IoT@Work

WP4 – PILOT SCENARIO AND TECHNOLOGIES TESTING

D4.2 – EVALUATION REPORT: EVALUATION RESULTS

Reference: IoT@Work/WP4/D4.2/D4.2_IoT@Work_Evaluation_report_evaluation_results_v1.3.doc

Category: Report

Deliverable Responsible: Siemens

Author(s): A. Houyou (Editor), K. Fischer (Siemens); A. Zanella, G. Vivo (CRF), D. Rotondi, S. Piccione, M. Comolli, F. Corvino, (TXT), H. Trsek, J. Imliaz (inIT); J. Claessens (EMIC)

Reviewer(s): All

Due Date: M23

Deliverable Date: 08/03/2013

Revised: 04/07/2013

Project Start Date: 01/06/2010

Project Duration: 36 Months

Status: Final

Availability: Public
Executive Summary

This deliverable provides an overview of the pilot and validation activities that have taken place since D4.1. The performance indicators, developed from the different technical, socio-economic points of view to estimate the impact of the project results is revisited in the early part of the document. The pilot details and validation framework are also addressed in this deliverable. The two developed pilots are a result of the scenario-driven approach that has allowed us to both extract requirements, performance indicators, and develop the architecture. The pilots are developed around some key application scenarios that should capture the impact of our Plug&Work approach and IoT-centred architecture on manufacturing scenarios. The FIAT-based pilot, re-creates the manufacturing set-up in keeping the production unit unchanged and making additions in terms of intelligence around the production system. Such extensions include the ability to extend the production through a sensor system, that offers multiple users (involved in maintaining, supplying, managing the production devices) access to the field-level sensor data by directly running applications and their communication system embedded within the same shared infrastructure. Other aspects of Plug&Work related to flexibility both of the hardware and software of a production system are addressed in the inIT pilot. There, modular and self-configuring production systems can be exchanged, reconfigured, and extended on the fly. The scenarios and their respective test-cases explaining the role of the developed solutions within the project and their impact on the state of art production system are addressed as well.
Document History

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Date</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Draft</td>
<td>01/04/2011</td>
<td>G. Vivo, CRF</td>
</tr>
<tr>
<td>0.2</td>
<td>Draft</td>
<td>17/04/2012</td>
<td>G. Vivo, CRF; A. Houyou, Siemens</td>
</tr>
<tr>
<td>0.3</td>
<td>Draft</td>
<td>16/05/2012</td>
<td>G. Vivo, CRF</td>
</tr>
<tr>
<td>0.4</td>
<td>Draft</td>
<td>23/05/2012</td>
<td>G. Vivo, CRF; M. Comolli, F. Corvino, S. Piccione, D Rotondi, TXT</td>
</tr>
<tr>
<td>0.5</td>
<td>Draft</td>
<td>30/05/2012</td>
<td>G. Vivo, CRF</td>
</tr>
<tr>
<td>0.6</td>
<td>Draft</td>
<td>03/06/2012</td>
<td>H. Trsek, inIT,</td>
</tr>
<tr>
<td>0.7</td>
<td>Draft</td>
<td>01/07/2012</td>
<td>A. Houyou, Siemens, J. Imtiaz, inIT</td>
</tr>
<tr>
<td>0.8</td>
<td>Draft</td>
<td>02/09/2012</td>
<td>A. Houyou, Siemens; S. Piccione, D Rotondi, TXT</td>
</tr>
<tr>
<td>0.9</td>
<td>Draft</td>
<td>25/09/2012</td>
<td>A. Houyou, Siemens, G. Vivo, CRF, S. Piccione, D Rotondi, TXT, J. Claessens, EMIC</td>
</tr>
<tr>
<td>1.0</td>
<td>Draft</td>
<td>05/10/2012</td>
<td>A. Houyou, Siemens, G. Vivo, CRF</td>
</tr>
<tr>
<td>1.1</td>
<td>Draft</td>
<td>24/10/2012</td>
<td>A. Houyou, Siemens, G. Vivo, CRF</td>
</tr>
<tr>
<td>1.2</td>
<td>Final</td>
<td>08/03/2013</td>
<td>A. Houyou, Siemens</td>
</tr>
<tr>
<td>1.3</td>
<td>Final</td>
<td>04/07/2013</td>
<td>A. Houyou, Siemens</td>
</tr>
</tbody>
</table>

Summary of Changes

<table>
<thead>
<tr>
<th>Version</th>
<th>Section(s)</th>
<th>Synopsis of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>All</td>
<td>TOC, repartition of work and general template</td>
</tr>
<tr>
<td>0.2</td>
<td>All</td>
<td>Overview, integration of first contributions</td>
</tr>
<tr>
<td>0.3</td>
<td>2</td>
<td>Finalisation of introduction and Section on KPI usage</td>
</tr>
<tr>
<td>0.4</td>
<td>3, 4</td>
<td>Reworked description of the CRF pilot, included section on the implementation of the ENS and Capabilities Framework</td>
</tr>
<tr>
<td>0.5</td>
<td>All</td>
<td>Overall revision of the CRF pilot description</td>
</tr>
<tr>
<td>0.6</td>
<td>3, 4</td>
<td>Overall revision of the inIT pilot description</td>
</tr>
<tr>
<td>0.7</td>
<td>4</td>
<td>Scenarios updates</td>
</tr>
<tr>
<td>0.8</td>
<td>4</td>
<td>Scenarios updates</td>
</tr>
<tr>
<td>0.9</td>
<td>2, 4</td>
<td>Revisions and updates of the sections around the results of the review by Siemens, inIT, EMIC, TXT, CRF</td>
</tr>
<tr>
<td>1.0</td>
<td>All</td>
<td>Finalizing use-cases and scenario analysis</td>
</tr>
<tr>
<td>1.1</td>
<td>All</td>
<td>Review ready version, deciding on KPI comments from the reviewers</td>
</tr>
<tr>
<td>1.2</td>
<td>All</td>
<td>Submission ready version (additional input to KPIs is finally moved to D4.3)</td>
</tr>
<tr>
<td>1.3</td>
<td>All</td>
<td>Revised submission due to slight formatting errors.</td>
</tr>
</tbody>
</table>
Contents

1 Introduction ................................................................................................................................. 9
   1.1 Objectives and scope of IoT@Work and WP4 Pilot Scenario and Technologies Testing ................................................................. 9
   1.2 Objectives and scope of this Deliverable D4.2 ................................................................. 9
   1.3 Structure of this document ......................................................................................... 10

2 KPI detailing and usage for the IoT@Work validation ........................................ 11

3 Detailed descriptions of the IoT@Work Pilots ................................................. 15
   3.1 Welding cell pilot (Recreating the FIAT Manufacturing Setup) ........................................... 15
   3.2 Hybrid manufacturing process pilot (inIT) ................................................................... 20

4 IoT@Work scenarios ............................................................................................................. 23
   4.1 Reference Application and Scenarios ........................................................................ 23
       4.1.1 Building a Multi-tenant Sensor-enabled Smart Factory ........................................ 23
       4.1.2 Technology Integration and Pilot Development .................................................. 24
   4.2 Smoke detection and fan control scenario ................................................................. 25
       4.2.1 Scenario overview .......................................................................................... 29
       4.2.2 Technical details ....................................................................................... 30
   4.3 Remote OEM Monitoring Scenario .......................................................................... 31
   4.4 Plug&Work of sensor/production units (inIT) ............................................................ 33
       4.4.1 Auto-configuration Service ...................................................................... 33
       4.4.2 Scenario overview ................................................................................... 34
       4.4.3 Scenario details ....................................................................................... 35

5 Conclusions ............................................................................................................................. 36

6 References ............................................................................................................................ 37
List of Figures

Figure 2.1: Flow chart detailing the policy of evaluation for the individual KPIs........ 14
Figure 3.1. CRF Pilot architecture ........................................................................ 18
Figure 3.2. inIT pilot architecture ........................................................................... 21
Figure 4.1 – An overview of the proposed Event Notification Service approach ..... 27
Figure 4.2 – CRF Cell Layout and ENS Placement .................................................. 26
Figure 4.3 – Integrating Event Notification System in the Pilot.............................. 29
Figure 4.4 – Integrating Event Notification System and Slice Management ......... 31
Figure 4.5 – Network Slice Management System ...................................................... 32
Figure 4.6 – Scenario at the Lemgo Model Factory .................................................. 34
Figure 4.7 – Architectural view for Module 6 for scenario at the Lemgo Model Factory ............................................................................................................. 35
## List of Tables

- **Table 2.1 – Summary of the IoT@Work KPIs** ................................................................. 13
- **Table 2.2 – Impact of KPIs on domains of relevance and high level aspects** ........ 13
- **Table 3.1 – CRF Pilot: general information** ................................................................. 15
- **Table 3.2 – CRF Pilot: detailed information** ................................................................. 19
- **Table 3.3 – CRF Pilot: actors** ...................................................................................... 19
- **Table 3.4 – inIT Pilot: general information** ................................................................. 20
- **Table 3.5 – inIT Pilot: detailed information** ................................................................. 21
- **Table 3.6 – inIT Pilot: actors** ...................................................................................... 22
- **Table 4.1 – Analysis of the Scenarios from D4.1** ....................................................... 25
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVB</td>
<td><strong>IEEE 802.1 Audio Video Bridging</strong>, a set of IEEE 802 standards for Quality-of-Service in a LAN (802.1Qat, 802.1Qav, 802.1as)</td>
</tr>
<tr>
<td>BPI</td>
<td>Basic Performance Indicator</td>
</tr>
<tr>
<td>CEP</td>
<td><strong>Complex Event Processing</strong> (CEP) refers to the processing of events such as creation, reading, transformation, and filtering. A CEP Engine is a system that (i) identifies specific events from a stream of events according to some patterns; (ii) selects the appropriate reaction to the identified events; (iii) executes the selected reaction. A CEP Engine can be used for supporting predictive maintenance and system monitoring.</td>
</tr>
<tr>
<td>Customer</td>
<td>Ultimate consumer, user, client, beneficiary or second party, see [1].</td>
</tr>
<tr>
<td>EMU</td>
<td>Energy Management Unit</td>
</tr>
<tr>
<td>ENS</td>
<td><strong>The Event Notification Service</strong> (ENS) is a functional component of the IoT@Work Architecture drafted in deliverable D1.2, which acts as a collector and distributor of events coming from different sources and dispatched to different listeners. It is in charge of collecting and dispatching the events used to carry out (remote) monitoring and maintenance activities.</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Ethernet is a family of computer networking technologies for local area networks (LANs).</td>
</tr>
<tr>
<td>Features</td>
<td>Features are identified properties of a software product, which can be related to the quality characteristics [2].</td>
</tr>
<tr>
<td>Functional Requirements</td>
<td>A <strong>functional requirement</strong> specifies what the system must be able to do, the functions it should perform. Functional requirements are associated with specific functions, tasks or behaviours the system must support. This term is used at both the user requirements analysis and software requirements specification phases in the software life cycle. Functional requirements capture the intended behaviour of the system. This behaviour may be expressed as services, tasks or functions the system is required to perform.</td>
</tr>
<tr>
<td>IEEE 802</td>
<td><strong>IEEE 802</strong> refers to a family of IEEE standards dealing with local and metropolitan area networks, e.g. Ethernet (IEEE 802.3) or Wireless LAN (IEEE 802.11).</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>KPI</td>
<td>Key performance indicators</td>
</tr>
<tr>
<td>LMF</td>
<td>The <strong>Lemgo Model Factory</strong> is a small scale system that represents a hybrid manufacturing process.</td>
</tr>
<tr>
<td>Methodology</td>
<td>A <strong>methodology</strong> is a set of instructions (provided through text, computer programs, tools, etc.) that is a step-by-step aid to the user [3]</td>
</tr>
<tr>
<td>Non-Functional Requirement</td>
<td>A <strong>non-functional requirement</strong> specifies an aspect of the system other than its capacity to do things. Examples of non-functional requirements include those relating to performance, accessibility, usability, etc.</td>
</tr>
<tr>
<td>OPC UA</td>
<td><strong>Object Linking and Embedding for Process Control – Unified Architecture</strong>: A new set of standards that incorporates all the functionality of the former OPC standards (and more), but does so using cross platform web services and other modern technology.</td>
</tr>
<tr>
<td><strong>Pilot</strong></td>
<td>The real context of industrial users in which IoT@Work solutions are applied. Pilots are employed to test in a limited context the goodness of a use case solution before its extension on the whole enterprise reality.</td>
</tr>
<tr>
<td><strong>PLC</strong></td>
<td><strong>Programmable logic controller</strong> is a control system that has a user-programmable memory for storage of instructions to implement specific functions such as I/O control, logic, timing, counting, report generation, communication, arithmetic, and data file manipulation.</td>
</tr>
<tr>
<td><strong>PROFINET</strong></td>
<td><strong>PROFINET</strong> is the open industrial Ethernet standard of PROFIBUS &amp; PROFINET International (PI) for automation. It is a specific industrial Ethernet protocol with real-time capabilities for realizing distributed control systems. It replaces the commonly used field buses.</td>
</tr>
<tr>
<td><strong>QoS</strong></td>
<td><strong>Quality-of-Service</strong></td>
</tr>
<tr>
<td><strong>SCADA</strong></td>
<td><strong>Supervisory Control And Data Acquisition</strong> refers to industrial control systems: computer systems that monitor and control industrial processes or infrastructures.</td>
</tr>
<tr>
<td><strong>Software quality</strong></td>
<td>The totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs [2].</td>
</tr>
<tr>
<td><strong>SUD</strong></td>
<td><strong>System Under Discussion / System Under Description</strong></td>
</tr>
<tr>
<td><strong>Test Procedure</strong></td>
<td>The step-by-step process needed in order to verify that the product meets all the requirements identified.</td>
</tr>
<tr>
<td><strong>Use Case</strong></td>
<td>A collection of possible sequences of interactions between the system under discussion and its actors, relating to a particular goal. A use case defines a goal-oriented set of interactions between external users/systems and the system under consideration or development. Use cases allow capturing functional requirements for a Business case.</td>
</tr>
</tbody>
</table>
| **Validation** | 1. "Confirmation by examination and provision of objective evidence that specifications conform to user needs and intended uses, and that the particular requirements implemented can be fulfilled. The primary focus of validation is customer satisfaction."
2. Confirmation by examination and provision of objective evidence that particular requirements for a specific intended use are fulfilled" [2] |
| **VPN Gateway** | **Virtual private network gateway** to allow remote access to the whole plant, e.g. for maintenance purposes. |
| **WLAN AP/STA** | **A Wireless LAN access point or station** provides a wireless connection for the corresponding module. |
1 Introduction

This document represents Deliverable D4.2 of the FP7 “IoT@Work” project (start 1 June 2010 – end 31 May 2013). D4.2 is the second deliverable of the project’s Pilot Scenario and Technologies Testing work package (WP4). It provides a consolidated overview on the pilots, together with the validation of the project developed technologies around a subset of the scenarios already given in D4.1 [6]. The revision of the KPIs and the “Framework for Economic, Social and Technical Validation of Selected Pilot”, are also reported in this document.

1.1 Objectives and scope of IoT@Work and WP4 Pilot Scenario and Technologies Testing

This section highlights the framework of the IoT@Work WP4 activities, giving a general overview of the objectives of the validation WP. Even though it is already included in other deliverables, it is included here so that the present deliverable is self-contained. The IoT@Work project is researching and developing the technologies required for enabling Internet of Things (IoT)-based applications and processes in the manufacturing domain. An IoT architecture should make it easier and quicker for production processes to adapt to new business models and processes. Process and industry automation, however, have strong demands for reliable communication and security guarantees, which an IoT architecture has to support directly. Today, deployment and commissioning of complex production processes and Internet-enabled applications interacting with production systems still require a time-consuming and error-prone manual network configuration process. This is due to the need for maintaining a high level of predictability, safety, and security of the production process and avoiding both safety-critical failures and costly production interruptions. The IoT@Work project intends to deliver tools and runtime mechanisms based on IoT technologies to significantly simplify commissioning, operating, and maintaining complex production processes. This includes self-configuration mechanisms, enabling what we call secure Plug&Work IoT. This also includes mechanisms to make networks and resources better adapt to “changes” and support dynamic and secure adaptation, where the change could be due to infrastructure change, failures, or even process adaptations.

Work Package WP4 focuses on the pilot scenario and technology testing in IoT@Work. The goal of WP4 is to provide a methodology and an IT architecture in order to test and validate, through the piloting activities, the results coming from the research and development activities carried out in the project.

1.2 Objectives and scope of this Deliverable D4.2

This deliverable reports the results of Task 4.2 – “Pilots development”. These pilots have been designed by following a business-driven approach, to make practically available a demonstration platform where to validate the industrial pilot scenarios of IoT@Work. The reference domain is the one of IoT-enabled maintenance and bootstrapping industrial internet in a manufacturing environment that will produce and validate transferable pilot applications.

The practical implementation of the IoT@Work technologies in this industrial pilot will prepare the ground for future industrial breakthroughs in the project’s application scenarios. To accomplish this achievement, two pilot areas have been prepared in two separate manufacturing validation locations in the laboratories of the Production Facilities department of Fiat Research Center and in the Lemgoer Modellfabrik...
located at the Centrum Industrial IT (CIIT) department of the Institute Industrial IT (inIT). Both validation locations include the materials, the specific facilities and equipment necessary to accomplish the goals of each individual pilot.

In this document a consolidated overview of the pilots is provided, together with the guidelines for applying the validation procedures to the use cases of the project.

1.3 Structure of this document

- KPI detailing and usage for the IoT@Work validation: this section summarises the overall validation framework and provides some more details and recommendations on the KPI usage during the final validation activities. Here the focus is on the application of the methodology for the evaluation of the industrial pilot applications, to produce a comprehensive set of validation results under the responsibility of the implementation teams.

- Detailed descriptions of the IoT@Work Pilots: here the detailed identification and description of the pilot scenarios are provided in two distinct sections. Each section presents also a pilot-specific glossary and list of actors, which will be further detailed in the subsequent test-case descriptions.

- IoT@Work Scenarios: this section provides the description of concrete scenarios in each of the Pilots, including a link to the main technology frameworks from partners or third parties.

- Conclusions: this section presents the concluding remarks on the achievements reported in this deliverable.
2 KPI detailing and usage for the IoT@Work validation

As reported in deliverable D4.1 [6], the overall steps of validation for the IoT@Work are summarised as follows:

1. Collection of the high level objectives, user needs, requirements and risks as identified by the activities carried out in the initial stage of the project; then the relevant entities are selected, grouped and clustered;

2. Identification of one or more Basic Performance Indicators (BPIs), for each entity. This phase consists in the systematic identification of the parameters allowing to assess whether - and possibly also in which measure - the given objective/need is satisfied/addressed by the identified reference use case(s). These BPIs will be assessed/evaluated both with and without the IoT@Work solution. This collection and identification phase is therefore exhaustive and provides a set of BPIs that can be used as a starting set for the identification of the relevant KPIs;

3. Synthesis of the KPIs. This phase consists of a generalization work, where the BPIs are clustered into a limited number of related groups;

4. Analysis of the KPIs. Here the identified KPIs are analyzed to evaluate how relevant they are to the requirements of the project;

5. Evaluation of the impact on the validation domains. Here the impact on the Economic, Social, Technical Security, Quality and Safety aspects is evaluated;

6. Application of the validation framework. Here the methodology is applied to the selected test-cases in order to evaluate (in quantitative and verifiable terms) the benefits of the IoT@Work solutions: test-cases are run with and without applying the project solutions and the KPIs are evaluated. Each test-case covers a set of KPIs that can be quantified according to each test-case through the “AS-IS” and the “TO-BE” situation. The way to quantify the improvement is each test-case through the selected KPIs is in progress and will be updated in the version of the document.

7. Analysis of the validation results. Here the different KPIs measures are compared, analyzed and interpreted.

The first five phases consisted in the definition of the methodological framework for the IoT@Work validation, and have been reported in deliverable D4.1 [6]. In the present document the focus is on the application of the methodology for the evaluation on the industrial pilot applications, to produce a comprehensive set of validation results under the responsibility of the implementation teams (step 6). The concluding step 7 (analysis of the validation results) will be carried out in the third year of activities of IoT@Work, and it will be presented in the final deliverable of WP4 (the D4.3 – “Evaluation report: evaluation results and expert feedback”).

The analysis of the project requirements led to the identification of 13 KPIs, which relate to the high-level objectives and user needs of our system. These KPIs were further clustered into different groups of impact, such as social, economic, technical, etc. Both pilots, described in the subsequent Section of this deliverable, will use them to validate the project results as compared to the state of the art.

These KPIs, are reported in Table 2.1. The application of the KPIs in the validation activities provides immediate and consistent access to critical data, which supports a persistent relationship between the performance of the system (to be assessed in the project pilots before and after the usage of the IoT@Work solutions) and the high-
level objectives of the project. In addition, they allow for having immediate access to
detailed, concise and easily accessible data to the system performance, which
enables them to provide insight and enable corrective action in case the components/devices/systems perform in a

<table>
<thead>
<tr>
<th>KPI n.</th>
<th>KPI Definition</th>
<th>KPI Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configuration/Management tool programming costs</td>
<td>Costs depending on the need to carry out software alignment or programming of the network configuration or the network management tools</td>
</tr>
<tr>
<td>2</td>
<td>Downtime costs</td>
<td>Costs occurring due to the need of putting down the system, for a period of time which is more or less extended, but not negligible</td>
</tr>
<tr>
<td>3</td>
<td>Investment costs</td>
<td>Costs of investment, both direct - e.g. the cost of an entire new system/device, in the case it is needed to buy it from scratch, or indirect – in the case the introduction of the addressed features are possible only if other components/infrastructures need to be upgraded</td>
</tr>
<tr>
<td>4</td>
<td>Manpower costs</td>
<td>Costs of the manpower, to be quantified both for the installation/configuration/maintenance (fixed costs) and for the operation (variable costs)</td>
</tr>
<tr>
<td>5</td>
<td>Number of backwards system step to restore a safe starting situation</td>
<td>Number of backward steps needed to restore the system up to a safe starting situation; this KPI should measure the impact of an eventual roll-back procedure, in case of the need to restore the system into a previously working state</td>
</tr>
<tr>
<td>6</td>
<td>Number of configuration errors</td>
<td>The number of individual mistakes occurring in the configuration of the system/device/component under analysis; this KPI is useful to evaluate the complexity of the configuration procedure</td>
</tr>
<tr>
<td>7</td>
<td>Number of other sub-systems/sites impacted</td>
<td>The number of other systems, production cells, or production sites that are directly affected by the introduction of the feature dealt with</td>
</tr>
<tr>
<td>8</td>
<td>Number of un-needed local stops</td>
<td>The number of local stops of the system/device/component under analysis, additional respect to the ones strictly needed to put the system into its normal working state, e.g. due to additional safety requirements, for an extended trial stage of a machinery, etc.</td>
</tr>
<tr>
<td>9</td>
<td>Range of remote access</td>
<td>This KPI can be described in terms of the scale to adopt in order to designate the different levels of geographic determination; for instance the values to attribute to this KPI could be: 0 - No capabilities are available to access the system/device 1 - Very limited capabilities of local access (hardwired, by using dedicated and proprietary interfaces) 2 - Local access (hardwired, by using open protocols/interfaces and completed by easy to use support applications), but with the need to be physically close to the system in order to carry out the operation 3 - Limited capabilities of remote access (wireless remote access based on short range protocols/interfaces – e.g. bluetooth, or wired, based on the available network services) 4 - Remote access within the plant/cell, assisted by dedicated applications, without the need to be physically close to the system in order to carry out the operation 5 - Web based remote access from outside the plant/cell</td>
</tr>
<tr>
<td>10</td>
<td>HMI easiness of use</td>
<td>If applicable, it quantifies the level of easiness of use for the HMI part of the system under analysis</td>
</tr>
<tr>
<td>11</td>
<td>Number of manual steps</td>
<td>Number of the individual elementary and manual operations that are needed to implement the feature dealt with</td>
</tr>
</tbody>
</table>
Similarly with the procedure adopted for defining the KPI #9, this performance index can be described in terms of the scale to adopt for designating the different levels of security considered in the validation phase; also in this case different metrics can be adopted and in the context of the proposed work a dedicated scale will be defined for the IoT@Work validation trials. The security check of the overall approach is also carried out within WP3 specifically using a threat analysis carried out for the system comparing the situation before and after integrating the IoT@Work technologies and system approach.

It is an index related to the impact on the quality aspects: it designates the number of defective parts produced as a consequence of the local stops of the system/device/component under analysis, with the same delimiting characterization of the KPI #8.

Table 2.1 – Summary of the IoT@Work KPIs

<table>
<thead>
<tr>
<th>KPI Description</th>
<th>Economic</th>
<th>Social</th>
<th>Technical</th>
<th>Security</th>
<th>Quality</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration/Management tool programming costs</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Downtime costs</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Investment costs</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manpower costs</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of backwards system step to restore a safe starting situation</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of configuration errors</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of other sub-systems/sites impacted</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of un-needed local stops</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range of remote access</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMI easiness of use</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of manual steps</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level of security (IoT@Work metric)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of scrap parts due to un-needed local stop</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2 – Impact of KPIs on domains of relevance and high level aspects

For the practical usage of the selected KPIs in the project pilots, it is worth to note that these performance indicators are not conveyors of abstract information. On the contrary, they are meant to provide precise parameters and information on the use cases running in the project pilots; in this regard a specific emphasis and recommendation for the proper application of the validation framework is given, by explicitly pointing out their purpose:

- The IoT@Work KPIs will provide a set of measures, enabling to monitor performance, to track progress towards the achievement of objectives, to demonstrate results, and to take corrective actions in case of discrepancies respect to the expected benefits;

- They can be expressed as qualitative or quantitative measures, of different nature and physical metrics depending on the context of application (economic, social, technical, etc.). Once applied in the validation tests, they must always follow the SMART (Specific, Measurable, Achievable, Relevant, and Time-bound) formulation and content of the gathered information;

The flow chart in Figure 2.1 graphically represents the guidelines of usage and can serve as a reminder of the evaluation policy to be applied when the validation tests for the individual KPI will be executed.
Figure 2.1: Flow chart detailing the policy of evaluation for the individual KPIs

- **Relevance.** In the stage of the evaluation of the performance, the first item to assess should be the relevance of the given KPI. This answers the question: to what extent are the goals, defined for the individual KPIs, justified in relation to the needs?

- **Effectiveness.** Then the effectiveness of the KPI should be measured. This will be carried out by answering the question: to what extent have the expected goals been achieved?

- **Efficiency.** Afterwards the efficiency of the given KPI usage will be analyzed. In this case the question to answer will be: have the goals been achieved at the lowest cost/effort?

- **Utility.** Finally the topics of the utility will be faced. The question to answer will be: are the goals or unexpected effects contributing to a net increase of the benefits in the specific domain (economic, social, technical, etc.)?

Deliverable D4.3 – “Evaluation report: evaluation results and expert feedback” will report on the application of the Validation Framework on the Pilot Scenarios. During the final phases of the validation activities the whole evaluation method will be applied on the industrial pilots, whose detailed description is provided in the following Section.

After thorough analysis the decision has been made to move the results of the operational mapping of the KPIs to the use cases demonstrating single or integrated IoT@Work technologies to the D4.3. This evaluation will also list the AS-IS and TO-BE values linked to the KPIs according to both mappings.
3 Detailed descriptions of the IoT@Work Pilots

The pilot description below is a suitable way to:

- Understand the problem at stake in terms of expectancies from the end users and demonstrate the impact of the IoT@Work solutions;
- Elicit relevant KPIs for the test-cases applied to the pilot;
- Highlight the relevance of IoT@Work, in terms of solutions available for the test and their impact on the process, measured on the selected metrics.

It provides a proper identification of the pilot scenario and its description in two distinct sections. Each section presents also a pilot-specific glossary and list of actors, which will be further detailed in the subsequent test-case description.

The pilot description documentation includes the following sections:

“1. Pilot overview”: here we provide basic information about the pilot, such as name for reference, issues considered, and the solution/s that will be tested on these issues;

“2. Pilot description”: here we detail the test-case representing a real work situation;

“3. Actors”: here we describe the actors that are involved in the pilot (i.e. human resources, applications, systems, etc. involved in the real work situations).

3.1 Welding cell pilot (Recreating the FIAT Manufacturing Setup)

1. Pilot overview - General information

<table>
<thead>
<tr>
<th>Owner</th>
<th>Body shop manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot name</td>
<td>Welding cell</td>
</tr>
<tr>
<td>Applicable Test-cases</td>
<td>Initial Configuration of industrial settings/ Remote Maintenance/ Management of repurposing/ changing</td>
</tr>
<tr>
<td>Scenario cluster</td>
<td>Agile / Large Scale Manufacturing / Remote maintenance</td>
</tr>
</tbody>
</table>

Table 3.1 – CRF Pilot: general information

2. Pilot description

THE OBJECTIVES

The main goal is to provide a platform in which flexibility and adaptability of the industrial network can be demonstrated. The purpose is to provide a physical deployment of the IoT@Work architecture, in a laboratory environment, in order to evaluate and validate the overall project achievements. The transferring of these achievements to the design of new production cells and lines will decrease the configuration/management tool programming costs, the number of manual steps and in general the manpower costs. On the side of the implementation of production systems, it will contribute to reduce downtimes and negative correlation between cells/sub-systems/sites.

FOCUS OF THE PILOT

A simplified welding station is replicated at the CRF laboratories. Welding is an operation generating smoke, so the smoke level has to be monitored and the environment needs to be ventilated. Such a station includes a welding robot (and relative controllers/interfaces to the ICT layer), a fan for the ventilation of the cell, smoke and temperature sensors, together with a vibration sensor for

1 As defined in D1.1[7].
implementing a suitable control of the fan and the applicative scenarios described in Section 4
hereafter. An energy monitoring unit can be used to analyse the energy usage of the principal
components within the cell (for instance the PLC itself and the fan unit).
Other devices that are normally used for vision guided welding applications (for instance a 2D range
finder) can be connected to the inner Profinet ring (i.e. the cell sub-network) and used, depending on
the applications to implement on the pilot.
The welding robot of the IoT@Work pilot is a SMARTLASER COMAU unit, equipped with a C4G
control rack and a ND-YAG welding laser source. It is composed by 2 arms, with 7 DOF in total. The
laser fiber and optics are integrated in the robot arms; its operating area and some of the mechanical
features of the SMARTLASE unit are reported in the following figures:

THE CHOICE

The description of this pilot is based on the understanding that the plants for production of goods are
complex systems, operating into contexts and conditions that are much more dynamic and subject to
changes, sometimes frequently and having strong impact on the number and types of the produced
goods. In this context, in order to contribute by means of the specific experience of CRF in the domain
of the automotive design and development, the pilot is referring to a welding cell, similar to the ones
that are present at the Mirafiori plant 2 (for instance the ones of the production line of the Alfa Romeo
“Mito” car) and aligned to the standards of the FIAT group related to the setting-up of new production
lines. The pilot is being located in the laboratories of CRF, where most of the relevant parts of this
welding cell have been installed, as described in the sections below. This choice has been taken
based on the following considerations.

In the FIAT plants, the access to the production areas is severely restricted, among others for safety
reasons to the production and maintenance operators. Consequently, the access to the production
welding cells for extended time periods for R&D purposes, such as the ones foreseen in the
IoT@Work validation plan, is not allowed. We consider here that an existing welding cell has to
continuously operate, in safety conditions, for the exclusive need of fulfilling the production duties; it
thus has a strong need of: maintenance plans, sensors, diagnosis, machinery check and planned or
unplanned substitution of machinery. These issues are all solved with the choice of locating the most
relevant components of the cell and all of the validation equipment in the CRF laboratories, where
many of the cell components and devices (for instance the laser welding robot itself) are already
available

2 The Miafiori plant is located in Turin, Italy.
Furthermore, by locating the IoT@Work pilot at the CRF laboratories, the validation tests and activities can be easily set up and conducted in a comfortable environment, where the involved technicians have no constraints of access, availability of the equipment and limitations of time for their interventions.

SUPPORTED SCENARIOS

The scenarios and validation story boards supported by the CRF pilot are discussed in detail in the Section 4 and in the Table 4.1 of the present deliverable; an excerpt is reported hereafter, in order to ensure a self-contained description of the pilot features.

- **Plug&Work Production Unit Scenario - Insertion of an Energy Monitoring Unit in the production cell**: it can be applied for the validation of the Plug&Work features of a device in the IoT@Work architecture. Data regarding the electrical consumption of the fan unit or the PLC as a whole are acquired, as shown in Figure 3.1, by means of a dedicated Energy Monitoring Unit (EMU – SIEMENS Sentronpac 4200). The Use Case for the validation will consist in the analysis of the Plug&Work behaviour, based on the IoT@Work KPIs, with and without the adoption of the specific network and communication services developed in WP2.

- **Predictive Maintenance Scenario - Dynamically configurable fan control**: dedicated data acquisition loops will be developed and operated as concurrent tasks, feeding the Cell PLC (an IM-151 8F I/O Controller CPU) with data produced by smoke, temperature and vibration sensors. A dedicated ventilating fan control loop (an additional concurrent task, driving the SINAMICS G120 Inverter) will be implemented in order to drive the AC asynchronous motor of the fan with dynamically adjustable parameters. The Use Case for the validation will be the monitoring of the events generated by the concurrent tasks running, based on the IoT@Work KPIs, with and without the adoption of the specific network and communication services developed in the project.

The applicative tasks running in the pilot for its production purposes consist of typical Laser Welding and robot controlling operations. The CRF pilot includes the principal systems and devices of a real laser welding industrial line, by means of which the benefits of the IoT@Work approach will be assessed in a realistic manner and the overall WP4 validation activities will be carried out. In the pilot a sensor data acquisition and processing system is also present: a 2D range finder, conveying a large amount of information on the Industrial Ethernet (Profinet) network will be used to acquire precise measures from the welding field. The stream of measures, time-stamped, will be transferred to a data processing PC which is external to the cell. The monitoring of a regular behaviour of these applications and systems will permit to verify that the validation scenarios and study cases do not interfere with the regular activities of the production cell.
Figure 3.1. CRF Pilot architecture

In Figure 3.1 the motor fan can be:
- activated by a certain time profile associated with the production rate.
- supplied at a reduced level for normal operations
- activated at full speed when the smoke level in the workstation is over the threshold

The fan’s consumption will be monitored and a predictive failure alarm can be given in case of increase of current consumption and temperature of the motor.

Predictive alarms can be faced through direct intervention or addition of fine level sensors (vibration signature detection). Added sensors need to be Plug&Work in the Profinet IoT system.

The fan can be remotely deactivated if the temperature of the motor is too high.

Regarding EMUs:
- one EMU, the Sentronpac 4200, is in the cell for energy monitoring of the major consuming systems (the PLC as a whole or the motor fan alone);
- when the EMU is inserted and plugged it performs network access control and then acquires a net address;
- the EMU collects energy data from the PLC or the motor fan and sends them to the local CPU and the server PC database;
- data from energy consumption are used for simple data logging and preventive alarm generation (shift of performances).
Several sensors will operate in the cell:
- the 2D range finder will carry out 3D positioning and/or quality management tasks in order to generate tuneable data loads of Ethernet packets on the network;
- the vibration monitoring will send data (buffered or in stream) to the local CPU and the server PC will perform signature analysis;
- information from all sensors will be available on the Profinet for different users (using different access rights) and will need to be Plug&Work, in the sense that they can be used only for a limited time and need to be inserted in a new location according to a Plug and Play and Walk Away strategy.

The System Monitoring will consist of:
- an additional server PC with SW to visualize the system state;
- monitor devices, events from network access control, eventually resource usage.

The Network characteristics are:
- Profinet protocol compatibility;
- small ring to show resilience and QoS concepts;
- should additionally have a policy control, a network access control and a resource management;
- shall ensure that the QoS requirements of the sensors are met and adjusted even when the data stream of the thermal camera competes on some link regarding bandwidth.

Configuration can be adjusted depending on use cases’ needs.

### Table 3.2 – CRF Pilot: detailed information

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant manager</td>
<td>Responsible of the plant. He/She receives production data and warnings through SCADA system</td>
</tr>
<tr>
<td>Body shop area manager</td>
<td>Responsible of the plant. He/She receives production data and warnings through SCADA system and direct intervention requests</td>
</tr>
<tr>
<td>Maintenance operator</td>
<td>Operator working on the line for maintenance purposes</td>
</tr>
<tr>
<td>Maintenance area responsible</td>
<td>Responsible for the maintenance of machinery in the local area</td>
</tr>
<tr>
<td>Remote maintenance operator</td>
<td>Responsible for maintenance of machinery through a remote connection</td>
</tr>
</tbody>
</table>

### Table 3.3 – CRF Pilot: actors

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
</table>
3.2 Hybrid manufacturing process pilot (inIT)

1. Pilot overview - General information

<table>
<thead>
<tr>
<th>Owner</th>
<th>Plant manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot name</td>
<td>Hybrid manufacturing process</td>
</tr>
<tr>
<td>Applicable Test-cases</td>
<td>Initial Configuration of industrial settings/ Remote Maintenance/ Management of repurposing/ changing</td>
</tr>
<tr>
<td>Scenario cluster</td>
<td>Agile/ Large Scale Manufacturing / Remote maintenance</td>
</tr>
</tbody>
</table>

Table 3.4 – inIT Pilot: general information

2. Pilot description

THE CHOICE

The rationale for the second pilot is based on the fact that the factories of the future should be able to produce a large variety of sophisticated products and, at the same time, offer a high flexibility and short cycle times. Such an approach should ensure a reliable, energy- and cost-efficient production, as well as setting up the facility with reduced costs and time by means of deploying the IoT@Work architecture.

THE PROBLEM

The Lemgo Model Factory (LMF) is a hybrid manufacturing process, i.e., it is composed of process automation and discrete manufacturing elements, in a small scale. It is used to test new automation technologies in a real industrial manufacturing process. The last production module can be added or removed during run-time, leading to an adaptable system. However, the whole procedure is currently pre-configured and used for demonstration purposes only.

THE OBJECTIVES

In order to address the abovementioned problem, the main goal of this pilot is to provide a platform in which flexibility and adaptability by means of using the IoT@Work architecture can be demonstrated. The main objectives are to decrease the configuration/management tool programming costs, the number of manual steps and in general the manpower costs, and on the side of the production system to avoid downtimes and negative correlation between cells/sub-systems/sites.

TEST-CASE IN FOCUS:

The LMF is built in a modular structure. Currently, it consists of 6 modules which fulfil different tasks within the whole production process. The first module is used for delivery of raw material (corn) and as a storage cell for bulk goods. From the first cell, the raw material is transported into another storage container, where a conveyor belt is used to transport the corn further. These components form the second production module. The third module includes a proportioning device based on a scale that can portion the amount of corn very precisely by using a feeding screw. Afterwards the raw material is filled into small bottles in module 4 and then transported via another conveyor belt into module 5 which consists of a pick and place robot, which empties the bottles and fills the corn into a tube that transports the corn by compressed air to module 6. In the last module, the raw material is processed. This uses hot air to heat up the corn until it becomes popcorn. Moreover, the product (popcorn) is filled either into a large storage container or into cups.

3 As defined in D1.1[7].
Figure 3.2. inIT pilot architecture

The LMF can already be adapted with respect to different needs. The last production module which is responsible for popcorn production and filling of the product is attached in a flexible way. This means that the module can be connected to and disconnected from the remaining plant. If it is connected, the previous module empties the bottles that contain the corn. This raw material is then handed over to the production module. In case of a disconnected production module, the full bottles arriving at the pick and place robot will be sent to a storage area until they are processed.

However, the current solution was already implemented and foreseen during design time, i.e. different states of the plant were considered during development. In the LMF states including and excluding both manufacturing cells are planned. Therefore, the PLC that controls the LMF is able to distinguish between both states by detecting an IO signal that is set if the plug for the fieldbus of both flexible modules is connected. In this case the control software recognizes the changed plant structure and adjusts the program flow to a new flow. Based on this concept, the flexible production is achieved for demonstration purposes.

The flexible module will be used to show the advancements related to the IoT@Work project results. That is, rather than to achieve modularity with pre-defined states recognized and pre-supported in each of the modules, we want to achieve modularity by plugging in something that wasn't foreseen before. For this, the module can be equipped with different development platforms which are used in the project. Furthermore, changes in other modules of the LMF can be applied if necessary.

Table 3.5 – inIT Pilot: detailed information
### 3. Actors of test-case (Roles, Systems, Resources)

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant manager</td>
<td>Responsible for the plant; receives production data and warnings through SCADA system</td>
</tr>
<tr>
<td>Manufacturing engineer</td>
<td>Responsible for planning, designing, setting up, modifying and optimising a manufacturing process</td>
</tr>
<tr>
<td>Maintenance engineer</td>
<td>Responsible for managing plant maintenance and resources in order to realize the plant and production as efficient as possible</td>
</tr>
<tr>
<td>Remote maintenance operator</td>
<td>Responsible for maintenance of the plant from a remote location</td>
</tr>
</tbody>
</table>

*Table 3.6 – inIT Pilot: actors*
4 IoT@Work scenarios

There are two main aspects of evaluation and demonstration in the IoT@Work project. The first one relates to providing a general proof of concept for the whole project approach and attempts to evaluate the significance of and implication of an IoT-centered architecture on a whole range of scenarios within the manufacturing environment. The second part of the evaluation concentrates on how specific developed technologies compare to the state of the art. This section is part of the first evaluation aspects, which is scenario and test-case driven.

In this deliverable, we use the same scenarios developed in D4.1 [6] for further developing test-cases and showcases as part of our proof of concepts and implementation work. The feasibility and demonstration effect of the scenarios and the ease of integration as part of the two pilots are used to further select a subset of scenarios for further discussion.

This section also explains the relationship between the selected scenarios and some of key technologies developed and implemented within this project. The first results of this activity are explained around one of the main scenarios that has been achieved so far. The remaining integration and evaluation work will be reported in later deliverables.

4.1 Reference Application and Scenarios

The first evolutionary step towards a shared agile factory could be demonstrated by enabling access to today’s external stakeholders in order to interact with an IoT-enabled manufacturing system. These stakeholders could include the suppliers of the production tools (e.g. machines, robots), as well as the production logistics (e.g. material flow, supply chain management), and maintenance and re-tooling actors.

The IoT@Work project proposes an IoT-based architecture that challenges the hierarchical and closed factory automation pyramid, by allowing the aforementioned stakeholders to run their services in a multi-tenant, flat production system. This means that the services and applications of tomorrow do not need to be defined in an intertwined and strictly-linked manner to the physical system, but rather run as services in a shared physical world. The room for innovation in the application space could be increased in the same degree of magnitude as this has been the case for embedded applications or Apps, which have exploded since the arrival of smart phones (i.e. the provision of a clear and well standardized interface to the embedded hardware of a mobile phone to be accessed by all types of Apps).

One key enabler to this ICT-driven smart and agile manufacturing lies in the way we manage and access the physical world, where the sensors, the actuators, and also the production unit should be accessed, and managed in the same or at least similar IoT standard interfaces and technologies. These devices are then providing their services in a well structured manner, and can be managed and orchestrated for a multitude of applications running in parallel.

4.1.1 Building a Multi-tenant Sensor-enabled Smart Factory

The reference application targeted in the IoT@Work project is an example of how IoT-enabled manufacturing infrastructure could provide context-aware and reliable data reflecting the status of an existing manufacturing system to multiple stakeholders. The targeted scenario would include the retrofitting and extension of an existing FIAT manufacturing assembly-line to become smarter and more capable of tracking its assets and condition. A set of condition monitoring sensors such as temperature, vibration, energy, and smoke sensors are attached to the line in order
to sense the environmental condition of the production line. These sensors are engineered around the production system and are offered as a platform for multiple stakeholders to run their condition monitoring applications.

The shop floor of such a factory that is owned today by FIAT is pre-engineered and designed and never changed. The production units are provided and integrated by third party suppliers such as COMAU or Siemens, who are also subcontracted for maintenance tasks of their supplied lines. The majority of the control software and applications interacting with the production are defined at design phase in a closely and intertwined manner with the physical production system. The same applies to the factory owner FIAT that also owns the data and who is able to collect it through its private network and SCADA system. The reference scenario allows FIAT to entrust the task to a third party, who is allowed to extend the existing factory to include a set of sensors that facilitate better condition monitoring and environmental applications to be developed around the factory.

Both FIAT and the line suppliers are treated as consumers of a new range of IT services that enable each stakeholder to access in run-time his/her respective data directly from the newly deployed sensor infrastructure around the field level. The different stakeholders, including FIAT, are able to run a whole new range of smart manufacturing applications that can interact with the manufacturing system not using an offline process but doing so at runtime. Examples would include:

1. Run-time access to material consumption by logistics chain.
2. Access by machine suppliers to condition monitoring events related to their respective machines.
3. Linking sensor data for condition monitoring (such as vibration sensors, thermal cameras, and on-machine sensors) to back-end condition-monitoring simulators.
4. Access to energy consumption data and environmental sensors to control air conditioning, smoke extraction.

The IoT@Work project aims at providing the ICT enablers offering the above applications and the stakeholders access to the manufacturing set-up, while making sure that they do not interfere with the existing production system or with each other. We provide in Table 4.1 a list of the scenarios and test-cases, considered within the scope of IoT@Work, and their link to the IoT@Work enablers.

### 4.1.2 Technology Integration and Pilot Development

The developed scenarios in D4.1 [6] describe hypothetical system models of what IoT-enabled manufacturing would look like. These system models have been used in an iterative way at several stages of the project, to first acquire the functional and non-functional requirements that have driven the IoT@Work technologies and solutions. The same scenarios are used and deepened to show the possible impact of the project approach through key performance indicators, as this is shown in D4.1. The same scenarios in deliverable D4.1 have also helped shape the pilots and the experimental setup that is built during the lifetime of the project.

The scenarios originating from D4.1 are summarized in Table 4.1, along with the results of the scenario analysis.
### Table 4.1 – Analysis of the Scenarios from D4.1

<table>
<thead>
<tr>
<th>Scenario family (D4.1 “test-case”)</th>
<th>Scenarios (D4.1 “use case”)</th>
<th>Validation storyboards</th>
<th>Affected Pilots</th>
<th>Current Status and Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial configuration of industrial settings</td>
<td>Configuration of a new line destined for a new car Model</td>
<td>Plug&amp;Work of production unit for fast re-configuration of production lines</td>
<td>Pilot 1</td>
<td>This scenario is no longer considered during validation.</td>
</tr>
<tr>
<td></td>
<td>Configuring of a modular line destined for a batch production process</td>
<td></td>
<td>Pilot 2</td>
<td>This scenario is no longer considered during validation.</td>
</tr>
<tr>
<td></td>
<td>Configuration of industrial networks for supporting fast changing logistics and supply-chain and production interfaces</td>
<td></td>
<td>Pilot 1 &amp; 2</td>
<td>This scenario now covers the reference application scenarios and is further elaborated for validation.</td>
</tr>
<tr>
<td>Remote maintenance and system monitoring</td>
<td>Remote monitoring of OEM equipment</td>
<td>OEM Maintenance worker performing targeted energy monitoring using mobile device</td>
<td>Pilot 1</td>
<td>This scenario is particularly considered for further validation (see section 4.3).</td>
</tr>
<tr>
<td></td>
<td>Rolling out device/ configuration update</td>
<td>Dynamically configurable smoke detection and fan control</td>
<td>Pilot 1</td>
<td>We particularly consider at least one application-level scenario for further validation (see section 4.2).</td>
</tr>
<tr>
<td></td>
<td>Predictive Maintenance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Continuous Device Integrity Monitoring</td>
<td>Storyboard about detecting malicious changes to device firmware or software.</td>
<td>Pilot 2</td>
<td>This scenario will be further elaborated for validation.</td>
</tr>
<tr>
<td>Management of repurposing / changing initial configuration of industrial settings</td>
<td>Insertion of an Energy Monitoring Unit in the production cell (Plug&amp;Work Production Unit)</td>
<td>Bootstrap energy monitoring devices for plant-wide energy usage assessment</td>
<td>Pilot 1 &amp; 2</td>
<td>This scenario now covers the reference application scenarios and is further elaborated for validation (in section 4.4).</td>
</tr>
<tr>
<td></td>
<td>Plug&amp;Work re-allocation of a vibration sensor from a given cell to a different one</td>
<td></td>
<td>Pilot 2</td>
<td>This scenario is no longer considered during validation.</td>
</tr>
<tr>
<td></td>
<td>PLC firmware update</td>
<td></td>
<td>Pilot 1 &amp; 2</td>
<td>This scenario is no longer considered during validation.</td>
</tr>
</tbody>
</table>

#### 4.2 Smoke detection and fan control scenario

For the CRF pilot implementation, the following describes an arrangement of the demo site devoted to demonstrating how some IoT@Work functionalities can be used to automatically monitor and control a fan in a welding cell, so as to assure the...
smoke level and temperature in the working environment are maintained within their respective optimal ranges and, as a side effect, reduce power consumption and inoperative time.

The monitoring and controlling applications run outside the production cell. Figure 4.1 depicts the cell layout.
sensing infrastructure run as an early prototype of an IoT-based platform for smart manufacturing applications

Figure 4.2 – An overview of the proposed Event Notification Service approach

Two network segments exist at the cell-level:

- the 1st one, based on Ethernet point to point connections, connects the COMAU C4G welding robot controller and the 3D Laser Sensor to the SCALANCE X208 Profinet Switch

- the 2nd one, consisting in a Profinet ring that connects the cell PLC (the CPU 317-2) and the I/O controller (the IM151-8F CPU) to the different sensors (smoke, temperature, vibration sensors and the SENTRONPAC 4200 EMU power management unit) and actuators (the SINAMICS G120 unit, driving the fan motor) that are used in the Pilot.

The third network segment above the cell (i.e. the field devices, and other periphery sub-systems such as the welding gun) connects the cell with the back-end servers which host some of the higher-level applications and IoT@Work components such as the directory service, the network controller, or the ENS middleware brokers.
As an entry point from the back-end servers into the cell, a gateway or proxy agent (**Cell Proxy** in the following) between the cell and the monitoring and controlling applications will be implemented in the cell PLC (the CPU 317-2).

The remaining legacy controller units offer access to the field-level data, and also host Profinet compliant control logic that interacts with the legacy I/O devices. This solution offers an easy way to integrate IoT-enabled applications into an existing system. The data is provided to back-end applications through the same technology used to provide field-level data to legacy SCADA system, i.e. through OPC servers. The IoT-enabled applications could also interact with the field devices through the same proxy.

In order to demonstrate the capabilities of a native IoT@Work device, we use an experimental hardware board that extends a traditional I/O device to include some functions developed within the project. The extended device would then replace a legacy I/O device within the cell. This is currently being examined as an option for the pilot integration as well. The cell’s PLC (the CPU 317-2) will manage the environment sensors and actuators and provide the **Cell Proxy** with the data required for event generation, and receive from this machine the commands for the actuators. The **Cell Proxy** will manage the forwarding of events to the IoT@Work Event Notification Service (ENS), receive the commands from the monitoring and controlling applications and act accordingly on the cell’s actuators.

The legacy field devices are connected through a Profinet network that is shown in the figure as a **Scalance X208** switch. We also intend to replace the Industrial Ethernet ring network between the field level and control level by our own emulated industrial network, which should demonstrate the network slicing mechanisms that allow to allocate different IoT-applications and multiple users to an own slice.

The proposed arrangement makes it possible to decouple the cell’s internal operation from the external one, minimizing the developments and impacts on the cell devices, while at the same time assuring the flexibility of the IoT@Work solutions, as well as allowing us to test these solutions in a real context.

The **Cell Proxy** will take care on interfacing the ENS, publishing one or more event streams (e.g., a stream that periodically reports the smoke sensor measures, a stream that regularly reports temperature sensor measures, etc.) and receive the commands to rearrange fan status and operation.

As evident from the above description there are **two channels** connecting the cell to the monitoring and controlling environment:

the **1st channel** (the **Upstream Channel**) is from the cell to the outside and reports measures (or other kinds of data) arranged in one or more streams of events. This channel assures that the **outside world** is kept informed of what is happening inside the cell; the **2nd channel** (the **Downstream Channel**) is from the **outside world** to the cell and is used to convey command for **actions** to be performed within the cell.

In the next section we provide more details on the overall arrangement.

---

4 For example using a CEP Engine for the implementation of the monitoring and control application makes possible to easily modify one or more of these applications or to use more applications, even in parallel, to analyse and monitor different issues. The development of such kind of applications using a CEP Engine can, sometimes, simply become the writing of a pattern search specification.
4.2.1 Scenario overview

Figure 4.3 provides a graphical overview of the overall deployment scenario. In the lower part of the figure the cell layout of Figure 4.1 is reported, while the upper half details the ENS and monitoring deployment.

![Diagram](image)

**Figure 4.3 – Integrating Event Notification System in the Pilot**

On the ENS side, the system envisages a set of components, as follows:

- The **ENS Authorization, Policy Decision Point (PDP)** and **Revocation** services that are used to check that applications or devices trying to establish connections to the ENS are actually authorized to do so;
- The **ENS Access Request Broker Service** that is used to convey access request to the **ENS Authorization** service. This is an AMQP broker\(^5\) that offloads the management of access requests from the operation broker;
- The **ENS Broker Service** is the operation AMQP broker service\(^6\) that actually manages the event streams and their dispatching to the interested applications;
- The **CEP Engine**\(^7\) server to host the monitoring and control applications.

All these components are connected to the plant Ethernet through which they are able to communicate among them or, indirectly, with the cell sensors and actuators.

---

\(^5\) It is actually a RabbitMQ server.

\(^6\) Made up of one or more clustered RabbitMQ servers.

\(^7\) Here it is suggested to use a CEP Engine for the implementation of the monitoring and control applications, because such kind of engines can reduce the development effort, and can significantly increase the flexibility and adaptability of the solutions.
via the *Cell Proxy*. The use of a separate network slice to support the ENS data traffic is also possible. Since we do not consider any time-critical ENS data, we assume that the whole ENS system including data sources and consumers are all edges within one network slice. The latter can provide best-effort data connectivity, while insuring that higher priority is guaranteed for process-relevant data, such as the Profinet traffic between the smoke sensors, the fan, and the controller.

The listed components run preferably on machines equipped with a Windows server OS (e.g.: Windows Server 2003 or Windows Server 2008).

The ENS components listed above are Java applications or, in the case of the RabbitMQ AMQP brokers, Erlang ones.

### 4.2.2 Technical details

The communication between the *Cell Proxy* and the ENS middleware is managed using a Java library that takes care of all technicalities related to the proxy authentication, ENS session establishment and management, and events publishing.

The ENS Java library can be used as a pure Java one, or as an OSGi bundle.

On the *Cell Proxy*, therefore, there is the need\(^8\) to develop the *inside-to-outside* gateway/proxy application\(^9\) that, on the one hand, acquires data from the cell's PLC and, on the other hand, publishes them as events in one or more event streams, using the aforementioned ENS Java library.

The processing power and RAM of the *Cell Proxy* machine heavily depends on the amount and frequency of events to be managed. If the events are published at a pace of a few events/sec, then a PC with a configuration like the one used for office activities is sufficient (e.g., 2-4 GB RAM and a CPU like an Intel i5 if using Microsoft Windows 7).

With reference to the services depicted in the upper half of Figure 4.3, the machines required to run these services have to meet the following requirements:

- Possibly a Windows server OS for all ENS related services (these being Java or Erlang applications they can also run on Unix-like OSes);
- The *ENS Authorization, Policy Decision Point and Revocation* and *ENS Access Request Broker Service* machines do not require high performance configuration. Typically they can have a 4 GB RAM, a 32 or 64 bit OS, 300 GB disk space;
- The *ENS Broker Service* machine(s) has/have to be dimensioned according to the envisaged workload, being the machine with the highest workload. For

---

8 A specific application will be developed acting as a gateway on the Downstream Channel; but this is independent from the ENS functionalities and can therefore be designed and developed as needed.

9 In case the events have to be arranged in different ENS streams (e.g. a 1\(^{st}\) stream devoted to publish smoke sensor events, a 2\(^{nd}\) one devoted to the temperature measures, a 3\(^{rd}\) one to the vibration measures, etc.) it is suggested to have different, even if conceptually similar, applications on the Cell Proxy each one devoted to handle a specific event stream; so that it will be possible to fine-tune both these applications and the management and provision of events on the ENS and monitoring applications side.
this/these machine it is suggested to have more RAM (i.e., 6-8 GB), faster CPUs, disk and network controllers\textsuperscript{10} and a 64 bit OS;

- The CEP Engine machine dimensioning depends on the kind of processing the monitoring and control applications need to do. The machine, therefore, can be like the one hosting the ENS Access Request Broker Service.

For the networking technologies the plant Ethernet can even be a 100Mbps solution. As for the machine dimensioning, the network bandwidth requirement is tied to the amount and nature of data to be transferred. Therefore, if a high-resolution video stream has to be supported, then the bandwidth requirement could be higher.

There is no specific real-time or bandwidth reservation requirement on the plant Ethernet infrastructure.

4.3 Remote OEM Monitoring Scenario

The first integrated demonstration that took place so far has shown two major functionalities: the network virtualization (the "slice" system as defined in D2.3 [5]) and the secure middleware messaging approach (Event Notification System, ENS). By those components we aim at visualizing Plug&Work properties of the system.

\textbf{Figure 4.4 – Integrating Event Notification System and Slice Management}

\textsuperscript{10} This is especially true if the camera in the cell has to send high definition images via an ENS event stream.
The basic scenario is defined as follows:

- A wireless mobile terminal joins the network and attempts to look up the events required by the manufacturing monitoring application. This will fail because the device is not yet assigned to a slice.
- The mobile device is authenticated as an eligible ENS client, allowed to access certain events from the factory floor.
- The Network detects connection and triggers the device configuration. IoT@Work Network Access Control (authorization and authentication of the device) is not shown to simplify the set up.
- The monitoring application can access the data and creates a graphical representation of the collected statistics.
- The slice is re-routed, once a network resource needs to be allocated to an application of higher importance.
- Optional steps include the QoS configuration of a slice, i.e. to limit available traffic of ENS or to guard ENS from bandwidth consumed by another slice.

The ENS slice is assumed to be already existing, and the ENS system already running. Therefore, the slice only has to be extended, in order to allow access by the mobile device to a slice, and to enforce this in the network. The mobile client is now connected to the ENS system, and can access the needed events. Within the demo, we use a configuration console to visually track the slice management system. This allows us also to understand the parameters and possibly to add some QoS characteristics to existing slices. The configuration of the network nodes is also shown in the console, replacing the automatic process that should occur between the application orchestration and the slice management in the architecture of the slice system.

A network admin can create a slice for the energy assessment system. Similar to the video surveillance system, this slice must include the energy monitoring devices and the pub/sub where the readings will be sent to. Perhaps similar to the remote service management case, the (physical) network addresses of the monitoring devices are not known in advance. When setting up the slice, the network admin knows the amount of monitoring devices, and can also assign abstract ID’s to them. When the monitoring device (or the monitoring agent on the device) registers for accessing the
slice, the physical network address (that was automatically assigned when the device was attached to the network) should somehow have been assigned with an approved abstract ID, and can then be authorized for accessing the slice.

4.4 Plug&Work of sensor/production units (inIT)

The aim of this scenario is to enable a flexible and adaptable production process in which a device can be attached to an industrial network and start functioning without manual interventions to allow Plug&Work capability. However, this will involve adaptations to industrial devices at multiple levels. An IoT@Work production unit could be an individual application, specific sensor enabled I/O device, or it could represent a sub-process module offering a set of services towards a modular production process. If such a unit is added to an existing production, it has to be engineered to become part of the system. A modular production process consists of several individual plant modules. Each module is equipped with a separate programmable logic controller (PLC) being programmed with its corresponding control software. The software needs specific information from other field devices (e.g. PLCs, robots or drives) depending on their functionality. This data is commonly referred to as process data and has to be exchanged in real-time to meet the requirements of the process. This scenario consists of field devices offering functionalities through services and exchanging their process data in real-time over Profinet. For example a temperature sensor can offer the service GetTemperature. There may be two ways to exchange temperature values, either by using a Profinet-based expedited real-time channel (that has to be configured first), or by a standard non real-time TCP channel by means of a vendor neutral transport protocol like OPC UA.

Profinet in its current state needs some manual configuration effort before the attached devices can start their operations. The network configuration must be supplied manually to an engineering tool, which configures all devices of the RTE. In addition, the engineering tool defines which data must be transferred between the distributed devices. However, in a Plug&Work environment, a device must discover the functionalities of other devices by itself and negotiate the data exchanges with other devices. For devices of multiple vendors it is necessary to offer standardized methods for a consistent discovery and the exchange of configuration information. These requirements can be fulfilled by a Service-Oriented Architecture (SOA), which is a major focus of interest within the factory automation industry, from the device level to the high-level IT. Devices in an SOA offer their functionalities/information as services that can be used by other devices. Currently there are two industrially standardized technologies to support the deployment of SOA at device level, namely the Device Profile for Web Services (DPWS) and OPC Unified Architecture (OPC UA). In this scenario we will focus on OPC UA to be adopted for auto-configuration of Profinet for process data communication.

4.4.1 Auto-configuration Service

Profinet needs at least one controller which configures all connected devices. In an auto-configuration scenario the controller must gather all necessary information by itself without manual interventions. To achieve this in phase 1 a device is newly connected to the network. To ensure IP connectivity it must get an IP address. This could be done by a DHCP server or Profinet specific service. In phase 2 the discovery process takes place, OPC UA describes a specific mechanism to enable discovery. In this phase the controller is informed that a new device is connected. The controller also needs Profinet specific information from the device for the auto-configuration phase. This information is described in a device description file (GSD) which belongs to each device. During the discovery the contents of the GSD file must
be made available to the RTE configuration service. An IoT@Work production unit has an OPC UA server offering a basic set of services, including device description. A controller includes an inbuilt OPC UA client, which uses a standard TCP channel to communicate with the configuration service running on the device and learn a configuration URL. This URL could refer to the device vendor’s database, hosted on some web-server, from where the controller can download the GSD file, thus completing phase 3. In phase 4, the RTE configuration service parses the received file. Afterwards it configures the connected devices and process control applications accordingly. The RTE configuration service takes over the tasks which an engineering tool would have in a traditional solution.

4.4.2 Scenario overview

The local PLC will communicate with a central control program, which is coordinating the production units of the whole modular production plant and runs the main control logic. The PC based host environment of a local PLC could be integrated with other IoT@Work technologies, such as feeding ENS with data from one or two sensors (events) or communicating QoS requirements with a slice enforcement point instance for assigning application traffic to the correct slice. Furthermore an additional OPC UA server could also be integrated with the local PLC to provide aggregated services as a single production unit, to the external systems that may be interested in process data for relatively less time critical applications (e.g. monitoring).

![Figure 4.6 – Scenario at the Lemgo Model Factory](image-url)
4.4.3 Scenario details

The basic scenario is:

- A production module joins the network, it has a dependency on the workplace environment to offer it functionality (e.g. right temperature and humidity level).
- A sub module “Environment Monitoring” offers functionality to provide such information. It is plugged into the production module. The sub module is composed of IO devices with various sensors.
- A PC based local PLC detects the newly integrated sub module and auto-configures it. The local PLC is responsible for the control logic of the production module and its sub modules.
- All the legacy and IoT@Work enabled devices in the production module are connected with the local PLC through an industrial switch. The switch may be replaced with the emulated slice management cloud.
- After configuring the sub module, the local PLC will be able to learn the environment data by two means, first cyclically using Profinet, secondly optional ad-hoc using OPC UA.
- The local PLC representing the production module offers flags stating ready/online/offline states. There may be two options to offer these flags to the central control program, firstly by using legacy OPC to set some variables of central PLC. Secondly, the whole production module is offered as a Profinet device to the system, and these flags are exposed as IO data. So that the central control program will be informed of inclusion of new production unit.
- The local PLC may also expose environment data as an OPC UA service to the external system, so that an external monitoring application equipped with OPC UA client, connected with the network infrastructure can communicate directly to the production module or even to an individual IoT@Work sensor node.
5 Conclusions

This deliverable has presented the results of Task 4.2 – “Pilots development”. The purpose of the pilots is to an industrial setup for both development and testing as well as show-casing purposes. The reference industrial scenarios and applications used to demonstrate the results of the project can be described as the ability to support multi-tenancy in an IoT-enabled manufacturing environment. The production system is ideally accessed simultaneously by a multitude of applications that are not necessary belonging to the owner of the manufacturing setup alone, but also multiple players acting around the production life-cycle, such as the machine suppliers, the line integrators for maintenance reasons, or production processes and logistical supply chain.

The demonstrated scenarios include examples of such novel IoT-applications that can be deployed and bootstrapped in a controlled manner and during run-time of the production. The IoT-enabled system allows new applications to better interact with the data needed from the shop-floor, and to allocate access to physical resources in a coordinated manner.

The implementations of the IoT@Work technologies are progressively integrated around each use-case or application scenario in a real manufacturing set-up. For this purpose, a pilot area has been prepared in a segregated manufacturing validation location in the laboratories of the Production Facilities department of Fiat Research Center. A similar facility but for testing and integration purposes is also made available in the Lemgo Model Factory laboratories. These two areas include the materials, the specific facilities and equipment necessary to accomplish the goals of the individual integration scenarios.

Concerning the validation planning and methods, the framework for the economic, social and technical validation of the selected pilot has been described in the deliverable D4.1 [6]. In the present document the KPI details and impact on each scenario is revisited. The impact of the IoT@Work technologies is measured in each use-case through the evaluating the affected KPIs quantitatively and qualitatively in terms of “AS-IS” and “TO-BE”. The elaboration of KPI metrics per use-case is currently in progress and will be integrated in a later version of this document. Furthermore, a consolidated report of the pilots has been provided, together with the guidelines for applying the validation procedures to the use cases of the project.

The implementation phase of the validation activities has been addressed by reporting the details of the IoT@Work scenarios, including the development and test platforms that have resulted from Implementation efforts from the partners. Details are also given on the Experiments and Technology Tests: in particular with the deepening on the Slice Management in IoT@Work, together with the related Use Cases.

11 This decision has now been to move this results to D4.3.
6 References


