Project Acronym: SatisFactory
Project Full Title: A collaborative and augmented-enabled ecosystem for increasing satisfaction and working experience in smart factory environments
Grant Agreement: 636302
Project Duration: 36 months (01/01/2015 - 31/12/2017)

DELIVERABLE D2.1.2
SatisFactory System Architecture

Deliverable Status: Final
File Name: SatisFactory-D2.1.1-v2.0-ISMB-SatisFactory System Architecture.pdf
Due Date: April 2016 (M16)
Submission Date: April 2016 (M16)
Task Leader: ISMB

<table>
<thead>
<tr>
<th>Dissemination level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
</tr>
<tr>
<td>Confidential, only for members of the Consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°636302
### The SatisFactory project consortium is composed of:

<table>
<thead>
<tr>
<th>Consortium</th>
<th>Description</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CERTH⁴</td>
<td>Centre for Research and Technology Hellas</td>
<td>Greece</td>
</tr>
<tr>
<td>SIGMA</td>
<td>Sigma Orionis SA</td>
<td>France</td>
</tr>
<tr>
<td>FIT</td>
<td>Fraunhofer-Gesellschaft zur Foerderung der Angewandten Forschung E.V</td>
<td>Germany</td>
</tr>
<tr>
<td>COMAU</td>
<td>Comau SPA</td>
<td>Italy</td>
</tr>
<tr>
<td>EPFL</td>
<td>École Polytechnique Fédérale de Lausanne</td>
<td>Switzerland</td>
</tr>
<tr>
<td>ISMB</td>
<td>Istituto Superiore Mario Boella sulle tecnologie dell’informazione e delle telecomunicazioni</td>
<td>Italy</td>
</tr>
<tr>
<td>ABE</td>
<td>Atlantis Engineering AE</td>
<td>Greece</td>
</tr>
<tr>
<td>REGOLA</td>
<td>Regola srl</td>
<td>Italy</td>
</tr>
<tr>
<td>GlassUP</td>
<td>GlassUp srl</td>
<td>Italy</td>
</tr>
</tbody>
</table>

**Disclaimer**

This document reflects only the author’s views and the European Union is not liable for any use that may be made of the information contained therein.

---

⁴ Project Coordinator
## Authors List

### Leading Author (Editor)

<table>
<thead>
<tr>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sottile</td>
<td>Francesco</td>
<td>ISMB</td>
<td><a href="mailto:sottile@ismb.it">sottile@ismb.it</a></td>
</tr>
</tbody>
</table>

### Co-authors (in alphabetic order)

<table>
<thead>
<tr>
<th>#</th>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Arena</td>
<td>Damiano Nunzio</td>
<td>EPFL</td>
<td><a href="mailto:damiano.arena@epfl.ch">damiano.arena@epfl.ch</a></td>
</tr>
<tr>
<td>2</td>
<td>Brizzi</td>
<td>Paolo</td>
<td>ISMB</td>
<td><a href="mailto:brizzi@ismb.it">brizzi@ismb.it</a></td>
</tr>
<tr>
<td>3</td>
<td>Cultrona</td>
<td>Pietro</td>
<td>COMAU</td>
<td><a href="mailto:pietro.cultrona@comau.com">pietro.cultrona@comau.com</a></td>
</tr>
<tr>
<td>4</td>
<td>Jentsch</td>
<td>Marc</td>
<td>FIT</td>
<td><a href="mailto:marc.jentsch@fit.fraunhofer.de">marc.jentsch@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>5</td>
<td>Kanlis</td>
<td>Alexandros</td>
<td>CERTH</td>
<td><a href="mailto:alexkanlis@iti.gr">alexkanlis@iti.gr</a></td>
</tr>
<tr>
<td>6</td>
<td>Kasinathan</td>
<td>Prabhakaran</td>
<td>ISMB</td>
<td><a href="mailto:kasinathan@ismb.it">kasinathan@ismb.it</a></td>
</tr>
<tr>
<td>7</td>
<td>Khaleel</td>
<td>Hussein</td>
<td>ISMB</td>
<td><a href="mailto:khaleel@ismb.it">khaleel@ismb.it</a></td>
</tr>
<tr>
<td>8</td>
<td>Mautino</td>
<td>Sara</td>
<td>GlassUP</td>
<td><a href="mailto:sara@glassup.net">sara@glassup.net</a></td>
</tr>
<tr>
<td>9</td>
<td>Mochament</td>
<td>Konstantinos</td>
<td>CERTH</td>
<td><a href="mailto:k.mochament@iti.gr">k.mochament@iti.gr</a></td>
</tr>
<tr>
<td>10</td>
<td>Papadopoulos</td>
<td>Nikos</td>
<td>ABE</td>
<td><a href="mailto:n.papadopoulos@abe.gr">n.papadopoulos@abe.gr</a></td>
</tr>
<tr>
<td>11</td>
<td>Perdikakis</td>
<td>Apostolos</td>
<td>EPFL</td>
<td><a href="mailto:apostolos.perdikakis@epfl.ch">apostolos.perdikakis@epfl.ch</a></td>
</tr>
<tr>
<td>12</td>
<td>Rasheed</td>
<td>Hassan</td>
<td>FIT</td>
<td><a href="mailto:hassan.rasheed@fit.fraunhofer.de">hassan.rasheed@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>13</td>
<td>Serra</td>
<td>Antonio</td>
<td>Regola</td>
<td><a href="mailto:a.serra@regola.it">a.serra@regola.it</a></td>
</tr>
<tr>
<td>14</td>
<td>Tovar</td>
<td>Orlando</td>
<td>ISMB</td>
<td><a href="mailto:tovarordonez@ismb.it">tovarordonez@ismb.it</a></td>
</tr>
<tr>
<td>15</td>
<td>Tregnaghi</td>
<td>Gianluigi</td>
<td>GlassUP</td>
<td><a href="mailto:giovanni.tregnaghi@glassup.net">giovanni.tregnaghi@glassup.net</a></td>
</tr>
<tr>
<td>16</td>
<td>Tsolakis</td>
<td>Apostolos</td>
<td>CERTH</td>
<td><a href="mailto:tsolakis@iti.gr">tsolakis@iti.gr</a></td>
</tr>
<tr>
<td>17</td>
<td>Turinetto</td>
<td>Maurizio</td>
<td>Regola</td>
<td><a href="mailto:m.turinetto@regola.it">m.turinetto@regola.it</a></td>
</tr>
<tr>
<td>18</td>
<td>Vafeiadis</td>
<td>Thanasis</td>
<td>CERTH</td>
<td><a href="mailto:thanvaf@iti.gr">thanvaf@iti.gr</a></td>
</tr>
<tr>
<td>19</td>
<td>Vergori</td>
<td>Paolo</td>
<td>ISMB</td>
<td><a href="mailto:vergori@ismb.it">vergori@ismb.it</a></td>
</tr>
<tr>
<td></td>
<td>Vlachogiannis</td>
<td>Evangelos</td>
<td>FIT</td>
<td><a href="mailto:evlach@fit.fraunhofer.de">evlach@fit.fraunhofer.de</a></td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------------------</td>
</tr>
<tr>
<td>21</td>
<td>Voutetakis</td>
<td>Spyros</td>
<td>CERTH</td>
<td><a href="mailto:paris@cperi.certh.gr">paris@cperi.certh.gr</a></td>
</tr>
<tr>
<td>22</td>
<td>Zikos</td>
<td>Stylianos</td>
<td>CERTH</td>
<td><a href="mailto:szikos@iti.gr">szikos@iti.gr</a></td>
</tr>
<tr>
<td>23</td>
<td>Ziogou</td>
<td>Chrysovalantou</td>
<td>CERTH</td>
<td><a href="mailto:cziogou@cperi.certh.gr">cziogou@cperi.certh.gr</a></td>
</tr>
</tbody>
</table>
## Revision Control

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>ISMB</td>
<td>April 2015</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>02</td>
<td>CERTH, EPFL, Regola, FIT, ABE, COMAU, ISMB</td>
<td>April 2015</td>
<td>Technology Exploration</td>
</tr>
<tr>
<td>0.3</td>
<td>CERTH, EPFL, Regola, FIT, ABE, ISMB</td>
<td>May 2015</td>
<td>Main Components' Architectures</td>
</tr>
<tr>
<td>0.4</td>
<td>ISMB</td>
<td>June 2015</td>
<td>Overall Satisfactory Functional Architecture</td>
</tr>
<tr>
<td>0.5</td>
<td>ABE, ISMB, Regola</td>
<td>July 2015</td>
<td>Information View</td>
</tr>
<tr>
<td>0.6</td>
<td>ISMB, All partners</td>
<td>July 2015</td>
<td>Deployment View</td>
</tr>
<tr>
<td>0.7</td>
<td>CERTH</td>
<td>August 2015</td>
<td>Technical Use Cases Instantiation</td>
</tr>
<tr>
<td>0.8</td>
<td>ISMB, FIT, Regola, ABE</td>
<td>August 2015</td>
<td>Architecture Perspectives</td>
</tr>
<tr>
<td>0.9</td>
<td>ISMB</td>
<td>August 2015</td>
<td>Draft version, sent to partners for internal review</td>
</tr>
<tr>
<td>1.0</td>
<td>ISMB/ CERTH</td>
<td>August 2015</td>
<td>Submission to the EC</td>
</tr>
<tr>
<td>1.1</td>
<td>EPFL, FIT, ABE, ISMB</td>
<td>March 2016</td>
<td>Review of the Main Components Architectures and APIs</td>
</tr>
<tr>
<td>1.2</td>
<td>ISMB, ABE, CERTH</td>
<td>March 2016</td>
<td>Updated the Functional Architecture</td>
</tr>
<tr>
<td>1.3</td>
<td>ABE, Regola, FIT</td>
<td>March 2016</td>
<td>Updated the Information View architecture</td>
</tr>
<tr>
<td>1.4</td>
<td>FIT</td>
<td>March 2016</td>
<td>Updated the Middleware description</td>
</tr>
<tr>
<td>1.5</td>
<td>All partners</td>
<td>April 2016</td>
<td>Updated Technical Use Cases Instantiation (i.e., sequence diagrams)</td>
</tr>
<tr>
<td>1.6</td>
<td>ISMB</td>
<td>April 2016</td>
<td>Updated Deployment View architecture</td>
</tr>
<tr>
<td>1.7</td>
<td>ISMB</td>
<td>April 2016</td>
<td>Draft version, sent to partners for internal review</td>
</tr>
<tr>
<td>1.8</td>
<td>ISMB</td>
<td>April 2016</td>
<td>Internal reviewers’ comments addressed and ready for submission to the EC</td>
</tr>
<tr>
<td>2.0</td>
<td>ISMB/ CERTH</td>
<td>April 2016</td>
<td>Ready for submission to the EC</td>
</tr>
</tbody>
</table>
## Reviewers List

<table>
<thead>
<tr>
<th>#</th>
<th>Surname</th>
<th>First Name</th>
<th>Beneficiary</th>
<th>Contact Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Della Casa</td>
<td>Emiliano</td>
<td>GlassUP</td>
<td><a href="mailto:emiliano.dellacasa@glassup.net">emiliano.dellacasa@glassup.net</a></td>
</tr>
<tr>
<td>2</td>
<td>León Rojas</td>
<td>Sarah</td>
<td>FIT</td>
<td><a href="mailto:sarah.leon.rojas@fit.fraunhofer.de">sarah.leon.rojas@fit.fraunhofer.de</a></td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

Table of Contents .................................................................................................................. 7
List of Tables .......................................................................................................................... 11
List of Definitions & Abbreviations ....................................................................................... 12
Executive Summary ............................................................................................................... 15

1. Introduction ......................................................................................................................... 16
   1.1 Purpose, Context and Scope of this Deliverable .......................................................... 16
   1.2 Background ..................................................................................................................... 16

2. Architecture Design Approach & Methodology ................................................................. 18
   2.1 Bottom-up Process ......................................................................................................... 19
   2.2 Top-down Process .......................................................................................................... 21
   2.3 Architecture Design and Update Workshops ................................................................. 21
       2.3.1 Architecture Design Workshop One ................................................................. 21
       2.3.2 Architecture Design Workshop Two ................................................................. 21
       2.3.3 Architecture Design Workshop Three ................................................................. 22
       2.3.4 Architecture Update Workshop – Second Iteration ........................................... 22
   2.4 Documentation of the Architecture .............................................................................. 22

3. Technology Exploration ....................................................................................................... 23
   3.1 Introduction .................................................................................................................... 23
   3.2 Analysis .......................................................................................................................... 25

4. Functional View .................................................................................................................... 27
   4.1 Overall SatisFactory Functional Architecture .............................................................. 27
   4.2 Overview of the Main Components ............................................................................... 29
       4.2.1 Smart Sensor Network ......................................................................................... 29
       4.2.2 Integrated DSS .................................................................................................... 33
       4.2.3 Ontology Manager ............................................................................................... 35
       4.2.4 Context-Aware Manager .................................................................................... 37
       4.2.5 AR In-Factory Platform ....................................................................................... 44
       4.2.6 CIDEM .................................................................................................................. 47
       4.2.7 Operational Platform with Augmented Intelligence ........................................... 48
       4.2.8 Collaborative Tools ............................................................................................... 49
       4.2.9 Gamification Framework ..................................................................................... 50
       4.2.10 Re-Adaptation Toolkit ....................................................................................... 51
       4.2.11 Training and Educational Platform ................................................................... 52
       4.2.12 Multi Modal and Augmented HMI .................................................................... 55

5. Middleware as a Supporting Infrastructure ......................................................................... 58
   5.1 Introduction to the LinkSmart Middleware ................................................................... 58
   5.2 LinkSmart Architecture ............................................................................................... 58
   5.3 Event Manager ............................................................................................................... 59
       5.3.1 Event Aggregator .................................................................................................. 60
   5.4 Device Manager ............................................................................................................ 61
   5.5 LinkSmart Miniaturization ........................................................................................... 62
   5.6 Device Manager Integration ......................................................................................... 63

SatisFactory project ■ GA #636302
Figure 2. Initial SatisFactory Conceptual Architecture as Presented in the DoW.

Figure 3. List of Identified Components and Sub-components and Partners Responsibilities.

Figure 4. Component Diagram of the Overall SatisFactory Platform.

Figure 5. SSN Component Diagram.

Figure 6. Robust Communication Infrastructure Component Diagram.

Figure 7. Multi-Radio Communications Component Diagram.

Figure 8. Integrated DSS Component Diagram.

Figure 9. Ontology Manager Component Diagram.

Figure 10. Context-Aware Manager Component Diagram.

Figure 11. Localization Manager Component Diagram.

Figure 12. Gestures & Content Recognition Manager Component Diagram.

Figure 13. Multiple-Media Manager Component Diagram.

Figure 14. AR In-Factory Platform Component Diagram.

Figure 15. CIDEM Component Diagram.

Figure 16. Operational Platform with Augmented Intelligence Component Diagram.

Figure 17. Collaborative Tools Component Diagram.

Figure 18. Gamification Framework Component Diagram.

Figure 19. Re-adaptation Toolkit Component Diagram.

Figure 20. Training and Educational Platform Component Diagram.

Figure 21. High level Multi Modal & Augmented HMIs Component Diagram.

Figure 22. Specialized UI Component Diagram.

Figure 23. LinkSmart Example Device Network.

Figure 24. LinkSmart Component Diagram.

Figure 25. Event Manager Sequence Diagram.

Figure 26. Device Manager Functional View.

Figure 27. SatisFactory Deployment View Architecture.

Figure 28. Example of SatisFactory Application Domain Model.

Figure 29. SatisFactory Information Flow.

Figure 30. UC1.2 Sequence Diagram.

Figure 31. UC1.3 Sequence Diagram.

Figure 32. UC2.1 Sequence Diagram.

Figure 33. UC2.2 Sequence Diagram.

Figure 34. UC3.2 Sequence Diagram.

Figure 35. UC4.1 Sequence Diagram.

Figure 36. UC4.3 Sequence Diagram.

Figure 37. UC5.2 Sequence Diagram.

Figure 38. UC5.3 Sequence Diagram.

Figure 39. UC6.3 Sequence Diagram.
Figure 40. Overall FACT4WORKERS Architecture [Facts4Workers] .......................................................... 96
Figure 41. Overall mainDSS Architecture [mainDSS] .............................................................................. 98
Figure 42. Overall LINKEDDESIGN Architecture [LinkedDesign 2012] .................................................. 100
Figure 43. Overall PLANTCockpit Architecture [PlantCockPit Architecture] ...................................... 102
Figure 44. Application Components [Hardiman et al. 2010] ................................................................ 104
Figure 45. Application Portfolio Catalogue [Hardiman et al. 2010] ....................................................... 105
Figure 46. First Overview of the MIRROR Architecture [Mirror Deliverables] ................................. 107
Figure 47. Adapt4EE Measurement Framework [Adapt4EE D2.1] ........................................................ 108
Figure 48. Adapt4EE Simulation Framework [Adapt4EE D2.1] ............................................................. 108
Figure 49. INERTIA Conceptual Architecture [Jimeno et al.] .............................................................. 110
Figure 50. ebbits Architecture [ebbits] ................................................................................................. 113
LIST OF TABLES

Table 1. Selected SatisFactory Relevant Projects for the Technology Exploration .................................. 23
Table 2. Robust Communication Infrastructure Main APIs. ................................................................. 33
Table 3. Integrated DSS Main APIs. .................................................................................................... 34
Table 4. Ontology Manager Main APIs. ................................................................................................ 36
Table 5. Localization Manager Main APIs. .......................................................................................... 39
Table 6. Gesture & Content Recognition Manager: MQTT APIs........................................................ 40
Table 7. Gesture & Content Recognition Manager: Frame Server APIs ................................................. 41
Table 8. Multiple-Media Manager: Connection APIs ........................................................................... 43
Table 9. Multiple-Media Manager: MQTT APIs .................................................................................. 43
Table 10: Multiple-Media Manager: Signalling APIs ......................................................................... 44
Table 11. AR In-Factory Platform Main APIs and Services ................................................................. 46
Table 12. CIDEM Main APIs. ............................................................................................................. 48
Table 13. Operational Platform with Augmented Intelligence Main APIs .......................................... 49
Table 14. Collaborative Tools Main APIs. ............................................................................................ 50
Table 15. Gamification Framework Main APIs. .................................................................................. 51
Table 16. Re-adaptation Toolkit Main APIs ........................................................................................ 52
Table 17. Training and Educational Platform main APIs ................................................................... 54
Table 18. Multi Modal and Augmented HMI Main APIs .................................................................... 57
Table 19. LinkSmart Main APIs. ......................................................................................................... 64
Table 20. SatisFactory Components HW Requirements. .................................................................... 68
### LIST OF DEFINITIONS & ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AR</td>
<td>Augmented Reality</td>
</tr>
<tr>
<td>BIM</td>
<td>Building Information Modelling</td>
</tr>
<tr>
<td>BPEL</td>
<td>Business Process Execution Language</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Modelling</td>
</tr>
<tr>
<td>CB</td>
<td>Collaboration Broker</td>
</tr>
<tr>
<td>CIDEM</td>
<td>Common Information Data Exchange Model</td>
</tr>
<tr>
<td>CMMS</td>
<td>Computerised Maintenance Management System</td>
</tr>
<tr>
<td>DAC</td>
<td>Device Application Catalogue</td>
</tr>
<tr>
<td>DER</td>
<td>Distributed Energy Resources</td>
</tr>
<tr>
<td>DM</td>
<td>Device Manager</td>
</tr>
<tr>
<td>DoW</td>
<td>Description of Work</td>
</tr>
<tr>
<td>DSM</td>
<td>Demand Side Management</td>
</tr>
<tr>
<td>DSS</td>
<td>Decision Support System</td>
</tr>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>EFFRA</td>
<td>European Factories of the Future Research Association</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FoFPPP</td>
<td>Factory of the Future Public-Private Partnership</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicators</td>
</tr>
<tr>
<td>LEAP</td>
<td>Linked Engineering and Manufacturing Platform</td>
</tr>
<tr>
<td>LM</td>
<td>Localization Manager</td>
</tr>
<tr>
<td>M2M</td>
<td>Machine to Machine</td>
</tr>
<tr>
<td>MES</td>
<td>Manufacturing Execution System</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failures</td>
</tr>
<tr>
<td>MTTR</td>
<td>Mean Time To Failure</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>OSF</td>
<td>Open Semantic Framework</td>
</tr>
<tr>
<td>OSGi DA</td>
<td>OSGi Device Abstraction Layer</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer to Peer</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PWAL</td>
<td>Physical World Adaptation Layer</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>OM</td>
<td>Supply Chain Management</td>
</tr>
<tr>
<td>SFP</td>
<td>SatisFactory Platform</td>
</tr>
<tr>
<td>SME</td>
<td>Small and Medium-sized Enterprise</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SSN</td>
<td>Smart Sensor Network</td>
</tr>
<tr>
<td>UC</td>
<td>Use Case</td>
</tr>
<tr>
<td>Abbr</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra Wide Band</td>
</tr>
<tr>
<td>VR</td>
<td>Virtual Reality</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>WSAN</td>
<td>Wireless Sensor and Actuator Networks</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

This deliverable describes a reviewed version of the SatisFactory architecture updated at M16 (second iteration). There will be a final update at M27 (third iteration).

The first part of the document presents the methodology used to achieve and document the architecture that has been defined as a result of three main steps namely, technology exploration, bottom-up and top-down. In the technology exploration phase both relevant EU funded projects and regional ones have been analysed in order to identify critical aspects for the SatisFactory architecture that should be taken into account. In the bottom-up phase, technology partners have described in detail their main components and their expertise requested by SatisFactory, after that on the basis of the requirements from Task 1.1 a first version of the SatisFactory architecture has been defined. In the top-down phase, applications and platform services have been preliminary tested by means of UML sequence diagrams by taking as input use cases defined within Task 1.3. The documentation of the architecture has been based on the standard IEEE 1471 “Recommended Practice for Architectural Description for Software-Intensive Systems” [IEEE 1471, 2000]. It implies a process based on a set of relevant architecture viewpoints. For SatisFactory three functional viewpoints have been defined, namely functional view, deployment view and information view.

In the functional view the components, their functionality, and their interactions are described. The main identified architecture components are:

- Smart Sensor Network, which includes sensors, wearable devices as well as a Robust Communication providing reliable communication in harsh radio propagation environments.
- Decision Support System (DSS) component, which provides decision on the basis of feedback coming from the shop floor, changes to manufacturing operations and processes, as well as maintenance operations and schedules.
- Safety related modules, such as Localization Manager, Gesture & Content Recognition Manager and Multi-Media Manager.
- Satisfaction and training related components, such as Collaborative Tools, Gamification Framework, Augmented Reality (AR)-In Factory and the Training Education Platforms.
- Finally, a visualization layer has been foreseen, which includes Visual Analytics Module, AR-Glasses User Interface (UI), and Web-based UIs.

The deployment view describes how and where the system will be deployed, which physical components are needed, what are the dependencies, hardware requirements and physical constraints. The information view describes the application domain models and the data flow as well as the distribution.

Finally, a number of use cases have been instantiated through sequence diagrams. The purpose of these sequence diagrams is to clarify how the SatisFactory platform will work and which components are relevant to achieve different tasks.
1. **INTRODUCTION**

This section outlines the purpose, background, and context of this deliverable as well as the structure of the remaining document.

1.1 **PURPOSE, CONTEXT AND SCOPE OF THIS DELIVERABLE**

This deliverable defines an updated architecture for the SatisFactory platform. It is worth remarking that the requirements for the architecture can change during the course of the project, because some aspects of the architecture will undergo verification during development and therefore the architecture described here, cannot be considered to be final or complete. There will be a final updated version of the architecture description at M27.

Within the SatisFactory work package (WP) structure, Task 2.1 (Reference Architecture Design and Technology Exploration) is responsible for analysing the relevant state of the art as well as specifying the SatisFactory system architecture. Having completed the previous steps in WP1 (Domain Analysis and Requirements Engineering), providing an initial set of requirements, this deliverable defines the system architecture, preparing for prototypal implementation to be carried out by the technical work packages within WP3 and WP4.

The architectural description includes aspects related to the identification of the major system components, how they should interact and how their external interfaces should be defined.

This document is structured as follows. Section 2 introduces the methodology for defining and documenting the architecture. Section 3 presents an analysis of technologies and relevant SatisFactory projects. Sections from 1 to 8 present three views of the architecture, namely functional view, information view and deployment view; these are then instantiated for specific technical use cases in section 9. The document also possesses two annexes: Annex 1 reports in detail the analysis of the relevant SatisFactory projects, while Annex 2 presents the description of all subcomponents brought by partners that have been used to support the architecture definition process.

1.2 **BACKGROUND**

The SatisFactory Platform (SFP) aims to make traditional factories more attractive, supported by continuous training of their employees, stimulating team interactions and capitalising the created knowledge and experience in every level of their organization. Going into details, the SFP collects, aggregates and analyses real-time data from heterogeneous sensors, privacy preserving infrared and depth cameras deployed in the shop floor, interacts with Augmented Reality (AR) glasses and novel HMIs through a fundamental component that is the Middleware. In order to distribute the gathered knowledge efficiently and improve the well-being of both the employees and organisation, three main tools are considered, namely, an integrated DSS, a real-time training environment and finally an independent, pervasive data communication network. Additionally, the SFP addresses workers’ safety through proper context aware modules, monitors the production facilities for detecting flaws and problems, and triggers re-adaptation as means for rectifying anomalies and improving throughput. Finally, in order to present all the above information and control capabilities,
intuitive and easy to use interfaces, among them AR environments are used. Gamification methods are applied for motivating workers for unpopular tasks. The SFP will be demonstrated in three different pilots belonging to the three end users (i.e., COMAU, SUNLIGHT, CERTH), where each one targets at different aspect of the operation and activities.
2. **Architecture Design Approach & Methodology**

This section presents the approach and methodology that have been followed by Task 2.1 to define the architecture. It is reminded that Task 2.1 uses three temporal iterations (i) M3-M8, (ii) M15-M16 and (iii) M27-M28. The first iteration defines the first version of the architecture while the other two iterations are used to further improve the architecture, taking into account inputs from both WP1 and the following technical work packages: WP3, WP4 and WP5.

In the first iteration, the architecture definition process has involved three main phases (*technology exploration, bottom-up and top-down*) as depicted in Figure 1. As it can be observed, the process takes two main inputs from WP1: technical requirements reported in D1.1 [SatisFactory D1.1] (available at M4) and use cases (UCs) reported in D1.2 [SatisFactory D1.2] (available at M7). In the *technology exploration* phase, relevant research projects and adopted solutions have been analysed in order identify critical aspects of the SatisFactory architecture that should be taken into account. In the *bottom-up* phase, the initiation of the architecture definition process has been carried out based on technologies and software modules brought by partners that have been considered necessary for the SatisFactory platform. Additionally, the architecture has been built based on the LinkSmart Middleware, which has also guided us with its pre-existent structure. Finally, the *top-down* phase is driven by the use cases that have been used to test the role and interaction among the main components.

![Figure 1. Architecture Design Methodology Diagram Related to the First Iteration.](image)

The second iteration of the architecture (M15-M16), revisions included in this deliverable, takes into account inputs from the technical work packages (i.e., WP3, WP4 and WP5), the updated requirements from T1.1 (available at M14) and the updated business scenarios (BSC) and use cases (UCs) from T1.2 (available at M15). Finally, the third iteration of the architecture (M27-28) will take into account inputs from the technical work packages (i.e., WP3, WP4 and WP5), the final requirements from T1.1 (available at M26) and the final business scenarios (BSC) and use cases (UCs) from T1.2 (available at M27).

Later this section presents in more detail the last two phases of the first iterations. In particular, section 2.1 presents the *bottom-up* process, showing how we have gathered and combined individual
technologies and components that the partners brought into the project; additionally, on the basis of the technical requirements presented in D1.1 [SatisFactory D1.1], we have also identified missing modules and functionalities that are required by SatisFactory and that will be developed and integrated along the project. As a result, this phase produced more detailed views on the initial SatisFactory conceptual architecture, which has been presented in the DoW (see Figure 2). Section 2.2 afterwards deals with the top-down process. This phase has been driven by UCs presented in D1.2 [SatisFactory D1.2]. During these two phases we have held three workshops that lead us step by step toward the definition of the architecture that is presented in this deliverable.

2.1 **Bottom-up Process**

This phase (M4-M6) has aimed to collect and categorize the technologies and software components that the individual partners of the SatisFactory project brought in with them. In addition, the partners’ expertise has been quickly identified and used as best as possible in this first process. This has also helped us to identify gaps in the architecture that needed to be filled in order to achieve the platform envisioned by the SatisFactory project.

As first step of this phase, a template has been defined in order to collect a short description of all components and sub-components brought by the partners. In particular, this template has aimed to collect the following component information: description of the main functionalities, related services, dependencies, inputs needed and outputs provided.

Altogether 13 main components have been identified. Some of them also included sub-components, in total 36. The full list of components and subcomponents along with associated responsibilities is reported in Figure 3.
For each subcomponent a description has been provided according to the provided template by the responsible partner. All the collected sub-component descriptions can be found in Annex 2. As next step, we have started putting the components into an initial architecture, identifying services and dependencies within the platform. We have also added new components in order to cover all the required SatisFactory functionalities.

The result of this process is presented in section 4.1, which includes the defined overall SatisFactory functional architecture. As it can be observed from Figure 4, the SatisFactory overall architecture has been subdivided in different layers, as specified initially, where components are of different nature and offer different functionalities. Having identified this issue, we have clarified the role of each layer in section 4.1.

<table>
<thead>
<tr>
<th>Main component</th>
<th>Sub-component</th>
<th>Responsible partner</th>
<th>Contributing partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Sensor Network</td>
<td>ISMB</td>
<td>CERTH, COMAU, SUNLIGHT</td>
<td></td>
</tr>
<tr>
<td>New IoT Sensors with Robust Communication</td>
<td>ISMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UWB Devices</td>
<td>ISMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cameras Sensor Network</td>
<td>CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automation System - Legacy Sensors</td>
<td>COMAU, SUNLIGHT, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR Glasses</td>
<td>GLASSUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other AR Devices</td>
<td>REGOLA, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legacy and Novel HMIs</td>
<td>GLASSUP, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middleware</td>
<td>FIT</td>
<td>ISMB, CERTH</td>
<td></td>
</tr>
<tr>
<td>Middleware-Core</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Manager</td>
<td>ISMB, CERTH, FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Manager</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Aggregator</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated DSS</td>
<td>ATLANTIS</td>
<td>CERTH, ISMB</td>
<td></td>
</tr>
<tr>
<td>DSS-Core</td>
<td>ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShopFloor Feedback Engine</td>
<td>ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incident Management Tools</td>
<td>CERTH, ISMB, ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance Toolkit</td>
<td>ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual &amp; Real-Time Data Analytics Module</td>
<td>CERTH, ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ontology Manager</td>
<td>EPFL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSV Engines</td>
<td>EPFL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSV Web Services</td>
<td>EPFL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content-Aware Manager</td>
<td>FIT</td>
<td>CERTH</td>
<td></td>
</tr>
<tr>
<td>Localization Manager</td>
<td>ISMB, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple Media Manager</td>
<td>ISMB, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gesture &amp; Content Recognition Manager</td>
<td>ISMB, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR In-Factory Platform</td>
<td>REGOLA, GLASSUP, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF Visualization Tool (GlassUP Version)</td>
<td>GLASSUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF Visualization Tool (Mobile Version)</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF Creation Tool</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF &quot;On-The-Job&quot; Creation Tool</td>
<td>REGOLA, CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIDEM</td>
<td>CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational Platform with Augmented Intelligence</td>
<td>CERTH, ATLANTIS, REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaborative Tools</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social Interaction and Cooperation (Applications)</td>
<td>FIT, REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gamification Framework</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gamification-Based Engine &amp; Tools</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re-Adaptation Toolkit</td>
<td>ISMB, CERTH, ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human Resources Re-Adaptation Toolkit</td>
<td>ISMB, CERTH, ATLANTIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Educational Platform</td>
<td>REGOLA</td>
<td>GLASSUP</td>
<td></td>
</tr>
<tr>
<td>AR OF Visualization Training Tool (GlassUP Version)</td>
<td>GLASSUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF Visualization Training Tool (Mobile Version)</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data Analytics Tool</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR OF Creation Tools (by AR Platform)</td>
<td>REGOLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi Modal &amp; Augmented HMIs</td>
<td>GLASSUP, FIT, CERTH, GLASSUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Analytics Module</td>
<td>CERTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glasses UI</td>
<td>GLASSUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web-based UrI</td>
<td>FIT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.2 **Top-down Process**

This phase (M7-M8) has been strongly focused on each component’s role and their interaction. To this purpose, we have taken advantage of the use cases identified within Task 1.3 and reported in D1.2 [SatisFactory D1.2], and build the related sequence diagrams by considering the components already identified. This has allowed us to stress the services each component would provide, and sketch the interactions between them. Using this approach, and taking into consideration functionalities to be provided, we have also refined and discussed high level components APIs.

2.3 **Architecture Design and Update Workshops**

Along the bottom-up and top-down processes, three online architecture design workshops have been organized, involving technical as well as end-user partners, in order to iteratively define the overall SatisFactory architecture and all its views. In addition, an architecture update workshop has been organized in order refine the architecture in the second iteration of the task. In the following sections we summarize these workshops.

2.3.1 **Architecture Design Workshop One**

Date: May 12\(^{th}\), 2015.

In the first workshop, which was held during the bottom-up process, we reviewed all the components and sub-components brought by the partners. In addition, we identified missing components and technologies on the basis of technical requirements identified within WP1 and reported in D1.1 [SatisFactory D1.1] as well as taking into consideration all the SatisFactory functionalities reported in the DoW. After that, going to the direction of the first definition of the overall architecture, partners responsible of main components have been asked to provide the related internal architecture also showing the interaction with the other SatisFactory main components.

2.3.2 **Architecture Design Workshop Two**

Date: June 17\(^{th}\), 2015.

During the second workshop, which was held during the bottom-up process, the first version of the overall SatisFactory architecture was presented and discussed with the aim to further refine it. First of all, during this workshop, the capability of the LinkSmart Middleware component was highlighted, that is able to incorporate heterogeneous physical devices into applications by offering easy-to-use web service interfaces irrespective of devices’ network technology. After that, it was remarked that the communication among the SatisFactory components can be enabled by means of the “Event Manager” sub-component, which belongs to the Middleware. Doing so, it was observed that scalability issues may arise if all components use the Event Manager for their interaction. As an alternative, it has been proposed to use a hybrid approach, where some components use the Middleware (i.e. the Event Manager sub-component) to exchange data and/or to signal any type of events, whereas others use a direct communication, by means of ad-hoc APIs, to exchange big amount of data, such as video streams.
2.3.3 **Architecture Design Workshop Three**

Date: July 09th, 2015.

The top-down process has been initiated by the third workshop within which we also started the discussion about the definition of the deployment and information views of the architecture. As mentioned above, the top-down process has taken as main input the use cases identified within Task 1.3, which have been reported in D1.2 \[SatisFactory D1.2\]. Among the 23 identified UCs, only 10 have been selected to be analysed by means of sequence diagrams. The UCs selection has been performed such that the main components of SatisFactory appear in at least one sequence diagram. It has been agreed with the whole consortium to draw the sequence diagrams regarding the remaining UCs during the next iteration of the SatisFactory architecture, i.e. between M13 and M14.

2.3.4 **Architecture Update Workshop – Second Iteration**

Date: March 31st, 2016.

The second iteration of the task (M15-M16) has been focused on the architecture update taking into account: refined inputs from the technical work packages (i.e., WP3, WP4 and WP5), the updated requirements from T1.1 (available at M14) as well as the updated business scenarios (BSCs) and use cases (UCs) from T1.2 (available at M15). In particular, the three architectural viewpoints have been updated along with the main components and related main APIs.

2.4 **DOCUMENTATION OF THE ARCHITECTURE**

The process used for documenting the architecture in this deliverable is based on the standard IEEE 1471 "Recommended Practice for Architectural Description for Software-Intensive Systems" [IEEE 1471, 2000]. This standard establishes a methodology for the architectural description of software intensive systems. One main part of this methodology is the use of viewpoints: collections of patterns, templates and conventions for constructing one type of view. One example is the functional viewpoint, which contains all functions that the system should perform, the responsibilities and interfaces of the functional elements and the relationship between them. These functions can be described using UML diagrams. In this initial version of the architecture, we have decided to define the three most important viewpoints: functional, information and deployment viewpoints.

- Functional viewpoint (section 4): This viewpoint describes the functional elements needed to meet the key requirements of the architecture. It will present proposals in a descriptive way and UML component diagrams will assist in the understanding of the proposal. It will describe responsibilities, interfaces, and interactions between the functional elements.
- Deployment viewpoint (section 6): This viewpoint describes how and where the system will be deployed and what dependencies exist, considering for example hardware requirements and physical restraints.
- Information viewpoint (section 0): The information viewpoint describes the data models and the data flow as well as the distribution. The viewpoint also defines which data will be stored and where. The description of where data will be manipulated is also a part of this viewpoint.

Finally, to address quality properties and cross-cutting concerns, architectural perspectives are presented as well in section 8. Typical examples for SatisFactory are: satisfaction (section 8.1), safety (section 8.2) and scalability (section 8.3) perspectives.
3. **TECHNOLOGY EXPLORATION**

### 3.1 INTRODUCTION

The purpose of this section is to explore the appropriate technologies that are relevant to SatisFactory and use these results as input for the architecture definition process. Moreover, critical architectural aspects are identified that should be taken into account during the architecture definition. In particular, this section presents a H2020 project funded under the same call FoF04 and other relevant FP7 funded projects and regional ones, where members of the SatisFactory consortium had been involved in.

Table 1 provides a higher level overview of the selected relevant projects with information such as research area, duration of the project, and partners involved with their core role in the project in the context of SatisFactory project.

A more detailed analysis of the relevant research projects is reported in the Annex 1.

<table>
<thead>
<tr>
<th>Name of Project</th>
<th>Project Information</th>
<th>Involved Partner</th>
</tr>
</thead>
</table>
| FACTS4WORKERS (Worker-Centric Workplaces in Smart Factories) | **Research area:** Developing smart factories that are attractive to workers  
**Tech. Area:** H2020-FoF4  
**Duration:** Start: 1.12.2014 – 48 months  
**Website:** [http://facts4workers.eu/](http://facts4workers.eu/) | None from SatisFactory |
| mainDSS (maintenance Decision Support System) | **Research area:** Eurostars Project (international research for R&D intensive SMEs)  
**Tech. Area:** 1.2.15 Knowledge Management, Process Management **Market Area:** 8.7 Industrial Services  
**Duration:** Start: 7/2010 – 40 months  
**Role of Partner:** Coordinator / Main Technical Contributor  
**Website:** [www.maindss.eu](http://www.maindss.eu) | ABE |
| LinkedDesign | **Research area:** FP7-2011-NMP-ICT-FoF  
**Tech. Area:** FoF-ICT-2011.7.4 - Digital factories  
**Market Area:** Manufacturing design and product lifecycle management  
**Call:** **Duration:** Start: 09/2011 – 48 months  
**Role of Partner:** EPFL was the leader of Ontology and Inference system for design and manufacturing domain work package. EPFL contributed to the overall architecture and requirements, as well as to the | EPFL |
<table>
<thead>
<tr>
<th>Project</th>
<th>Research area</th>
<th>Role of Partner</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlantCockpit</td>
<td>Research area: EU FP7 FoF ICT 2010</td>
<td>Role of Partner: EPFL was the leader of Requirements and Standardization work packages. EPFL was also involved in Data and Process modelling. EPFL contributed to the overall architecture and KPI interdependences concept and reference model</td>
<td>EPFL</td>
</tr>
<tr>
<td>CoSpaces (Innovative collaborative work environments for engineering and design)</td>
<td>Research Area: Collaborative Working Environments</td>
<td></td>
<td>FIT</td>
</tr>
<tr>
<td>Mirror (Mirror - Reflective Learning at Work)</td>
<td>Research Area: Technology-enhanced learning</td>
<td>Role of Partner: Application Partner. The application case will be in the Civil Protection Organisation in Turin (Torino).</td>
<td>Regola</td>
</tr>
<tr>
<td>Adapt4EE (Occupant Aware, Intelligent, Adaptive Enterprises)</td>
<td>Project name: Adapt4EE</td>
<td>Role of Partner: CERTH was the coordinator of the project. Furthermore, the occupancy extraction and modelling components and the occupancy simulation engine implemented within this project have been designed and developed by CERTH, while the middleware module and its sub-components have been implemented by Fraunhofer.</td>
<td>CERTH</td>
</tr>
<tr>
<td>INERTIA (Integrating Active, Flexible and Responsive Tertiary Prosumers into a)</td>
<td>Research area: ICT systems for energy efficiency (FP7-ICT-2011-6.2)</td>
<td>Role of Partner: CERTH was the coordinator of the project and has designed and developed the occupancy</td>
<td>CERTH</td>
</tr>
</tbody>
</table>
## 3.2 Analysis

Critical architectural aspects of the selected relevant projects and their technical advantages in the context of SatisFactory are briefly described as follows.

- **FACTS4WORKERS** [Facts4Workers] uses the information and communication technology to improve the manufacturing process regarding flexibility, efficiency, reliability; further, this essential information is supported by optimized information and communication technology, self-learning working environment, and in-situ learning for the workers while operating the machines.

- **mainDSS** [mainDSS] project tries to maximize the operational availability of assets and minimize the life-cycle costs in order to improve the competitive business environment by improving the maintenance departments and decisions; thus, an effective communication between managerial and technical departments is established by prioritizing the necessary maintenance pillars. In SatisFactory an improved shop floor feedback and decision making system will be provided for gains in productivity, workers wellbeing and comfort. Also it will reuse the maintenance and production facilities ontology, the decision support engine, as well as the algorithms for producing recommendations for improving intra-factory procedures and maintenance operations developed in the previous project.

- **LinkedDesign** [LinkedDesign] project decouples the actual process execution from the design and engineering of products, and the manufacturing processes. The project provides a solution that allows the integration of the holistic view on data across the full product lifecycle for ICT by developing methodologies and novel integration tools. SatisFactory is going to use the framework developed before as a basis for context-aware services, most importantly for the acquisition of context information and its presentation in a way that can be properly interpreted and managed.

- **PlantCockpit** [PlantCockPit] project develops a central environment for monitoring and controlling of all intra-logistical processes within a manufacturing environment. PLANTCockpit provides production supervisors, foremen, and line managers with the required visibility to make well-informed decisions for optimizing plant processes in a quick and efficient way. SatisFactory will use the plant visualization outcome of PlantCockpit and enrich it with a semantic framework which will support the context-aware platform for an improved shop floor environment.
• CoSpaces [CoSpaces] project uses technologies in manufacturing and design for ground-breaking innovations in context-aware interfaces, natural interfaces, and “human-centric” workspaces supporting a range of collaboration scenarios within product lifecycles. The developed platform supports product design and manufacturing in geographically dispersed teams in distributed virtual engineering enterprises. It helps the team members to participate in decision-making, view designs, propose modifications, and access reference material in ways that are specific to their disciplines, and allows them to interact using a range of devices depending on their current location and situation. In SatisFactory, the platform will be enhanced by social networking and gamification aspects.

• The vision of MIRROR is to empower and motivate employees to learn by reflection of tacit work practices and personal experiences. MIRROR shall help employees capture experiences and collaboratively develop creative solutions for problems that need to be solved immediately [Mirror 2015]. This will be achieved by complementing personal and organisational learning environments (which mainly rely on knowledge being explicitly available) with highly personal MIRROR applications for individual, social, creative, game-based as well as organisational reflection and real-time learning.

• Adap4EE aims at augmenting the contemporary architectural envelope by incorporating business and occupancy related information thus providing a holistic approach to the planning, design & evaluation of energy performance of construction products at an early design phase [Adap4EE]. Adap4EE aims to deliver and validate a holistic energy performance framework that incorporates architectural metadata and environmental parameters (i.e., BIM), critical business models (i.e., BPM), treating occupants as the central reference point [Adap4EE]. The new device managers for occupancy already implemented will be transferred to SatisFactory as valuable input for the extensions needed in the project.

• INERTIA [Jimeno et al.] project aims at providing an overlay network for coordination and active grid control, running on top of the existing grid and consisting of distributed and autonomous intelligent Commercial Prosumer Hubs. This way, it will address the present “structural inertia” of DG by introducing more active elements combined with the necessary control and distributed coordination mechanisms.

• The ebbits project² provides a semantic approach to IoT and hence introduces an innovative bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real world. The ebbits platform is based on SoA (Service Oriented Architecture), so virtualizing every device into a service and allowing these services to semantically discover, configure, and communicate with each other. Furthermore, the ebbits platform aims at supporting interoperable applications to process context-aware data, separated in time and space, acquiring information and real-world events and to handle end-to-end business workflows, including comprehensive consumer demands. ebbits brings to SatisFactory, the expertise in the development of secure and reliable network and communication infrastructure and the control of large-scale complex systems in manufacturing lines interconnecting legacy systems with smart dependable multi-radio wireless sensor networks.

---

² www.ebbits-project.eu
4. **FUNCTIONAL VIEW**

This chapter provides the overall functional architecture of the SatisFactory platform and an overview of its main components, their functionalities and interactions.

4.1 **OVERALL SATISFACTORY FUNCTIONAL ARCHITECTURE**

Figure 4 shows the overall SatisFactory functional architecture that has been designed on the basis of the requirements, business requirements/use cases identified within WP1 and reported in deliverables D1.1 [SatisFactory D1.1] and D1.2 [SatisFactory D1.2], respectively.
Figure 4. Component Diagram of the Overall SatisFactory Platform.
As it can be seen from Figure 4, the overall architecture is organized in five layers, namely Physical Layer, Decision Layer, Facility Layer, Service Layer and Attractive User Interface (UI) Layer. The Physical Layer coincides with the Smart Sensor Network component that includes all the physical components deployed on the shop floor and equipped with a communication interface (e.g., cameras, Wireless Sensor Networks (WSNs), Ultra Wide Band (UWB) wearable devices, AR devices, HMIs, Automation System – legacy sensors as well as new sensors with robust communication). The bidirectional communication with the existing shop floor devices takes place through the SSN.

The Decision Layer corresponds to the Integrated Decision Support System (DSS) component, which provides decision on the basis of feedback coming from the shop floor, such as incidents, changes to manufacturing operations and processes, as well as maintenance operations and schedules. Thus, this layer provides actionable knowledge and recommendations to the Facility Layer, which includes the Ontology Manager, the Context-Aware Manager, the AR-In Factory Platform and the SatisFactory Repository (i.e., the CIDEM). In particular, the Facility Layer provides safety and AR related data to the upper layer, named Service Layer. This layer includes the Operational Platform with Augmented Intelligence, the Collaborative Tools, the Gamification Framework, the Re-adaptation Toolkit and the Training Education Platform components. The Service Layer provides services that aim to improve workers’ satisfaction (through collaboration and training functionalities) as well as to provide re-adaptation functionalities. All the three layers (i.e., Decision, Facility and Service) interact with and provide visualization data to the Attractive UI Layer, which includes UI components such as: Visual Analytics Module, AR Glasses UI, other AR UIs, and Web-based UI. The glue of the SatisFactory platform is represented by the Middleware module, which handles the heterogeneity of physical devices and allows messages and events to be exchanged among the SatisFactory components and services. The Middleware includes three main subcomponents, namely: Device Manager, Event Aggregator and Event Manager. In particular, the Device Manager provides the instruments to simplify the integration of heterogeneous physical devices while the Event Manager implements a publish-subscribe based notification for events exchange. For large amount of data, ad-hoc direct message exchanging among components is also possible.

4.2 OVERVIEW OF THE MAIN COMPONENTS

This section introduces the main components of the SatisFactory platform presenting their main functionalities, interactions and APIs.

4.2.1 Smart Sensor Network

The Smart Sensor Network (SSN), as depicted in Figure 5 consists of heterogeneous devices that allow the interaction between the physical world (i.e., the shop floor) and the SatisFactory platform. Some devices specified by the identified use cases are cameras (e.g., depth, thermal and IR), UWB devices (fixed and wearable) for workers’ localization, AR glasses, legacy/novel HMIs, Automation Systems with integrated legacy sensors (e.g., flow, pressure, temperature, etc.), and new IoT sensors equipped with a robust communication module offered by the Robust Communication Infrastructure presented in the following section.
4.2.1.1 **Robust Communication Infrastructure**

This component\(^3\) guarantees IoT communication robustness and correct data delivery in harsh radio propagation environments. This component hosts the sensing and actuation applications that integrate shop floor data and events into the cyber-world following an IoT approach. This component contributes to the supply of information to the SatisFactory knowledge-base in order to enable context-aware applications.

In order to ensure dependability, this component employs the following set of enabling features:

- Multi-radio connectivity: exploiting multiple network interfaces in a complementary way to guarantee communication robustness in harsh radio propagation environments.
- Intelligent functions: leveraging on cognitive techniques to enable self-configuration capabilities, such as discovery, adaptation and healing.
- Cooperative methods: exploit a number of collaborating IoT devices, in terms of sensing and computation capabilities, in order to achieve complex tasks more accurately and efficiently, as well as to improve context extraction.
- Secure data exchange: ensuring information integrity in order to reliably identify scenarios that may compromise the safety of workers.
- Power management: enabling devices to adopt different operation profiles for handling processing and communications capabilities.

The architecture of the Robust Communication Infrastructure component is depicted in Figure 6. Two main components are observed: the Robust Communication Gateway and the Wireless Sensor and Actuator Network (WSAN).

The Robust Communication Gateway is responsible for (i) collecting and processing the WSAN data, (ii) administering the WSAN operations at network and application levels and (iii) exposing WSAN resources and capabilities to the SatisFactory framework, via the middleware. The Robust Communication Gateway hosts the following functions:

- Gateway Workflow Management: coordinates the operations of the Gateway's internal functions.
- Device Resource Abstraction: interacts with the raw data and protocols of the WSAN and creates virtual resources corresponding to those of the real devices.
- Multi-Radio Communications: administers the utilization of the network interfaces at the WSAN, based on application requirements, network performance and spectrum conditions. This function consists of a number of sub-components as shown in Figure 7.

---

\(^3\) More details about this component will be provided in D4.1 [SatisFactory D4.1].

---

D2.1.2 - SatisFactory System Architecture ■ April 2016 ■ ISMB

SatisFactory project ■ GA #636302
- Security Mechanism: hosts encryption based security method at the network or transport layers, such as IPSec or DTLS respectively, in order to ensure the integrity of the data exchanged between the WSAN and the Gateway.
- Application Data Processing: provides application-related events and information, obtained from the shop floor, to consumer applications at the SatisFactory framework.
- Gateway Server: provides the SatisFactory framework an access to the WSAN resources via the Shop Floor Monitoring & Control API.

The Wireless Sensor and Actuator Network (WSAN) comprises the logical and physical modules that carry out the user application at the networked embedded devices level. It consists of the following functions:

- Sensing and Actuation: enables interaction with the physical quantities of the shop floor, obtaining information and applying control/corrective actions.
- Energy Management: evaluates the operational and application status of the embedded devices and controls the power supply to the connected peripherals.
- Radio Communications: carries out the data exchange between the embedded devices and the Gateway. The function also determines the network interface and the radio parameters to be used based on instructions from the Multi-Radio Communications function of the WSAN Gateway.
- Security Mechanisms: interacts with the Security Mechanisms function of the WSAN Gateway in order to secure the data exchange.
- Self-Configuration: implements intelligent algorithms at device level to dynamically adapt its local behaviour according to context information, such as network discovery and healing, network interface status and utilization and other application-related actions.
- Object Cooperation: supports the execution of different tasks collaboratively among embedded devices, including sensing, actuation, computation and communication in order to perform complex tasks more efficiently.
Figure 6. Robust Communication Infrastructure Component Diagram.

Figure 7. Multi-Radio Communications Component Diagram.
Table 2 shows the main APIs of the Robust Communication Infrastructure.

**Table 2. Robust Communication Infrastructure Main APIs.**

<table>
<thead>
<tr>
<th>Robust Communication Framework APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shop Floor Monitoring &amp; Control API</strong></td>
<td>This interface exposes the resources and capabilities of the underlying system to the SatisFactory platform. It enables SatisFactory system operators to display information related to the sensing application events and parameters as well as networking status. It also provides actuation and control capabilities related to the application and the network.</td>
</tr>
<tr>
<td><strong>Robust Communication API</strong></td>
<td>This is an internal interface that enables the exchange of raw data between the WSAN Gateway and WSAN devices. In the uplink, this interface delivers sensing and networking related data to the Gateway. While in the downlink, this API distributes application control and actuation commands to the embedded devices.</td>
</tr>
</tbody>
</table>

### 4.2.2 Integrated DSS

The Integrated Decision Support System is responsible for providing feedback to the decision makers regarding immediate actions needed in response to shop floor incidents, together with changes to manufacturing operations and processes and also maintenance operations and schedules. In order for the component to be able to provide actionable knowledge and recommendations, it needs inputs from the Smart Sensor Network (through the Device Manager), the overall model of production and maintenance activities (through CIDEM) and context – aware knowledge stemming from the Ontology Manager. The outputs of the component are mainly events (e.g. reaction to incident detection, re-adaptation of the production processes, re-scheduling of maintenance activities, etc.), which through the Event Manager can trigger actions in any other component of SatisFactory (e.g. Context-aware and AR-enabled HMIs) which has subscribed for notifications of these events. Based on the above, the main components that interact with Integrated DSS are Device Manager, Event Manager, CIDEM and Ontology Manager. The high level functional architecture of the Integrated DSS component is depicted in the following figure:
### Table 3. Integrated DSS Main APIs.

<table>
<thead>
<tr>
<th>Integrated DSS API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SSN Measurements</strong></td>
<td>This is a required interface. The maintenance toolkit, which is a sub-component of the Integrated DSS, monitors and supervises in real time the production processes in order to diagnose possible problems, flaws or malfunctions and triggers events for activating maintenance procedures or safety mechanisms. In order for the above to happen, the maintenance toolkit needs a constant flow of measurements, stemming mainly from Legacy Sensors. These measurements will be provided by an interface of the Device Manager, which must expose Events that signal capturing of new measurements and also from the CIDEM which should provide service methods using the RESTful architectural pattern, that return a collection of measurement events for a given sensor over a specific period of time.</td>
</tr>
<tr>
<td><strong>Context-Aware Knowledge</strong></td>
<td>This is a required interface. In order for the Integrated DSS to provide recommendations, the component must be able to “understand” the environment into which it operates. In other words, the decision support system must be able to get hold of implicit and explicit knowledge regarding manufacturing processes and operations, maintenance schedules and procedures, production plans and details, worker activities and availability. This knowledge must be based in well-defined vocabularies, which could provide the ability for reasoning and inference, so a SPARQL endpoint which will return RDF triples of de-referenceable URI’s will be expected.</td>
</tr>
<tr>
<td><strong>UI and AR Events</strong></td>
<td>This is a supplied interface, which will be used mainly by HMI’s. The shop floor feedback engine will take as an input incidents detected by the Incident Management Tools, pair them with corresponding</td>
</tr>
</tbody>
</table>
response procedures and provide feedback to the shop floor in terms of actionable knowledge and recommendations. This feedback will have the form of events, which will be published to any subscriber interested in executing some action based on that event. Context aware and AR-enabled HMIs will mainly consume these events, in order to either inform the end users for specific incidents, that have happened on the shop floor, or support the decision makers for specific actions needed to be decided on production processes or maintenance procedures.

| Production Model | This is a required interface. The Production Activities tool, which is a sub-component of Integrated DSS, needs to detect the operating status of production lines in order to decide if there are flaws or malfunctions on parts of it. In such a case it must initiate an alarm, which will finally publish a number of events to all registered subscribers. These tools need an API provided by the CIDEM, which will expose a number of methods returning static and dynamic data regarding products, production facilities and processes in order to build the overall production model. |

4.2.3 **Ontology Manager**

The Ontology Manager will be implemented using the Open Semantic Framework architecture. The Open Semantic Framework (OSF) is an integrated software stack using semantic technologies for knowledge management. It has a layered architecture that combines existing open source software with additional open source components developed specifically to provide a complete semantic technology framework. The premise of the entire stack is based on the RDF (Resource Description Framework) data model. RDF provides the ready means for integrating existing structured data assets in any format, with semi-structured data like XML and HTML, and with unstructured documents or text.

The OSF framework is made operational via ontologies that capture the domain or knowledge space, matched with internal ontologies that guide OSF operations and data display. This design approach is known as ODapps, for ontology-driven applications. Mediating the natural semantic differences that arise between people, departments and other actors in the information space is done by employing best practices for ontology and vocabulary construction. Main characteristics of this component are given below.

- Data is generally exposed (and universally available) as linked data
- SPARQL endpoints and APIs are generally RESTful in design
- The overall architecture of the Ontology Manager component is modular

The OSF Web services are generally RESTful in design and are based on HTTP and Web protocols and open standards. For the needs of SatisFactory, there is a set of web services covering functionality in CRUD, Revision, Search, Authentication, Dataset Management, Converter and Ontology management.

The functionality of the web services layer is based on controlling and interacting with the OSF Engines. Using the common RDF data model means that all web services and actions against the data only need to be programmed via a single, "canonical" form. Simple converters convert external, native data formats to the RDF form at time of ingest; similar converters can translate the internal RDF form back into native forms for export (or use by external applications). This use of a
"canonical" form leads to a simpler design at the core of the stack and a uniform basis to which tools or other work activities can be written.

The OSF Engines governs the index and management of all OSF content. Documents are indexed by the full-text search engine, while information about their structural characteristics and metadata are stored in an RDF Database, called a "triple store". The schema aspects of the information (the ontologies) are separately managed and manipulated with their own W3C standard application, the OWL API. At ingest time, the system automatically routes and indexes the content into its appropriate stores. Another engine is available for semi-automatic assistance in tagging input information and other natural language processing (NLP) tasks.

Lastly, the proposed high-level architecture for the Ontology Manager (see Figure 8) conceives of 1 input and 2 outputs. In fact, the Ontology Manager consumes data from the CIDEM, the OSF Engines work in background, while the OSF Web Services provide an interaction interface with the context-aware engine and the DSS.

<table>
<thead>
<tr>
<th>Ontology Manager API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP API</td>
<td>The Ontology Manager RESTful Web services layer may be accessed directly via API or command line, or may be controlled and interacted with using standard content management systems (CMSs). All web services are exposed via APIs and SPARQL endpoints. Each request to an individual web service returns an HTTP status and optionally a document of result sets. Each results document can be serialized in many ways, and may be expressed as either RDF, pure...</td>
</tr>
</tbody>
</table>

Figure 9. Ontology Manager Component Diagram.
XML, JSON, or different flavours of irON to be included, such as analysis or advanced inference engines.

4.2.4 Context-Aware Manager

The Context-Aware Manager component supports workers’ safety and comfort by leveraging location information as well as media and gesture recognition data. As depicted in Figure 10, the context-aware manager component includes three enabling modules, namely the Localization Manager, the Gesture & Content Recognition Manager and the Multiple-Media Manager, which are presented in the following subsections.

![Figure 10. Context-Aware Manager Component Diagram.](image)

4.2.4.1 Localization Manager

The Localization Manager (LM) component guarantees worker’s safety functionalities leveraging accurate positioning service and virtual fencing logics. The position information is estimated by UWB wearable devices that implement a Bayesian method based algorithm. Furthermore, hybrid and cooperative approaches are used to get a more robust positioning service. The virtual fencing logics are based on a Point in Polygon algorithm. This algorithm takes as input the estimated positions associated to workers and checks if they are inside or not pre-defined polygons. These polygons represent forbidden and potential dangerous areas in the shop-floor. The Localization Manager is able to generate geo-fencing events/alerts when a worker approaches a forbidden/dangerous area as well as data history related to localization data.

As mentioned in [SatisFactory D1.2], this module supports Use Case (UC) applications regarding to the online recognition of workers activities (UC-3.1), identification of worker’s and equipment’s location (UC-3.4) and monitoring and decision support of operations and maintenance procedures (UC-4.3). Moreover, the LM supplies information to the SatisFactory knowledge-based so as to enable the platform services. In order to provide virtual fencing logics and positioning services the LM needs location estimations from UWB wearable devices (through the LinkSmart middleware) and the forbidden/dangerous zones (through the CIDEM).

The LM architecture is depicted in Figure 11. As it can be observed, four main components are included: the handler for local forbidden areas, the handler for local resource catalogue, the handler for local service catalogue and the logics manager.

The handler for local forbidden areas is responsible for (i) updating the forbidden areas locally, (ii) defining a new forbidden area, locally and at the CIDEM according to data analysis received from middleware and (iii) sending a message to SatisFactory components when a new forbidden area is created.

---

*SatisFactory project ■ GA #636302*
The handler for the local resource catalogue is responsible for (i) updating the resource catalogue locally and (ii) sending error messages through middleware when a resource is lost.

The handler for the local service catalogue is responsible for updating its service catalogue in the middleware every certain time in order to notify other resources/services that LM services are alive.

The logics manager comprises the logical modules that implement the applications at the facilities layer level. It consists of the following functions:

- **Virtual fencing:** Receives localization estimations from UWB wearable devices, checks if they are registered in the resource catalogue, after that, it checks if the workers (associated to the UWB devices) are inside of any forbidden areas. Alert messages are generated and sent through the middleware, when any worker enters or exits from any forbidden area. Furthermore, an error message is sent through the middleware if the LM receives localization data from an UWB wearable device not registered in the resource catalogue.

- **Path optimization:** Takes as input localization estimations from UWB wearable devices and a target position in the shop-floor to be reached. The LM generates and sends messages to workers through the middleware suggesting in which direction to go and informing them when the final position has been reached.

- **Time optimization for repairing a malfunction:** Takes as input a malfunction alarm, searches the closest worker with the required skills and sends him messages indicating the direction to reach the malfunction position.

![Figure 11. Localization Manager Component Diagram.](image)

Table 5 shows the main APIs of the Localization Manager component.
Table 5. Localization Manager Main APIs.

<table>
<thead>
<tr>
<th>Localization Manager APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Forbidden areas</td>
<td>This interface is defined by the CIDEM and it is used to retrieve the predefined Forbidden areas. This API is invoked when a new forbidden area has been inserted by any SatisFactory component.</td>
</tr>
<tr>
<td>Insert new Forbidden area</td>
<td>This interface is defined by the CIDEM and it is used to insert a new forbidden area when the Handler for local forbidden areas detects a new one.</td>
</tr>
<tr>
<td>Publish to middleware</td>
<td>This interface is defined by the MQTT protocol. This API is invoked when the Localization Manager publishes messages to its subscribers.</td>
</tr>
<tr>
<td>Publish to client</td>
<td>This interface is defined by the MQTT protocol. The MQTT broker sends the messages to the LM from publishers to which the LM is subscribed.</td>
</tr>
<tr>
<td>GET from Middleware</td>
<td>This interface is defined by the REST protocol. This API is invoked when the Localization Manager updates its local resource catalogue.</td>
</tr>
<tr>
<td>POST to Middleware</td>
<td>This interface is defined by the REST protocol. This API is invoked when the LM updates its local service catalogue in the LinkSmart Service catalogue.</td>
</tr>
</tbody>
</table>

4.2.4.2  Gesture & Content Recognition Manager

This component is devoted to the analysis of video streams (both RGB and depth) at the level of the Smart Assembly Station (SAS). This software module is instantiated once for every SAS. It provides high level information that can be leveraged by other modules of the SatisFactory ecosystem to deliver advanced services for an effective support of workers.

Relying on input from a composite RGB and time-of-flight (ToF) sensor – such as Microsoft Kinect 2.0 – and adopting advanced computer vision and machine learning algorithms, the GCRM can reliably identify the presence of workers in the SAS in conjunction with other relevant statistics. Services offered are:

- Presence detection and people count: GCRM is able to detect the occupancy of the SAS and assess the worker count up to 8 people.
- Safety gear detection: GCRM can assess if the worker is wearing safety equipment such as helmet, gloves and jacket based on their appearance.
- Fall detection: GCRM performs the continuous monitoring of the worker to detect fall events thus enabling a prompt reaction to accidents.
- Gestures detection: GCRM detect worker gestures from a predefined set to allow contactless human interaction with other SatisFactory modules such as the Digital Andon installed in the SAS.

As depicted in Figure 12, the GCRM interacts with the SatisFactory ecosystem mainly through the LinkSmart middleware and provides most of its services as MQTT messages over distinct topics.
Figure 12. Gestures & Content Recognition Manager Component Diagram.

The next table details the content of MQTT messages in JSON format.

Table 6. Gesture & Content Recognition Manager: MQTT APIs.

<table>
<thead>
<tr>
<th>GCRM: MQTT APIs</th>
<th>Description</th>
</tr>
</thead>
</table>
| Keep Alive API  | The message is sent every 20 seconds and contains the following data:  
|                 | • Sensor framerate  
|                 | • GCRM processing framerate  
|                 | • ID of the SAS  
|                 | • People count  
| Presence Detection API | The message is sent every time GCRM detect a change in the number of people in the SAS. It contains the following data:  
|                 | • Current people count  
|                 | • Previous value of people count  
| Safety gear detection API | The message is sent whenever a change is spotted in gear detection. It contains the following data for each of the items (helmet, gloves and jacket):  
|                 | • ID of the item  
|                 | • Status (detected / not detected)  
|                 | • Status changed w.r.t. previous message (true/false)  
|                 | This message is sent in retain mode  
| Fall detection API | This message is sent every time a workers fall is detected. It has no fixed payload, since the binary information of the fall event is represented by the message itself  
| Gesture detection API | This message is sent every time a worker gesture is detected. It contains the ID of the gesture.  

Message data detailed in the previous table is intended as a minimal set that could be extended, if necessary.
Beyond Link Smart, the GCRM interacts with the SAS instance of Multiple-Media Manager (M3) by providing frames and frame metadata to encode and redistribute. So GCRM has a dedicated API to act as a frame server. The video streams are two: RGB and infrared. In order to ensure real time delivery of uncompressed Full HD frames GCRM and M3 are hosted on the same machine and communicate via named pipes (one for RGB and one for infrared). Pipes are created by M3 and GCRM connect to them.

### Table 7. Gesture & Content Recognition Manager: Frame Server APIs.

<table>
<thead>
<tr>
<th>GCRM: Frame Server APIs</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Frame server RGB**    | For each RGB frame the GCRM writes on the pipe a 72 byte structure of frame metadata containing:  
  - Sensor ID  
  - Diagonal field of view  
  - Horizontal field of view  
  - Vertical field of view  
  - Relative time  
  - People count  
  - Bounding box of the most relevant person in the frame if any  
  - Distance from the sensor of the most relevant person in the frame if any  
  - Frame exposure time  
  - Frame gain  
  After this GCRM writes on the pipe the uncompressed frame data: 1920 x 1080 pixels. Each pixel has 4 channels (red, green, blue, alpha) for a total of 4 bytes per pixel. |
| **Frame server IR**     | For each IR frame the GCRM writes on the pipe a 64 byte structure of frame metadata containing:  
  - Sensor ID  
  - Diagonal field of view  
  - Horizontal field of view  
  - Vertical field of view  
  - Relative time  
  - People count  
  - Bounding box of the most relevant person in the frame if any  
  - Distance from the sensor of the most relevant person in the frame if any  
  After this GCRM writes on the pipe the uncompressed frame data: 514 x 424 pixels. Image is grayscale with 8 bit per pixel. |

### 4.2.4.3 Multiple-Media Manager

The Multiple-Media Manager a software module that can be broken down in the following subcomponents:

- One or more Multiple-Media Manager streaming servers;
- The Multiple-Media Manager central unit.
The intended goal of the Multiple-Media Manager streaming servers is to provide a reliable distribution infrastructure for video streams from heterogeneous sources. The server acts as a proxy encoding each video frame received from sources in the Motion-JPEG (MJPEG) format and providing streaming capabilities to clients, through HTTP sessions.

For the needs of SatisFactory, the component is used to distribute video information about each Smart Assembly Station (SAS) as provided by the Gesture & Content Recognition Manager (GCRM). The streaming server can also buffer the encoded frames to allow a deferred video when particular conditions are matched.

Furthermore, the server can enrich the processed video with auxiliary information embedded in JPEG images using the Exchangeable Image File Format (Exif). At the moment the following information are available:

- Generic information: date of creation, image number, information about the M3 software version, ...;
- Smart Assembly Station (SAS) information: station position and id, ...;
- People information (provided by GCRM): people count, subject area (bounding box) and subject distance, ...;
- Camera information (provided by GCRM): fields of view, exposure time, ...;

Using the Re-adaptation Toolkit, the shop-floor supervisors can receive MJPEG streams from each Multiple-Media Manager streaming server, both real-time or deferred, in order to facilitate reaction time in the occurrence of certain events.

When specific events such as fall alarms are triggered, the Multiple-Media Manager server automatically encodes the video and uploads it to the central unit in order to collect historical information about incidents and later provide them on demand.

The central unit also allows VOIP interactions between employees and supervisors, ensuring interoperability and portability between the heterogeneous devices. This result is achieved using WebRTC technology. It defines mechanisms to enable browser-to-browser communications without the need of either internal or external additional plugins, not dedicated hardware components.

While most browsers natively support features like media acquisition from input devices, transcoding and rendering, an external signaling server is needed to exchange control messages. It is needed to setup the communication sessions between peers. The signaling server structure is not defined by WebRTC and must be developed specifically to cope with the architecture where it will be deployed.

The Multiple-Media Manager central unit relies on WebSockets to establish a persistent connection between peers to allow real-time events exchange. All events are JSON textual messages with a specific format that can contain either a request or a response. A specific sequence of requests and responses is called session. The server provides two services that differ between messages formats and sequences (WebRTC signaling service, presence service).

The server is in charge of delivering each message to the proper receiver and discards all out of sequence messages. After the handshake termination all audio/video data is exchanged between the peers without other mediations by the Multiple-Media Manager server.

Often a direct connection is not possible due to the presence of a firewall and Network Address Translation (NAT) devices. The ICE (Interactive Connectivity Establishment) protocol is used by WebRTC to deal with this issue. The Multiple-Media Manager signaling service also delivers all the messages generated by ICE handshaking.
Multiple-Media Manager has a dedicated API that allows connection to various data sources. The next table details the connection API.

### Table 8. Multiple-Media Manager: Connection APIs.

<table>
<thead>
<tr>
<th>Multiple-Media Manager: Connection APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw format on named pipe</td>
<td>This Multiple-Media Manager acts as a named-pipe server receiving frames from source in one of the following formats: RGB24, BGR24, RGB32, BGR32, ARGB, ABGR, RGBA, BGRA, GRAY8, GRAY16. If required, metadata information also can be received before each frame with the format described in 4.2.2.</td>
</tr>
<tr>
<td>Raw format on socket</td>
<td>Same as previous, but the Multiple-Media Manager acts as a TCP socket server</td>
</tr>
<tr>
<td>Network streams</td>
<td>The Multiple-Media Manager acts as client receiving frames from both IP-enabled video camera and unicast or multicast RTSP/RTP streams. The Multiple-Media Manager support several video formats by integrating the FFmpeg libraries.</td>
</tr>
</tbody>
</table>

The Multiple-Media Manager also interacts with other components through the Link Smart middleware, publishing on MQTT topic various information in JSON format. Messages are detailed in the table below:

### Table 9. Multiple-Media Manager: MQTT APIs.

<table>
<thead>
<tr>
<th>Multiple-Media Manager MQTT APIs</th>
<th>Description</th>
</tr>
</thead>
</table>
| Status messages | These messages are sent to notify the status of each source. A source can have three statuses:  
  - Connected: the source is active and is sending data;  
  - Trying: the server is trying to connect with the source;  
  - Disconnected: the source is offline. |
| Stat message | This message is sent periodically and contains information about the frame rate of each source |
| Encoding and upload messages | This messages contains information about the status of both: video encoding process and video upload to the central unit after fall alarms |
| Singling and presence messages | This messages are sent to notify about WebRTC signalling events and peers presence status |

As described above, each message sent and received from the Multiple-Media Manager signaling service by a peer must have a specific JSON format that always contains the following information:
The following table reports the message API available at the current stage:

<table>
<thead>
<tr>
<th>Multiple-Media Manager: Signalling APIs</th>
<th>Description</th>
</tr>
</thead>
</table>
| Presence message                       | These messages are sent to notify the status of each peer. A peer can have three statuses:  
- online: the peer is available to start a call;  
- away: the peer, at the moment, is away from station;  
- offline: the peer is not available.  
These messages don’t need a specific target. The server broadcasts it to all connected peers. |
| Call message                           | These messages allow to exchange information between peers to:  
- invite a peer to a call;  
- accept or reject a call;  
- terminate a call;  
- codec negotiation (SDP);  
- network configuration (ICE);  
To initiate a call session the invite message must contain the unique identifier of the called peer; |

Figure 13. Multiple-Media Manager Component Diagram.

4.2.5 AR In-Factory Platform

This component has the responsibility to provide highly specialized services, maximizing the prerogatives of Augmented Reality, in order to support workers at their workplace in a more effective and appealing way. The services offered are distinguished in two specific groups: 1) support to employees in conducting specific operating procedures; 2) support to employees by
providing them advanced information and content. The AR Platform allows to provide these services for all main types of procedures required by the project (Training, Assembly, Maintenance), and for the various situations in which these procedures and information content may be of added value (standard and/or planned activities, management of emergency situations). The services of the first group make innovative instruments available to the end user which support the implementation of operative procedures. Following an approach both "on the job" (based on Augmented Reality capabilities), and in "simulated environment" (totally based on Virtual Reality), these tools guide users in their activity by presenting the step by step procedures interactively, by providing guidance on tools, materials, and components involved and to use. The presentation took place beneath the services offered by the Attractive Interface Layer, thus from innovative HMIs. In particular, such as the other architectural high level components, the Multi Modal HMI of the AR Platform should be considered a strictly integrated component of it, rather than an external independent component.

The services of the second group are designed instead to convey, within the visualization tools made available by the AR Platform, information and content both pre-packaged and generated at runtime by other components of the architecture (first among all those related to the Service Layer, obtained and/or constructed by services exposed from Facility Layer and from Middleware). The ecosystem of SatisFactory in fact, includes a wide variety of information and content that is important to notify the end users as a result of processes operating separately (e.g. alarm notifications, requests for collaboration, information from a supervisor, DSS results, etc.). By providing a specialized service addressed to convey information and specific content to the Multi Modal HMI of the AR Visualization Tools", the AR Platform allows to overcome this complexity and variety, ensuring not only a full compatibility with the current information, but anticipating the potential future evolution of the other components. This service, called “View Channel Service", provided by Middleware to all the other architectural components, allows them to provide any kind of information, messages and simple content to end users in an asynchronous way and with a specific view layout based on the declared priority.

From the point of view of its composition, the AR In-Factory Platform includes a certain number of end-use applications, that can be divided into:

(a) creation tools of operative procedures for the Augmented Reality: AR OP Creation Tool and AR OP "on-the-job" Creation Tool;

(b) tools for interactive visualization of procedures in the AR on several devices: AR OP Visualization Tool (GlassUPVersion), AR OP Visualization Tool (Mobile Version) which moreover expose services related to the delivery of advanced information and content indicated above.

The AR Creation Tools allow in particular, to describe operating procedures following a specific formalism (AR OP Model) and related file format (XML based). Then, these procedures can be packed in bundles, including all related resources (images, 3D models, etc.) and saved into CIDEM in order to be directly downloadable and usable by AR Visualization Tools.

Although not expressly stated in global architectural layout of SatisFactory, it should be emphasized that these tools, parallel to the integration with the Middleware component, directly support different physical devices (AR Glasses, Tablet, etc.) in order to obtain a real-time visualization, normally bound to applications of AR and VR.

As is evident, this component depends on and interacts in its work with various other components of the architecture. Below is shown a high-level architectural detail of the component discussed:
The following is a high level explanation of the main interfaces, both those made available and those which request other components.

**Table 11. AR In-Factory Platform Main APIs and Services.**

<table>
<thead>
<tr>
<th>AR In-Factory Platform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operative Procedure Management</td>
<td>This is a provided service. It allows an external component to exploit the high specialization of AR Viewers relative to the management and interactive visualization of operative procedures. Among the main features of this API are: (a) activation and loading of a specific operative procedure on the AR Visualization Tool with which the external component interacts; (b) activation and management of an integrated logging system, by which trace all the useful information accumulated during the progress of the procedure itself (e.g.: timestamps, events, interactions, etc.);</td>
</tr>
<tr>
<td>View Channel Management</td>
<td>This is a provided service. It allows the AR Visualization Tool to become the front-end to many specialized services supplied by higher level (Service Layer) and by lower-level (Decision Layer and Facility Layer), without knowing them, that is without the necessity to implement specific integration with this kind of components The service implements a specific paradigm for the submission of information based on the concept of &quot;View Channel&quot;. It is a way to deliver content specific, pre-packaged and/or</td>
</tr>
</tbody>
</table>
generated at runtime by the external components, which are displayed interactively using the HMIs made available by AR Visualization Tools. The service decouples the contents to display by way of presenting them to the end user. Among the main features of this service we find: (a) opening, management and closure of a specific View Channel; (b) sending a data packet within the View Channel and describing the content to be displayed interactively; the manageable contents depend by the ability of Middleware (Linksmart) to move information. Thus main contents that can be managed are: messages (text, audio), icons, images, etc.; while streamed data (e.g. video) can't be covered; (c) control of the running method linked to View Channel and synchronization with the external components.

<table>
<thead>
<tr>
<th>AR Operating Procedures &amp; Resources</th>
<th>This is a required interface. The CIDEM Component must provide the services through which saving and loading the data packets describing the operating procedures in the form that can be used by AR Tools and, at the same time, also provides all the resources associated with these procedures (e.g., 3D models, images, video, etc.).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Devices Management</td>
<td>This is a required interface. The Middleware Component must provide services to obtain data from the Smart Sensor Network component. The AR Platform expects that these data are not in “raw” form but, as required by the project, are already integrated and homogenized.</td>
</tr>
</tbody>
</table>

4.2.6 **CIDEM**

The CIDEM (Common Information Data Exchange Model) is responsible for storing the static and the real-time dynamic information acquired from various heterogeneous sources in the shop floor in a uniform way. All the shop floor information will be translated and presented in an understandable form, so as all SatisFactory components use them. It will contain information about the shop floor static information such as building architectural map, actors, procedures, equipment related to the project, installed sensors, Augmented Reality (AR) models that will be used, etc. Furthermore, it will include the dynamic aspects related to the shop floor, including events and context information (e.g. location, time, etc.), alerts, measurements, events related to re-adaptation and training activities, AR events, events related to gamification processes and social platform, etc. The CIDEM will mainly interact with LinkSmart middleware, which will collect all the information from the shop floor, translate it to a common vocabulary and store it to the CIDEM. So, all SatisFactory components will be able to store their output data (events, measurements, etc.) to the CIDEM through the middleware. Furthermore, CIDEM will be able to provide the historical data and information related to the shop floor to all SatisFactory components. This information will also be performed through the middleware. In other words, each SatisFactory component will send a data request event to the middleware, which in sequence will communicate with CIDEM, retrieve the data and send it back to the component that made the request. The high level functional architecture of CIDEM is depicted in the following figure:
Table 12. CIDEM Main APIs.

<table>
<thead>
<tr>
<th>Integrated DSS API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data &amp; metadata</td>
<td>This is a required interface. In order for the CIDEM to store all the shop floor information, it will be provided to middleware services so as to store raw data, measurements, events and metadata to the repository. This information will be based on well-defined vocabularies.</td>
</tr>
<tr>
<td>Historical data &amp; models</td>
<td>This is a required interface. In order for the CIDEM to send historical information to the SatisFactory components (after having request them from the middleware), it will provide services to the middleware to retrieve these data. This information will be based on well-defined vocabularies.</td>
</tr>
</tbody>
</table>

4.2.7 Operational Platform with Augmented Intelligence

The Operational Platform with Augmented Intelligence is responsible for providing real-time diagnostics and control actions to the operators at nominal conditions of shop floor operation. The platform communicates directly with the middleware component and the Decision Support System (through the middleware) in order to derive the optimum actions based on the current state of the machines and the worker’s behaviour. It is responsible for providing an overview of Key Performance Indicators (KPIs), personalized notes and tasks that are adjusted per worker. Additionally, the AR platform will integrate and present the Ontology Manager and the Collaborative tools through an easy to use and intuitive interface. The high level functional architecture of the Operational Platform with Augmented Intelligence is depicted in the following figure:
Figure 16. Operational Platform with Augmented Intelligence Component Diagram.

Table 13. Operational Platform with Augmented Intelligence Main APIs.

<table>
<thead>
<tr>
<th>Operational Platform with Augmented Intelligence API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data, metadata, historical data &amp; models</td>
<td>This is a required interface. The Operational Platform with Augmented Intelligence will request and retrieve all the needed historical and real-time data from the CIDEM through the middleware. All this information will be useful for the analysis of the shop floor operation.</td>
</tr>
<tr>
<td>Metadata</td>
<td>This is a required interface. All the information extracted from the platform (e.g. real-time diagnostics, control actions to the operators, etc.) will be sent for storage and further use by other components to the CIDEM through the middleware.</td>
</tr>
<tr>
<td>KPIs, Personalized notes &amp; tasks adjusted per worker</td>
<td>This is a required interface. All the KPIs, personalized notes and tasks adjusted per worker, as estimated and extracted by the platform, will be sent to the SatisFactory components and to the CIDEM for storage through the middleware.</td>
</tr>
</tbody>
</table>

4.2.8 Collaborative Tools

The Collaborative Tools incorporate the Social Interaction and Cooperation component. This provides a platform that workers can use to interact with their (remote) colleagues. It can be used to cooperate on work related issues, e.g., requesting for help for a particular problem or for sharing off-topic interests. It is designed for usage inside the shop floor, i.e. users will use the system when not sitting on their computers in the office environment. This requires the equipment of mobile or ubiquitous user interfaces which are elaborated within the adaptable user interfaces task. The final choice of the particular user interface will be made based on the specific requirement of the social interaction and cooperation use case.
At the current stage, the Social Interaction and Collaboration component is being realized through the Suggestions application. The Suggestions application uses the Input/Output API in order to communicate to the user interface and the “Connect to other SatisFactory Components” API in order to connect to the repository component, through the middleware.

Table 14. Collaborative Tools Main APIs.

<table>
<thead>
<tr>
<th>Collaborative Tools APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect to other SatisFactory Components</td>
<td>This is a required interface. The social collaboration platform needs to communicate with other components. These components are used to capture input for the social interaction and collaboration platform or to provide output, usually to the user interface that is viewed by the end user. The input can also come from that interface or it can be data that is imported or captured from external systems such as sensors or intra-factory systems. The middleware provides the necessary functionality for network connection and interoperability. Furthermore, the Event Manager of the Middleware is used for event-based data transmission.</td>
</tr>
<tr>
<td>Get Location</td>
<td>This is a required interface. When the social interaction and collaboration platform processes data that is based on the location of workers or when the input or output interface is dependent on the location of the workers, this location data is requested from the Localization Manager.</td>
</tr>
<tr>
<td>Input/Output</td>
<td>This is a required interface. It incorporates the interfaces between the social interaction and collaboration platform and the user interfaces. The adaptable user interfaces provide functionalities to present the output from the collaborative tools to the end user or to gather input from them.</td>
</tr>
</tbody>
</table>

4.2.9 Gamification Framework

The Gamification Framework is responsible for applying gamification in various aspects of Satisfactory. Its aim is to motivate users to perform better in the domain of every external system. Different external systems can use the Gamification APIs provided in order to participate to the global factory’s game. Each external system that needs to participate in the game has to implement the Gamification Management API. During deployment the external system is being registered to the gamification framework. Each game can consist of various tasks and for each task there has been predefined a number of points. Workers can take part in this game either individually or in the form of teams.
After the initialization of the connection and configuration of the game, the external system based on its internal business logic (allocate tasks / points) and user's behavior can add points to a user account (avatar) or team stored in the CIDEM component (through Gamification Repository API). These points are then accumulated into a grand total, which shows whether the workers have surpassed their previous score or not. The collective information is displayed on the Digital Display Screen through the middleware. The workers can also view their personal scores and achievements via a mobile app (Gamification App API).

The gamification framework consists of the following components along with the provided interfaces.

![Gamification Framework Component Diagram](image)

**Figure 18. Gamification Framework Component Diagram.**

<table>
<thead>
<tr>
<th>Gamification APIs</th>
<th>Framework</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamification Management API</td>
<td>Framework</td>
<td>This is a required interface. This API exposes functionality that allows the configuration of the global game including the deployment and management of external systems. Additionally this will provide functionality for the management of user / team accounts.</td>
</tr>
<tr>
<td>Gamification App API</td>
<td></td>
<td>This is a required interface. Through this API external systems provide information to users and teams about their status.</td>
</tr>
<tr>
<td>Gamification Repository API</td>
<td></td>
<td>This is a provided interface. The interface will provide access to CRUD services to the CIDEM for the gamification's entities.</td>
</tr>
</tbody>
</table>

### 4.2.10 Re-Adaptation Toolkit

The toolkit for the re-adaptation of existing facilities and human resource balancing is developed within Task 2.4 and further details will be reported in D2.4 [SatisFactory D2.4] due at M20. In the following a brief description is provided.

The Re-adaptation toolkit is a portable application that provides in a single point of aggregation, heterogeneous information coming from the shop floor. It is meant for supervisors and is mainly divided in two views. The first one represents live data from the shop floor level such as:

- visualize the gbXML map of the shop floor
- localize technicians based on their location
- visualize status of smart assembly stations
- provide live videos of connected locations from available sources
- provide notifications of events from the shop floor
- assist with contextual information when incident detection situations are triggered
The second one is a practical visualization for HR supervision of the work scheduling component integrated in the Integrated Decision Support System dedicated to workload balance.

Table 16. Re-adaptation Toolkit Main APIs.

<table>
<thead>
<tr>
<th>Re-adaptation Toolkit APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-adaptation toolkit Interface API</td>
<td>The interface will be detailed in D2.4 with punctual specifications. This API will allow portability and scalability of the component. In fact, it will detail the connection mechanisms between the two logically separated entities: the rendering and the backend.</td>
</tr>
<tr>
<td>Re-adaptation toolkit IDSS API</td>
<td>This is a required interface. To communicate with the IDSS.</td>
</tr>
<tr>
<td>Re-adaptation toolkit Repository API</td>
<td>This is a dedicated interface to communicate with the CIDEM Repository.</td>
</tr>
</tbody>
</table>

In the following figure a functional view of the re-adaptation toolkit and its interactions are depicted.

![Re-adaptation Toolkit Component Diagram](image)

More details about this component will be found in [SatisFactory D2.4].

4.2.11 Training and Educational Platform

This component has the responsibility to provide dedicated services to support the activities of training and education not only to workers and machinery operators but also to manufacturing process supervisor. The main feature of this platform is the ability to manage, in an innovative way, the creation of content for training and, especially, its use. The Training Platform in fact, optimizes its effectiveness by focusing on the technologies of Augmented Reality, on the exploitation of a new generation of HMI and on the adoption of an approach to training "on the job".

D2.1.2 - SatisFactory System Architecture ■ April 2016 ■ ISMB

SatisFactory project ■ GA #636302
This component is developed on top of AR In-Factory Platform, but is able to take advantage of the specialized services offered by several other members of the Facility Layer and of the Middleware. Thanks to the integration with the platform of AR, the Training component inherits all of its major powers such as: a) support to the description and presentation of procedures related to all areas required (Training, Manufacturing, Maintenance); b) support to the creation and use of standard procedures related both to common or planned activities, and to emergency situations; c) support to the creation of condition-based procedures; d) support to the functionalities offered both on-the-job and in a simulated environment; e) use of specific content perfectly integrated into HMIs inherited by AR Tools.

Beyond the functionality inherited from the platform of AR, this component can provide several highly specialized services, deemed necessary in order to increase the level of involvement of the trainees; they are: a) support to a bidirectional communication between the trainees and their supervisor (or, in any case, between the novice and the expert), in order to increase the passage of knowledge; b) support to a transfer of information and data between the tools made available to the trainees and their supervisor; c) implementation of features specifically related to the management of a training session: performance measurement, comparison with previous sessions, analysis of results and provision of indices and specific information, etc.

From the point of view of its composition, the Training Platform includes the following highly specialized tools for training: a) AR OP Visualization Training Tools (Mobile version and GlassUP version); b) Training Data Analytics Tool; AR Op Creation Tools (formally part of AR Platform but directly included into Training Platform).

Different instances of the visualization training tools have the opportunity to communicate with each other, by using collaboration tools services, allowing to provide the services described above and, specifically the bidirectional communication between trainees and supervisors. Finally, in this case as well it is useful to emphasize that the applications of training, as well as being able to interact via the Middleware with the Physical Layer, directly support physical devices, in order to maintain their maximum performance always.

Figure 20 shows the high level architectural detail of the component discussed, in which the connections with the other components are highlighted:
Figure 20. Training and Educational Platform Component Diagram.

The Training Platform does not expose specific interfaces to other components of the architecture. Instead it depends necessarily on interfaces requests to several other system components. The following is a brief breakdown of these interfaces.

Table 17. Training and Educational Platform main APIs.

<table>
<thead>
<tr>
<th>Training and Educational Platform APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operative Procedure Management</td>
<td>This is an interface requested to the AR In-Factory Platform. This API must allow access and exploit the highly specialized services related to the management and visualization of operative procedures, included in the platform of AR. Moreover, the same API should provide good support to the logging of procedures performed, containing all of the essential information on which the Training Platform builds its specialized services (e.g. benchmarking of achieved performance).</td>
</tr>
<tr>
<td>View Channel Management</td>
<td>This is an interface requested to the AR In-Factory Platform. This API must allow all content specifically designed for training to vehicular at runtime, towards the visualization tools made available by the AR platform.</td>
</tr>
<tr>
<td>AR Operating Procedures &amp; Resources</td>
<td>This is an interface requested to CIDEM. The latter must provide services through which you can save and load the data packets describing the training operating procedures and simultaneously all the resources associated with these procedures (e.g. 3D models, images, video).</td>
</tr>
<tr>
<td>Physical Devices Management</td>
<td>This interface is required to the Middleware. The latter</td>
</tr>
</tbody>
</table>
must provide services through which you can get data from the component Smart Sensor Network. The Training Platform is expects that this information is not provided in the "raw" form but, as per the requirements of the project, is already integrated and homogenized.

4.2.12 Multi Modal and Augmented HMIs

This component has the responsibility to provide the best user experience for all specialized applications developed within SatisFactory. The scope of this module is not only to provide as much as possible a uniform look-and-feel and optimized for different typology of considered devices, which takes into account the ergonomic issues (e.g. minimizing the cognitive effort for users), but also to reduce the need for the developers of the applications to re-invent user interaction solutions for similar situations. Thus resulting in a cost reduction and shorten delivery time. Even more, it tries to follow the most innovative approaches currently applied in the best known experiences, documented in literature, in this area.

Each software deliverable of the project, which has to do with a user, needs a proper and specific HMI. Nevertheless, in order to maximize the innovative approaches and technologies of SatisFactory, it is very important to uniform the HMIs to be developed by fitting them with the adopted physical device, the specific use cases and the chosen "attractive and gamification" approaches.

The key approach in this case is to provide a comprehensive specification for the requested HMIs, that each software applications and systems have to realize considering their specific implementation environment. Anyway, high specialized architectural components (e.g. AR In-Factory Platform) can share services to simplify the implementation of concrete HMIs for the demanding external components. The fulfillment of this requirement is in particular covered by the so called "View Channel Service" provided by the AR Platform.

Only for the practical purposes of this documentation (namely not introducing here the specific subdivision provided by the all the other components and/or applications developed in Satisfactory), it is useful to consider this component as a unique specialized module, strongly reusable and integrable into all specific applications and systems provided with SatisFactory. By doing this, it is possible here to describe it as a set of specific sub-components and APIs.

Hardware UI Component

It is the sub component directly related to the capabilities of the physical devices used to implement the HMIs for SatisFactory applications and systems. It is built in features provided by:

- AR Glasses (by GlassUP)
- Wearable Devices (by GlassUP and 3th parts)
- Mobile Devices (for instance Tablet, Smartphones etc.)
- Desktop PCs

It also includes management of the following aspects:

- Worker's Input:
  - Touch [Type and Press] (e.g. Touch Screen of a Table, keyboard)
  - Speech (e.g. Microphone)
  - Body [body movement, gesture] (e.g. Camera, Motion Tracking System)
- Output to Worker:
Multi Modal UI Component

It’s the sub component, implemented as a set of highly specialized software modules, which can support most of the common multimodal interfaces, by combining different hardware-provided input modes such as touch, speech, touch (tactile), gestures, head and body movements. The advantage of using these multi modal inputs is that the multiple modalities increase usability and provide the users with more possibilities to switch from one input mode to another that may be better suited to a particular task or setting. Complementarily the same component manages all output devices, by guaranteeing a specific content to be provided to the worker, fits the best channel possible (audio, visual, etc.).

GUI Component

This subcomponent implements specific graphical user interfaces, useful to support the designed HMIs, allowing to apply all innovative approaches of VR, AR and gamification requested by the project. It’s based on a Model-View pattern for a flexible structure in which the interface is independent from and indirectly linked to application functionality, so the GUI can be easily customized.

Specialized UI Components

They represent modules, designed and implemented as specialization of above shown UI components. Their presence is caused from the several innovative approaches and functionalities requested by the project. These components are directly built on top of other architectural ones.
(e.g. AR In-Factory Platform, Collaborative Tools, etc.). The most important Specialized UI components are:

- AR UI
- Collaboration/Communication UI
- Visual Analytics Module UI

![Diagram of Specialized UI Component](image)

**Figure 22. Specialized UI Component Diagram.**

The following is a high level explanation of the main interfaces provided by Multi Modal & Augmented HMI component.

<table>
<thead>
<tr>
<th>Multi Modal and Augmented HMI Main APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Channel Management</strong></td>
<td>This is a provided interface. It allows to obtain data acquired by physical devices and managed by Multi Modal UI, in order to handle them. From an application point of view, this API uncouples physical HMIs and specialized multi modal services from the application layer. The handling of GUI can directly call this API as well.</td>
</tr>
<tr>
<td><strong>Output Channel Management</strong></td>
<td>This is a provided interface. It allows to send data, generated by specialized architectural modules, to physical devices, passing through the management of Multi Modal UI. From an application point of view, this API uncouples physical HMIs and specialized multi modal services from the application layer. The handling of GUI can directly call this API as well.</td>
</tr>
</tbody>
</table>
5. MIDDLEWARE AS A SUPPORTING INFRASTRUCTURE

In SatisFactory, the LinkSmart Middleware will be used to integrate existing automation systems, sensor networks, and augmented reality devices with their solutions to be developed within the scope of the project. Below we present an introduction and architecture of the LinkSmart middleware.

5.1 INTRODUCTION TO THE LINKSMART MIDDLEWARE

The LinkSmart middleware has its origins in the FP6 IP Hydra [Hydra]. It was developed under the lead of Fraunhofer FIT. After the project ended, the resulting Hydra middleware was renamed to LinkSmart and it was released open source. From that time on, the middleware has been continuously enhanced by different international partners, among them FIT and ISMB.

The main purpose of LinkSmart is to give application developers a tool, which handles the heterogeneity of physical devices for them. As a result, developers can easily use web service interfaces for controlling any kind of network technology, such as Bluetooth, RF, ZigBee, RFID, Wi-Fi, etc. Further features of LinkSmart are secure peer-to-peer (P2P) communication, device and service discovery, and respective developer tools.

LinkSmart uses a service-oriented architecture (SOA) approach in order to offer support across platforms and network boundaries. This higher level generalizes core features, such as security, trust and reflective properties. LinkSmart has been evaluated in several applications in ubiquitous computing domains including eHealth and (Energy-aware) Smart Homes [Al-Akkad et al., 2009], [Eikerling et.al, 2009], [Jahn et al., 2010], [Reiners et al., 2009].

5.2 LINKSMART ARCHITECTURE

LinkSmart divides between devices, which can directly run the middleware, and those devices, which are too resource-constrained to run it. The latter ones are connected via a dedicated gateway device to the LinkSmart network. These gateways have a wired or wireless connection to the resource-constrained device. The gateways run Device Proxies, which handle the communication to the resource-constrained device and translate it to LinkSmart. From the service point of view, the communication with a proxy is identical to a communication with a native LS device. Devices which are not able to directly run LinkSmart can be further classified into D1, D2 and D3 devices.

Main components in LinkSmart are managers. A LinkSmart manager encapsulates a set of operations and data that realize a specific functionality. Figure 23 shows a typical LinkSmart deployment including components and interactions. It includes six LinkSmart connected devices: Three of them can directly run LinkSmart. The other three are connected to the same LinkSmart gateway.

Each connected device runs a Network Manager (NM), which registers all services belonging to the device. All communication between devices runs through the Network Managers. So, network services are accessed through their own Network Manager. The Network Managers are connected to Security Managers (SM) and Crypto Managers (CM), which secure, encrypt and decrypt the communication.
5.3 **Event Manager**

The Event Manager provides a means of communicating events between components. For this, it implements a publish/subscribe mechanism. In this approach, the Event Manager plays the role of the event broker. That is the central component which manages subscriptions, provides a method for publishing events and finally, it distributes events to subscriber components. This way, the Event Manager achieves a decoupling in space, time and synchronization. The high level interaction of LinkSmart Event Manager with various other components of the SatisFactory architecture is depicted in Figure 24.

![Event Manager Diagram](image-url)
Figure 25 shows the typical interaction with the Event Manager on a theoretical, realistic example from the SatisFactory domain. In this example, two components, namely the Localization Manager and the Glasses UI, need to react to location sensor data events. So, both subscribe at the Event Manager under the topics “localizationSensorData”. Every time the location sensor raises a new sensor data event, it publishes this event to the Event Manager. The Event Manager then takes care to notify all subscribers of this event type, in this case, the Localization Manager and the Glasses UI. These two components can then further process the data; for instance, they can display the data.

5.3.1 Event Aggregator

SatisFactory components on a physical layer are basically data sources to high level services. These data sources use LinkSmart for data propagation and this data is supposed to be stored into the CIDEM repository. The Event Aggregator is the component that subscribes to the Event Manager for such data and stores it into the CIDEM repository. The Device Managers speaks only MQTT protocol for communication and doesn’t need to rely on the REST interface of the CIDEM repository for data storage.
5.4 **DEVICE MANAGER**

The Device Manager (hereafter DM) will provide the instruments to simplify heterogeneous physical devices virtualization through a common interface and technology-agnostic access to the lower layer. These devices have different capabilities, communication technologies and features. A device virtualization is so needed to help the seamless integration of these devices with other SatisFactory components and features. More specifically, the need is to overcome the fragmentation of vertically-oriented closed systems, architectures and application areas and move towards open systems and platforms that support multiple applications. The DM will take care of the compliance with current and upcoming standards/reference architectures and semantics-enriched representations, like: OSGi Device Abstraction Layer (OSGi DA), ETSI M2M (ETSI M2M), OneM2M (OneM2M), HGi Device (HGiD), VORTO (VORTO), SAREF ontology (SAREF) and DogOnt ontology.
The Device Manager will be able to expose devices accordingly through the combination of 3 main possible, ontology based, representations of any device, which can be summarized as: services provided by a device, events generated by a device or variables which can be configured or can be accessed by the device user. The device representation will also describe the nature of the devices itself and the characteristics of the shared information (e.g.: type and unit of measured data).

At the current stage of the SatisFactory project, the device manager is not yet architecturally defined in detail. Full specifications will be available within Task 4.5 “Integration of social collaboration and sensing environments for enhancing collective awareness”, which starts at M13. At that stage, the devices requiring inclusion through the DM will be identified: this is relevant because the approach described before is not always applicable, due to some technological/physical limitation of devices or due to specific software devices’ interface/protocol/API constraints.

5.5 **LinkSmart Miniaturization**

One goal, which we aim at in SatisFactory, is the LinkSmart miniaturization. This has two purposes. Firstly, we want to make LinkSmart more lightweight, so that it can better run on low resource platforms and so that is can run on more of those platforms than now. Secondly, we want to facilitate the usage of LinkSmart for application developers and improve their user experience.

For this purpose, LinkSmart will replace various old, cumbersome and proprietary protocols by up-to-date, lightweight and free equivalents. Firstly, LinkSmart is originally using SOAP web services. We will exchange this by the far more easy to handle REST paradigm. Coming along with this, the XML-based WSDL as description language will be replaced by the lightweight JSON format. Based on the successful experience of adopting a web-based approach, LinkSmart uses REST as the core architectural principle and RESTful Web Services as the main approach to implement core middleware services. Middleware is one of the key elements of an ICT platform as it integrates...
heterogeneous IoT devices and ICT systems that the platform services build upon. The tasks of the middleware include the following:

- Providing modelling abstractions of real-world ICT data sources, IoT devices and sensor systems;
- Enabling search and discovery of these devices and their resources by applications and services;
- Providing unified APIs and protocols for historical and (near) real-time sensor data access.

The core middleware constitutes a set of middleware services that provide APIs for applications and services working with ICT data sources, such their discovery and data access. These services include:

- **Event Manager Service** provides a message bus for efficient asynchronous communication of sensor data streams implementing the publish/subscribe communication pattern. This service supersedes the old Event Manager that used a proprietary event format and the SOAP protocol. It has been replaced by the de facto standard protocol MQTT.

- **Resource Catalogue** exposes a lightweight JSON-based RESTful API and provides a registry of integrated ICT data sources, their basic meta-information and deployment configuration, including information on how their data can be accessed. The Device Managers are supposed to register the available devices and their resources so that applications and services can discover these devices and learn how to communicate with them.

- **Service Catalogue** has similar functionality as the described above Resource Catalogue with the difference that it provides a registry for middleware and SatisFactory services. Service Catalogue enables search and discovery of available platform services by attributes, such that applications and services can dynamically discover required services without prior configuration of the endpoints. In addition to that, the endpoint of the Service Catalogue API can be advertised on the network using DNS-SD, which enables automatic system discovery in the SatisFactory environment.

### 5.6 Device Manager Integration

As described above, enabling the communication with physical layer components or devices via open protocols commonly used by applications and services, as well as to actively populate the information about them is the task of Device Manager. More specifically, Device Managers are the software components providing the device integration in the LinkSmart Middleware and implementing the following functionality:

- Manage devices and their resources in the Resource Catalogue service
  - Publish registrations to (the) Resource Catalogue(s)
  - Continuously update these registrations
  - Remove the registrations on devices failures and graceful shutdown of the Device Manager
  - Optionally, expose local read-only Resource Catalogue API with managed resources

- Provide communication with devices over the network via standardized protocols
  - Expose APIs of devices/resources via open APIs and protocols (HTTP/REST, MQTT, etc)
- Implement native APIs/protocols of devices internally (if needed)

The introduced above functional specification of the Device Connector component leaves open its implementation details, as it largely depends on the device technology being integrated.

5.7 **LinkSmart Main APIs**

The following table contains the list of APIs which are exposed by the LinkSmart middleware. For each API, a brief description has been provided and major characteristics have been highlighted.

<table>
<thead>
<tr>
<th>Middleware API</th>
<th>Description</th>
<th>Required Interface</th>
<th>Provided Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device-Specific Sensor Access</strong></td>
<td>Each individual sensor type requires either a LinkSmart proxy or must run the Network Manager component to be accessible from the LinkSmart network. In both cases, the device specific input and output interfaces are made available as virtualized devices in the Device Manager.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Get Domain Model</strong></td>
<td>The Device Manager needs information about the available resources that need to be incorporated, such as communication technology and access methods, as well as their interconnection to one another.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Abstracted Device Access</strong></td>
<td>It represents the abstracted access to individual devices on a standardized virtualization layer. This incorporates services access and addressability. Furthermore, the interface consists of functionalities for the whole group of devices. Other SatisFactory components can list the available resources and search within the list.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Publish</strong></td>
<td>When SatisFactory components raise an event, they publish it at the Event Manager using this method. The Event Manager publishes the event to all the components which subscribed to the topic of the event. An event consists of a topic, the message payload and QoS parameters.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Subscribe</strong></td>
<td>SatisFactory components can subscribe to a topic for which they want to receive events. The Event Manager publishes an event to the subscribed components every time an event with the particular topic is raised.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td><strong>Notify</strong></td>
<td>Every component which has subscribed to an event topic must implement this interface. It is called by the Event Manager every time an event with the subscribed topic is raised.</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
<td>Result</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Get Semantic Device Data</td>
<td>The device information that the Device Manager gets from the Domain Models is augmented by semantic information about the resources provided by the Ontology Manager.</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Store</td>
<td>Every event, which is running through the Event Manager is stored at the Repository for later processing and logging. Event data is transferred from Event Manager to Repository through this interface.</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
6. **Deployment View**

The deployment view focuses on aspects of the system that are important after the system has been tested and is ready to go live. This view defines the physical environment in which the system is intended to run, including:

- Required hardware environment (e.g. processing nodes, network interconnections, etc.)
- Technical environment requirements for each node
- Mapping of software elements to the runtime environment
- Third-party software requirements
- Network requirements

The deployment view needs to document the required deployment environment of the SatisFactory platform, which depends on the pilot areas and their topology. This chapter provides a first overview, covering the known hardware requirements of the software modules and used tools.

A first component diagram indicating the deployment view of the SatisFactory components is depicted in Figure 27. The smart sensor network is comprised by a variety of heterogeneous sensors (bottom of the figure) including the preinstalled factory’s automation system, which provides information about the production activities and the status of the factory’s infrastructure. Furthermore, it is comprised by the UWB wearable and UWB anchor, as well as other sensors (such as gyrometers, accelerometers etc.) which will be used within the SatisFactory project. All these sensors/devices will be connected to the Middleware Device Manager through corresponding gateways, which will forward the information from the shop floor to the Middleware. The Device Manager, running in a dedicated PC named *IoT Gateway*, is equipped with all the necessary drivers, so as to understand and interpret the multi-sensorial information. The Device Manager is interconnected with the Middleware Event Manager, which could be located on the same PC (i.e., *IoT Gateway*) or even on another workstation as it is depicted in Figure 27. The integrated DSS is interconnected with the Middleware Device Manager in order to acquire data stemming from the smart sensor network. This information is subsequently processed, generating recommendations which are then provided to users or other subsystems in the form of Events through the Event Manager.

The Augmented Reality (AR) glasses could be thought of, in this stage, as another sensor, which will be connected to the Device Manager with a dedicated Radio Module GW and exchange messages/events with the rest of the SatisFactory components through it. At the time of writing, the type of AR glasses radio technology had not been selected yet. This activity will be done within Task 4.1.
Figure 27. SatisFactory Deployment View Architecture.
All SatisFactory components, which will be connected among each other with the Middleware Event Manager, could be installed either on the same PC workstation or distributed to a number of PCs. Figure 27 illustrates the case where the SatisFactory components are distributed to various PC workstations, e.g. one for the integrated DSS, one for the Ontology Manager, Collaborative tools, Operational Platform with Augmented Intelligence, Repository, Localization manager, Multimedia manager, Gesture content recognition manager, AR In-Factory platform and the Training and Educational platform. All these SatisFactory components compose the cloud-based SatisFactory infrastructure and will be interconnected among each other with a dedicated or not intranet, which could be either wired, wireless or even a combination of both.

It is worth mentioning that HMIs are directly connected to the Device Manager and exchange messages to the SatisFactory components through it. Finally, cameras, which will be used within SatisFactory, will be handled as special devices. Since the data produced by cameras is huge, Middleware and Repository are not able to handle them. So, they will be connected directly with the components that will need their information, and only the metadata (e.g. incidents, locations, etc.) will be forwarded to the middleware and to the repository.

6.1 SatisFactory Components Hardware Requirements

Table 20 lists the preliminary HW requirements concerning the main SatisFactory components.

<table>
<thead>
<tr>
<th>Name</th>
<th>Support</th>
<th>Hardware Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>FIT</td>
<td>A PC workstation will be needed with the following minimum requirements:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- i5 CPU is preferable;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- at least 4GB RAM;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 500GB hard disk;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Wi-Fi connection;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ethernet connection;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Windows 7 operating system.</td>
</tr>
<tr>
<td>Ontology Manager</td>
<td>EPFL</td>
<td>Minimum requirements of the workstation:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Supported Linux Distributions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CentOS 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CentOS 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ubuntu 14.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PHP 5.4 or higher</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 64 Bit Operating System</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Access to internet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 5 GB of disk space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 GB of RAM</td>
</tr>
<tr>
<td>Collaborative Tools</td>
<td>FIT</td>
<td>See Middleware requirements.</td>
</tr>
<tr>
<td>Integrated DSS</td>
<td>ABE</td>
<td>A PC workstation will be needed with the following minimum requirements:</td>
</tr>
</tbody>
</table>

D2.1.2 - SatisFactory System Architecture ■ April 2016 ■ ISMB
SatisFactory project ■ GA #636302
| Platform (Creation Tools) | REGOLA | **AR In-Factory Platform**
|---------------------------|--------|---------------------------------|
|                           |        | A PC Workstation will be needed with the following minimum requirements:
|                           |        | - i5 CPU (last Generation);
|                           |        | - 4GB RAM;
|                           |        | - 500GB hard disk;
|                           |        | - NVidia GPU GTX 650 2GB
|                           |        | - Wi-Fi connection;
|                           |        | - Ethernet connection;
|                           |        | - Windows 7 operating system.

| Platform (Visualization Tools – Mobile version) | REGOLA | **AR In-Factory Platform**
|------------------------------------------------|--------|---------------------------------|
|                                                 |        | Tablet with the following minimum requirements:
|                                                 |        | - QUAD-CORE arm Cortex-A9 (set ARMv7)
|                                                 |        | - 3GB RAM
|                                                 |        | - 16GB ROM
|                                                 |        | - Camera Front 8MP
|                                                 |        | - Wi-Fi connection;
|                                                 |        | - Bluetooth connection
|                                                 |        | - USB connection
|                                                 |        | - Display 10”
|                                                 |        | - Android 4.4 (KitKat) operating system

| Platform (Visualization Tools – GlassUP version) | REGOLA | Definition of GlassUP hardware requirements is still in progress.

| **CIDEM/Repository** | CERTH | See Middleware requirements.

| Operational Platform with Augmented Intelligence | CERTH | See Integrated DSS requirements.

| Training Educational Platform (Creation Tools) | REGOLA | See AR In-Factory Platform – (Creation Tools) requirements.

| Training | REGOLA | See AR In-Factory Platform – |
### Educational Platform (Visualization Tools – Mobile Version)

<table>
<thead>
<tr>
<th>Training Educational Platform (Visualization Tools – GlassUP version)</th>
<th>REGOLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>See AR In-Factory Platform – (Visualization Tools – GlassUP Version) requirements.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multi modal &amp; Augmented HMIs and AR devices</th>
<th>GlassUp (AR glasses)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Wi-Fi connection;</td>
<td></td>
</tr>
<tr>
<td>• Wired Box connected to the glasses for batteries and UI with commands (provided by GlassUp)</td>
<td></td>
</tr>
<tr>
<td>• USB connection for direct connections to third devices (PCs, tablets, smartphones, other)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localization Manager, Multi-Media Manager, Gesture &amp; Content Recognition Manager</th>
<th>ISMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>A PC workstation will be needed with the following minimum requirements:</td>
<td></td>
</tr>
<tr>
<td>• i7 CPU is preferable;</td>
<td></td>
</tr>
<tr>
<td>• at least 8GB RAM;</td>
<td></td>
</tr>
<tr>
<td>• 500 GB hard disk;</td>
<td></td>
</tr>
<tr>
<td>• Wi-Fi connection;</td>
<td></td>
</tr>
<tr>
<td>• Ethernet connection;</td>
<td></td>
</tr>
<tr>
<td>• USB connection;</td>
<td></td>
</tr>
</tbody>
</table>

### 6.2 Existing Software and Hardware Requirements

#### 6.2.1 Middleware

The LinkSmart middleware does not have any specific hardware requirements. A basic infrastructure of the core services (Resource Catalogue, Service Catalogue, Event Manager, and Device Manager) can be set up on one or more state-of-the-art workstations. Recommended requirements for such workstations are: i5 CPU, at least 4 GB RAM and 500GB hard disk.

#### 6.2.2 Human Tracking Component

Human tracking components are comprised by a depth camera network interconnected with an aggregated system, which is able to acquire all the detected information from the cameras. Each camera should be connected to a computer with hardware characteristics similar to the integrated DSS characteristics presented above (i5 CPU is preferable, at least 4GB RAM, 500 GB hard disk, Wi-Fi connection, Ethernet connection, USB connection, Windows 7 operating system). Furthermore, all these computers should be connected to a server with similar characteristics either wired or wireless (depending on the pilot area). Of course, all PCs should support this connection protocol.
This connection could be dedicated or not, but the server should be also connected with the rest of the SatisFactory components, in order to exchange messages and events with the middleware.

6.2.3 AR In-Factory Platform Component

In order to accomplish all advanced services that it has in charge, the AR In-Factory Platform needs to have all the necessary data by directly asking them from the Middleware. Thus it is strongly uncoupled from specific hardware requirements. The real dependencies on physical devices are instead associated to the specialized components that are in charge of the implementation of specific services.

To know the software and hardware requirements for the services provided by the other specialized components, see the paragraph related to them. Here a list of all main services from which AR In-Factory Platform needs data:

- Gesture Recognition
- Object Recognition
- Frame Acquisition
- Video Acquisition
- Smart Sensor Value Evaluation
7. INFORMATION VIEW

7.1 APPLICATION DOMAIN MODEL

The information view is based on the abstraction of the shop floor in terms of semantic IoT resources corresponding to “virtual entities” in the IoT Architecture Reference Model (ARM) [IoT-A D1.2]. On top of that, we added the required classes and interconnections supporting the SatisFactory applications.

Due to the user-centric concept of the project, in this first iteration of the architecture we have provided an initial overview of the SatisFactory domain model focused on the training feature of the platform, see Figure 28. However, it can be extended to support other features such as collaboration, safety and re-adaptation, but this has been left for the other two subsequent iterations.

As it can be observed from Figure 28, each physical entity, which is represented by the IoT device, is extended by sensors and devices that will be deployed in the shop floor. Each sensor and device hosts some resources that can be accessed through IoT services. The resources are software components that provide information about physical entities or enable the controlling of devices. Physical entities (i.e. the IoT devices) are represented in the digital world via a virtual entity.

Note that virtual entities are associated to a single IoT device that they represent. While ideally there is only one IoT device for each virtual entity, it is possible that the same IoT device can be associated to several virtual entities. Each virtual entity must have one and only one identifier (ID) that identifies the represented object.

Focusing on the upper part of the domain model, each user (e.g. actor, machine, and tool) can interact with IoT devices and invoke the associated services.
Figure 28. Example of SatisFactory Application Domain Model.
7.2 **Overview of Information Flow**

Figure 29. SatisFactory Information Flow.
Figure 29 depicts the information flow representing the data exchanged among the main components of the SatisFactory platform. Note that sequence diagrams, which have been drawn on the basis of the use cases [SatisFactory D1.2] identified within WP1, are presented in section 9 of this document.

A brief explanation of each data is reported as follows.

- **VisualizationData**: Data to be presented to the end user.
- **HMI-InputData**: GUI-collected data, such as touch data (type & press) or gesture data (body or hand motion) or speech data (microphone).
- **HMI-OutputData**: Text, image or sounds data.
- **RawData**: Raw data coming from sensors, cameras, etc. deployed in the shop floor
- **GamifiedData**: Contains the necessary data needed for visualizing results and info to the user in a pleasant and comfortable way.
- **SuggestionsData**: Suggestions data coming from workers
- **DSSEvent**: Contains data extracted by the integrated DSS component.
- **TrainingNotification**: A notification about the need of a new training. Can refer either to a foreman (to confirm the training) or to a trainee (to start the training).
- **TrainingData**: Data from the repository related to training (also the workplace practice), such as texts, images, videos. Also contains Training Analytics Data, Collaborative Data and Channeled Information (by AR-Platform).
- **TrainEvaluationResults**: Results related to the training evaluation procedure, such as whether the trainee succeeded or not, indicating what are the failures.
- **AR-TrainingProceduresData**: Training Procedure Description and Training Procedures Resources (e.g. 3D Models, Images, Videos, Motion Data, etc.).
- **TrainingSessionsData**: Sessions Data, Analytics Data, Results Data (retrieve or store).
- **CamerasData & LegacySensorsData**: Data from cameras and legacy sensors.
- **AR-OperatingProcedures**: Operating Procedures Description and Operating Procedures Resources (e.g. 3D Models, Images, Videos, Motion Data, etc.)
- **OperatingProceduresNotification**: A notification about operating procedures.
- **OperatingProceduresData**: Data from the repository related to operating procedures, such as texts, images, videos. Also contains Operating Procedures Analytics Data, Collaborative Data and Channeled Information (by AR-Platform)
- **DepthImages**: Contains the depth images acquired by the depth sensor network.
- **OccupantFlows**: Contains data about the flows/ trajectories of the occupants in the monitored area.
- **ThermalImages**: Contains the thermal images acquired by the thermal sensor network.
- **ThermalEvent**: Contains data about events related to areas and equipment whose temperature has been suddenly and abnormally increased.
- **IncidentEvent**: Contains data about the incident events that have already happened or potentially will happen in the area covered by the depth and thermal sensor network.
- **ShopFloorInfo**: Contains the data about the shop-floor, such as building geometry, procedures, actors, sensors, etc.
- **ProductionInfo**: Contains data about the production operations in the shop-floor.
- **ShopFloorData**: Contains data about the shop-floor.
- **ShopFloorEvent**: Contains data about the events that are recorded from the multi-sensorial network installed in the shop-floor.
- **ShopFloorOverview**: Overview of KPIs related to the shop-floor operation.
- **PersonalizationResults**: Personalized notes and tasks that are adjusted per worker, based on the evaluation of the current status of the shop-floor operation.
- **ActionData / RecommendationData**: Data concerning actions or recommendations on actions.
- **StreamingPointInfo**: The request to acquire streams from a specific areaID or cameraID. The device manager is the component that maps the cameraID (or the areaID) with the end point for streaming (generally a URI).
- **ContextData**: Data coming from Ontology Manager.
- **UWBLocationData**: Data coming from UWB devices.
- **geoFencingEvent**: Virtual geofencing (e.g. for avoiding human presence on unauthorized paths and areas).
- **SafetyEvent**: Safety-related events (e.g. automatic detection of personnel not wearing required safety equipment).
- **SafetyData**: Safety-related data (e.g. data related to an incident or to a human presence in not authorized area).
- **View Channeled Data**: Represent all kind of data that can be conveyed to AR In-Factory Platform at runtime and asynchronously visualized inside of the AR Tools HMI by following a specific priority (e.g. DSS notifications, Collaboration Tools data, etc.)
8. Perspectives

8.1 Satisfaction Perspective

The SatisFactory project approach, to improve operator satisfaction, is based on the design of user centred and value-added applications which increase the attractiveness of the working environment as well as the quality of the working conditions.

In order to properly involve both end-users and stakeholders in the evaluation process, SatisFactory solutions will be simultaneously evaluated by all involved actors. This will be possible through the social interaction platform, which will support multimedia enhanced interactions among employees and will encapsulate the context-aware knowledge schemas (section 4.2.3) available through the middleware. A set of collaborative tools (section 4.2.8) will be deployed leveraging gamification techniques. These will, for example, enable employees to earn rewards and compete as groups, enhancing the overall user experience and driving greater participation and engagement through a friendly game competition. In the same scenario, a digital add-on system will be integrated with both panels and mobile devices (section 4.2.12), providing to operators, information on the basis of back-end elaborated task scheduling and productivity benchmarking. The SatisFactory platform will also leverage prompt abnormal events management (section 5.3), as well as robust and reliable wireless connectivity (section 4.2.1) for real-time services in hostile environment, for the benefit of the users’ efficiency and feeling of satisfaction towards the production environment. Finally, the usage of innovative interactive platform (section 4.2.5), in combination with all the previously mentioned actions, shall positively stimulate user gratification and collaboration between employees, further improving the effectiveness and the acceptability of the platform.

Satisfactory will introduce the use of the Augmented Reality in a structured way, focusing primarily on the creation of a “Development Framework” through which applications will be developed to support the production environment and training activities that are in scope of the project. This framework will introduce the adoption of a development approach, oriented to increase the attractiveness of work environments and employees’ satisfaction. The AR framework will be based on the development of a set of specialized tools, divided in two categories: the Creation Tool, specifically designed to describe optimized contents for AR (3D models, animations, texts, images, etc.), and the Visualization Tool that will act as an enabler to visualize the created contents on a wide range of devices (i.e. mobile and wearable devices). Besides being used as standalone applications, designed to directly support assembly and maintenance activities (AR In-Factory Platform), the tool will also act as a framework to develop more specialized applications on the top of it. These applications will be mainly related to the Operational Platform with Augmented Intelligence, Training & Educational Environment and Collaborative Tools.

The real challenge taken up by SatisFactory will be to take into account difficulties emerged in recent years correlated to the application of AR technologies to the industrial environment. In fact, if the number of AR applications for marketing and entertainment is now so high that it is impossible to be counted them, in the industrial field the adoption of AR has quickly proved its limits, mainly due to the difficulty of designing effective HMIs. In fact, only in a real working environment, especially by analysing the physical and behavioural constraints of the workplace, it is possible to exploit the most out of these innovative technologies. From this point of view, the project will aim in particular to obtain the greatest possible benefits from the use of AR glasses (GlassUP) and
wearable devices associated with them, through the design and implementation of advanced multi-modal HMIs (touch, speech, display, gesture recognition). Finally, the presentation of the information in the most attractive and effective way will be a challenge for the AR applications, alongside the minimization of physical interactions required by the user. Therefore, the satisfaction of final operators will not only be achieved with the use of cutting-edge technological settings, but also maximizing synergies between them.

8.2 **SAFETY PERSPECTIVE**

The SatisFactory project takes care of innovative instruments supporting safety and ergonomics of the working areas and of the single worker. One of these instruments is the context recognition module based on localization. The location information can be obtained by the data fusion of information coming from heterogeneous sensors (e.g., UWB and inertial sensors) as well as privacy preserving cameras. For this purpose, the context manager interacts with enabling modules such as the Localization Manager, the Multiple-Media Manager and the Gesture & Content Recognition Manager. These modules will act as enablers in order to deploy safety enhanced procedures at the shop floor level.

The Localization Manager will provide workers’ position leveraging Ultra Wide Band (UWB) wearable devices and related indoor localization algorithms for real time indoor localization. Other localization schemes are based also on the processing of video streams coming from privacy preserving cameras. The workers’ localization will be used to enhance safety at the shop floor level, providing features such as virtual geofencing and its information will act as an enabler to create the data fusion infrastructure model that the SatisFactory ecosystem is aiming to deploy.

The Multiple-Media Manager will programmatically enrich localization data with visual information, especially to early trigger the incidents detection. Also other devices, like thermal cameras, will be considered as possible, non-intrusive privacy preserving instruments, in order to provide appropriate sources of data concerning workers’ behaviours. Many operations will be performed through the real time analysis of the aforementioned data, preventing workers from going into potential dangerous areas (i.e. virtual geofencing), incorrect procedures or any other danger. The module in charge of handling these situations is the Gesture & Content Recognition Manager. It will analyse data coming from the Multiple-Media Manager in order to guarantee a fast response and trigger safety mechanisms in reaction of safety related events, propagating ad-hoc messages to the Event Manager.

On the top of that, the context aware detection engine will provide two categories of safety approaches, namely proactive and reactive. The proactive actions can detect potential incident events (through real-time data analysis), while the reactive ones can detect incidents when they occur and as a consequence the corresponding predefined countermeasures are triggered.

Just to provide an example, a proactive measure can detect and alert a risky condition that might occur when a worker enters a forbidden/dangerous area. Therefore, this procedure is based on the location of workers, the geometry of the plant and the geometry of the forbidden areas. The geometry of the forbidden areas can be set by the safety manager and changed at any time. However, some dangerous area can be automatically set by the Context Manager module on the basis of particular events that may happen on the shop floor. For instance, fire or explosions are detected by sensors whose positions are known. As a consequence, a dangerous area is automatically created on the basis of those sensors data and a related alarm or escape procedures will be activated. The Context Manager will
leverage on the Event Manager in order to simplify and harmonize the sharing of the safety related information among the SatisFactory platform.

Finally, the attention given to the security mechanisms, which is adopted by the intelligent IoT infrastructure over wireless networks, will ensure the integrity of the collected information and thereby guarantee accurate identification of risky scenarios which could compromise workers’ safety.

8.3 **Scalability Perspective**

8.3.1 **Event Manager**

The LinkSmart middleware is capable of operating in P2P environments and one such example of deployment has been demonstrated in the ALMANAC platform [ALMANAC] where real-time data from heterogeneous sensors and actuators is collected and processed to support Smart City processes. In SatisFactory, the Event Manager is used for two different communication patterns. The first one is called client-to-centre pattern where psychical layer components send messages to high-level layers. In this pattern, the clients do not typically need to send messages to each other. The second pattern is a client-to-client pattern, where components of high-level layers (Facilities, Services etc.) send messages to each other. Given this communication scenario and number of components, SatisFactory deployment is considered to be relatively small and the Event Manager can easily handle communication among SatisFactory components. However, the Event Manager can also employ a “bridging” technique by forming a cluster of Event Managers to handle very large number of connected clients and messages.

8.3.2 **Ontology Manager**

The overall philosophy in architecting the Ontology Manager (OM) is to provide a Web-based, scalable framework for integrating data and content from a variety of sources. The OM corresponds to what is known as a web-oriented architecture (WOA).

A WOA builds on aspects of many of the largest properties on the web, with proven scalability and extensibility. As used in OM, these proven web aspects are enhanced by adhering to open standards from the W3C (World Wide Web Consortium) in the areas of semantic technologies and vocabularies.

The OM provides a standardized content storage and management environment that is web-accessible, scalable and distributed. The content that can be hosted within the OM includes documents (unstructured data), metadata (semi-structured data), conventional database information (structured data) and multimedia metadata. While this content can exist in multiple native formats in the wild, it is converted to the common RDF format that enables the development of standard (“canonical”) tools and operations to act upon this content.

8.3.3 **Decision Support System**

The Integrated Decision Support System will incorporate and implement a number of different algorithms that will combine sensor data from various sources, context-aware knowledge, production models and events and will finally produce a set of recommendations that may include actions over maintenance operations and schedules or worker activities, production re-adaptation, human resources workload management. The total architecture of the Integrated DSS (presented in section 4.2.2) allows handling increased workloads in order to provide the desired level of scalability by using both scale-up and scale-out techniques. Scale-up is the technique of increasing the
available resources of the node that hosts the software application. The implemented algorithms being both computationally- and memory-intensive, the number of available CPU-cores can be increased and the main system memory expanded up to the physical limits of the HW, if a deterioration of response times is detected. This action will radically benefit the overall performance of the system and finally provide an acceptable level of scalability. If the limits of the HW are reached, then the system architecture, which is mainly service oriented, will provide us the ability to scale-out, meaning that we can add a number of new nodes hosting the services. Each new event or incident detected could trigger the execution of the appropriate recommendations providing the algorithm in the less loaded node, with the help of a load balancer. This scale-out technique is feasible, due to the fact that the system does not need to internally hold the state of each request, meaning that subsequent requests for recommendations from the same source can be handled by different nodes. The result of the scale-out functionality will raise the scalability of the system in levels higher than those anticipated for the SatisFactory project.

8.4 **COMMUNICATION RELIABILITY PERSPECTIVE**

The Wireless Sensor and Actuator Networks (WSANs) provides the SatisFactory framework with an interface to the physical world. This system is expected to provide monitoring and control functions related to manufacturing environment. It will generate, collect and provide data related to people, machines and environment to the SatisFactory core components where it can be processed and exploited to enable context-aware functions. Typically, manufacturing environments are considered harsh for radio propagation, due to a possibly large number of signal sources as well as the dense deployment of metallic objects. Moreover, different manufacturing scenarios and machinery operate under a variety of requirements with respect to the type and amount of collected data, and therefore, the WSANs are required to support these scenarios with the expected level of performance in order to adapt to the constraints derived from a wide range of applications. In this context, the dependability of the WSAN is strongly related to the dependability of its communication links.

Accordingly, SatisFactory WSANs are equipped with multi-radio communication capabilities, based on short-range wireless technologies such as IEEE 802.11 Wi-Fi, IEEE 802.15.1 Bluetooth, 802.15.4a UWB, and IEEE 802.15.4 6LoWPAN. Due to their different physical layer characteristics, complementary utilization of heterogeneous wireless technologies can suppress the weaknesses of individual radios, and thereby ensure a higher level of robustness. In addition, this would improve the WSAN flexibility and ability to support different application requirements in terms of the offered bandwidth, transmission latency, etc.

SatisFactory WSANs will feature multi-radio interface management leveraging a cognitive approach, by retrieving and monitoring the networks’ status and adaptively adjusting their configuration parameters. In addition, self-configuration and self-healing features will be supported by intelligent mechanisms to extend network availability and maintain end-to-end connectivity. Moreover, data reliability and integrity, in terms of correctness and timeliness, will be ensured exploiting cooperative mechanisms and secure communications.
9. TECHNICAL USE CASES INSTANTIATION

This section details some selected use cases by means of simple sequence diagrams in order to clarify how the SatisFactory platform behaves to deliver some specific features of interest. In particular, 10 UCs out of 23 have been chosen from D1.2 [SatisFactory D1.2], representing application scenarios such as:

- Reconfiguration of process flows and actions for flexible redesign of production procedures
- Recognition of incidents and path optimization for workers movement on the shop floor
- Preventive maintenance management system
- Manual assembly operations support
- Training platform for production process
- Online supervision of the operation and workforce resources of pilot plants
- Repair or restore an electromechanical malfunction

The UCs described by the sequence diagrams have been selected aiming to include all main components of the SatisFactory platform. The sequence diagrams have been refined taking into account the updated requirements and UCs from WP1 as well as inputs from technical WPs such as WP2, WP3 and WP4.

The following subsections present the sequence diagrams for the 10 selected UCs. The descriptions of the UCs have not been included, as they have been already presented in D1.2 [SatisFactory D1.2].
9.1 **UC1.2: STORAGE OF THE SHOP FLOOR INFORMATION AND DATA FROM THE MULTI-SENSORIAL NETWORKS**

Figure 30. UC1.2 Sequence Diagram.
9.2 **UC1.3: Analysis of Real-time and Historical Info from the Shop Floor**

![UC1.3 Sequence Diagram](image)

Figure 31. UC1.3 Sequence Diagram.
9.3 **UC2.1: IN-FACtORY TRAINING AND SUPPORT OF WORKERS USING A FLEXIBLE LEARNING PLATFORM**

![Sequence Diagram](image)

Figure 32. UC2.1 Sequence Diagram.
9.4 **UC2.2: Validation of Training Actions Performed at the Shop Floor**

![UC2.2 Sequence Diagram]

Figure 33. UC2.2 Sequence Diagram.
9.5 UC3.2: INCIDENT IDENTIFICATION BASED ON DYNAMIC EVOLVING OPERATIONS

Figure 34. UC3.2 Sequence Diagram.
9.6 **UC4.1: PROVIDE MAINTENANCE WORK PLANS AND ACTIONS RELATED TO HUMAN-CENTRIC ACTIVITIES**

The sequence diagram is reported in the following page.
9.7 **UC4.3: Monitoring and Decision Support of Operations & Maintenance Procedures**

The sequence diagram is reported in the following page.

![Sequence Diagram](image-url)
9.8 **UC5.2: PLATFORM FOR SUGGESTIONS FOR IMPROVEMENT**

![Sequence Diagram](image)

**Figure 37. UC5.2 Sequence Diagram.**
Figure 38. UC5.3 Sequence Diagram.
9.10 **UC6.3: Knowledge Sharing Among Workers Based on Advanced Reasoning**

**Figure 39. UC6.3 Sequence Diagram.**
10. Conclusions

This document included two main contributions from Task 2.1, the first one concerns the technology exploration, where both relevant EU funded projects and regional ones were analysed, while the second one concerns the updated version of the SatisFactory architecture. We remind that a final version (third iteration) of the architecture is due at M27.

The first version of the architecture, which was available at M8, was defined by the following three consecutive steps. In the first one, called technology exploration, relevant projects and technologies were analysed in order to identify critical aspects for the SatisFactory architecture that were taken into account in the next steps. In the second one, called bottom-up phase, a number of components and subcomponents, belonging to the set of partners’ assets, were identified and considered necessary for the SatisFactory platform. In addition, missing components and corresponding functionalities were identified as well. Overall 11 main components and 38 subcomponents were collected. Each responsible partner presented the components’ internal architecture, functionalities, and interaction with other main components. Following this approach, the SatisFactory architecture was defined and documented in three different views (namely, functional view, deployment view and information view) according to the standard IEEE 1471 “Recommended Practice for Architectural Description for Software-intensive Systems” [IEEE 1471, 2000]. The third step, called top-down phase, was focused on each components role and their interaction. In particular, 10 use cases out of 23, identified within Task 1.3 [SatisFactory D1.2], were selected to build the related sequence diagrams by considering the identified components. This has allowed emphasizing the services each component would provide, and sketching the interactions between them. Architectural perspectives were discussed to address quality properties and cross-cutting concerns such as satisfaction, safety and scalability.

The second version of the architecture was defined in the period M15-M16. In particular, the overall architecture was carefully reviewed by taking into account inputs from the technical work packages (i.e., WP3, WP4 and WP5), the updated requirements from T1.1 (available at M14) and the updated business scenarios (BSCs) and use cases (UCs) from T1.2 (available at M15). The updated version of this document contains a revision of components descriptions as well as an update of the overall SatisFactory architectural viewpoints (Functional Viewpoint, Deployment viewpoint and Information viewpoint).
REFERENCES


[Adapt4EE D2.1] D. Adapt4EE D2.1, ”D2.1 Adapt4EE Conceptual Architecture.”


[SatisFactory D1.1] SatisFactory consortium, “User group definitions, end-user needs, requirement analysis and deployment guidelines”, delivered at M4, April 2015.


[SatisFactory D2.4] SatisFactory consortium, “HR Workload Management toolkit” to be delivered at M20, August 2016.


[SatisFactory D4.1] SatisFactory consortium, “Development of intelligent IoT-infrastructure for the smart-sensor network” to be delivered at M20, August 2016.


ANNEX 1 – SatisFactory Relevant Projects

A1.1 FACTS4WORKERS

Production is moving constantly away from European high-wage countries to so-called “best-cost” countries or to locations with low energy cost. To fight this trend the European industry is challenged to develop intelligent added-value concepts for the field of production.

Over the idea “Improved training and increase of investment in factories will help bring Europe new and better jobs”; FACTS4WORKERS puts the worker into the centre of future-oriented production concepts in order to render manufacturing jobs more attractive and help Europe to become more competitive [Vehicle et al. 2014].

Goal of the project:

The goals of the project, address to three different fields: Research, Smart Factory and The human component; they are described below [Vehicle et al. 2014].

Research:
The project initiates a new industrial era, which is characterized by the so-called “Smart Factory”. The “Smart Workers” in those production sites are ideally supported by information and communication technology in order to improve the manufacturing process regarding flexibility, efficiency, and reliability. This results in a local benefit in competition, and (central) European production locations can be secured in the long term.

Smart Factory:
In a “Smart Factory”, the production site of the future, the focus lies on the worker as the most flexible element involved in the manufacturing process. The worker becomes a “production knowledge worker” and is supported by optimized information and communication technology, self-learning working environment, and in-situ learning while operating the machinery. The digitalisation is not just to factories; it will affect entire added value networks. This is achieved via systems consisting of various components (IT, software, mechanical parts, etc.) that communicate via the internet or other means of communication.

The human component:
In terms of “Smart Factory”, there also has to be a focus on the role of the worker as the human component and key factor in the manufacturing process. Here there is a new concept, “knowledge work”, which is defined by an entirely new, complex, and autonomous work environment. Furthermore, “Smart Workers” develop new ways of continuous improvement of knowledge exchange on their own at the work place. The objective is to understand, how people work and learn, how they interact with new technologies, and how they can create an added value to the industry by working at an attractive and demanding work place. The answers to these questions are the key to successful and human-centred solutions of information and communication strategies within manufacturing processes. Increasing satisfaction and motivation, the productivity will increase by 10 per cent.

Related architectures:
The overall architecture is shown in Figure 40 below.

![Overall FACT4WORKERS Architecture](image)

The proposed solution is a worker-centric smart factory, satisfying the workers’ goals. FACTS4WORKERS develops a modular smart factory infrastructure, interlinking a number of building blocks, described as follows [Facts4Workers]:

- **SmartWorker-Centric HCI/HMI building blocks**: It provides workers with novel ways to interact with information and knowledge inside the working environment adopting the latest devices.
- **Worker Centric Service Building Blocks**: It provides the content for the selected worker-interface, unfolding the treasure of manufacturing data, information and knowledge to the worker via APIs.
- **Smart Factory Infrastructure**: It is the back-end infrastructure including the latest developments in data enrichment and aggregation, including semantics and linked data, datafication and analytics, and visualisation frameworks.

**Smart Factory Data**: It is the hidden “treasure” to be unfolded by our developed technologies and services, ranging from multiple data sources and data formats.
A1.2 mainDSS

The mainDSS project focuses on the necessity of maintenance. Consultants and experts always would like to extract sound information from the corporate Computerised Maintenance Management System (CMMS) in order to assess maintenance data, records, etc. and then apply criteria and performance indicators, combining them with their expertise and experience, and conclude into their recommendations and audit reports for the management. The project developed an IT system to be used by industries themselves as a plug-in approach to interface with CMMS repositories.

Goal of the project:
Nowadays, in competitive business environment one of the key challenges is to maximize the operational availability of assets and minimize the life-cycle costs. Maintenance offers significant opportunities in reducing cost and increasing production outcome, being a good choice to improve the competitive business environment. However, it is considered as a “black hole” where costs are generated without having straightforward ways of measuring returns; industries have to address three main problems [mainDSS]:

1. Lack of an effective communication between managerial and technical departments. Management executives are unfamiliar with the required concepts for understanding life cycle maintenance needs (targets to set, KPIs to measure) and assessing the impact of long-term maintenance investments.

2. Maintenance teams focus on repair-maintenance, instead of establishing a proactive-oriented maintenance department and prioritizing the necessary maintenance pillars. Usually the enterprises use support from maintenance consultants, but they are often unaffordable, especially for industrial Small and Medium-sized Enterprises (SMEs).

3. CMMS platforms focus on day to day maintenance performance and neglect holistic assessment of maintenance department performance; there are not any effective suggestions for improvements or means to comparatively assess the performance of the industry.

In order to solve these problems, mainDSS offers a plug-in approach, which dynamically makes a bridge with CMMS repositories and retrieving all necessary data in order to know, the current maintenance performance of the company.

Related architectures:
The overall mainDSS architecture is shown in Figure 41 below:
- **Maintenance ontology**: It is used to model infrastructure components (e.g. assets), CMMS model entities (e.g. Bill of Materials) and common maintenance operations (MTBF & equipment replacement). The ontology represents concepts (e.g. MTTR) and their relationships, and facilitates applied operations.

- **Auditing module**: It is responsible for the management of various auditing criteria and the persistent storage and retrieval of auditing results; it is used to assess effectiveness of maintenance operations by asking if the spare parts with stock level are below the established minimum, if the MTTR is too high or the consolidated cost of a machine failure. It provides to the management the financial and quality reports based on best practices and Key Performance Indicators (KPIs) compliant to international standards; complimented by a Progress Control module to monitor the progress of the maintenance department (i.e. comparative assessment of current and historical data, at selected time periods).

The specific tasks that the module is responsible for are the following:

- Provision of user interfaces that give the ability to the end user to define auditing criteria, along with an interface to manage them (e.g. edit an existing criterion, add a new one, etc.), to save/reload them, etc.
- Implementation of repository mechanisms that store and retrieve auditing results, per company and period of time.

- **Decision Support (DSS) module**: It provides recommendations and defines priorities for leveraging maintenance performance and offering proposals for ameliorating identified shortcomings; for instance, if the MTBF is below the accepted threshold, then it may suggest increasing preventive maintenance, replacing equipment, training technicians or hiring more experienced technicians. This module is the point in the system where: visions, dimensions and recommendations are managed, DSS engine rules and policies are configured and decision support engine runs in order to provide recommendations to the end users.

- **Bridging module**: Allows reliable import of data from CMMS repositories, this is the innovative plug-in approach. It is the link between the mainDSS auditing module and the client CMMS, its
main function is providing the information bridge between the two systems and controlling the way information flows. The primary user of the bridging module is the consultant implementing the mainDSS, which can perform the following tasks:

- Configure various aspects of the plug-in that collects maintenance data
- Configure details of the service and the interface
- Configure the end-user alerts mechanism

A1.3  **LINKEDDESIGN**

LINKEDDESIGN project concentrates on the problem that it is not allowed to have an integrated, holistic view on data across the full product lifecycle for the current ICT landscape for manufacturing. Especially the design and engineering of products and manufacturing processes is decoupled from the actual process execution. A tight integration of all tools used throughout a product lifetime is usually not feasible, and this requires therefore new design methodologies and novel integration tools. The project provides a solution that allows the integration of the holistic view on data across the full product lifecycle for ICT.

**Goal of the project:**

LinkedDesign will boost the productivity of today’s engineers by using the Linked Engineering and Manufacturing Platform (LEAP), capable of providing an integrated, holistic view on data, persons and processes across the product lifecycle as a vital resource for the outstanding competitive design of novel products and manufacturing processes.

LEAP is conceived as an integrated information system for manufacturing design. It merges all product lifecycle information relevant to drive engineering and manufacturing processes, independent of their format, location, originator, and time of creation. Besides the unified access to the integrated information, it provides specific knowledge exploitation solutions such as design decision support systems to analyse the integrated information.

In detail, LinkedDesign project provides User-centric lifecycle information management, Context-driven access to merged information and Knowledge and cross-discipline collaborations between users by novel approaches for collaborative engineering. In addition, it offers tight feedback connections to existing engineering tools in order to push back formalised knowledge to enable automated design of elementary product components.

**Related architectures:**

In Figure 42, it is shown, the overall architecture, which is described in more detail in [LinkedDesign 2015].

All concepts and components developed have been evaluated using three application prototype focused on the following topics:

- Manufacturing Design based on Plant Lifecycle Costs.
- Knowledge Re-use and Collaboration in Automated Design for Offshore Engineering.
- Metrology-driven Manufacturing in the Automobile Industry.
Some experimental results of using the LinkedDesign solution are the achieved estimated reductions of:

- Up to 50% in time spent for search and knowledge acquisition through the improved (collaborative) information access.
- Over 20% in time to market through support and automation of product and manufacturing design provided by the engineering platform.

The framework for knowledge exploitation within LinkedDesign has included:

1. Algorithms for the product and plant lifecycle data analysis that are applied in a decision support system to optimize the design of products/processes.
2. Algorithms of the statistical analysis to correlate errors detected with the 3D metrology with corresponding process configurations.
3. Algorithms for customer opinion mining that incorporate domain expert knowledge provided by the LinkedDesign Knowledge Network.

The algorithms could be considered as machine learning that automatically recognizes complex patterns and makes intelligent decisions based on the given data. However, LinkedDesign scenarios involve engineers, manufacturers or other users as essential stakeholders. The solution does not target pure machine learning algorithms based on artificial intelligence, but goes for decision support systems that analyse and process the given data and expose a set of high-quality solutions to the user.

A1.4 **PLANTCOCKPIT**

The Factories of the Future Public-Private Partnership (FoFPPP) initiative aims at helping EU manufacturing enterprises, in particular SMEs, to adapt to global competitive pressures by developing the necessary key enabling technologies across a broad range of sectors. It will help European industry to meet increasing global consumer demand for greener, more customized and
higher quality products through the necessary transition to a demand-driven industry with less wastage and a better use of resources. In order to contribute to the FoFPPP initiative, this project develops a central environment for monitoring and controlling of all intra-logistical processes. PLANTCockpit provides production supervisors, foremen, and line managers with the required visibility to make well-informed decisions for optimizing plant processes [PlantCockPit webpage].

Goals of the project:
These days, numerous methods, systems, and tools exist to facilitate production management, optimize resource utilization, and process efficiency. However, current Enterprise Resource Planning (ERP) systems, Manufacturing Execution Systems (MES), Supervisory Control and Data Acquisition (SCADA), and special-purpose solutions are rarely integrated with each other and typically provide no more than point-to-point interfaces between selected functionalities [PlantCockPit webpage].

The sporadic point-to-point integrations do not fulfil the requirements of today's dynamic markets where enterprises have to quickly judge complex situations, react to unexpected events, and make far-reaching decisions. With the growing focus on sustainability, complexity grows even further as production supervisors have to manage energy and material consumption, carbon footprint, and waste output in addition to classical KPIs like process efficiency, asset utilization, quality, scrap rate, and costs. Efforts to find the optimum for yield, quality, speed, or energy consumption often result in local optima, far from the ideal solution. Optimization must start at global bottlenecks within the plant or supply network, which can only be identified if overall process transparency is given. Only a tight integration of all systems will provide the visibility and process integration needed to truly recognize the potentials and optimize intra-logistics processes; be it with respect to yield, quality, energy consumption, or waste [PlantCockPit webpage].

PLANTCockpit is a central environment for monitoring and control of all intra-logistical processes. The outputs will help production supervisors, foremen, and line managers with the required visibility to make well-informed decisions to optimise plant processes. In order to do that, it includes the holistic visibility of the plan, the current status, deviations, exceptions, and bottlenecks. PLANTCockpit suggests a model for integrating heterogeneous shop floor management systems including ERP, MES, SCADA, condition-based maintenance, energy management, and other special-purpose systems. Virtually all manufacturing operations are multi-vendor environments which makes a harmonization of all systems a highly challenging goal. PLANTCockpit focuses on defining standard interfaces and a reference model for integrating the most prominent manufacturing processes. Current shop floor integration standards such as ISA 95, OAGIS, OPC Unified Architecture, MTConnect will be used as starting points.

The following are the main goals of the PLANTCockpit project with their respective activities to achieve them [PlantCockPit webpage]:

1. Visibility and integration across all manufacturing processes, layers, and systems:
   a. Manufacturing ontology, data and process model
   b. Standardised interfaces
   c. Flexible information integration and process orchestration
   d. Propagation and aggregation of alarms and events
   e. Visual integration
2. **Optimized production and logistics processes:**
   a. Data gathering and bottleneck determination
   b. Optimization through simulation and advanced planning

3. **Increased energy efficiency and reduced waste production:**
   a. Monitoring
   b. Contextualized analysis
   c. Asset optimization through intelligent maintenance

**Related Architectures:**

Figure 43 presents the PLANTCockpit architecture. The architecture is done adopting the layer-based concept, in order to separate different functional aspects and facilitate customization and configuration. It allows the use of different technologies for the implementation, provides easy extensibility for new functionalities, focuses on the major functional aspects concerning production and logistic cockpits and it is driven by the user requirements.

![PLANTCockpit Architecture](image)

The connection between the layers and different components is provided by message-based uniform and standardized interfaces. Each layer provides various functional components with specific functionalities and connections between them. All the layers and their inner components are described in detail in [PlantCockPit Whitepaper].
A1.5 CoSpaces

Ambient Interfaces and ubiquitous computing are characterised by the integration of the increasingly complex technological landscape with human environment, while removing the perception of the computer in the tasks and activities that are undertaken. The evolution of IT over the last decades has led to the development of individual CAD/CAE workstations. While the computational power available to Engineering professionals has grown exponentially, the collaborative dimension of the workspace has been largely underdeveloped. CoSpaces provides an evolutionary path towards new and more collaborative work environments. The users of CoSpaces technologies in manufacturing and design, working in cooperation with their suppliers, are able to configure their own collaborative workspaces and utilise ground-breaking innovations in context-aware interfaces, natural interfaces, and “human-centric” workspaces supporting a range of collaboration scenarios and product lifecycles. The developed platform supports product design and manufacturing in geographically dispersed teams in distributed virtual engineering enterprises. It helps the team members to participate in decision-making, view designs, propose modifications, and access reference material in ways that are specific to their disciplines, and allow them to interact using a range of devices depending on their current location and situation [CoSpaces 2015].

Goal of the project:
The CoSpaces project has evaluated collaboration at individual, team and enterprise levels, and developed collaboration models emphasizing applications of problem solving, creativity, participatory and knowledge based design in innovative collaborative work environments. CoSpaces is a distributed software framework that supports easy creation of collaborative work environments for distributed knowledge workers and teams in collaborative design and engineering tasks.

CoSpaces features a collaborative platform for viewing a workspace. Several experts have a domain specific view on the workspace at the same time. This enables them to discuss topics together. Also augmented reality technology is integrated into the platform. It allows experts to experience the workspace from the viewpoint of an end-user already during the design phase.

The CoSpaces project addresses the following objectives [CoSpaces 2015]:

- Evaluate collaboration at individual, team and enterprise levels, and develop collaboration models emphasizing applications of problem solving, creativity, participatory and knowledge based design in innovative collaborative work environments.
- Create an innovative distributed software framework that will support easy creation of collaborative work environments for distributed knowledge workers and teams in collaborative design and engineering tasks.
- Validate the distributed software framework for creating different classes of collaborative working styles required for collaborative design and engineering in the Aerospace, Automotive and Construction sectors.

Related architectures:
The CoSpaces application components are illustrated in Figure 44.
Figure 44. Application Components [Hardiman et al. 2010].

- **Portal**: It is the interface to the users. Its backend, the Collaboration Broker (CB) communicates with the other framework components and processes data and input from and for the user.
- **The Dynamic Session Management (DSM)**: It is responsible for the dynamic parts of the collaboration, i.e., meetings.
- **Group Management, Knowledge Support, Positioning and Identification**: They provide various kinds of information, which is required during the collaboration.
- **Resource Management (RM)**: It is responsible for the availability of data to the users.
- **Application Controller (AC)**: It manages the applications during a meeting and the communication between them.

In a corporate enterprise architecture, the purpose of an application portfolio catalogue is to identify and maintain a list of all the applications in the enterprise. The current application model for CoSpaces is depicted in Figure 45.
A1.6 **MIRROR**

Overall Objective of “MIRROR” is to develop a **personal performance mirroring and improvement solution**. MIRROR combines instruments of personal competency and potential analysis, reflection, interaction and learning based on activities within a work process. MIRROR helps to identify improvement potentials of oneself to increase the productivity and business performance. Benchmarking with a peer group within a social network is in the core of MIRROR. People who share the same work experience and job roles can use MIRROR across companies for temporary or continuous personal improvement and learning support. By this, MIRROR integrates personal learning with personal improvement strategies and performance management [Mirror 2015].
From a technological perspective, MIRROR will offer instruments and tools to log and track personal business activities and compare the activities with other peers in a social network. For activity tracking, MIRROR uses tools on the desktop as well as on mobile tracking devices. The tools build on results of APOSDLE [APOSDLE webpage] for activity tracking and of PROLIX [PROLIX 2015] for matching people competencies as well as analysis of competency gaps. MIRROR will provide a personal visual desk board for training planning, gap analysis, benchmarking, performance comparison etc. MIRROR will use semantic technologies for competencies and learning to analyse business activities, to match profiles of people with working attitudes and to bring all this into a social network of people with same interests and roles in business. MIRROR will also provide game elements for learning in the group within the social network.

The idea behind the project is that the motivation to learn in business is highly related to individual objectives and how people see themselves in comparison to others in the same work environment or job role. On the job-training with technologies is still a very new method, there are yet not enough learning and creativity tools that support learning at the work place in a systematic way. In Sports-Training, systematic analysis of time spend to train – related with a goal to reach, collecting data like time, pulse frequency, speed, etc. allows to measure and benchmark the training success and get fitness data. MIRROR transfers this approach into a business environment for training and learning by self-reflection.

Training and learning from others means to learn from them by knowing how they act, what they do, how they do it compared to oneself. With MIRROR people will see how they can change their working style, knowledge and competencies in order to be better and more productive in business and for their own company. MIRROR will be a personal system, so data will be only accessible for oneself, if allowed for others in an anonymous way; it is not a managed environment from a company for the employees. MIRROR can be used for example in the following scenarios:

- Sales professionals in the automobile industry can find out, how other sales professionals work, what they know, how they structure their working days, how much time they spend for their activities etc. Based on this, MIRROR will allow creating groups of people with the same working profile and providing learning recommendations and coaching service to improve the group.

- Shop floor personal in the retail business can capture over a time of 3-6 weeks every activity in detail. The data will be aggregated and made comparable with others in the same business. Graphical activity charts will show, how the own business day is being structured and compared to others. Based on these data, individual recommendations for changes in the attitude of work, timing, knowledge, etc. will be given by the system.

Related architectures:

An overview of the Mirror architecture is shown in Figure 46 below.
Adapt4EE aims to develop and validate a holistic energy performance evaluation framework that incorporates architectural metadata (i.e., BIM), critical business processes (i.e., BPM) and consequent occupant behaviour patterns, enterprise assets and respective operations as well as overall environmental conditions. The Adapt4EE framework, having as a central point of reference the occupancy behaviour (presence and movement) aligns energy consumption points to all interrelated enterprise aspects (business processes, enterprise assets and utility state and operations) [Adapt4ee].

In this context, Adapt4EE developed an enhanced semantic enterprise model that treats, learns and manages the enterprise environment as an intelligent agent, perceives environmental state using multi-type sensors and information modalities. The Adapt4EE Model incorporates business processes and occupancy data. It also constitutes a formal model for enterprise energy performance measuring, monitoring and optimization. The model is calibrated during the training phase based on sensor data captured during operation and then applied and evaluated in real-life every day enterprise operations. More specifically the Adapt4EE Enterprise Models allows for the proactive identification of optimum local adaptations of enterprise utility operations, based on predictions of possible occupancy patterns and respective business operations and energy profiles. To this end, Adapt4EE provides a set of Open Business Reference Models as well as respective methods and tools for modelling and simulating the real life operation of construction products, thus allowing for more accurate and realistic evaluation of the energy impact of alternative design and planning decision at an early design phase, prior to realization.

Related architectures:
The overall Adapt4EE conceptual architecture is comprised of two main frameworks, the measurement and the simulation framework [Adapt4EE D2.1], which are visually depicted in Figure 47 and Figure 48.

**Figure 47. Adapt4EE Measurement Framework [Adapt4EE D2.1].**

**Figure 48. Adapt4EE Simulation Framework [Adapt4EE D2.1].**

- The agent-training module: It is fed with semantically enriched information gathered by the Adapt4EE Middleware and it is able to analyse the actual building space utilization having as main catalyst the occupants (presence and flows).

Simulation Framework:
- **Occupancy prediction module:** It is responsible for predicting the occupancy of a building and its space utilizing the occupancy information from the measurement framework and the building information.
• **Agent-based simulation module:** It processes and controls the generation of simulated space utilization of the building, taking into account the business processes occurring in the specific enterprise, the user behaviour modelling (as provided by the Occupancy Prediction Module) and the current space layout (i.e., BIM).

• **User Activity Simulation Module:** It incorporates the information from the Agent-based Simulation Module and from the Occupancy Prediction Module, and generates a spatiotemporal model of building space utilization by users.

• **Building performance simulation module:** The extracted detailed occupancy schedule for the building spaces (multi-agent based activity building simulation) along with the enterprise energy profile models feed the Building Performance Simulation Module, that is able to estimate the energy performance of the building taking into account both the architectural envelope itself and the facility users and the activities performed by them.

**Visual analytics module:** It implements different energy-related views of the building under design and provides the capability to compare and assess the energy efficiency of multiple design alternatives.

### A1.8 INERTIA

INERTIA adopts the Internet of Things/Services principles to the Distribution Grid Control Operations. Semantically enhanced DER will be the main constituents of the INERTIA active distribution grid management framework. They will constitute active and flexible components (generation as well as consumption) that carry contextual knowledge concerning their local environment. Within INERTIA, DER will form dynamic clusters that essentially comprise self-organized networks of intelligent active nodes that will efficiently distribute and balance global and local intelligence. This self-organized overlay network of connected DER will allow for the seamless management and control of the active grid as well as the efficient coordination and exploration of single and aggregated technical prosumer potential (generation and consumption) to participate in energy balancing and other distributed grid related services.

Global Operational and Technical parameters of the electricity grid will be seamlessly and continuously translated into real-time Local Control Strategies. INERTIA will offer fine grained control (equivalent or even higher than existing Direct Control Programmes) while at the same time ensure privacy and autonomy on the local level, fully respecting prosumer preferences and needs. INERTIA will promote the efficient integration of flexible demand with distributed generation within the smart grid as the means to tackle the problems resulting from the continuous and massive integration of distributed intermittent and non-controllable renewable sources.

More specifically, the main objectives of INERTIA are the following:

- **Objective 1:** Innovative DSM framework, for the efficient integration of local demand and generation (DER) to the smart grid and utilization of DER flexibility for the optimal coordination of the Distribution Grid.

- **Objective 2:** Semantically Enhanced and Multi-Dimensional DER (Generation and Consumption) Profiles.

- **Objective 3:** Integrated Prosumer Profiles, Incorporating Behaviour, Activity Flows and Environmental Knowledge.

- **Objective 4:** Multi-Agent Based Distributed Grid Management Framework.
- **Objective 5**: Multi-Directional Communication and Control Framework for Horizontal and Vertical Integration of Active Components (DER).
- **Objective 6**: Holistic Energy Performance Models and Respective Business/Contractual Models for the S&M Cis.

**Related architectures:**

The overall INERTIA conceptual architecture is depicted in Figure 49 and presented in the paper [Jimeno et al.]. The INERTIA control network on the highest level will consist of two types of peers/“Hubs”, while appropriate intelligent agents will be running on both hubs, underlying the self-organized distributed grid and load control operations:

- **Local Control Hub**: Corresponding to the level of building areas, buildings or building complexes, the Local Control Hub will autonomously and automatically manage in real-time local control actions and respective information.
- **Aggregator Hub**: Through this Hub the Aggregators will be provided with real-time monitoring and control tools, in order to efficiently organize their customer loads portfolio and define specific Active Demand strategies and respective services based on market needs, acting as an intermediary between suppliers, network operators and prosumers belonging to their portfolios.

![INERTIA Conceptual Architecture](Image)

The main components of the INERTIA framework are described below:

**Local Control Hub:**

- **Semantically Based Middleware**: It provides a multi-directional communication and control interface of individual and aggregated DERs (local loads and generation) with the Distribution Grid.
- **Occupancy Flow Modelling and Prediction**: It is responsible for extracting real-time building occupancy flows based on data stemming from the multi-sensorial cloud, as well as for occupancy modelling and prediction.
- **DER Flexibility Modelling**: It provides integrated models, extending the typical two dimensional stochastic profiles in an attempt to deliver Context Aware Multi-Dimensional Load Flexibility Profiles.
• **User Profiling:** It monitors and learns transparently occupants’ behaviour, in order to extract user preference models.

• **Performance Modelling:** It models and calculates the necessary parameters (KPIs) contributing to the Holistic Flexibility decision making.

• **Holistic Flexibility Analysis, Aggregation and Forecasting:** It has the responsibility of determining the control strategy which should be applied by analysing and correlating data coming from other internal frameworks.

• **Building Monitoring and Control:** It is responsible for monitoring building operation and applying control strategies.

• **Ambient User Interface:** Personalized Ambient User Interfaces will be developed in order to address final occupants’ and facility managers’ needs.

**Aggregator Control Hub:**

• **DER Flexibility Analysis, Aggregation and Forecasting:** It provides visualization and interaction mechanisms to the Aggregator for multidimensional analysis, correlation and efficient management of prosumer profiles.

• **Grid Coordination and Demand Response Activation:** It is the communication interface between the Aggregator Control Hub and the DSO system.

• **Multi-Agent Based DER Prosumers Control and Optimization:** It ensures the efficient execution of Global Demand Control Strategies through coordinated, distributed local control actions on selected clusters of the aggregator’s portfolio.

**Common Aspects:**

**Common Information Model (CIM):** It defines the high level domain model/ontology comprising the basic elements underlying the INERTIA framework

---

**A1.9 EBBITS**

The ebbits project provides semantic approach to IoT and hence introduces an innovative bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real world. The ebbits platform enables these main innovative features:

• **Semantic interoperability** between heterogeneous physical world technologies and the enterprise systems, also outlining scalable network architectures and featuring opportunistic communication paradigms.

• **IoT entity Virtualization,** through an addressing layer based on unique identifiers, it enables discovery features, through semantic techniques and attribute-based services descriptions.

• **Innovative Data and Event Management,** through a P2P (peer-to-peer) Event Management Architecture, which leverages on the publish/subscribe communication paradigm to handle rule-based service orchestration.

• **Centralized and Distributed Intelligence,** which allows the definition of data fusion frameworks and adopting ontology-based context models to promote self-awareness mechanisms.
• **Semantic Knowledge Infrastructure**, connecting many conventional data sources to semantic models, with support of hybrid querying and real-time reasoning.

• Support for Business Process Life Cycle Management, including process taxonomy and optimization metrics.

Related architecture

The ebbits project advances the LinkSmart middleware⁵, the main result of an EU project called Hydra. It combines a service-oriented architecture, peer-to-peer networking technologies and semantic Web Services, addressing the interoperability issue. On the top of LinkSmart, the ebbits platform builds its five main layers (as depicted in the overall architecture diagram in Figure 50):

- **Physical World layer**: includes real-world devices (e.g. robot controller, PLCs, energy measuring sensors, WSANs), which are accessible via proper drivers and adapters. The key component at this layer is the Physical-World Adaptation Layer (PWAL), and adaptation component responsible for exposing device resources and data access capabilities in a unified way.

- **Internet of Things layer**: offers “Thing” enabling functionality needed for IoT resource management, including resource discovery provided by Internet of Service Layer. Key components are based on the LinkSmart middleware.

- **Internet of Services layer**: provides services and resource catalogue functionality, as well as event processing capabilities. Key components at this layer are the IoT Service Manager, Event Processing Agent, and Thing Discovery Manager.

- **Business System Mediation layer**: provide interfaces for applications and enterprise systems and offers the abstractions of things, services and people for IoT enabled applications. Key components are Thing Manager, People Manager, Context Manager and Service Proxy.

- **Inter Business layer**: is the highest layer of the ebbits platform architecture focused on support traceability and information management for IoT enabled entities, such as products or different types of physical goods. Key component is Product Service Orchestration Manager that resolves and invokes services in order to assemble a product life cycle.

Orthogonal to the identified layers are components that offer infrastructure functionality, such as the Network, Security and Opportunistic Manager. In addition, the Ontology Manager implements a semantic model to collect information regarding devices, services, events, as well as obtaining knowledge supporting the ebbits decision-making components, such as the Context Manager.

**Main ebbits platform components**

The ebbits platform architecture is composed by several system modules, each one with specific functionalities is briefly presented in [Kostelnik et al. 2011; Brizzi et al. 2013]. The main components are:

- **Network Manager**: this component is the entry and exit point of the LinkSmart middleware. There must be only one instance of it, in every device, where the middleware is running. This component provides the entry point of the middleware as a Web Service interface. Furthermore, it creates the overlay P2P network needed to

---

⁵ https://linksmart.eu
interconnect the components of the middleware. It provides the architecture needed to address LinkSmart Web Services using unique IDs and it provides the transport mechanism over the P2P network, needed to invoke the LinkSmart Web Services (SOAP Tunnelling) using their IDs.

Event Manager: it is the component used to implement the publish/subscribe paradigm on the top of the functionalities provided by the LinkSmart middleware. It provides an application-level selected multicast that decouples senders and receivers in time, space, and data (i.e., sender and receivers do not need to be up at the same time, do not need to know each other’s network addresses and do not need to use the same data schema for events they send). It leverages the JXTA Pipe Abstraction, which provides a virtual connection between two pipes’ endpoints: the input pipe and output one. Pipe connections are independent from pipe location and connectivity, thus enabling to bypass network issues like firewall or NAT. Publish-subscribe mechanism is implemented through JXTA since a pipe endpoint can subscribe to messages sent from a publisher endpoint. Using this manager, the developers can create applications based
on asynchronous events; the Event Manager can be used both to send events and to subscribe itself to the topic of interest.

- **Ontology Manager**: the purpose of this component is to provide a unified interface for using the LinkSmart Device Ontology and all related modules, within the LinkSmart middleware. The manager maintains run-time instances of the LinkSmart devices.

- **Context Manager**: a unified interface to access the context of entities in the environment, it decouples the application from the context providers (sensors), which consequently eases the application developer tasks. The context management consists of an Entity Manager and Rule Manager, both depending on the Event Manager to retrieve sensor data from the proxies.

- **Entity Manager**: exploits the Ontology Manager to retrieve the metadata of the sensor readings, such as location, measurement unit, reading interval, and other attributes. This manager is responsible to store the current context attributes of the entities. For this purpose, the developers must define which entities and context attributes are to be monitored by the Context Manager. The application developers can define this in the header of the rules, which then are passed to the rule engine. In this way, the Entity Manager can monitor only the state of the system that is needed, optimizing the performance of the Context Manager.

- **Rule Manager**: The Context Manager recognizes situational context based on rules defined by the developers and stored by the Rule Manager component. The rules may also define a higher level of context information, which must be inferred based on several sensor information, and their semantics defined by the ontology. In the rules, there is also the definition of action that should be performed when certain context is recognized.

- **Physical World Adaption Layer (PWAL)**: The PWAL is the adaption layer between the Physical World sensors and Internet of Things platform, used to provide a common way to interconnect physical entities (including sensors, devices, protocols and communication channels), which are not part of LinkSmart or of other ebbits components. The main role of the PWAL is to abstract any device connected to the platform, exposing its capabilities and output data and hiding the specific implementation details (hardware and software). Specifically, the capabilities exposed are the events generated, the services offered and the direct access to information stored into hardware resources. In this way, the application that uses the device can interact with it, without knowing details about its implementation. The capabilities of the device are exposed through the PWAL driver, which provides, on one side, an abstraction to the device specific technologies and, on the other side, a uniform interface for other ebbits components, ensuring that devices can be integrated seamlessly with the other IoT components of the platform.
ANNEX 2 – SATISFACTORY SUB-COMPONENTS DESCRIPTION

Sub-component Name
New IoT Sensors with Robust Communication

Description
This component comprises wireless networks of new IoT devices, featuring robust connectivity for delivering relevant data and events across the shop floor. This component is essential for providing the necessary information to the SatisFactory knowledge-base and thereby enable context-aware applications.

Services
- Robust wireless connectivity
- Secure communications
- Complex network management

Dependencies
- Automation Systems - I/O Field Network
- Ultra Wide Band (UWB) Localization Devices
- Depth Sensor Network
- Thermal Sensor Network
- Device Manager
- Event Manager

Inputs needed
- Factory layout according to the use-case definition
- System architecture
- End-user requirements
- Technical requirements

Outputs provided
- Dependable communication API
- Physical-world monitoring and control API

Related Tasks
- T4.1: Development of intelligent IoT infrastructure to enable context-aware applications
- T4.2: Hardware Development of context-aware HMIs
- T4.4: Development of intelligent notification and control middleware
• T4.5: Integration of social collaboration and sensing environments for enhancing collective awareness

SatisFactory Platform Layer

• Physical Layer

Type of exposed APIs (if exists)

REST APIs
Sub-component Name
UWB Devices

Description
The UWB Devices can be of three types (i.e., tag, anchor and gateway) and are used in indoor environments to provide location information. Typically, the anchors are placed in fixed and well know positions in the region of interest. The tags are worn by workers and/or installed on mobile objects whose positions need to be estimated. During the estimation process the tags continuously interact with anchors. In particular, each tag performs range measurements with respect to the anchors that are in the connectivity range. After that, a localization algorithm, which runs on the tag itself, uses range measurements and anchor positions to estimate the tag's position. Finally, the gateway is the end-point of the localization devices; it receives localization data from tags and forwards them to the Localization Manager through the Middleware by using a serial interface.

Services
- Location information
- Ranging information

Dependencies
- Middleware
- Localization Manager

Inputs needed
- Coordinates of UWB anchors
- Ids of anchors
- Localization update rate

Outputs provided
- Ranging data (distance between tag and anchor, tag Id and anchor Id)
- Localization data (tag Id, flag position estimated, estimated coordinates expressed according to a relative reference system, number of ranging measurements, anchor connectivity, localization accuracy)

Related Tasks
- Task 3.3: Context-aware incident detection engine to increase worker's comfort
- T4.1: Development of intelligent IoT infrastructure to enable context-aware applications

SatisFactory Platform Layer
- Physical Layer

Type of exposed APIs (if exists)
- At tag level: UWB-based messages
- At gateway level: serial messages
Sub-component Name
Cameras Sensor Network

Description
Cameras Sensor Network may be composed of depth cameras and/or thermal cameras. It is responsible for providing the system privacy preserving information to the system. This information will be mainly focused on the analysis of the human movements and actions in the shop floor.

Services
- Acquisition of human movements/actions in the shop floor

Dependencies
- Middleware
- Incident detection tool

Inputs needed
- No input needed

Outputs provided
- Trace of the detected humans in the shop floor
- Trajectories of the detected and tracked humans in the shop floor
- Lightweight statistics about the humans in the shop floor (e.g. how many people lay in each space)

Related Tasks
- Task 1.4: Common Information Data Exchange Model (CIDEM)
- Task 3.3: Context-Aware Incident Detection Engine to Increase Worker’s Comfort

SatisFactory Platform Layer
- Physical Layer

Type of exposed APIs (if exists)
Each camera will be connected with a system able to detect and track humans. This system will provide all the necessary information to the middleware through TCP/IP connection (either wired or wireless).
Sub-component Name
Automation System – Legacy Sensors

Description
The Automation System subcomponent is able to automatically control a factory process by means of a network of interconnected sensors and actuators. It includes a software module, called Plant Data Exchange, that supports the transmission and transformation of the shop floor data to ISA-95 complaint format (based on B2MML). It combines diverse data from shop-floor automation systems and the IoT sensors to support decision making, in events related to production activities, in smart factory incidents, etc. It is used in two modes. First, as a component of the automation system at in-house process system. Secondly, it provides a service to the SatisFactory platform through the middleware or directly to SatisFactory components that are able to communicate based on Restful services or MQTT protocol.

Services
- Collection of all necessary shop floor data and distribution to the proper SatisFactory components
- Receive feedback for notifications or warnings to be displayed to the existing HMIs at the shop floor

Dependencies
- Common Information Data Exchange Model (CIDEM)
- Middleware
- Integrated Decision Support System (iDSS)

Inputs needed
- Shop floor data (from the automation system and/or the legacy sensors)

Outputs provided
- Notification in case of alarms at the pilot plants
- Information about the nominal operation of the pilot plants

Related Tasks
- Task 3.5: Shop floor feedback engine and integrated Decision Support System
- Task 5.3: Deployment of industry lab use case

SatisFactory Platform Layer
- Physical Layer

Type of exposed APIs (if exists)
Set of REST and MQTT APIs for the communication with the middleware, the CIDEM and the iDSS. Also, a TCP and OPC API for retrieving the shop floor data for the automation systems.
Sub-component Name
AR Glasses

Description
Glassup AR Glasses is an augmented reality eyewear device allowing the operator to see information coming from the company’s information system directly on the eyewear’s lens, for example data about manufacturing processes or a maintenance procedure to perform. The eyewear has also a camera (for on the job training activities or remote assistance) and a port to plug different type of sensors, like thermo-cameras, temperature sensors, gas sensors, etc.

Services
- Augmented reality information displayed directly on the glasses lens

Dependencies
- AR In-factory platform, Middleware

Inputs needed
- Data coming from SatisFactory components such as Integrated DSS and sensors through the Middleware
- Events coming from SatisFactory components such as AR-In Factory Platform, Gesture & Content Recognition Manager, Localization Manager.

Outputs provided
- Augmented info overlaid on operators’ field of view: guidance and instruction.
- Video streaming and audio or images to SatisFactory components (e.g., AR-In Factory Platform, Gesture & Recognition Manager)

Related Tasks
- Task 4.2: Hardware Development of context-aware HMIs
- Task 4.3: Interactive augmented reality services and data analytics techniques
- Task 4.5: Integration of social collaboration and sensing environments for enhancing collective awareness

SatisFactory Platform Layer
- Physical Layer

Type of exposed APIs (if exists)
- The physical radio interface might be Wi-Fi or Bluetooth Low Energy
- Open REST-like API interface
Sub-component Name
Middleware Core

Description
This component is heart to middleware functionality and provides an abstraction layer for transparent and location independent communication between different middleware services. The services and device manager makes registration into middleware to be part of middleware overlay network.

Services
- Service registration
- HTTP Tunnelling for location independent access
- Discovery of middleware services

Dependencies
- Event Manager
- Device Managers

Inputs needed
- SatisFactory tools
- Domain Model
- Device Registration Document

Outputs provided
- Output is a registration document that is used by the Device manager and Event Manager to access other Middleware services

Related Tasks
- T.4.4: Development of intelligent notification and control middleware

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
REST API for service registration and discovery
HTTP API for tunnelling
Sub-component Name
Device Manager

Description
The Device Manager provides the instruments to simplify heterogeneous physical devices virtualization through common interfaces. These devices have different capabilities, communication technologies and features. A device virtualization is needed to help the seamless integration of these devices with other SatisFactory components and features (e.g. DSS systems, Ontology Manager, etc.).

Services
- Location information
- Geo-fencing events
- Other services

Dependencies
- Semantic Knowledge Management Components
- Other Middleware sub-components

Inputs needed
- Device related information (communication technology, access methods)
- Device APIs

Outputs provided
- Device abstraction reference model
- Virtual instances of available/DM compliant devices in the shop floor
- Events and/or Services and/or Memory maps of available/DM compliant devices
- Unique addressability of available/DM compliant devices

Related Tasks
- T4.4: Development of intelligent notification and control middleware
- T4.5: Integration of social collaboration and sensing environments for enhancing collective awareness

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
Set of REST APIs which allow to retrieve the list of the devices abstracted and to interact with them.
Sub-component Name
Event Manager

Description
This component provides a publish/subscribe-based notification mechanism to other SatisFactory components and services. Mainly the data captured from smart and other sensor types is collected, processed and stored in a repository through the Event Manager.

Services
- Real Time diagnostic
- Contextualization of raw data
- Incident related dynamic and persistent data

Dependencies
- Device Managers
- CIDEM
- Ontology Manager

Inputs needed
- SatisFactory tools
- Domain (Shop floor) Model
- Interfaces for Smart Sensor Network
- Interaction with Repository

Outputs provided
- An Event type that is understandable to Device Managers, repository and other SatisFactory tools (e.g., Ontology Manager)

Related Tasks
- T4.4: Development of intelligent notifications and control middleware

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
MQTT based protocol for notification
Sub-component Name
Event Aggregator

Description
It is responsible of storing events received from physical layer components through Event Manager into CIDEM repository service. It subscribes to the Event Manager for event notification and uses CIDEM repository REST interface for storing captured data.

Services
- Incident related dynamic and persistent data

Dependencies
- Event Manager
- CIDEM Repository

Inputs needed
- List of MQTT topics for subscription

Outputs provided
- CIDEM compliant representation of Sensor Measurement

Related Tasks
- T4.4: Development of intelligent notification and control middleware

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
None
Sub-component Name
DSS Core

Description
The DSS Core subcomponent is responsible for orchestrating all the rest of the subcomponents of the Integrated DSS component, in order to ensure that the output of each subcomponent is using the right channel and the appropriate format in order to be directed to external components and that external components can subscribe and be actually notified by the various events raised from DSS subcomponents.

Services
- Accept safety and maintenance related events from the Maintenance Toolkit and route them to components that have subscribed to them.
- Receive actionable knowledge recommendations from the shop floor Feedback engine and route them to the appropriate HMIs.
- Handle RESTful API requests and route them to the appropriate module

Dependencies
- Device Manager
- Operational Platform with Augmented Intelligence
- All DSS sub-components

Inputs needed
- Corrective Maintenance events
- Safety related event triggers
- Production re-adaptation recommendations from Maintenance Toolkit
- Actionable knowledge and Recommendations from Shop Floor Feedback Engine

Outputs provided
- UI Events for AR Platform, Device Manager and other HMIs.
- Structured recommendations for actions

Related Tasks
- Task 1.3: Use cases and Scenarios
- Task 2.4: Toolkit for re-adaptation of production facilities and HR workload balancing
- Task 3.5: Shop floor feedback engine and integrated Decision Support System
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators
- Task 6.4 VR-enabled end-users training to SatisFactory solution

SatisFactory Platform Layer
- Decision Layer
Type of exposed APIs (if exists)

The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform. Server side push content technologies will be used in order to provide near real-time alerts to connected clients for incidents and the relative recommendations.
Sub-component Name
Shop Floor Feedback Engine

Description
The detected incidents will be paired with the corresponding response procedures, in order to give feedback to the shop floor in terms of actionable knowledge and recommendations, in cooperation with the DSS core module.

Services
- Information on incidents from the shop floor
- Feedback to the shop floor on how to treat incidents, how to react

Dependencies
- Incident Management Tools
- Middleware
- DSS core
- Visual Analytics module

Inputs needed
- Proactive or reactive type of incidents by Incident Management Tools
- Recommendations per incident from CIDEM

Outputs provided
- Information to the DSS core
- Information to the workers at the shop floor level through Device Manager.

Related Tasks
- Task 1.3 Use cases and Scenarios
- Task 3.3 Context-aware incident detection engine to increase worker's comfort
- Task 3.5 Shop floor feedback engine and integrated Decision Support System
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators
- Task 6.4 VR-enabled end-users training to SatisFactory solution

SatisFactory Platform Layer
- Decision Layer

Type of exposed APIs (if exists)
RESTful APIs will provide access to information and extracted shop floor related knowledge. Server side push content technologies will be used in order to provide near real-time alerts to connected clients for incidents and the relative recommendations.
Sub-component Name
Incident Management Tools

Description
This subcomponent represents a substantial role in the everyday activities of actors involved in use case scenarios in all pilot demonstrations supporting workers’ safety and comfort. The main goal of the module is the detection of proactive and reactive incidents on the shop floor. The system detects and alerts the possibility of a risky condition (proactive incident) and incidents after their occurrence (reactive) based on real-time data. In both cases, the tool performs the corresponding predefined countermeasures.

Services
- Monitoring of activities on the shop floor
- Detection of an incident (proactive & reactive)
- Enable alerts regarding the detected incident
- Enable of countermeasures related to the detected incidents

Dependencies
- Middleware
- Ontology Manager
- DSS-core module

Inputs needed
- Smart Sensor Network data (through middleware)
- Events from the Localization manager
- Events from the Multi-media manager
- Events from the Gesture & Content Recognition manager

Outputs provided
- Events regarding incidents (type of event, location, actors, etc.)

Related Tasks
- Task 1.4: Common Information Data Exchange Model (CIDEM)
- Task 2.1: Reference Architecture Design & Technology Exploration
- Task 2.2: Knowledge Modelling to Support the Human Resource Optimization
- Task 3.5: Shop Floor Feedback Engine and Integrated DSS
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators

SatisFactory Platform Layer
- Decision Layer
Type of exposed APIs (if exists)

The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform.
Sub-component Name
Maintenance Toolkit

Description
This subcomponent represents a substantial role in the everyday activities of actors involved in use case scenarios in all pilot demonstrations. The basic responsibility of the module is to monitor and supervise in real time the production processes and diagnose possible problems, flaws or malfunctions while triggering events for activating maintenance procedures or safety mechanisms.

Services
- Monitoring of maintenance in production facilities by interpreting the measurements coming from the Sensor Network or the Thermal and Optical Sensors
- Scheduling of preventive maintenance actions
- Identifying the need and organisation of corrective maintenance activities
- Enabling Condition based maintenance
- Triggering activation of safety mechanisms

Dependencies
- Operation & Maintenance procedures
- Asset-Machinery-Production Data
- DSS Core

Inputs needed
- Incident Events
- Logging from workers and supervisors
- Smart Sensor Network (mainly legacy sensors)
- Asset-Machinery-Production Model

Outputs provided
- Recommendations for balancing workload density
- Events triggering maintenance procedures need
- Events triggering activation of safety mechanisms

Related Tasks
- Task 1.3: Use cases and Scenarios
- Task 2.4: Toolkit for re-adaptation of production facilities and HR workload balancing
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators
- Task 6.4 VR-enabled end-users training to SatisFactory solution

SatisFactory Platform Layer
• Decision Layer

**Type of exposed APIs (if exists)**
The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform.
Sub-component Name
Visual & Real-Time Data Analytics Module

Description
The real-time analytics module is responsible for correlating, combining and analysing heterogeneous data acquired from the smart sensor network in order to evaluate the shop floor and production operations, supporting comparative assessment of worker’s situation and suggesting role-based actions.

Services
- Analysis of the shop floor heterogeneous information
- Evaluation of the shop floor and production operations

Dependencies
- Middleware
- DSS-core module

Inputs needed
- Smart Sensor Network data & models (through middleware)

Outputs provided
- KPIs regarding the performance of the shop floor operation
- Assessment events regarding the shop floor operation
- Assessment event regarding the production operation

Related Tasks
- Task 1.2: Models for Actors and Procedures Interconnections
- Task 1.3: Use cases and Scenarios
- Task 2.2: Knowledge Modelling to Support the Human Resource Optimization
- Task 3.5: Shop-floor Feedback Engine and Integrated DSS
- Task 4.3: Interactive Augmented Reality Services and Data Analytics Techniques
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators

SatisFactory Platform Layer
- Decision Layer

Type of exposed APIs (if exists)
The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform.
Sub-component Name
OSF Engines

Description
The OSF Engines governs the index and management of all OSF content (RDF data and Ontologies). Documents are indexed by the full-text search engine, while information about their structural characteristics and metadata are stored in an RDF Database, called a "triple store". Another engine is available for semi-automatic assistance in tagging input information and other natural language processing (NLP) tasks.

Services
- Management of OSF contents
- Full-text search engine
- Semi-automatic assistance in tagging input information
- Other NLP tasks

Dependencies
- CIDEM

Inputs needed
- CIDEM XML Schema
- CIDEM XML Data

Outputs provided
- RDFized Data

Related Tasks
- T2.2. Knowledge Modelling to support the human resource optimization
- T3.1 Semantically-enriched framework for analysis and design of dynamically evolving shop floor operations

SatisFactory Platform Layer
Facility Layer
Sub-component Name
OSF Web Services

Description
The OSF Web services are generally RESTful in design and are based on HTTP and Web protocols and open standards. For the needs of SatisFactory, there is a set of web services covering diverse functionalities toward the management and interaction with the Ontology Manager.

Services
- Controlling and interacting with the OSF Engines
- Simple RDF Data conversion
- CRUD (Create, Read, Update, Delete) functions
- Revision
- Search
- Authentication
- Dataset Management
- Ontology Management

Dependencies
- CIDEM
- DSS
- Context-Aware Engine

Inputs needed
- OM Ontology Models
- RDF datasets

Outputs provided
- Dataset exposition widgets
- Data search results
- Ontology Model Editor/Revision

Related Tasks
- T2.2. Knowledge Modelling to support the human resource optimization
- T3.1 Semantically-enriched framework for analysis and design of dynamically evolving shop floor operations

SatisFactory Platform Layer
- Physical Layer

Type of exposed APIs (if exists)
RESTful APIs
Sub-component Name
Localization Manager

Description
The Localization Manager provides location information of objects and workers based on the identifier of the associated UWB devices. This module receives either ranging or localization data from the UWB modules through the middleware. Moreover, it provides geo-fencing events when a worker approaches a forbidden area.

Services
- Location information
- Geo-fencing events

Dependencies
- Ontology Manager
- Middleware
- UWB Localization Node
- Operational Platform with Augmented Intelligence

Inputs needed
- Location or ranging data from localization UWB modules through the middleware
- Shop floor indoor map
- Coordinates of UWB reference nodes
- List of forbidden areas information

Outputs provided
- UWB-based location data expressed according to relative and absolute reference systems.
- List of UWB devices deployed in the shop floor (including mobile and reference UWB devices)
- Geo-fencing event (e.g., inside the forbidden area A at time t0)
- Data history related to location data and geo-fencing events

Related Tasks
- Task 3.1: Semantically-enriched framework for analysis and design of dynamically evolving shop floor operations
- Task 3.3: Context-aware incident detection engine to increase worker’s comfort
- T4.1: Development of intelligent IoT infrastructure to enable context-aware applications

SatisFactory Platform Layer
- Facility Layer
Type of exposed APIs (if exists)
MQTT based APIs
Sub-component Name
Multiple Media Manager

Description
The Multiple-Media Manager is a component responsible for handling information concerning video streams. If necessary and required, the video can propagate contextual information as metadata embedded in the video streams. Privacy preservation is also in scope of this component, where required.

Services
- Video streaming services
- Embedded metadata

Dependencies
- Ontology Manager
- Middleware

Inputs needed
- Real-time video streams from cameras
- Context-manager information to propagate relevant contextual information with the video streams

Outputs provided
- Multiple video streams
- Contextual information embedded as metadata (if needed)
- Adaptive streams based on network condition (if hostile environment is detected)

Related Tasks
- Task 2.4
- Task 3.1
- Task 3.3

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
Web-Service APIs are provided to subscribers in order to receive dedicated video streams. Adaptation of video quality based on network conditions is also in scope of this component.
Sub-component Name
Gesture & Content Recognition Manager

Description
The Gesture & Content Recognition Manager is a component in charge of performing a complex set of analysis on input streams and it reacts to a certain level of configurable alerts, triggering context-aware events. The component works under the assumption that the background can be mapped and this mapping action can be performed with regular scheduling.

Services
- Gestures recognition
- Geo-fencing events

Dependencies
- Ontology Manager
- Middleware
- Multimedia Manager

Inputs needed
- Blob streams from the Multimedia Manager
- Context-manager information to propagate relevant contextual information with the frames

Outputs provided
- Gestures triggered events
- Geo-fencing events
- Historical data based on geo-fencing events
- Historical data based on gestures triggered events
- Contextual information with frames provided alongside violation of paths

Related Tasks
- Task 2.4:
- Task 3.1: Semantically-enriched framework for analysis and design of dynamically evolving shop floor operations
- Task 3.3: Context-aware incident detection engine to increase worker’s comfort

SatisFactory Platform Layer
- Facility Layer
Type of exposed APIs (if exists)

REST APIs to provide images of the background with an overlay of the violator path, alongside all raw data for post processing and visual search purposes. The same component can also be relocated for providing privacy preserving statistics, in order to mark most crossed intersections, areas and paths.
Sub-component Name
AR OP Visualization Tool (Mobile Version)

Description
This Visualization Tool provides a rich set of visualization and interaction functionalities in order to present previously prepared AR OP directly “on the job”. Nevertheless it can provide the same information in a simulated framework based on VR. By using these tools, operators can follow a specific procedure step by step, obtain support for their current task and choose which kind of information and which level of detail they want about the procedure. Moreover, the tools themselves can become a “view channel” to show relevant information available at runtime from other components (e.g. Middleware, DSS, Event Manager. Etc.). Key features of the Visualization Tools are: a) high scalability on physical devices (e.g. tablet, smartphone, wearable devices); c) large number of presentation modes (e.g. in Augmented Reality, Virtual Reality, Text only, Hybrid, etc.).

Services
- Support of assembly and maintenance operations in AR for the operational environment
- Support to asynchronous visualization of common or emergency indications to the operators, deriving from external modules (e.g. DSS)

Dependencies
- AR OP Creation Tools
- SatisFactory Repository (CIDEM)
- Middleware
- Hardware Devices Capabilities & Performances

Inputs needed
- Augmented Reality Operating Procedure (AR OP)
- Data obtained from Middleware at runtime
- (No needs but possibilities) Data available from external modules

Outputs provided
- Audio/Visual information shown by implemented HMIs
- Log Data file for each executed procedure to save in CIDEM and to provide to Visual Analytics Tools

Related Tasks
- Task 4.3: Interactive augmented reality services and data analytics techniques

SatisFactory Platform Layer
- Facility Layer
Type of exposed APIs (if exists)

Since the Visualization Tools are stand-alone mobile Apps, they can communicate with the other SatisFactory modules mainly via Middleware. The exposed APIs in fact, are implemented as two specialized services, registered on Middleware, provided in order to exploit Operative Procedure Management and View Channel Management.
Sub-component Name
AR OP Visualization Tool (GlassUP Version)

Description
This version of Visualization Tools provides the necessary porting of above discussed mobile version to GlassUP’s devices. In particular it presents a new dedicated HMI, in order to maximize the capabilities of AR devices of GlassUP (glasses, video camera, input device in primis). Furthermore it introduces a new strategies in filtering and presentation of contents by respecting the possibility of display of the devices themselves.

Services
- Support of assembly and maintenance operations in AR for the operational environment
- Support to asynchronous visualization of common or emergency indications to the operators, deriving from external modules (e.g. DSS)

 Dependencies
- AR OP Creation Tools
- SatisFactory Repository (CIDEM)
- Middleware
- Hardware Devices Capabilities & Performances

Inputs needed
- Augmented Reality Operating Procedure (AR OP)
- Data obtained from Middleware at runtime
- (No needs but possibilities) Data available from external modules

Outputs provided
- Audio/Visual information shown by implemented HMIs
- Log Data file for each executed procedure to save in CIDEM and to provide to Visual Analytics Tools

Related Tasks
- Task 4.3: Interactive augmented reality services and data analytics techniques

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
Since the Visualization Tools are stand-alone mobile Apps, they can communicate with the other SatisFactory modules mainly via Middleware. The exposed APIs in fact, are implemented as two
specialized services, registered on Middleware, provided in order to exploit Operative Procedure Management and View Channel Management.
Sub-component Name
AR OP Creation Tool

Description
The Creation Tool allows to convert original Operating Procedures, often described with different formalisms, in a new format, called AR OP (Augment Reality Operating Procedure) ready to be used by the AR Visualization Tools developed to support both training and operational environments. The Creation Tools provide functionalities designed to manually describe a new AR OP by a dedicated approach. This tool supports the technical actor who is in charge of the new AR OP creation, by providing him power wizards, resources and inventory management, node-based UI to describe structure and properties of each operation, step, action and relationship, etc.

Services
- Augmented Reality Operating Procedure creation
- Storage of AR OP in Satisfactory Repository (CIDEM) ready to use with AR Visualization Tools

Dependencies
- SatisFactory Repository (CIDEM)

Inputs needed
- Original Operating Procedures (e.g. specific use-cases by COMAU, Sunlight, CERTH) and all related contents (e.g. images, 3D models, videos, etc...)

Outputs provided
- Augmented Reality Operating Procedure (AR OP), ready to be used by AR Visualization and AR Training Visualization Tools

Related Tasks
- Task 2.5: Development of “on job” training/education environment
- Task 4.3: Interactive augmented reality services and data analytics techniques

SatisFactory Platform Layer
- Facility Layer

Type of exposed APIs (if exists)
AR Op Creation Tool doesn’t expose any interface.
Sub-component Name
AR OP "On-The-Job" Creation Tool

Description
This Creation Tool allows to improve the above discussed version, in order to support a new, "on the job" approach in description of the operating procedures themselves. By using this tool directly in the work area in which the operating procedure will be performed, the procedure designer, together to an expert and/or a trained worker, can carry out the description of a new AR OP by following a specific operative protocol, based on the alternation of manual input to the software tool and the direct execution of the actions to do. Other than the necessary operative protocol, the real enhancement of this tool, is the ability to manage data and information directly captured from the work environment at runtime, by using specific sensors of the Smart Sensor Network. The most important managed data are: motion captured data (e.g. worker’s actions), images (e.g. for components, equipment, tools, actions), videos (e.g. actions), audios (e.g. further information and/or warnings).

Services
- Augmented Reality Operating Procedure creation
- Storage of AR OP in Satisfactory Repository (CIDEM) ready to use with AR Visualization Tools

Dependencies
- Satisfactory Repository (CIDEM)
- Middleware

Inputs needed
- (if present) Original Operating Procedures (e.g. specific use-cases by COMAU, Sunlight, CERTH) and all related contents (e.g. images, 3D models, videos, etc.)
- (potentially) Semantic information about procedures and/or part of them, present in CIDEM
- Data and information acquired on the job

Outputs provided
- Augmented Reality Operating Procedure (AR OP), ready to be used by AR Visualization and Training Tools

Related Tasks
- Task 2.5: Development of “on job” training/education environment
- Task 4.3: Interactive augmented reality services and data analytics techniques

Satisfactory Platform Layer
- Facility Layer
Type of exposed APIs (if exists)

AR Op Creation Tool doesn't expose any interface.
Sub-component Name
Operational Platform with Augmented Intelligence

Description
It is responsible for providing real-time diagnostics and control actions to the operators at nominal conditions of shop floor operation. The platform communicates directly with the middleware component and the DSS system in order to derive the optimum actions based on the current state of the machines and the worker’s behaviour. It is responsible for providing an overview of KPIs, personalized notes and tasks that are adjusted per worker. Additionally the AR platform will integrate and present the Ontology Manager and the Collaborative tools through an easy to use and intuitive interface.

Services
- Providing real-time diagnostics
- Providing real-time control actions
- Overview of KPIs, personalized notes and tasks that are adjusted per worker

Dependencies
- Ontology Manager
- Middleware
- Integrated DSS
- Collaborative Tools
- CIDEM

Inputs needed
- Smart sensor network data & models (through middleware)
- Semantic information from the Ontology Manager
- Events and actions from the Integrated DSS
- Information from the Collaborative tools
- Information from the AR in-factory platform

Outputs provided
- Real-time diagnostics and control actions
- Overview of the KPIs
- Personalized notes and tasks that are adjusted per worker
- Presentation of semantic context through intuitive interfaces
- Presentation of collaborative tools through intuitive interfaces

Related Tasks
- Task 1.3: Use cases and Scenarios
- Task 1.4: Common Information Data Exchange Model (CIDEM)
- Task 2.2: Knowledge Modelling to Support the Human Resource Optimization
- Task 3.3: Context-Aware Incident Detection Engine to Increase Worker’s Comfort
- Task 3.5: Shop-Floor Feedback Engine and Integrated DSS
- Task 4.1: Development of Intelligent IoT Infrastructure to Enable Context-Aware Applications
- Task 4.3: Interactive Augmented Reality Services and Data Analytics Techniques
- Task 4.4: Development of Intelligent Notification and Control Middleware
- Task 5.1: Preparation and Component Integration to Shop Floor
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators

SatisFactory Platform Layer
- Service Layer

Type of exposed APIs (if exists)
The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform.
Sub-component Name
Social Interaction and Cooperation

Description
The Social Interaction and Cooperation component provides a platform that workers can use to interact with their (remote) colleagues. It can be used to cooperate on work related issues, e.g., requesting for help for a particular problem or for sharing off-topic interests.

Services
- Means of communication
- Sharing media and text
- Storage
- Search

Dependencies
- Middleware
- possibly Localization Manager
- Adaptable user interfaces

Inputs needed
- Network connection through the middleware
- Content from users
- Possibly location information of users

Outputs provided
- Communication channels
- Platform for sharing, storing and searching for media and text

Related Tasks
- T2.3: Social interaction and gamification development to increase attractiveness
- T3.4: Intrafactory collaborative applications for work process support

SatisFactory Platform Layer
- Service Layer

Type of exposed APIs (if exists)
Sub-component Name
Gamified-based Engine & Tools

Description
The gamified process support primarily integrates concepts that are typically known from games into work processes. Alternatively new processes can be created.

Services
- None in the sense of technical services

Dependencies
- Social interaction and cooperation
- Training educational environment
- Ontology Manager
- Gamification augmented reality
- Multimodal & augmented HMI's and AR devices

Inputs needed
- Existing processes
- Defined targets
- Technology capabilities

Outputs provided
- Process concepts

Related Tasks
- T2.3: Social interaction and gamification development to increase attractiveness
- T3.4: Intrafactory collaborative applications for work process support

SatisFactory Platform Layer
- Service Layer

Type of exposed APIs (if exists)
Sub-component Name

Human Resources Re-Adaptation Toolkit

Description

This component is a visual tool for the supervision of activities at the shop floor level. It represents the building map and the location of personnel inside the facility. It also provides important information about progress status of activities and embeds a notification system. A second view allows analysing the work schedule view with timely responsiveness to schedule re-adaptation and access to historical information.

Services

- Visualization of information coming from the shop floor level.
- Work schedule view for scheduled jobs.

Dependencies

- Gestures & Content Recognition Manager
- Multiple Media Manager
- IDSS
- Middleware
- Repository
- Localization Manager

Inputs needed

- Video streaming and deferred video
- Notifications
- Building map
- Content information
- Localization with reference on the map origin

Outputs provided

- Visualization of aggregated information
- WebRTC end-to-end communication (not yet integrated)

Related Tasks

- T2.4: Re-Adaptation of existing facilities and Human Resource workload balancing

SatisFactory Platform Layer

- Services Layer
- Visualization Layer

Type of exposed APIs (if exists)
Sub-component Name
AR OP Visualization Training Tool (Mobile Version)

Description
This tool provides a further specialization on AR OP Visualization Tool in order to include the typical human interaction between the actors involved in a training and educational session (e.g. teacher-trainee, supervisor-worker, experienced-beginner). Furthermore it directly obtains procedure data by AR Creation Tools and include a specific integration with Training Data Analytics Tools and with Collaboration Tools, in order to implement a complete Training Platform. The tools allow training “on the job” and can manage both several kinds of operative procedures. Training tools make use of all functionalities already implemented in the AR In-Factory Platform Tools, both for the creation and for the presentation of the procedures. Moreover training tools take more closely in account specific scenarios and role based environments.

Services
- Delivering "on the job“ training and educational functionalities
- Support to a bidirectional flow of data and information between different actors (e.g. supervisor-worker, experienced-beginner)
- Support to asynchronous visualization of common or emergency indications to the operators, deriving from external modules (e.g. DSS)

Dependencies
- SatisFactory Repository
- AR OP Creation Tools
- AR OP Visualization Tools
- Middleware
- Hardware Devices Capabilities & Performances

Inputs needed
- (Training) Augmented Reality Operating Procedure (AR OP)
- Data obtained from Middleware at runtime
- (No needs but possibilities) Data available from external modules

Outputs provided
- Audio/Visual information shown by implemented HMI
- Log Data file for each training session to save in CIDEM and to provide to Training Data Analytics Tools

Related Tasks
- Task 2.5: Development of “on job“ training/education environment
- Task 3.4: Intrafactory collaborative applications for work process support
- Task 4.3: Interactive augmented reality services and data analytics techniques
SatisFactory Platform Layer

- Service Layer

**Type of exposed APIs (if exists)**

Since the Visualization Tools are stand-alone mobile Apps, they can communicate with the other SatisFactory modules mainly via Middleware. The exposed APIs in fact, are implemented as two specialized services, registered on Middleware, provided in order to exploit Operative Procedure Management and View Channel Management. More specialized interfaces addressed to communication by Collaboration Tools, are going to be designed.
Sub-component Name
AR OP Visualization Training Tool (GlassUP Version)

Description
This version of Visualization Training Tool provides the necessary porting of above discussed mobile version to GlassUP’s devices. In particular it presents a new dedicated HMI, in order to maximize the capabilities of AR devices of GlassUP (glasses, video camera, input device in primis). Furthermore it introduces a new strategies in filtering and presentation of contents by respecting the possibility of display of the devices themselves.

Services
- Delivering “on the job” training and educational functionalities
- Support to a bidirectional flow of data and information between different actors (e.g. supervisor-worker, experienced-beginner)
- Support to asynchronous visualization of common or emergency indications to the operators, deriving from external modules (e.g. DSS)

Dependencies
- SatisFactory Repository
- AR OP Creation Tools
- AR OP Visualization Tools
- Middleware
- Hardware Devices Capabilities & Performances

Inputs needed
- (Training) Augmented Reality Operating Procedure (AR OP)
- Data obtained from Middleware at runtime
- (No needs but possibilities) Data available from external modules

Outputs provided
- Audio/Visual information shown by implemented HMI
- Log Data file for each training session to save in CIDEM and to provide to Training Data Analytics Tools

Related Tasks
- Task 2.5: Development of “on the job” training/education environment
- Task 3.4: Intrafactory collaborative applications for work process support
- Task 4.3: Interactive augmented reality services and data analytics techniques

SatisFactory Platform Layer
- Service Layer
**Type of exposed APIs (if exists)**

Since the Visualization Tools are stand-alone mobile Apps, they can communicate with the other SatisFactory modules mainly via Middleware. The exposed APIs in fact, are implemented as two specialized services, registered on Middleware, provided in order to exploit Operative Procedure Management and View Channel Management. More specialized interfaces addressed to communication by Collaboration Tools, are going to be designed.
Sub-component Name
Training Data Analytics Tool

Description
This component allows to analysis of data collected and stored during a training session, in order to provide analytical results about performances of trainees, influence of asynchronous events, comparison with previous sessions, etc. Even if part of functionalities of this component are addressed to the post processing of acquired data, some of them can be applied in real-time and give the chance to visualize important feedback to trainee at runtime.

Services
- Analysis of the acquired data during training sessions (e.g. timestamp, events, info, etc.), both in realtime and offline
- Visualization of derivated data and information about single and/or multi training sessions and on single-trainee basis and/or about trainees belonging to a specific category of users

Dependencies
- SatisFactory Repository (CIDEM)
- AR OP Visualization Training Tools
- (potentially only) Other components reachable via Middleware

Inputs needed
- Log Data provided by AR Visualization Training Tools
- Previously stored data and comparisons
- Other information present in CIDEM

Outputs provided
- Visual representation of trainees performance
- Visual representation of comparison between several training sessions, related to single trainee and group of comparable trainees
- Specific analytics results

Related Tasks
- Task 2.5: Development of “on job” training/education environment
- Task 4.3: Interactive augmented reality services and data analytics techniques

SatisFactory Platform Layer
- Services Layer

Type of exposed APIs (if exists)
The use of RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform.
**Sub-component Name**

AR OP Creation Tools (by AR Platform)

**Description**

The Creation Tool allows to convert original Operating Procedures, often described with different formalisms, in a new format, called AR OP (Augment Reality Operating Procedure) ready to be used by the AR Visualization Tools developed to support both training and operational environments. The Creation Tools provide functionalities designed to manually describe a new AR OP by a dedicated approach. This tool supports the technical actor who is in charge of the new AR OP creation, by providing him power wizards, resources and inventory management, node-based UI to describe structure and properties of each operation, step, action and relationship, etc.

**Services**

- Augmented Reality Operating Procedure creation
- Storage of AR OP in Satisfactory Repository (CIDEM) ready to use with AR Visualization Tools

**Dependencies**

- SatisFactory Repository (CIDEM)

**Inputs needed**

- Original Operating Procedures (e.g. specific use-cases by COMAU, Sunlight, CERTH) and all related contents (e.g. images, 3D models, videos, etc...)

**Outputs provided**

- Augmented Reality Operating Procedure (AR OP), ready to be used by AR Visualization and AR Training Visualization Tools

**Related Tasks**

- Task 2.5: Development of “on job“ training/education environment
- Task 4.3: Interactive augmented reality services and data analytics techniques

**SatisFactory Platform Layer**

- Facility Layer

**Type of exposed APIs (if exists)**

AR Op Creation Tool doesn't expose any interface.
Sub-component Name
Visual Analytics Module

Description
This subcomponent presents different state-related views of the supervised shop floor. It will provide the capability to compare and assess the situation of the workers in a visual and comprehensive way taking into account properties of human cognition, perception and reasoning. It will combine, correlate and visualize large, complex and heterogeneous data providing a multi-factorial exploration in the spatio-temporal domain, assisting end-users/managers to detect patterns, templates and crucial points that are difficult to detect otherwise.

Services
- Analysis of the shop floor heterogeneous information
- Evaluation of the situation of the workers
- Visualization in the spatio-temporal domain of the shop floor information

Dependencies
- Middleware
- DSS-core module

Inputs needed
- Smart Sensor Network data & models (through middleware)

Outputs provided
- Visualizations regarding the shop floor operation
- Visualization for the evaluation of the shop floor and its operation status

Related Tasks
- Task 1.2: Models for Actors and Procedures Interconnections
- Task 1.3: Use cases and Scenarios
- Task 2.2: Knowledge Modelling to Support the Human Resource Optimization
- Task 3.5: Shop-floor Feedback Engine and Integrated DSS
- Task 5.3 Deployment of Industry lab use case
- Task 5.4 Industrial pilot demonstrators

SatisFactory Platform Layer
- Decision Layer

Type of exposed APIs (if exists)
The use of a RESTful API ensures effective communication with heterogeneous systems, making the integration process simple and independent from the underlying operating platform. Furthermore, visualizations and interactions through a human friendly environment ensure the usability of the module in the underlying platform.
### Sub-component Name

Glasses UI

### Description

This subcomponent belongs to the main component of multimodal & augmented HMI systems and AR devices. Together with other HMI devices, UI glasses on one side enable Augmented Reality (AR) services both to real (AR) and virtual environments (VR) for a richer experience. The relative AR platform feeds with augmented information services both the Training Educational Environment and the Operational Platform with Augmented Intelligence.

### Services

- Augmented reality information displayed on the lens of the glasses.

### Dependencies

- AR In-factory platform

### Inputs needed

- Data to be displayed coming from SatisFactory components such as Integrated DSS and sensors through the Middleware and Event Manager
- Events coming from SatisFactory components (e.g., AR-In Factory Platform, Gesture & Content Recognition Manager and Localization Manager).

### Outputs provided

- Augmented info overlaid on operators’ field of view: guidance and instruction
- Video streaming and audio or images to SatisFactory components (e.g., AR-In Factory Platform, Gesture & Recognition Manager)

### Related Tasks

- Task 2.3: Social interaction and gamification development to increase attractiveness
- Task 4.1: Development of intelligent IoT infrastructure to enable context-aware applications
- Task 4.2: Hardware development for context-aware HMI systems (main reference task)
- Task 4.3: Interactive augmented reality services and data analytics techniques
- Task 4.5: Integration of social collaboration and sensing environments for enhancing collective awareness

### SatisFactory Platform Layer

- Attractive UI Layer

### Type of exposed APIs (if exists)

- Open REST-like API interface
Sub-component Name
Web Based UI

Description
This HTML5-based application will be available as a kiosk on a tablet (workers) and as a browser-app (manager). It will display a list of suggestions in the system and offer the possibility to add new suggestions. The manager web-app additionally will offer CRUD-functionalities as well as the possibilities to manipulate the status of the suggestion (accept, reject, modify). Also an up- and down voting system will be included.

Services
- Display suggestions
- Up-/Downvote suggestions
- Create suggestion
- Change suggestion status (manager)

Dependencies
- Gamification Framework

Inputs needed
- Previously stored suggestions

Outputs provided
- Suggestion list

Related Tasks
- T2.3: Social interaction and gamification development to increase attractiveness
- T3.4: Intrafactory collaborative applications for work process support

SatisFactory Platform Layer
- Attractive UI Layer

Type of exposed APIs (if exists)
None.