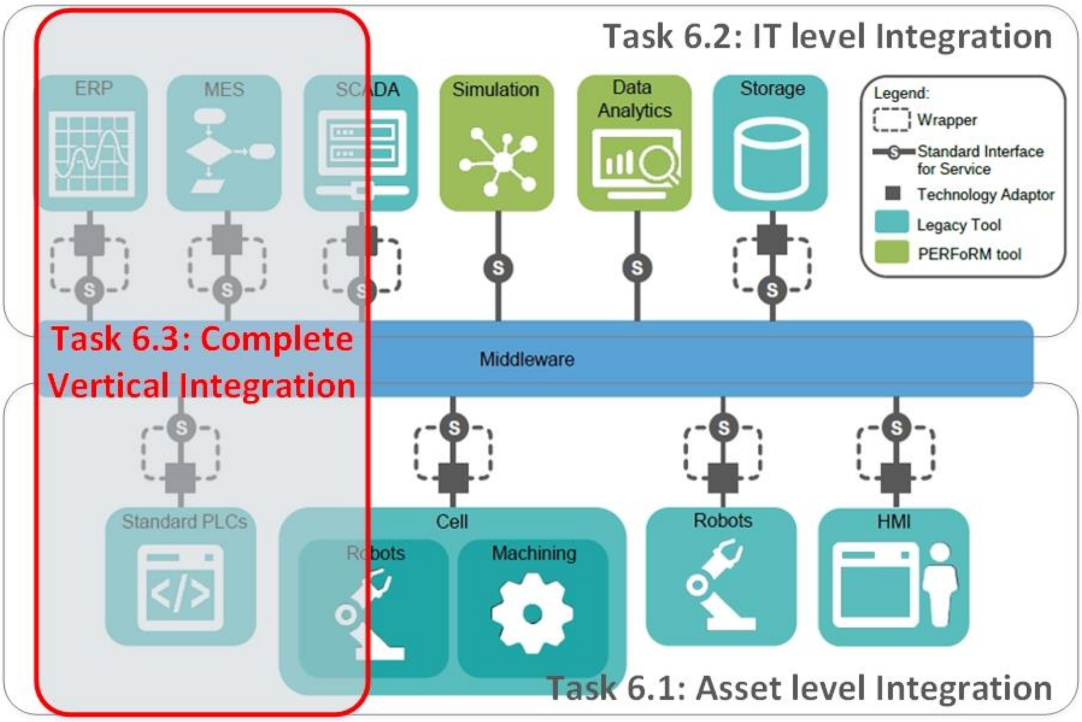

Figure 2. Scope of Task 6.3 in contrast to Tasks 6.1 and 6.2

1.1 Objectives of the document (MTC)

The objective of Task 6.3: "*Self-Adaptive and Reconfigurable Large Scale Systems*" is to demonstrate the self-adaptive and reconfigurable characterises of system in a large scale industrial environment. In contrast to Task 6.1: "S*elf-adaptive and reconfigurable machines and robots*" and Task 6.2: "*Self-adaptive and reconfigurable production modules*", this task is a complete vertical integration including the production machinery level as well as the higher level enterprise applications (see Figure 2). Task 6.3 comprises of two deliverables D6.3: "*Self-Adaptive Large Scale Demonstrator Design and Set-up*" and D6.6: "*Self-Adaptive Large Scale Demonstrator Documentation and Results*". The specific objectives of this document are listed below:

- Identify the scope, processes and requirements specific to each use case. A brief description of the use cases are provided in Sections 3, 4, 5 and 6.
- Identify the test scenarios relevant for each industrial use case. Details about the Environment requirements have been presented within sections 3.3, 4.3, 5.3 and 6.3 for the Siemens, Whirlpool, GKN and the IFeVS use cases respectively. The environment requirements were captured via a technical questionnaire prepared by IPB and Loccioni. Further information was captured via focused discussions with the individual use cases.

The next deliverable for this task (D6.6) will include details about the demonstrator and the test results. It is to be noted that some of the test scenarios identified for the industrial use cases may not be tested within the test beds (MTC and SmartFactory) due to lack of appropriate infrastructure, and consequently will be tested as a part of either WP5: "*Integration and Deployment Planning*" or WPs 7-10 (Siemens, IEFVS, Whirlpool and GKN respectively). The relation between Task 6.3 and the other work packages is illustrated in Figure 3. The left-hand side of Figure 3 illustrates the inputs for Task 6.3 and a brief description of each element is provided below:

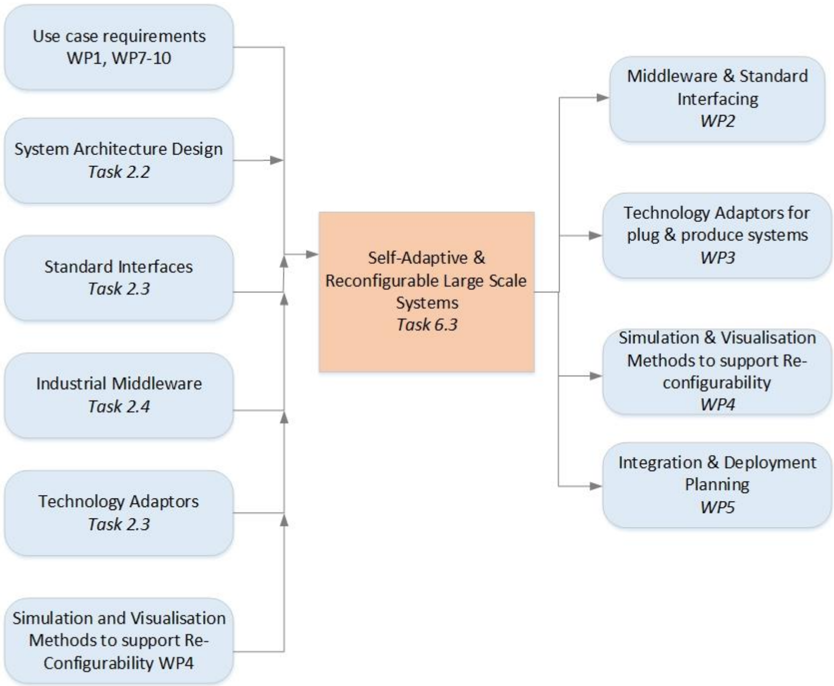

Figure 3. Interaction of Task 6.3 with other tasks and work packages

- The requirements of the four industrial use cases have been identified by WP1 via workshops and questionnaires, and these have been presented in deliverables D1.2 "*Requirements for Innovative Production System Functional requirement analysis and definition of strategic objectives and KPIs*" (POLIMI, 2016) and D1.3 "*Requirements review, evaluation and selection of best available Technologies and Tools*" (FhG-IPA, 2016). Additionally, details regarding the requirements for individual use cases has been illustrated in D6.1: "*Report on Self-Adaptive Machines Demonstrator Design and Set-up*" (MTC, 2016). A brief summary of the use cases has been presented in Sections 3, 4, 5 and 6 for the Siemens, Whirlpool, GKN and IFeVS use cases respectively.

- The system architecture design based on smart production components that are able to support seamless reconfiguration has been presented in deliverable D2.2: "*Definition of the System architecture*" (IPB, 2016).

- The standard interfaces are meant to provide seamless exchange of manufacturing data between varied entities across the entire value chain. (MTC, 2016). Further details on the standard interfaces can be found in D2.3: "*Specification of the Generic Interfaces for Machinery, Control Systems and Data Backbone*" (UNINOVA, 2017).

- An industrial middleware is being proposed as a part of this project to address the issue of interoperability whilst connecting diverse production systems and applications. Task 2.4: "*Industrial manufacturing middleware*" is researching different Middleware technologies in depth by defining multiple basic functionalities that a Middleware should provide to meet the requirements that are being set by the goals of PERFoRM.

- The simulation environment can be used offline to address planning problems such as reconfiguration or maintenance. It can also be used online considering current production conditions (e.g. change request in production line).

The outcomes of Task 6.1 will be utilised in different work packages such as WP2: "*Middleware and Interfacing"*, WP3: "*Technology Adaptors for plug & produce systems*", WP4: "*Simulation & Visualisation Methods to support Re-configurability*" for refining the technologies developed within the project. The outcomes are specifically important for WP5:"*Integration & Deployment Planning*" as this task is responsible for consolidating the results in the previous WPs and deploy the results within production systems at the industrial use cases. It is to be noted that the testing activities at the test beds within MTC and SmartFactory may not be representative of the industrial use case specifically for Task 6.3. These test cases that cannot be evaluated within the test beds will be tested as a part of WP5 at the industrial use case site(s).

1.2 Document outline (MTC)

A structured methodology has been used to define the tasks for Task 6.3, as presented in Section 2. Sections 3, 4, 5 and 6 present the objectives, processes and requirements specific for each use case has been described. Additionally, the architecture that needs to be evaluated have been briefly described.

The test scenarios envisaged for each use case have also been presented within Sections 3, 4, 5 and 6. The test and demonstration approach has been presented in Section 7 and includes descriptions of environment requirements, test schedule and potential risks associated with the test scenarios. Finally the conclusions and future work are presented in Section 8.

2 Methodology (MTC, SmartFactory)

A structured methodology for conducting the testing activities within Task 6.3 has been proposed (see Figure 4) to enhance the understanding of the tasks and ensure that the objectives of the task are met. The proposed methodology comprises the major tasks that need to be performed within Task 6.3. Additionally, the scope of D6.3 and D6.6 has also been highlighted in Figure 4. It is to be noted that the scope of this deliverable does not cover Steps 5, 6, 7 and 8. An explanation of each step within the methodology has been described below:

Step1: involves identification of the scope of Task 6.3 and this involved meetings with the different stakeholders (i.e. use cases and technology partners) to understand the boundaries of this task, especially to understand the difference between the other tasks (namely Task 6.1 and Task 6.2) of this WP.

Step2: involves the identification and documentation of test scenarios applicable to this task. The test scenarios for this task is suited to the process view, and as such involves the complete vertical integration within the PERFoRM architecture.

Step 3: Once the test scenarios were defined, these were discussed with the individual use cases to better define the scope of the test scenarios. This step is very critical as some of the test scenarios can be tested at the test labs (e.g. MTC, SmartFactory), whilst some of others need to be tested at industrial facilities, and this involves consideration of the infrastructure. The definition of where the test scenario can be tested best is defined as Step 3a in Figure 4.

Step 4a: involves the definition of the testing environment. The test environment is very specific to individual use cases. The current deliverable is restricted to identifying the test scenarios. The set-up of the demonstrator and the results of the actual testing will be illustrated within D6.6:"*Self-Adaptive Large Scale Demonstrator Documentation and Results*".

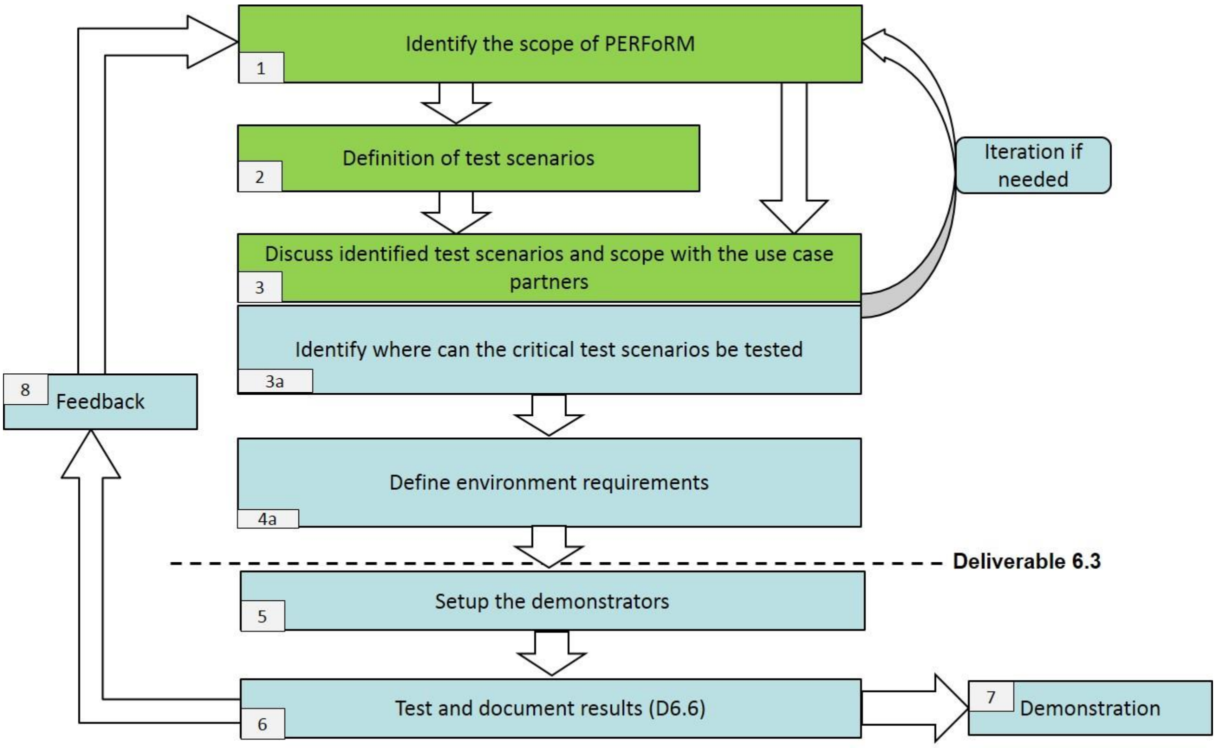

Figure 4. Task 6.3 Metholodogy

3 Siemens use case (Siemens, SmartFactory, MTC)

The Siemens use case is involved in the production of industrial compressors and gas separators. The PERFoRM architecture will be deployed at the Duisburg factory which is responsible for the manufacturing of tailored compressors trains for oil and gas applications, such as air separation units or for the Liquefied Natural Gas (LNG) production. At the moment, production of compressors is characterised by small lot sizes (1-30), machining, manual labour and a highly complex final assembly.

The objective of this use case is to identify disturbances in production. The use case has an overall aim to introduce predictive and condition based maintenance, which shall improve the flexibility of manufacturing by better shifting production tasks between different machines and reduce quality issues and production failures. It is envisaged that a failure or machine breakdown can lead to delays of the productions and missing parts to semi-finished products; further details on this use case can be found in D7.1: "*Siemens description and requirements of architectures for retrofitting production equipment*" (Siemens AG, 2016). Realising such a predictive maintenance scenario encompasses the monitoring of the current machine health, the generation of related maintenance tasks and the combination of production tasks and maintenance tasks in the overall production schedule. Within the PERFoRM project, the concepts need to be introduced for three comparable Carnaghi turning machines.

3.1 Process and requirements (Siemens, SmartFactory, Polimi, MTC)

The objectives of this use case can be categorised into two major groups as depicted in Figure 5. Firstly there is the condition monitoring and maintenance task process which includes the accessing of data from the databases and additional machine sensors, the analysis of this data to monitor the machine health and to predict possible machine failures and the maintenance task generation based on these predictions. Secondly, the (re)scheduling and evaluation process, including maintenance tasks, the evaluation of the proposed schedules by simulating the new schedules in production.

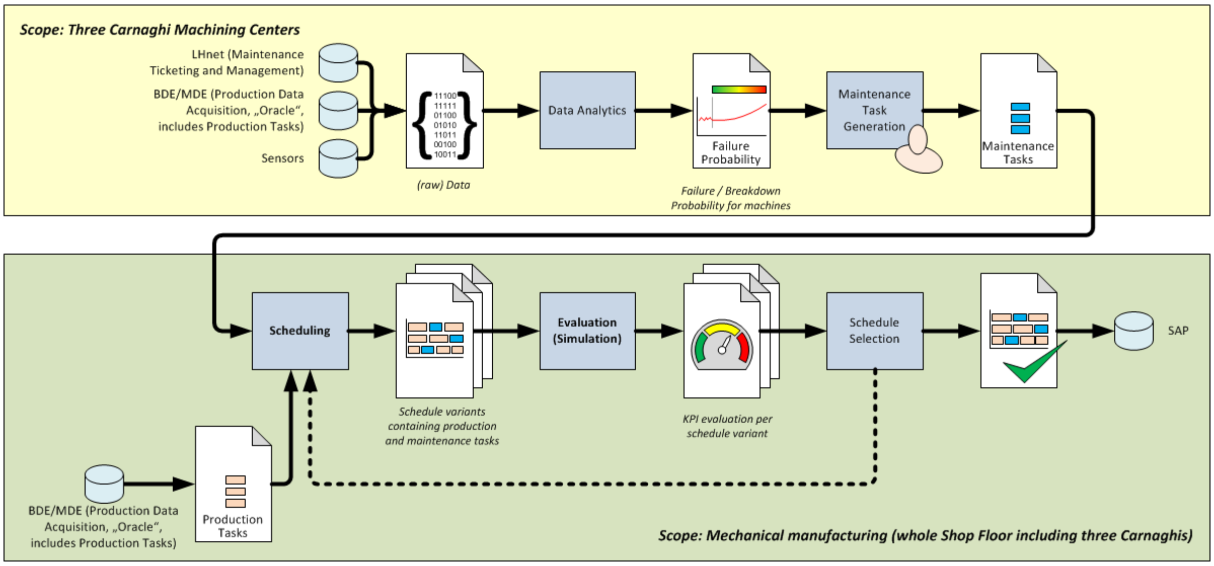

Figure 5. Siemens – Process Flow

The main functions of the predictive maintenance system introduced in the Siemens production site can be seen below:

a) Monitoring of machine conditions;
b) Prediction of possible machine failures;
c) Generation of maintenance tasks;
d) Scheduling system which treats production tasks and maintenance tasks equally;
e) Simulation which evaluates the new proposed schedules;
f) Transferring the new factory schedule to the SAP system manually;

3.2 Architecture and Description of Test Components (Siemens, SmartFactory, MTC)

The overall architecture of the Siemens use case is illustrated by Figure 6. It includes all the tools and its functionalities (described in Section 3.1 and illustrated by Figure 5) such as Data Analytics, Maintenance Task Editor, Scheduling System and the Simulation. The above mentioned tools interact and collaborate via the PERFoRM middleware.

The collection of machine and production data is done by a comprehensive system, encompassing a ticketing terminal (LHnet) and the MDE/BDE data which includes the production data from the ERP system and machine data. The monitoring of detailed machine conditions is done by additional sensors which publish their data to the middleware. It is to be noted that the selection of additional sensors will be conducted as a part of WP4 and WP5. As seen in Figure 6, the technology adaptors are used by the SQL and Oracle databases to ensure that these systems can connect to the PERFoRM compliant middleware. Technology adaptors (grey colour rectangles as seen in Figure 6) have been used to ensure seamless connectivity of these systems within the PERFoRM architecture. Additionally, standard interfaces (letter "S" as seen in Figure 6) will be used to enable plug-ability and interoperability. Further details on the technology adaptors and standard interfaces can be seen in D6.1: "*Self-Adaptive Machines Demonstrator Design and Set-up*" (MTC, 2016).

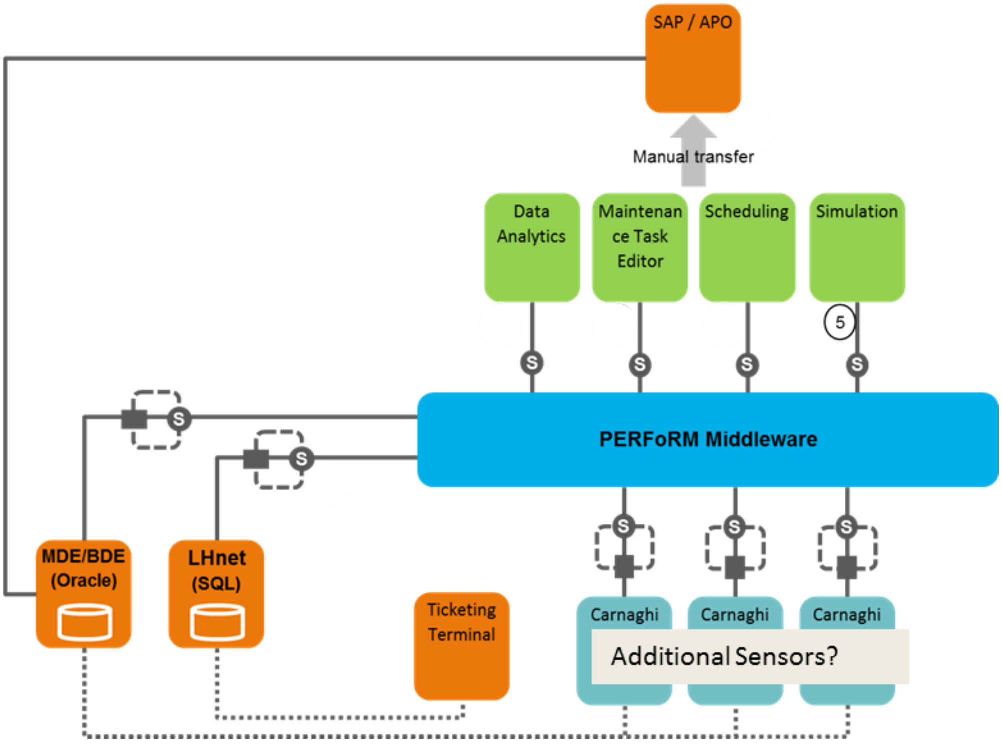

Figure 6. Siemens - System Architecture

3.3 Environment Requirements

The environment requirements for the Siemens use case can be seen below:

- **Oracle database** for the LHnet data.
- **SQL database** for the MDE/BDE data.
- **SAP APO**
- **Data Analytics** to detect and visualise the state of the components or the complete machine
- **Carnaghi** machines

3.4 Test scenarios (Siemens, SmartFactory, Polimi, MTC)

The scope of T6.3 involves testing and demonstrating at the process level, consequently the test scenarios should reflect the complete system behaviour and in particular the functionality of the whole system. As illustrated in Figure 5, the processes can be divided into two categories, condition monitoring and maintenance task generation, and the scheduling and evaluation. Thus, the test scenarios must validate that the functionality of those two elements are fulfilled. The test scenarios for the Siemens use case can be seen in the table below:

Table 1. Siemens test scenarios

Test Scenario ID	Test Scenario	Risk / Priority
TS-S-F-1	Validate that the data analytic tools can detect and visualise the state of components or the complete machine, when a maintenance task is required in the next time.	High

TS-S-F-2	Show that an operator can understand and use the information from the visualisation system.	Low
TS-S-F-3	Validate that the maintenance task editor is supporting all required maintenance tasks based on the input of an operator.	High
TS-S-F-4	Show that an operator is able to define maintenance tasks in the maintenance editor with the information from the data analytics visualisation system.	Low
TS-S-F-5	Validate that the maintenance task and the production task can be accessed by the scheduling system via the middleware.	Low
TS-S-F-6	Validate that the scheduling tool automatically collects the pending production and the maintenance tasks and that it proposes new schedules including both.	High
TS-S-F-7	Validate that the simulation tool can use the input form the scheduling system and that it starts the simulation to calculate optimal time slots for the maintenance task with different KPIs.	High
TS-S-F-8	Show that different kind of maintenance task can be manually transferred to the SAP system.	Low

3.3.1 Test scenarios description

The basic workflow for the following test scenarios, which is also indicated in the previous chapter, is as follow:

1. Sensor values and/or pre-computed data are being transferred from the machinery to the database via middleware, or are directly being visualised to the operator. Together with the BDE/MDE and the Failure Ticket System (LHnet) the data reflect the current machinery state.
2. The Data Analytics tool can access the data via the middleware, analyse it and visualise the result.
3. The results from the data analytics tool are being used for the manual creation of a new maintenance task in the maintenance task editor.
4. The scheduling tool accesses the maintenance tasks that have been created to propose different schedules including the production and the maintenance tasks.
5. The schedules are being evaluated by the simulation tool with different KPI outputs.
6. The “best” maintenance task is being transferred to the SAP system, including the timeslot.

TS-S-F-1

This test validates a scenario when a machine or component is in a state where a maintenance task is required soon. Within the scope of this task, the interaction among components at the machinery level, the architecture level (Middleware, PLM), and the IT-Tools is required. The data analytics tools must be able to access the machinery information to calculate the state of the machine or a component. The results of this calculation are visualised in a way that the maintenance team can use to create a new maintenance tasks.

TS-S-F-2

This scenario is a usability test, where the visualisation system from the data analytics tool is being used by the maintenance team. Testing activities should show that the visualization system effectively communicates the information to the operator.

TS-S-F-3

This scenario validates that all the required maintenance tasks can be created or selected for a Carnaghi machine by using the information from the visualisation system.

TS-S-NF-4

This scenario is a usability test to show that an operator or the maintenance team can use the maintenance editor to define new maintenance tasks with the information from the data analytics visualisation system.

TS-S-F-5

This scenario shows that the scheduling system can collect all the required information to calculate new schedules including the maintenance tasks and the production tasks. The scheduling tool can access the information via a PERFoRM compliant middleware using the standard interface.

TS-S-F-6

The scheduling tool is triggering itself. Therefore this scenario validates that the tool starts its operation automatically, collects the pending maintenance and production tasks via the middleware and it calculates new schedules including both kind of tasks.

TS-S-F-7

This scenario should validate that the simulation tool can access the information from the scheduling system or it correctly receives the information, so that it can start the simulation to calculate the optimal time slots for the maintenance task with different KPIs.

TS-S-F-8

This scenario shows that the created maintenance tasks including the timeslots can be manually transferred to the factory's SAP system.

4 Whirlpool use case (Whirlpool, Polimi, SmartFactory, MTC)

The Whirlpool use case focuses on the challenge associated with the lack of available real-time shop floor data and its integration with the production system for enabling controlling and planning. In particular, the Whirlpool use case aims at establishing a real-time monitoring system that will be able to correlate the dynamic behaviour of the factory to the Key Performance Indicators (KPIs) and static Key Business Factors (KBFs), such that a fast reconfiguration can be enabled. Further details on this use case can be found in D9.1: "*Whirlpool description of Requirements and Architecture Design*" (WHR, 2016).

Currently within the as-is architecture, data is gathered with different Quality and Production systems (MES tools) and the decision-making process is a combination of a different level of operations based on KPI elaboration without any real-time tool support. At present, medium to long term decisions are supported by the analysis of post-production data (excel, Minitab, etc.). Moreover, Production and Quality KPI are being communicated in a point-to-point way, i.e. only specific data is communicated to specific people at a specified time.

The production system is equipped with: statistical process control (SPC) (visual measurement system for the correct components alignment), PLC and robot alarm system (sound signals to warn of a line block), assembly kit (part-number collected in suitable kit in line with the logic KANBAN), and production control (display panel which shows the updated production status).

The main challenge of this use case is to overcome the following issues:

- KPIs are currently being analysed and monitored in isolation. Data is often segmented in "silos";
- Different KPI groups are used for different objectives: each function of the factory has complete visibility and possibility to make only direct correlation. For example, Quality department has ownership of Fall Off Rate (FOR) KPI, however they don't have access to data that is reuiredto create correlation between FOR and other potential influencing KPI (e.g OEE);
- Re-configuration activities are driven by a single KPI;
- No high-level simulation of factory exists and thus, reconfiguration can't be tested in a dry environment. This implies that all reconfiguration alternatives have to be implemented in the real factory and their effects should be measured with real data. This limitation reduces the number of alternatives that can be tested, as the implementation of the reconfiguration mechanism can put the whole factory at risk.

Finally, it should be highlighted that legacy sensors, PLC and SCADA systems cannot be modified in the short-medium term. Standard and well-established components, protocols and operating systems are required, taking into consideration that some equipment, logic and software services are not owned by Whirlpool.

4.1 Process and requirements (Whirlpool, Polimi, SmartFactory, MTC)

The production process is composed of 8 major steps:

1. Material Incoming: Raw material (steel and stainless steel coils) and mechanical, electrical and electronic components are being delivered at any time, on a daily basis. They are being checked and stored in a warehouse.
2. Cavity line Fabrication: The process aims at producing the internal cavity of the oven starting from carbon steel and then being painted or stainless steel, and includes three major steps:
   a) Cavity parts stamping: cavity parts (top and wrappers) are being stamped in hydraulic presses
   b) Cavity Assembly: cavity parts are being welded to form an open box
   c) Painting: cavities that have been made in carbon steel are then washed and painted.
3. Door fabrication and assembly: Door is composed of steel frame, shield, glass and a handle. The most critical step in this process is the gluing of glass together with the steel frame.
4. Part Fabrication and Silk Screening: Other metal parts are stamped from steel coils. Aesthetical parts such as the front panel are marked using a silk screening technology.
5. Final Assembly: Cavity, door and all other components are assembled in sequence in a continuous flow conveyor.
6. Testing: Ovens are tested from safety and functional point of view.
7. Repairing Bay: Products that are unable to pass the test are examined and repaired in a separated bay.
8. Packaging: ovens are finally packaged and sent to finished good warehouse.

The system objectives should essentially consist of a particular and dynamic system which will be able to manage the following actions:

- To recognise the input: the variable external issues (e.g. market demand);
- To identify the KBFs;
- To define a relevant model;

- To evaluate the output, i.e. the KPI value;
- To compare inputs, KBFs and KPIs in order to choose the actions needed to obtain the system reconfiguration and to improve the output;
- To apply a sensitivity analysis, which aims at defining the relationships (i.e. sensitivity factors) among the KBFs and KPIs;
- To validate the consistency of the model.

4.2 Architecture and Description of Test Components (Whirlpool, Polimi, Loccioni, IPB, MTC)

Whirlpool's system architecture, which is illustrated by Figure 7, displays the new architecture that focuses on the development of a reconfiguration support system, which will improve the current reconfiguration process by allowing a closer and faster correlation among actions and expected performance. The Robotic Cell Reconfiguration use case scenario performs an automatic leakage test. The second use case scenario represents the Cavity Fabrication and Value Stream Reconfiguration.

The system consists of a Universal Robot 10 (UR10), equipped with a microwave probe. A CAD file that needs to be sent to the robot (via the Middleware) and is being stored to a database. An adaptor that has been developed by Loccioni is responsible for accessing the file and translating it into a UR script.

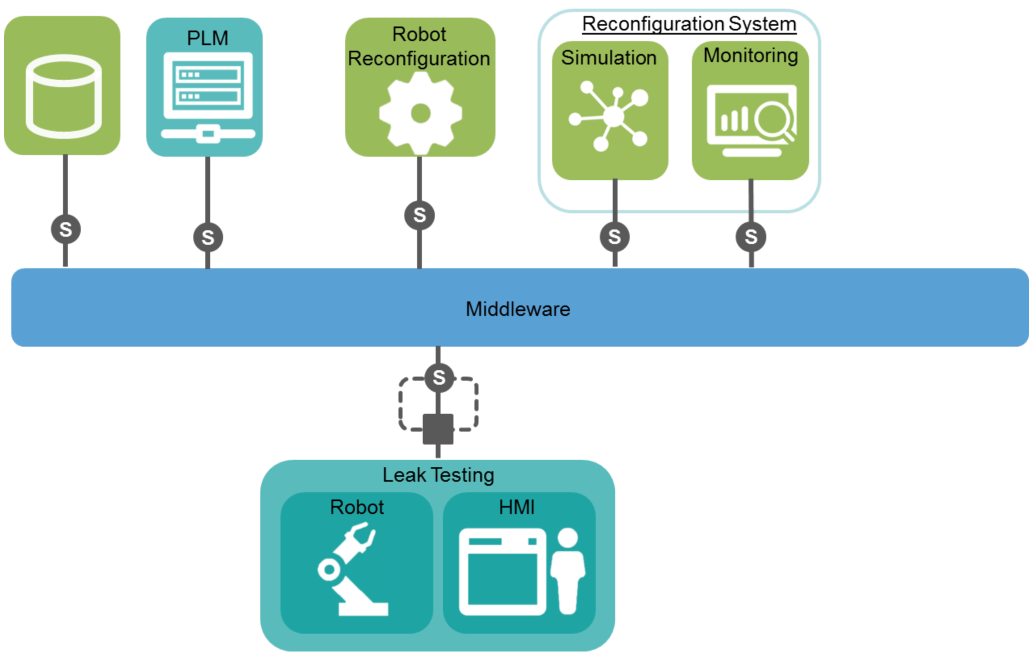

Figure 7. Whirlpool's system architecture

Whirlpool's system architecture can be seen below seen in Figure 8. Different systems interact with each other by exchanging data and information. The top layer identifies the IT application (Production Data Population and PLC) and provides shop floor data related to production parameters, product declaration and product reworked to the manufacturing database.
The bottom layer shows the IT application that is needed to create a digital manufacturing domain based on CPS and leads to the simulation, evaluation and visualisation of the production process behaviour

and therefore to the selection of the best combination among KPIs and KBFs in order to achieve process optimisation.

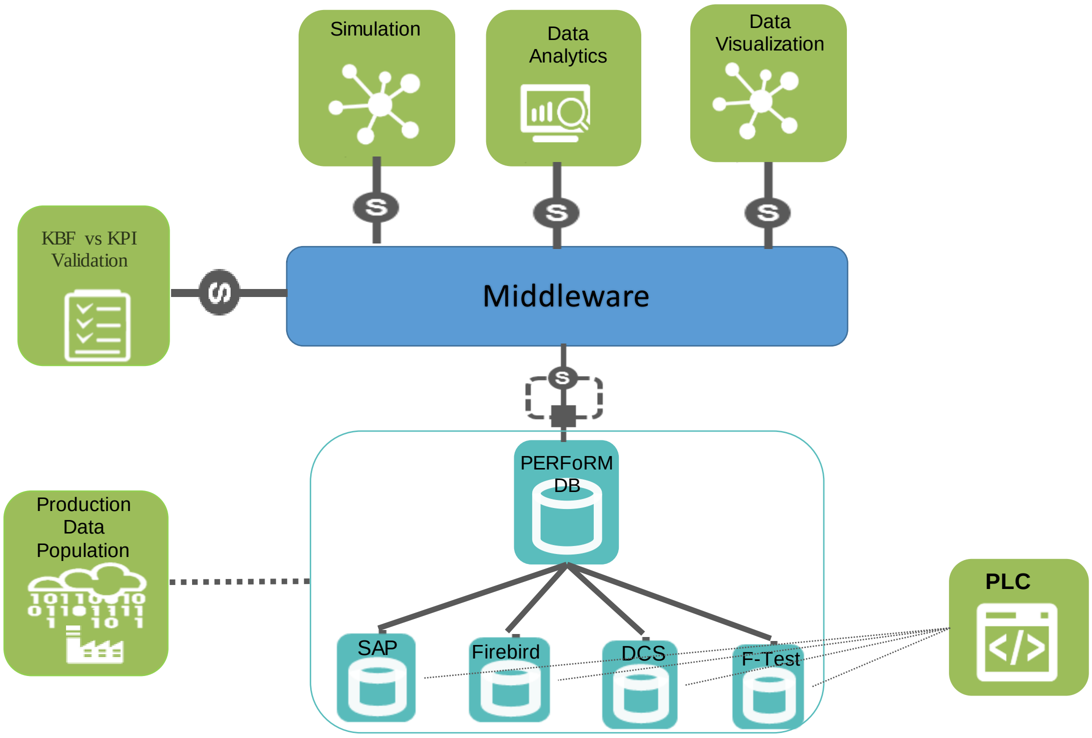

Figure 8. Whirlpool Manufacturing Information System Architecture

4.3 Environment Requirements

The environment requirements for the Whirlpool use case can be seen below:

- Tools deployed in **Virtual Machines** (VMWare)
- **UR10** robot
- Database **– SQL 2000**

4.4 Test scenarios (Whirlpool, Polimi, SmartFactory, Siemens, MTC)

The test scenarios for Whirlpool's use case can be seen in Table 2. The priorities have also been identified as seen in Table 2.

Table 2. Whirlpool's test scenarios

Test Scenario ID	Test Scenario	Priority
TS-WHR-F-1	Correct population of DBs with useful data to test activities	(only for validation process)
TS-WHR-F-2	Simulation activity on process production behaviour with KPI evaluation	High
TS-WHR-F-3	Data aggregation using Data Analytics in order to identify correct information about production process	High
TS-WHR-F-4	Visualisation activity needed to point out the evaluation of data coming from both simulation and field activities	High
TS-WHR-F-5	Data population of PERFoRM DB using PLCs and OPC-UA interface	Low
TS-WHR-F-6.1	Decision maker activity: a new set of KBFs are defined to optimise the process and to remake the simulation	High
TS-WHR-F-6.2	Decision Maker activity: Field configuration	High

4.4.1 Test scenarios description

TS-WHR-F-1

The overall reconfiguration process should be based on the shop floor data. Nevertheless, it is possible to provide specific data adapted only for test activity in order to validate how the PERFoRM DB communicates with the overall architecture and to verify how different IT services (Simulation, Data Analytics and Visualisation) can retrieve useful information.

TS-WHR-F-2

The simulation tool should be able to run the process behaviour in a digital domain, by evaluating and anticipating the production process indicators. This means that different testing activities need to be performed in order to validate if the digital model can fit within the real manufacturing environment, if the manufacturing data have been collected correctly and if the different KPIs are coherently evaluated.

TS-WHR-F-3

Data Analytics will be used in order to validate the ability of providing strategic information to the visualisation tool, taking also into account the results from the simulation activity.

TS-WHR-F-4

The data that have been collected and analysed need to be visualised by different stakeholders, so a strategic decision that will be aligned with plant’s operational goals can be proposed.

TS-WHR-F-5

This scenario aims at the verification of the connection among the PLCs and the different PERFoRM databases. In particular, it will validate the communication between the existing manufacturing databases and the PERFoRM database and also the interaction among the PLCs and the existing databases. In addition, this scenario should verify the feasibility to interact with OPC-UA.

TS-WHR-F-6

This scenario can be split into two different validation processes. The former aims at validating the decision maker capability of selecting and adjusting a new set of KBFs that will allow the process optimisation. The latter targets at testing the reconfiguration mechanism that will be applied at the shop floor level (i.e. machine disposition, dispatching management etc.) and aims at improving the overall production activity.

5 GKN use case (GKN, MTC, Paro, SmartFactory)

The GKN use case is represented by a factory that manufactures complex, high value jet engine components with very stringent requirements. GKN uses advanced material and processes in order to manufacture high quality jet engine components. The current level of automation is low and is based on isolated process automation cells with low level of process flow integration. The main objective of the GKN use case is to: (1) implement a Micro-Production Flow Cell that is able to reduce change over time and to realise different products and (2) be flexibility and re-configurability aspect to the whole industrial plant. Further details on this use case can be found in D10.1: "*GKN Use Case goals, KPIs and requirements*" (GKN, 2016). An innovative reconfigurable robotic cell will be developed within PERFoRM and installed at GKN's site.

5.1 Process and requirements (GKN, MTC, Polimi, SmartFactory)

In order to test and demonstrate the reconfigurable robotic cell, four different processes have been defined: brushing/ removal of oxides, marking, surface inspection, dimensional inspection (AB, 2016). An overview of the concept of the demonstrator, including the hardware and software details, can be seen in Figure 9.

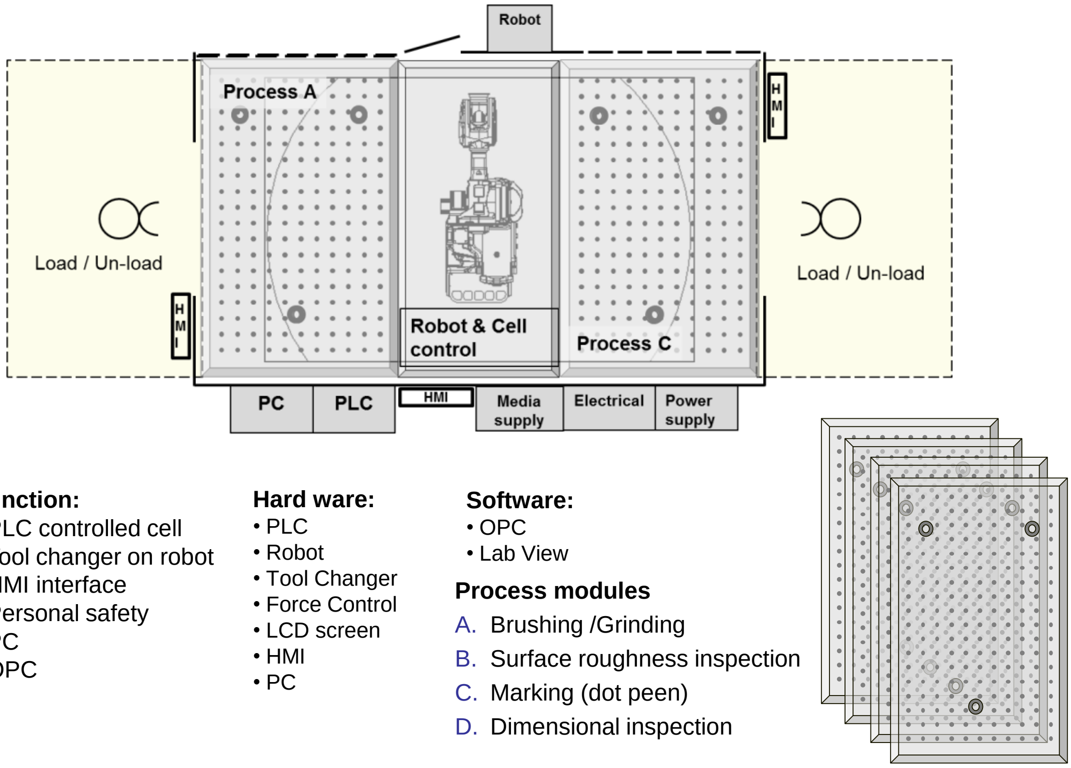

Figure 9. GKN's demonstrator concept

A system with flexible processes will allow the production of various types of products and variants within the same cell. As a result, the production will be adapted to customer demands and able to handle changes on planning and scheduling. In addition, flexibility in the tooling and fixture will allow quick changeovers, while a variety of different operations and geometries will be able to be produced by just one cell.

GKN's cell will be able to adjust to changes easily through the reconfiguration system. The main functions of the reconfiguration system that will be enabled through the implementation of the PERFoRM ML data model and the reconfiguration tool can be seen below (AB, 2016):

a) Change process modules in the cell – i.e. plug-out and plug-in another process module.
b) Modify or rebuild the cell and/or process module to another process function – i.e. a large change over that cannot be managed within the boundaries of flexibility. Each module's change will require an update of the process module definition and the data structure as well as an updated variant of the agent.
c) Introduce a new type of process module – i.e. a new process module for another type of process, or for a product that does not fit any current process module. Every time that a new process module will be added, a new agent will be created at the reconfiguration tool.
d) Move the process module to another cell – i.e. the function can be developed at one location and moved, or shared between different value streams. This will not be demonstrated within PERFoRM, but it is part of GKN's plan at the PTC in Trollhättan.
e) Move the entire cell to another location or production site – i.e. move the function and capacity to where it is needed and/or make changes to the layout to improve the value stream. GKN's long term vision involves change of cell's location, but this will not be demonstrated within PERFoRM.

GKN also identified specific KPIs that will be used to validate the developed solutions. The measurements have been defined as follows:

a) Production flexibility of similar product variants (i.e. built-in and managed by the cell control system) — "0"

b) Changeover / Set-up time for new fixture and/or tooling (i.e. to change and/or replace on the process module in the cell) — < 20 minutes

c) Changeover to a another process module (i.e. plug-out, replace process module and plug-in) — < 30 minutes

5.2 Architecture and Description of Test Components (GKN, MTC, IPB, Loccioni)

GKN's system architecture is illustrated by Figure 10. The devices and the various communication systems that are going to be implemented on GKN's use case are still under definition. The ABB 6640-180 IRC6 robot, which is most likely to be used as cell's robot, will be controlled by a Siemens S7-300 PLC. Each process will be controlled by its own PLC. The reconfiguration mechanism as well as the data visualisation system will upgrade the performance of the whole concept.

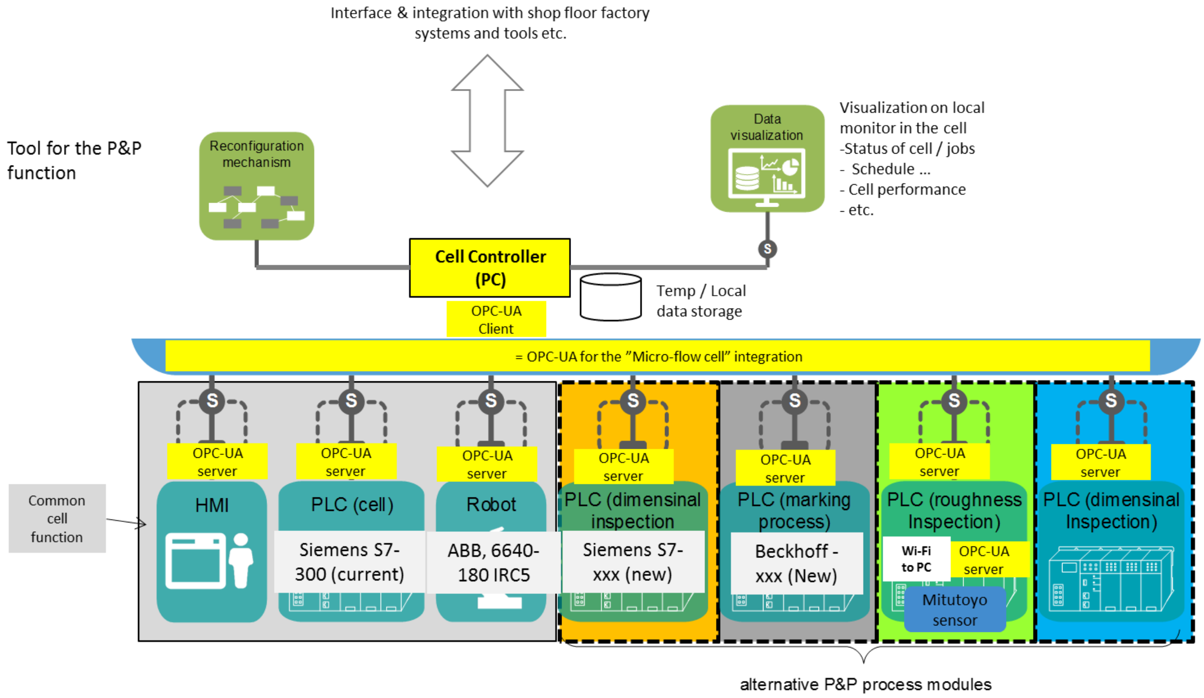

Figure 10. GKN's system architecture

An OPC-UA client will be built in order to allow Middleware functionality. Within PERFoRM two different adaptors are being developed. The first adaptor is being developed by Loccioni and allows the communication between a Mitutoyo sensor (used for roughness inspection) and the OPC-UA server.

Paro will be developing the second adaptor that will be used as an interface that will enable the communication between the OPC-UA and the Siemens S7 PLC.

The reconfiguration mechanism, that is being developed and tested by IPB, will provide new capabilities that will permit the system to change its own functionalities and adapt to different situations and larger changeovers.

5.3 Environment Requirements

The environment requirements for the GKN use case can be seen below:

- **OPC-UA** as the cell level middleware
- **Biz-Talk** as the factory middleware
- **S7-300** for the Cell Control
- A **DNC system** (HI-Fit) to download robot programs to the robot controller from a repository
- **Mitutoyo** roughness sensor
- **ABB 6640** robot
- Data Visualisation system for visualising each process model e.g. status.
- Scheduling system e.g. tools being developed in WP4

5.4 Test scenarios (GKN, MTC, Paro, SmartFactory, Loccioni, Polimi)

The test and demonstration scenarios for GKN can be seen in Table 3 below:

Table 3. GKN's test scenarios

Test Scenario ID	Test Scenario	Priority
TS-GKN-F-1	Verify that the program from HI-FIT will be able to be transferred to the robot through OPC-UA.	High
TS-GKN-F-2	Verify that the reconfiguration tool is able to detect which process has been plugged in to the right side and which to the left.	High
TS-GKN-F-2.1	The reconfiguration tool can detect a new module/ process, after this will be plugged in.	High
TS-GKN-F-2.2	New module receives the confirmation from reconfiguration tool and start sending data to the OPC-UA.	High
TS-GKN-F-3	The reconfiguration tool, after detecting the processes, can send the data to the scheduling system.	Medium
TS-GKN-F-3.1	The scheduling system receives the data and uses them to optimise the production schedule.	Medium
TS-GKN-F-3.2	The updated schedule will be sent back to the control execution through the OPC-UA.	Medium

TS-GKN-F-4	The specifications of the plugged "Process modules" are available to the HMI.	High
TS-GKN-F-4.1	The current status of the cell is tracked and visualised by software tools.	High
TS-GKN-F-4.2	During the communication with visualisation software, unexpected disconnection happens.	Low

5.4.1 Test scenarios description

TS-GKN-F-1

The robot program that is stored to the HI-FIT needs to access the robot. The script needs to be able to access the cell middleware (OPC-UA). The robot should send a request to the HI-FIT, asking for the program, while HI-FIT from its side, needs to receive the request and respond appropriatly, sending back the required information.

TS-GKN-F-2

The reconfiguration tool should be able to identify the type of the processes that have been plugged into the cell, as well as any change-overs in the plugged modules of the cell (TS-GKN-F-2.1). After each module will be identified by the reconfiguration mechanism, data will be able to be sent or received to/from the cell PLC through the OPC-UA (TS-GKN-F-2.2).

TS-GKN-F-3

After the identification of the plugged modules by the reconfiguration tool, the data can be passed to the scheduling system. The scheduling system after analysing the required data can optimise the schedule of the production (TS-GKN-F-3.1) and send it through the middleware back to the control execution process (TS-GKN-F-3.2).

TS-GKN-F-4

The specifications of the plugged process should be available at the connected HMI. The data should be visualised, using appropriate software, like the WinCC (TS-GKN-F-4.1). When a disconnection happens, the HMI should react with a specific way.

6 E-district use case (IFeVS, Comau, SmartFactory, Uninova, MTC)

The IFEVS use case is involved with the assembly of low cost full electric vehicles with high variants and high quality on low budget assembly lines. The objectives of this use case are to: (1) achieve a high degree of automation systems, (2) improve the efficiency and reproducibility of processes and enable highest product quality, (3) reduce re-work of sub-modules and part rejection and (4) minimise the variability of manual operations.

Further details on the requirements of this use case can be found in D8.1: "*Micro Electric Vehicles description and requirements of architectures in view of flexible manufacturing*" (I-FEVS, 2016), are defined in this section. High flexibility is needed to process all variants on the same cells (islands or work-areas), which at the moment can only be provided by human operators. The lot sizes can vary from few units in the case of vehicles transporting temperature controlled goods such as pharmaceutical products, up to tens of units for special freight delivery vans or several hundred for passenger vehicles. The higher degree of automation that can be achieved through PERFoRM's architecture, is expected to improve the efficiency and reproducibility of the processes and enable higher quality of products.

6.1 Process and requirements (IFeVS, Comau, SmartFactory, Uninova, MTC)

In the first period of the project, E-district's goal was to define the system architecture and to simplify the overall assembly process of making the same chassis with a novel approach aiming at:

- Variable-demand manufacturing,
- High-mix manufacturing,
- Manufacturing per which non-recurring engineering costs become a large portion of the overall product cost,
- Rate-dependent production.

The complete assembling line has been finalised and this has demonstrated an improvement to the original expected output (50 vehicles per day). All the necessary production steps have been studied from the flexible assembling of the chassis, the motorised powertrain, interior and exterior panels including testing areas such as:

- 3D geometry of the chassis,
- battery pack
- powertrain
- electric systems.

A solution for the complete microfactory plant has been studied to be energy independent.

The production process is composed of the following major steps:

1. Material Incoming: Raw material (high strength steel) and mechanical, electrical and electronic components are being delivered on a daily basis. They are being checked and stored in a warehouse.
2. Chassis fabrication: The process aims at producing the complete chassis of the vehicle and includes these major steps:
    - Laser cut of the steel tubes
    - The tubes are positioned on the templates in the working islands
    - Manual welding of the parts
    - Assembling of the submodules composing the full chassis
    - Geometrical testing of the chassis
    - Painting of the full chassis
3. Axle system assembling composed by:
    - Laser cut of the steel tubes
    - The tubes are positioned on the templates in the working islands

- Manual welding of the parts
- Painting of the full axle frame
- Assembling of the components composing the full axle system (motor,inverter, transmission etc.)
- Testing of the motorised axle system

Before the final demonstrator will be built at E-district's site, some testing activities related to the system level will be performed by Uninova.

6.2 Architecture and Description of Test elements (IFeVS, Comau, Uninova, MTC)

E-district's system architecture can be seen in Figure 11.The working island will be controlled by a Siemens IM-151 PLC that will be responsible for monitoring the various sensors that will be used on the welding station.

In addition, the MES system developed by XETICS will be used in this use case for the following tasks:

- Production scheduling
- Traceability of production
- Traceability of used materials
- Control of the production flow
- Visualisation and monitoring of the KPIs

The MES system will be connected to the middleware architecture that is being developed by Uninova and it is based on the Siemens WinCC OA platform. All the data will be stored on the decentralised cloud based architecture.

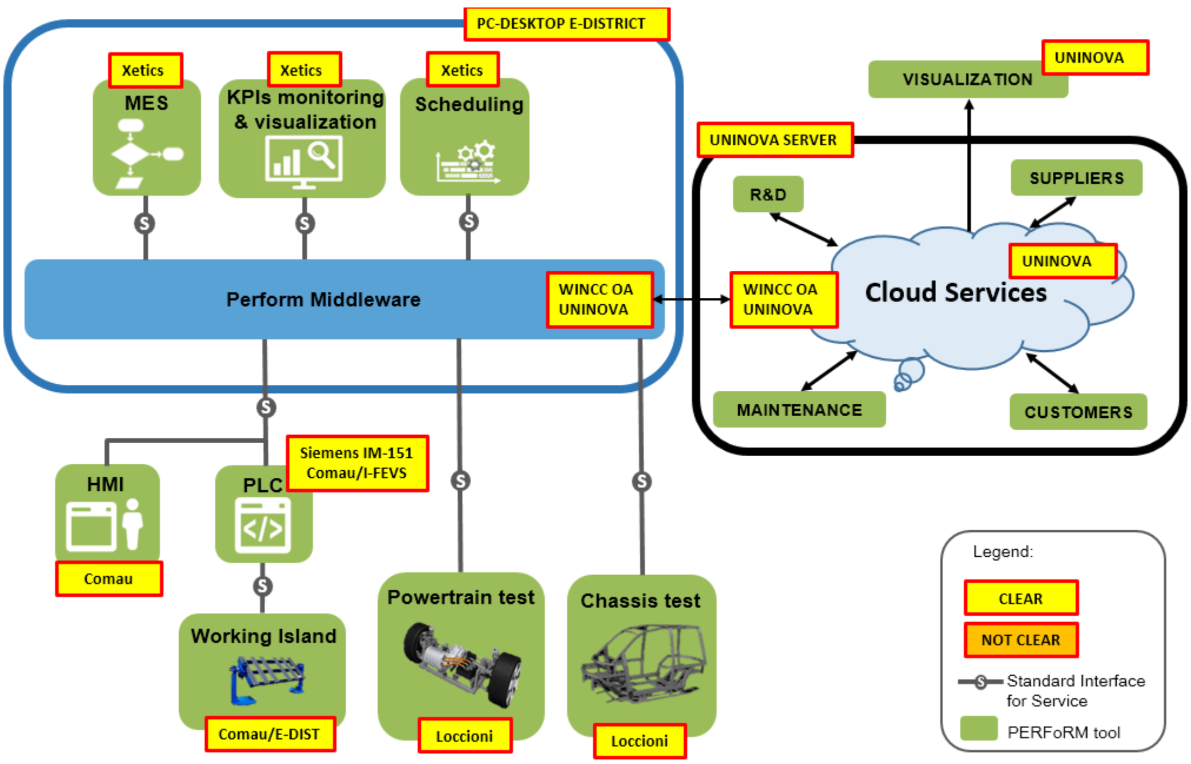

Figure 11. E-district block diagram

The testing activities that will be performed in terms of Task 6.3, will be based on the instantiation of the architecture, which is illustrated in Figure 12.

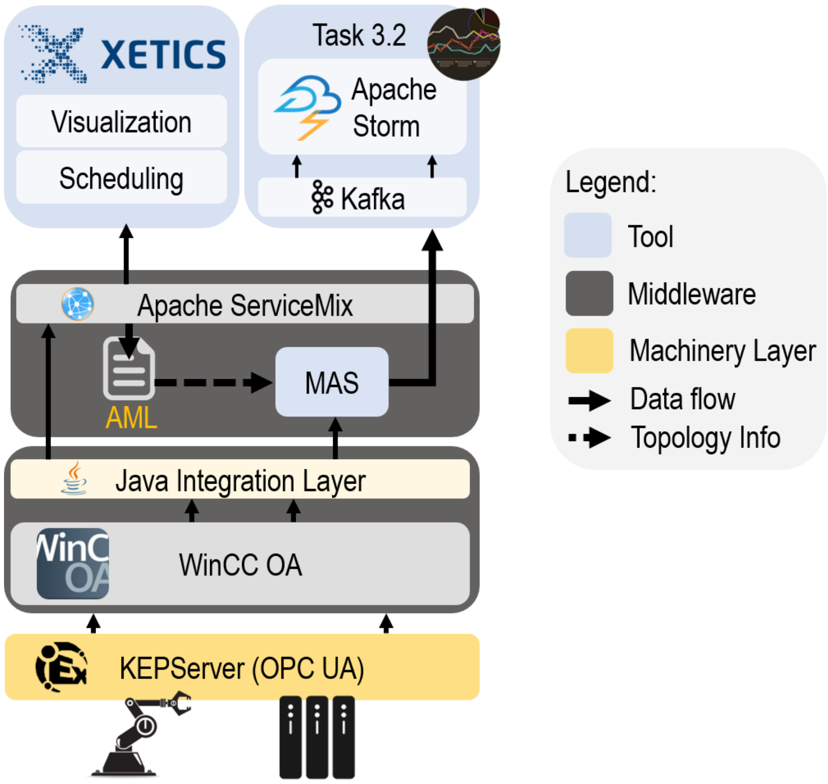

Figure 12. - E-district Test Scenarios Instantiation

In essence, the shop-floor data can be accessed via a KEPServer instance, an OPC UA server which connects directly to the WinCC OA middleware via a built-in driver. The middleware is integrated with the IT-level tools via a REST-based Java integration layer, allowing the data from the machinery level to be updated in an AutomationML (AML) file via the Apache ServiceMix.

Additionally, this data is also being collected by a multi-agent system in near real-time period, pre-processed and sent to a data analysis module. After the completion of the pre-processing task, the data is relayed to an Apache Kafka queue which streams it to an Apache Storm analysis module that enables the early detection and prevention of possible deviations from the expected operation parameters.

Finally, a scheduling and visualisation tool provided by Xetics will also be integrated into this scenario. The tool will be able to receive the orders as well as the topology and operations information through the Apache ServiceMix layer. The output will be a new production schedule, whilst providing the KPI monitoring and visualisation capabilities.

6.3 Environment Requirements

The environment requirements for the IFeVS use case can be seen below:

- **Comau** robot
- PLC **Siemens** IM-151
- **Xetics** MES, Scheduling
- **WinCC** OA Platform

6.4 Test scenarios (IFeVS, Comau, Polimi, SmartFactory, Xetics, Loccioni, HSEL, TUBS, Uninova, MTC)

Table 4. E-district's test scenarios

Test Scenario ID	Test Scenario	Priority
TS- E-DISTRICT -F-1	Verify that the shop-floor data can be accessed in WinCC OA via the KEPServer through OPC-UA.	High
TS- E-DISTRICT -F-1.1	Verify that the data is correctly read via OPC-UA in KEPServer.	High
TS- E-DISTRICT -F-1.2	Verify that the middleware (WinCC OA) can communicate with KEPServer via the built-in driver and access the data.	High
TS- E-DISTRICT -F-2	Verify the connection between WinCC OA and Apache ServiceMix via the Java integration Layer	High
TS- E-DISTRICT -F-3	The MAS can subscribe to values from WinCC OA	Medium
TS- E-DISTRICT -F-4	Apache ServiceMix can update the AML file	High
TS- E-DISTRICT -F-5	The data analysis tool can compute trends based on shop-floor data	Medium
TS- E-DISTRICT -F-5.1	Verify if the MAS can relay data to the Apache Kafka queue	Medium
TS- E-DISTRICT -F-6	The current status of the cell is tracked and visualised by the respective tools.	Medium
TS- E-DISTRICT -F-7	The Xetics scheduler can output a new production schedule based on the shop-floor data	Medium
TS- E-DISTRICT -F-7.1	The updated schedule can be shown to the operator in the HMI	Medium

6.4.1 Test scenarios description

TS-E-DISTRICT-F-1

The data needs to accessible via the WinCC OA middleware such that it is available to the IT level, such that the IT level can perform its tasks. As such, the OPC UA server instance (KEPServer) needs to be able to communicate with the hardware in the shop-floor via the OPC UA (TS-E-DISTRICT-F-1.1), and transfer the data to WinCC OA via the built-in driver (TS-E-DISTRICT-F-1.2).

TS- E-DISTRICT -F-2

In order to ensure that the data can be accessed by the tools plugged to the middleware at the IT-level through service calls, a seamless connection and data exchange between the middleware layers is necessary. This should be achieved through the REST-based java integration layer connecting the two.

TS- E-DISTRICT -F-3

As the main source of near real-time production data, the multi-agent based data acquisition needs to be able to subscribe to specific values exposed by the WinCC OA in order to relay them to the necessary tools.

TS- E-DISTRICT -F-4

A service hosted in Apache ServiceMix should be able to extract the system data from the WinCC OA and update the AML file describing the current system state, including the topology information, as well as the machine and product's production states.

TS- E-DISTRICT -F-5

Based on the shop-floor data streams relayed by the near real-time data acquisition MAS to Apache Kafka (TS-E-DISTRICT-F-5.1), the analysis network based on Apache Storm should be able to compute trends and detect possible deviations early in the production process, enabling the triggering of alarms as necessary to alert the operators.

TS- E-DISTRICT -F-6

The current status of the cell, based on the data extracted from the shop-floor, should be tracked and displayed by the respective visualisation tools.

TS- E-DISTRICT -F-7

As new production orders are issued, along with updated shop-floor data, the Xetics scheduler should compute new production schedules based on the freshly arrived data, publishing it to an HMI where it can be consulted by the operators (TS-E-DISTRICT-F-7.1).

7 Testing and demonstration approach (MTC, SmartFactory)

7.1 Testing Approach at MTC

7.1.1 Testing UR's adaptor

The first iteration of the test bed has been set up at the MTC including an UR10 robot (Machinery Level) and the PERFoRM middleware (Apache Service Mix) as seen in Figure 13.

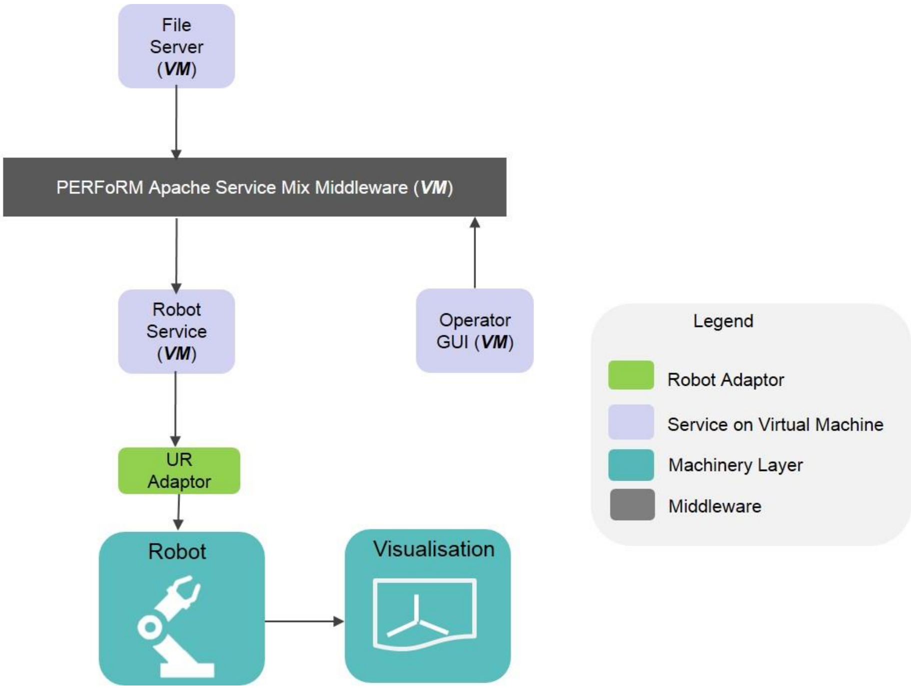

Figure 13. Initial set-up of demonstration using the PERFoRM compliant middleware (Apache Service Mix)

This demonstration is aligned to the Leak Test Scenario from the WHR use case (Figure 14). The demonstrator involves a file repository which stores the CAD files, and a technology adaptor which translates a particular CAD file to a UR Script and transfer this script to the robot. It is to be noted that this demonstrator can be used to test other Machinery level assets.

Figure 14. Universal robot with 3D printed probe for leak tests

For the purpose of the demonstrator, four virtual machines (VM) has been set-up to replicate the de-coupling between the individual systems. The VMs also allow a user to directly copy the VM image to individual host machines as well, and this can result in four different host PCs running the File Server, Operator Graphical User Interface (GUI), Robot Service and the Middleware respectively. The function of each of the VMs are listed below:

- File Server (VM): hosts the CAD files for performing the robot leak test
- Operator GUI VM: reads the list of CAD files from the file storage location and displays them such that an operator can select a particular file
- Robot Service VM: is used to pull a particular CAD file from the File Server and push it to a folder that the UR adaptor can read, fetch and translate it to UR Script
- Middleware VM: comprises of an Apache Service Mix

The testing approach and the sequence of operations for this demonstrator can be seen below:

- An operator selects a CAD file from the list displayed on the Operator GUI. This results in the Operator GUI service sending a trigger to the Robot Service, which then calls the FTP service requesting the file chosen by the operator.
- The selected CAD file from the File Server VM is being downloaded, communicating via the middleware, to a folder that the Robot service can access
- Once the Robot service gets the file, it pushes this file to the Adaptor
- The Adaptor translates the CAD file to a URScript and pushes the script to the robot.
- The linear movements and the rotation in x, y and z directions are recorded via using a test script. The actual and the planned movements are then and visualised using a Python script.

The actual flow of messages is illustrated by Figure 15. This demonstrator can be used to test other Machinery Level adaptors such the Mitutoyo surface roughness adaptor for the GKN use case. Additionally, this test demonstrator can also be extended to test other technology developments within WP2, WP3 and WP4.

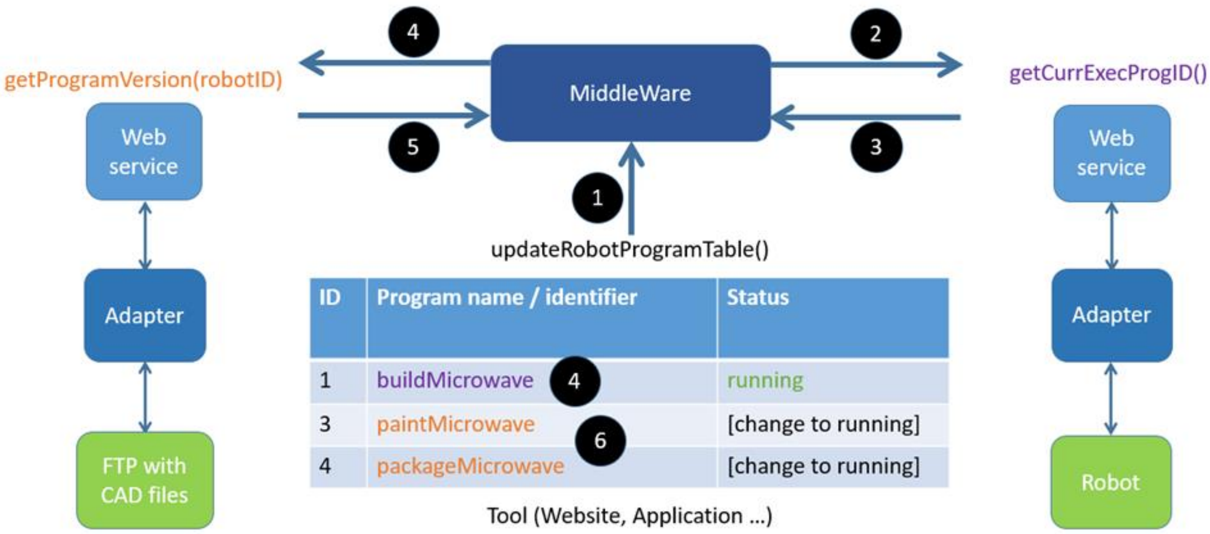

Figure 15. Demonstrator's architecture – Flow of messages

7.1.2 Testing Mitutoyo's adaptor

An adaptor has also been developed for a Mitutoyo sensor (used by GKN for the roughness inspection process). Mitotoyo’s sensor adaptor allows communication between the sensor and PERFoRM’s middleware, by making use of an OPC-UA server/ client architecture. In order to increase the cell’s flexibility, the sensor is communicating through WiFi connection with the PC (Loccioni, 2017), instead of a wired connection.

The first testing approach at MTC has been performed using simulated data that the adaptor could retrieve from a .tdms file. LabVIEW and the Datalogging amd Supervisory Control Module were installed in two different machines to replicate OPC-UA client/ server communication.

Further testing will be conducted by integrating the Mitutoyo sensor to the PERFoRM demonstrator architecture as seen in Figure 12.

7.1.3 Small Component Assembly Cell

The Small Component Assembly Cell SCALP system consists of two underslung KUKA robots, a servo conveyor and a number of process stations and tools. The system is programmed using a high level state machine, with each state running a cell sequence. The sequences contain low level instructions such as selecting individual robot programs, moving the conveyor stations and activating tool operations. Because the sequence scripts can be reused, this allows for quick reconfiguration of the cell to perform a new task.

Figure 16. Small Component Assembly Cell SCALP

The physical interface provided is Ethernet, connected to the SCALP cell’s internal network. Connected to the internal network is an industrial PC which runs the main control software for the cell. The software exposes several C# services that the GUI, a separate piece of software, uses to control the cell in response to the users instructions. However, this PC is not suitable for end users’ direct interface, but may be used by custom software to interact with the hardware. Moreover, there are a number of cameras in the SCALP box to stream live video to the outside world. Therefore the type of communication channel requires high bandwidth, near real-time capabilities and provides flexibility in the communication protocol.

In order to run programs, the SCALP cell needs to be in a safe state. This includes all doors and interlocks being closed, all emergency stops released and any errors acknowledged. Only then will the cell's controls be fully activated. This is an important safety feature and the SCALP control software works within this restriction. Thus, the cell needs to be left in a safe state when attempting to use the external control software. If program execution is attempted when the cell is not ready, this results in a 'FAILED' response being sent back to the interfacing technology (MTC, 2017). The SCALP cell can be used for validation and demonstration of the reconfiguration tool and the scheduling system that are being developed for GKN's use case.

7.2 Testing Approach at IPB

IPB is the main developer of the agent based reconfiguration system for the GKN use case. As part of the development flow, IPB has used its small scale production system (see the Deliverable 6.2 for more details) as a workbench for conducting preliminary tests and validation for the agent based reconfiguration.

The set of tests conducted allowed a preliminary validation of the capability of the agent based system to correctly collect the generated data by being used as an intermediary, which means an OPC-UA based infrastructure.

The summary of conducted tests can be shown as follows:

- Agents were deployed into a desk computer were an OPC-UA server was running, exposing all the data being generated in the production system;
- Agents connected and subscribed to key item events, aiming to verify that the agents are able to recognize changes (e.g., physical changes) in the system;
- The aforementioned events are simulated by powering-up and powering-down the system resources (emulating the plugging-in/out of the GKN's process);
- The multi-agent system, based on the events detection, must be able to exchange information in order to recognise what are the connected processes and where they are located (aiming to have a logical representation of the physical system structure);
- Key information must be displayed in a Human-Machine Interface (HMI), aiming to simulate information that can be showed to the human operator responsible for the management of the GKN production cell (the HMI must be populated with data coming from the OPC-UA infrastructure).

All of the above mentioned tests were conducted with success, validating, in a laboratory environment, the agent based reconfiguration tool connection with the IT infrastructure.

Naturally, further testing situations will be conducted and are already designed, namely those allowing to verify and assess the agent's capability to successfully be integrated in a PERFoRM compliant based system, particularly:

- Dynamic launch of the necessary set of agents based on the instantiated data model for the GKN use case;
- Verify the agent's connection with a Servicemix based middleware by subscribing the needed events in the MW;

- Test that the agents are capable to parse their configuration from the instantiated data model, allowing them to be self-configurable;

7.3 Testing Approach at SmartFactory (SmartFactory)

7.3.1 Test Environment

In order to validate the test scenarios defined within WP6, a new demonstrator, the "Mini Flexible Cell", is currently being set up at SmartFactory. The demonstrator is aligned to the GKN use-case, but should also allow test scenarios for the other use-cases of the PERFoRM project. The basic idea is to show Plug & Play concepts and technologies on both module and subsystem levels.

Hence, it is composed of a flexible cell including a robot module and different process modules that can be coupled via Plug & Play. Figure 17 shows the architecture of the demonstrator. More details can be found in D6.2 from Task 6.2. The basic setup will include a collaborative robot module (probably a Universal Robot 3) and two process modules. Furthermore, some active subsystems of the process modules will be designed as encapsulated components that can be easily added and removed thanks to adequate mechanical, electromechanical and IT interfaces.

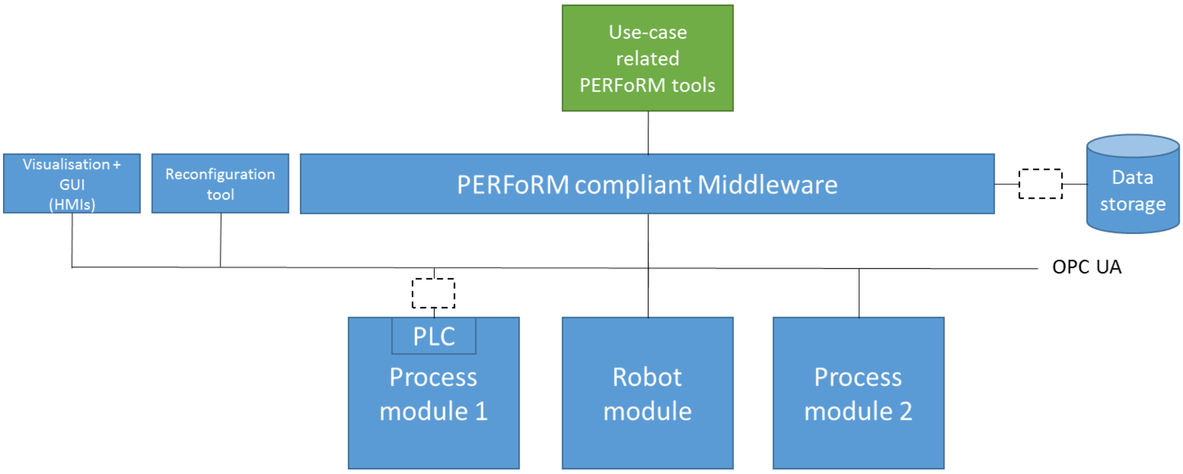

Figure 17: Basic architecture of the Mini Flexible Cell

The current idea is to design the two process modules in the following way:

- A fully automated one with active subsystems collaborating with the robot
- A manual one where the human collaborates with the robot

The first definition of the IT-architecture (see Figure 15) based on the concepts of PERFoRM, enabling a high coverage of required tests and demonstrations. The current concept to enable reconfiguration is by integrating the cell level with OPC-UA. Therefore the module controller has to be upgraded with an OPC-UA adapter, which is in line with the use case of GKN. The reconfiguration tool enables the Plug & Play capabilities of the cell by reading and writing to OPC-UA nodes or using OPC-UA functions. Other tools like a DNC-tool for the robot program download, Simulation or Monitoring tools can be attached to the integration layer by using the PERFoRM-Interfaces and a PERFoRM compliant middleware.

7.3.2 Testing Approach

Within Task 6.3, the focus of the testing approach in set on the overall workflows, from IT level to shop floor.

GKN use case – Reconfiguration of the cell

The test workflow for the GKN use case focusses on the DNC, Reconfiguration and Data Visualisation tools.

1. Data of the machines, in particular of the PLC, are stored in the database. They reflect the current state of the module
2. The modules data are visualised on the HMI´s Data Visualisation tool
3. The operator enters an order on the GUI - he selects a variant of the product
4. The order is stored in the database
5. The HMIs show the needed configuration of the cell and the modules layout according to the order
6. The operator configures the cell and the modules and confirms on the GUI
7. The layout of the cell and of the process modules is detected by the Reconfiguration tool
8. After confirmation of the Reconfiguration tool, the operator starts the process on the GUI
9. The robot downloads the appropriate program from the DNC and the process starts
10. The current status of the cell and of the modules is visualised on the HMIs (e.g. magazine status, order status)

Testing of further workflows

The Mini Flexible Cell could also allow to demonstrate or test some scenarios for other use-cases. Exemplarily, the integration of the KPI Monitoring and Visualisation tool could allow to measure and visualize some machines and process data (e.g. actual process time) and compare them to target data (e.g. target process time). Based on this, a new reconfiguration of the cell or of the modules layout could be proposed by the Simulation tool. Also, the Scheduling tool could be implemented to propose an optimal production schedule based on the entered orders and the machines data.

7.4 Test risks (all)

The risks for the approach highlighted in this document can be seen below:

- It is to be noted that validation of some of the test scenarios may not be performed as a part of WP6 due the lack of suitable infrastructure.
- Though some of the new technologies implemented within the PERFoRM project will be demonstrated within the test beds (at MTC and SmartFactory), the appropriate demonstration for Task 6.3 may be within the use-case production facilities.
- Additionally, some of the testing infrastructure may not be ready at the use case facilities within the stipulated time schedule for Task 6.3 (ends M24). These test scenarios can be tested as a part of WP5 or WPs7-10.
- Some of the assets identified as potential assets (MTC, 2016) for the testing at the test labs such as MTC and SmartFactory may not be available at the time of testing as the assets are shared between multiple projects.

8 Conclusion (MTC)

The objective of the PERFoRM architecture is to support a new generation of agile manufacturing systems based on the plug-and-produce concept, thus enabling the production of smaller lot sizes, more customised products, shorter lead times and shorter time-to-market. WP6 is responsible for de-risking the technologies developed for supporting the proposed architecture.

A brief detail of the process flow within the use cases has been presented. The architecture for each use case has been also presented. Additionally, test scenarios for the use cases have been presented. It is to be noted that some of test scenarios may be validated at the industrial use cases sites instead of the test beds (MTC, SmartFactory) because of the lack of suitable infrastructure.

9 References

AB, G. A. S., 2016. *GKN Use Case goals, KPIs and requirements,* s.l.: PERFoRM project.

FhG-IPA, 2016. *Requirements Review, evaluation and selection of best available Technologies and Tools,* s.l.: PERFoRM project.

GKN, 2016. *GKN Use Case Goals, KPIs and requirements - specifications of applications, functions and requirements for the "micro-flow cell",* s.l.: s.n.

I-FEVS, 2016. *Micro Electriv Vehicles description and requirements of architectures in viw of flexible manufacturing,* s.l.: PERFoRM.

IPB, 2016. *Definition of the System architecture,* s.l.: PERFoRM project.

Loccioni, 2017. *Adaptors implementation for at least three different prodution resources,* s.l.: PERFoRM.

MTC, 2016. *Report on Self-Adaptive Machines Demonstrator Design and Set-up,* s.l.: PERFoRM project.

MTC, 2017. *11287 - Shop Floor Connectivity and Interoperability - D3: Implementation Report ,* s.l.: s.n.

POLIMI, 2016. *Guidelines for Seamless Integration of Humans as Flexibility Driver in Flexible Production Systems,* s.l.: PERFoRM project.

Siemens AG, C., 2016. *Siemens description and requirements of architectures for retrofitting production equipment,* s.l.: PERFoRM project.

Siemens, 2016. *Report on Decentralized COntrol & Distributed Manufacturing Operation Systems for Flexible and Reconfigurable Production Environments,* s.l.: PERFoRM project.

UNINOVA, 2017. *Specification of the Generic Interfaces for Machiner, Control Systems and Data Backbone,* s.l.: s.n.

WHR, 2016. *Whirpool description of Requirements and Architecture Design,* s.l.: s.n.

Appendix

I. Acronyms

Abbreviation	Explanation
SOA	Service Oriented Architecture
XML	Extensible Markup Language
KPIs	Key Performance Indicators
KBFs	Key Business Factors
MAS	MultiAgent System
AML	AutomationML
FOR	Fall Off Rate
SCALP	Small Component Assembly Cell
PLM	Product Lifecycle Management