**DELIVERABLE D6.1 - Detailed Specification of Usage Scenarios and Planning of Validation Activities - First version**

<table>
<thead>
<tr>
<th>Deliverable Number</th>
<th>D6.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliverable Name</td>
<td>Detailed Specification of Usage Scenarios and Planning of Validation Activities - First version</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Type of Document</td>
<td>Report</td>
</tr>
<tr>
<td>Contractual date of delivery</td>
<td>30/09/2018</td>
</tr>
<tr>
<td>Deliverable Leader</td>
<td>iSPRINT</td>
</tr>
<tr>
<td>Status &amp; version</td>
<td>1.2</td>
</tr>
<tr>
<td>WP / Task responsible</td>
<td>WP6 (iSPRINT) / Task T6.1 (iSPRINT)</td>
</tr>
<tr>
<td>Keywords:</td>
<td>Usage scenarios, validation planning</td>
</tr>
<tr>
<td>Abstract (few lines):</td>
<td>The purpose of this deliverable is to provide detailed descriptions of each of the usage scenarios of the SecureIoT project, the related use cases and the validation planning.</td>
</tr>
</tbody>
</table>

**Deliverable Leader:** Sofoklis Kyriazakos (iSPRINT)

**Contributors:** LuxAI, FUJITSU, IDIADA

**Reviewers:** IDIADA, INRIA

**Approved by:**

---

This document is part of a project that has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 779899. It is the property of the SecureIoT consortium and shall not be distributed or reproduced without the formal approval of the SecureIoT Management Committee.
1 Executive Summary

The purpose of this deliverable is to provide detailed descriptions of each of the usage scenarios of the SecureIoT project, the related use cases and the validation planning. The detailed specification presented in this document defines the actors and systems involved, the interfaces between them, as well as how the SecureIoT services will be integrated and used in each use case. For each of them, multiple alternative flows are presented, along with the conditions under which the tests will be carried out. Further to that, a detailed methodology for executing alternative tests under different operational conditions and deployment configurations is presented. As part of this deliverable, the KPIs of each use case are detailed. Based on the detailed specification of the scenarios and their operational settings, the document presents preparatory activities needed prior to commencing the testing and validation. This includes ensuring the availability of software and hardware, training users on the usage scenarios, planning the proper IT environments for deploying the cyber-security solutions and more with the ultimate goal to ensure that the use case sites will be ready for validating the SecureIoT solutions and for carrying out the planned test cases.

Deliverable 6.1 presents the first version of the Detailed Specifications of Usage Scenarios and Planning of Validation Activities. The final version will be delivered 9 months later (M18) in deliverable 6.2.
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Contributor(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01/03/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Initial ToC</td>
</tr>
<tr>
<td>0.2</td>
<td>15/03/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Final ToC</td>
</tr>
<tr>
<td>0.3</td>
<td>14/05/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Document using deliverable template</td>
</tr>
<tr>
<td>0.7</td>
<td>17/08/2018</td>
<td>Sofoklis Kyriazakos, Pouyan Ziafati, Hendrik Eikerling, Daniel Calvo Alonso,</td>
<td>Expanded use case scenarios</td>
</tr>
<tr>
<td></td>
<td></td>
<td>David Evans</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>14/09/2018</td>
<td>Hendrik Eikerling</td>
<td>Added Industry 4.0 Input and Feedback</td>
</tr>
<tr>
<td>0.9</td>
<td>18/09/2018</td>
<td>Sofoklis Kyriazakos, Pouyan Ziafati, Razvan Craciunescu, Kyriazi, Dres, Hendrik</td>
<td>First complete version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Eikerling, Daniel Calvo Alonso, David Evans, Ali Paikan</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>26/09/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Final version for approval</td>
</tr>
<tr>
<td>1.1</td>
<td>27/09/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Final version after internal review</td>
</tr>
<tr>
<td>1.2</td>
<td>28/09/2018</td>
<td>Sofoklis Kyriazakos</td>
<td>Final version</td>
</tr>
<tr>
<td>1.3</td>
<td>28/09/2018</td>
<td>Mariza Konidi</td>
<td>Final version to be submitted</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Executive Summary ........................................................................................................... 2
2. Definitions, Acronyms and Abbreviations ......................................................................... 9
3. Introduction ......................................................................................................................... 10
   3.1 Relation with other deliverables .................................................................................... 10
4. Detailed Specifications of Usage Scenarios ........................................................................ 12
   4.1 Methodology .................................................................................................................. 12
   4.2 Specification iterations and planning .............................................................................. 12
5. Usage scenarios in SecureIoT ............................................................................................. 14
   5.1 Multi-Vendor Industry 4.0 ............................................................................................. 14
      5.1.1 Description ............................................................................................................... 14
      5.1.2 Use cases (abuse cases) involved ............................................................................ 15
   5.2 Connected Car and Autonomous Driving ...................................................................... 33
      5.2.1 Description ............................................................................................................... 33
      5.2.2 Use cases (abuse cases) involved ............................................................................ 38
      5.2.3 Overview of IoT Assets .......................................................................................... 60
   5.3 Socially Assistive Robots and IoT Applications .............................................................. 62
      5.3.1 Description ............................................................................................................... 62
      5.3.2 Scenarios and use cases description ....................................................................... 67
      5.3.3 Use cases specifications .......................................................................................... 74
6. Validation and testing ......................................................................................................... 105
   6.1 Validation and testing for Multi-Vendor Industrie 4.0 (MVI) ........................................ 105
      6.1.1 Methodology ........................................................................................................... 105
   6.2 Validation and testing for IoT Applications and Connected Car and Autonomous
       Driving (CCAD) ............................................................................................................. 108
      6.2.1 Methodology ........................................................................................................... 108
   6.3 Validation and testing for Socially Assistive Robots and IoT Applications (SAR) .... 116
      6.3.1 Methodology ........................................................................................................... 116
7. Integration Guidelines ........................................................................................................ 127
   7.1 Data Collection and Actuation Layer ............................................................................. 127
   7.2 Security Intelligence layer ............................................................................................. 128
7.2.1 Typical Action Types / Verbs

7.2.2 Specific API examples

7.3 SECaaS layer

7.4 General guidelines for 3rd party vendor integration

8 Conclusions

9 References
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>RELATION OF D6.1 TO OTHER DELIVERABLES</td>
<td>10</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>RELATION OF WP2 AND TASK 6.1</td>
<td>11</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>MULTI-vendor INDUSTRY 4.0</td>
<td>14</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>HIGH LEVEL OVERVIEW OF V2internet SCENARIO</td>
<td>35</td>
</tr>
<tr>
<td>FIGURE 5</td>
<td>EXAMPLE OF VEHICLE OBU FUNCTIONAL ELEMENTS</td>
<td>35</td>
</tr>
<tr>
<td>FIGURE 6</td>
<td>LOGICAL VIEW FOR EDGE LAYER (VEHICLE)</td>
<td>36</td>
</tr>
<tr>
<td>FIGURE 7</td>
<td>LOGICAL VIEW FOR CLOUD LAYER (FIWARE – IOT PLATFORM)</td>
<td>37</td>
</tr>
<tr>
<td>FIGURE 8</td>
<td>LOGICAL VIEW FOR APPLICATION LAYER (AI MODELS FOR SCENARIOS)</td>
<td>37</td>
</tr>
<tr>
<td>FIGURE 9</td>
<td>IDIADA IDAPT OBU</td>
<td>61</td>
</tr>
<tr>
<td>FIGURE 10</td>
<td>FIWARE BASED REFERENCE ARCHITECTURE FOR THE IOT PLATFORM OF T6.4</td>
<td>62</td>
</tr>
<tr>
<td>FIGURE 11</td>
<td>OVERVIEW OF THE SYSTEM AND INVOLVED ACTORS IN THE Socially Assistive Robots AND IOT APP SCENARIOS</td>
<td>66</td>
</tr>
<tr>
<td>FIGURE 12</td>
<td>QTRobot SOFTWARE STACK DIAGRAM</td>
<td>66</td>
</tr>
<tr>
<td>FIGURE 13</td>
<td>CC2U SOFTWARE STACK DIAGRAM</td>
<td>67</td>
</tr>
<tr>
<td>FIGURE 14</td>
<td>STEPS NEEDED TO RUN A SIMULATION USING SUMO TOOL</td>
<td>112</td>
</tr>
<tr>
<td>FIGURE 15</td>
<td>TECHNOLOGY ACCEPTANCE QUESTIONNAIRES FOR THE EXPERIMENT</td>
<td>123</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** ...................................................................................................................... 1
2. **Requirements** ...................................................................................................................... 5
3. **Design** ................................................................................................................................... 7
4. **Implementation** ..................................................................................................................... 9
5. **Validation** ............................................................................................................................ 11
6. **Conclusion** .......................................................................................................................... 13
7. **Appendices** .......................................................................................................................... 15

---

# Table of Tables

**Table 1:** Usage case descriptions vs. IEC 62559-2 ........................................................................ 13
**Table 2:** Usage Scenarios ........................................................................................................... 16
**Table 3:** Use case UC1.1 ........................................................................................................... 17
**Table 4:** Usage Scenarios ........................................................................................................... 21
**Table 5:** Use case UC2.1 ........................................................................................................... 22
**Table 6:** Use case UC2.1 ........................................................................................................... 27
**Table 7:** Use case UC3.1 ........................................................................................................... 28
**Table 8:** Use case UC3.2 ........................................................................................................... 31
**Table 9:** Main actors involved in connected car and autonomous driving use-cases ............... 33
**Table 10:** Use case UC1.1 ........................................................................................................ 38
**Table 11:** Use case UC3.1 ........................................................................................................ 42
**Table 12:** Use case UC3.2 ........................................................................................................ 45
**Table 13:** Use case UC3.3 ........................................................................................................ 48
**Table 14:** Use case UC3.4 ........................................................................................................ 51
**Table 15:** Use cases UC3.5 / 3.6 ................................................................................................. 54
**Table 16:** Use case UC3.7 ........................................................................................................ 57
**Table 17:** Summary of IoT assets for connected and autonomous driving scenarios ............... 60
**Table 18:** Use cases UC1.1 ........................................................................................................ 63
**Table 19:** Cognitive & physical games scenario (SAR-S1) and relevant use cases ..................... 69
**Table 20:** Monitoring & checkups scenario (SAR-S2) and relevant use cases ......................... 72
**Table 21:** Daily calendar and system admin scenario (SAR-S3) and relevant use cases .......... 73
**Table 22:** Companionship – user friendly interface to digital life scenario (SAR-S4) and relevant use cases .............................................................................................................. 74
**Table 23:** Use cases UC1.2 ........................................................................................................ 76
**Table 24:** Use cases UC1.3 ........................................................................................................ 78
**Table 25:** Use cases UC1.4 ........................................................................................................ 81
**Table 26:** Use cases UC2.1 ........................................................................................................ 83
**Table 27:** Use cases UC2.2 ........................................................................................................ 86
**Table 28:** Use cases UC2.3 ........................................................................................................ 88
**Table 29:** Use cases UC2.4 ........................................................................................................ 91
**Table 30:** Use cases UC3.1 ........................................................................................................ 94
**Table 31:** Use cases UC3.2 ........................................................................................................ 96
**Table 32:** Use cases UC3.3 ........................................................................................................ 99
**Table 33:** Use cases UC4.1 ........................................................................................................ 101
**Table 34:** Industry 4.0 scenario release plan ................................................................................ 105
**Table 35:** Industry 4.0 integrated usage scenario requirements to risk assessment and mitigation service .............................................................................................................................. 107
**Table 36:** Industry 4.0 requirements for programming support services ................................ 107
**Table 37:** Industry 4.0 requirements to IoT security knowledge base ...................................... 108
**Table 38:** Release plan for connected and autonomous driving scenarios (T6.4) ...................... 108
**Table 39:** Sprint plan for release #1 of connected and autonomous driving scenarios (T6.4) .... 109
**Table 40:** Car and autonomous driving use-cases requirement to risk assessment and mitigation service .............................................................................................................................. 113
**Table 41:** Car and autonomous driving use-cases requirement to compliance auditing service .... 114
**Table 42:** Car and autonomous driving use-cases requirement to programming support services ................................................................................................................................. 115
**Table 43:** Car and autonomous driving use-cases requirement to IoT security knowledge base ................................................................................................................................. 116
**Table 44:** Release plan for socially assistive robots and IoT applications scenarios (T6.4) ........ 117
TABLE 46: SPRINT PLAN FOR RELEASE#1 OF SOCIALLY ASSISTIVE ROBOTS AND IOT APPLICATIONS SCENARIOS (T6.4) ..........................118
TABLE 47: SOCIALLY ASSISTIVE ROBOTS AND IOT APPLICATIONS USE-CASES REQUIREMENT TO RISK ASSESSMENT AND MITIGATION SERVICE ..................................................................................................................................................123
TABLE 48: SOCIALLY ASSISTIVE ROBOTS AND IOT APPLICATIONS USE-CASES REQUIREMENT TO COMPLIANCE AUDITING SERVICE ..........125
TABLE 49: SOCIALLY ASSISTIVE ROBOTS AND IOT APPLICATIONS USE-CASES REQUIREMENT TO PROGRAMMING SUPPORT SERVICES .....126
TABLE 50: SOCIALLY ASSISTIVE ROBOTS AND IOT APPLICATIONS USE-CASES REQUIREMENT TO IOT SECURITY KNOWLEDGE BASE ..........126
## 2 Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBBA</td>
<td>Consensus-Based Bundle Algorithm</td>
</tr>
<tr>
<td>CC2U</td>
<td>CloudCare2U</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>QT</td>
<td>QTrobot</td>
</tr>
<tr>
<td>UC</td>
<td>Use Case</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle-to-Infrastructure</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-Vehicle</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
3 Introduction

The scope of the deliverable is to present the detailed description of each of the usage scenarios, the related use cases and validation planning. The methodology that has been followed is based on splitting a usage scenario into discrete scenarios that describe a wider range of activities and process that may occur. For each of the scenarios, there are numerous use/abuse-cases involved, which are describing a single action or process. This description is based on the presentation of the actors and systems involved, interfaces, the integration of SecureIoT services, pre-/post-conditions, main/alternative flow of events, use case diagram (UML), data collection process, common tools to capture data, security gaps & risks, threats and the desired security level. In addition, the validation process is presented with information on conditions, hardware/software involved, methodology, data captured, security gaps & risks and Key Performance Indicators (KPIs).

The deliverable is structured around seven sections. Sections 1-3 present the overview of the report with an executive summary, definitions, acronyms and abbreviations and the presentation of the deliverable in the introduction. Section 4 is the detailed specification of usage scenarios of SecureIoT and presents the methodology, the specification iterations and the planning. Section 5 is the major part of this document and presents the three usage scenarios of SecureIoT, namely Multi-Vendor Industrie 4.0; Connected Car and Autonomous Driving; Socially Assistive Robots and IoT Applications. Each of these scenarios is described, with information about the attacker motivational rationale, a benchmarking, presentation of metrics & KPIs in both ways of integration, the use/abuse cases involved and its validation. Section 6 elaborates on the validation methodology, while section 7 presents the generic integration guidelines. Finally, section 8 sums up the conclusions and section 9 lists the references.

3.1 Relation with other deliverables

Deliverable D6.1 is the first version of detailed specification of the SecureIoT usage scenarios and the planning of validation activities, delivered in month 9 (M9) of the project.
The final version will be available 9 months later (M18) in D6.2. In Figure 1, the relation of D6.1 other project deliverables in its pathway toward D6.2 is presented. The documents that are precedent to D6.1 are D2.1 and D2.2 that describe security requirements and the stakeholders’ analysis. Both WP2 deliverables (and relevant tasks) are linked with D6.1 and therefore Task 6.1, as presented in Figure 2. More specifically, in task 2.1 the threats in project reference scenarios are defined and requirements are derived. In task 2.2 the security/privacy and trust requirements are presented, as an outcome of a process that consider the major stakeholders. Task 6.1 provides the detailed specifications of the use cases, linked with the usage scenarios, as part of the project reference scenarios.

![Diagram](image)

**Figure 2: Relation of WP2 and Task 6.1**

Deliverable D6.1 is using input from other activities and tasks of the project, as presented above, but is also contributing to several WPs. The detailed description of the Use/Abuse Cases and the information about the security services to be used is an information that contributes to the design and implementation activities of WP2/3/4/5, while the integration guidelines (section 7) provide useful input about the functional requirements of SecureIoT and the SECaaS services.
4 Detailed Specifications of Usage Scenarios

4.1 Methodology
Use cases are building blocks for projects in software engineering and describe the developed system and its functionalities in static as well as dynamic aspects. According to [4] definition of use case is: *A use case is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system.*

Use cases and their functionalities are meaningful when presented in a structured and organized manner. One standard that is often used in IEC 62559-2 [5]. The full standard template has eight sections, namely:

1. Description of the use case
2. Diagrams of use case
3. Technical details
4. Step by step analysis of use case
5. Information exchanged
6. Requirements
7. Common terms and definitions
8. Custom information

4.2 Specification iterations and planning
In the case of SecureIoT, following hierarchy and terminology has been agreed:

- Project **reference scenarios** (more generic approach of SecureIoT)
  - **Usage Scenarios** (3 main scenarios)
    - Multi-Vendor Industrie 4.0
      - **Scenario (S)**: several scenarios that occur under each usage scenario
        - **Use-cases (UC)**: building blocks of each Scenario
    - IoT Applications and Connected Car and Autonomous Driving and IoT Applications
      - **Scenario (S)**: several scenarios that occur under each usage scenario
        - **Use-cases (UC)**: building blocks of each Scenario
    - Socio-technically Assistive Robots and IoT Applications
      - **Scenario (S)**: several scenarios that occur under each usage scenario
        - **Use-cases (UC)**: building blocks of each Scenario
The Use-cases is the smallest “block” used in SecureIoT and we describe it with a number of parameters. In Table 1 the use-case parameters are presented against IEC 62559-2 recommendations.

**Table 1: Usage Case descriptions vs. IEC 62559-2**

<table>
<thead>
<tr>
<th>UC descriptions in Secure IoT</th>
<th>IEC 62559-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of UC / ID</td>
<td>Description of the use case</td>
</tr>
<tr>
<td>Description/scope</td>
<td></td>
</tr>
<tr>
<td>Validation / KPIs</td>
<td></td>
</tr>
<tr>
<td>Pre-/post-conditions</td>
<td></td>
</tr>
<tr>
<td>Use case diagram (UML)</td>
<td>Diagrams of use case</td>
</tr>
<tr>
<td>Actors involved</td>
<td>Technical details</td>
</tr>
<tr>
<td>Systems involved</td>
<td></td>
</tr>
<tr>
<td>Interfaces</td>
<td></td>
</tr>
<tr>
<td>Integration of SecureIoT services</td>
<td></td>
</tr>
<tr>
<td>Main flow of events</td>
<td>Step by step analysis of use case</td>
</tr>
<tr>
<td>Alternative flows</td>
<td></td>
</tr>
<tr>
<td>Data collection</td>
<td>Information exchanged</td>
</tr>
<tr>
<td>Pre-/post-conditions</td>
<td>Requirements</td>
</tr>
<tr>
<td>Definitions</td>
<td>Common terms and definitions</td>
</tr>
<tr>
<td>Security gaps &amp; risks</td>
<td>Custom information</td>
</tr>
<tr>
<td>Threats</td>
<td></td>
</tr>
<tr>
<td>Security level</td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td></td>
</tr>
</tbody>
</table>
5 Usage scenarios in SecureIoT

5.1 Multi-Vendor Industry 4.0

5.1.1 Description

The integrated Industry 4.0 scenario is comprised of three distinct scenarios spanning different levels within the IoT context. The first level is the shop floor with machinery that produces a product and machine operators who control the machines and have access to human machine interfaces such as head mounted displays or IoT devices in general. In this case, the machinery involved is an injection molding machine producing electrical connectors.

The second level consists of the IoT Platform, which is hosted within the organization that owns the shop floor. The first and second level are considered the “Local” environment. Lastly, there can also be IoT services provided by external organizations. This third level is considered the “Outbound” environment.

On the shop floor level, the injection molding machine is controlled by the machine operator. In addition, it receives control instructions from the IoT Platform and send diagnostic data to it. The machine operator uses an IoT device with a human machine interface, i.e., a head mounted display. This device guides the machine operator and receives commands and feedback from the operator in turn, e.g., through speech inputs. It may also collect other data, such as location data and visual data.

To guide the machine operator, the head mounted display receives instructions from the IoT Platform and sends the machine operator’s input to the platform for processing. Responsible for
Data processing on the IoT platform are microservices, each with a specific set of responsibilities. For example, one microservice may be responsible for speech processing, while another is responsible for sending control commands to the machinery on the shop floor, and yet another guides the machine operator.

In addition to the local IoT Platform microservices, it may be possible to integrate third party services from partners outside of the local organization. These services may also influence the control signals from the IoT platform to the shop floor machinery.

One example of an outbound service is a product configurator. This configurator is used by engineers to virtually assemble a product from supplier provided parts. The result is a product configuration file, that can be sent to a production machine to assemble it. In this case, the product configuration file is sent to an extractor, which extracts parts specifications from the latter so that the injection molding machine can produce the required part.

5.1.2 Use cases (abuse cases) involved

5.1.2.1 Head-Mounted Display (Fujitsu)

The Head Mounted Display (HMD) displays information related to the production process such as guidance, warnings, and advice. Information can be displayed permanently or when context-sensitively, e.g., based on the machine operator’s location or the current step in a workflow. Note that the HMD is not an augmented reality device and is solely meant to display informational elements.

The HMD is independent on external hardware for operation. It runs its own operating system (Android) with applications for, e.g., object recognition, speech recognition, location-based services etc. Additional applications can be downloaded from platforms provided by trusted partners, enabling the HMD to be configured at low cost by using existing ecosystems. This also enables the ad-hoc installation of applications to quickly fulfill arising requirements. To avoid man-in-the-middle attacks through manipulated third-party applications, trusted certifications are required.
The connectivity is provided by a common WiFi module. It supports the 802.11n, g, and h standards, supporting both the 5GHz or 2.4GHz frequency and both the 20 and 40 MHz bandwidth.

The information distributed to the operator is generated in the IoT cloud backend using data provided by the HMD. This data can be based on object recognition or speech input from the wearer. This requires the HMD to record and transmit audio and video during its operation. In addition, the HMD collects location information using either the integrated GSM module or a location based on triangulation and the location of mesh network access points.

In conclusion, HMDs can be used to guide workers by giving context sensitive instructions based on their location, object recognition, and speech input. Thus, complex workflows can be presented to an operator in an accessible manner and directed using intuitive (speech) commands.

Table 2: Usage Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>Real time data push (display) and pull (gathering) for operators to supervise guidance and instructions handling of machine-, sensor- and IoT parameter proceedings.</td>
</tr>
</tbody>
</table>

Use cases involved (*)

<table>
<thead>
<tr>
<th>Use case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
<tr>
<td>Use case 1.2 (abuse)</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 1.3 (abuse)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC1.3</td>
</tr>
<tr>
<td>Description/scope</td>
<td>Operator’s guiding instruction is compromised and compromise feature sets e.g. (installing backdoor, scanning etc.) – variation of UC1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 1.4 (abuse)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC1.4</td>
</tr>
<tr>
<td>Description/scope</td>
<td>An Intruder is compromising data back channels and rerouting information from HMD’s acoustic and video sensors to foreign hosts and locations (spy feature) – variation of UC1.1</td>
</tr>
</tbody>
</table>

**Table 3: Use case UC1.1**

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>S1/UC1.1</td>
</tr>
</tbody>
</table>

**Actors involved**

- Machine operator

**Systems involved**

- Machine towards manual worker operates with
- HMD Sensors (acoustic, video)
- HMD Display

**Interfaces**
- IT network (wireless data transmission / operation)
- OT network (location based services)

Integration of SecureIoT services

- Operators instruction generation from backbone and sources
- Data gathering from the HMD device on purpose (scan, control and report)
- Data supervision and report for visualization on dashboards or further decision processes (predictive related issues)

Pre-conditions

- HMD device has been linked to operator / ID recognition
- HMD device fully engaged by established network connectivity
- Operator trigger signals are being acquired

Main flow of events

- Instruction distribution: Backbone is establishing based on location based information and setup requirements (e.g. product order) corresponding handling instruction. Microservices are established either locally or within defined ecosystem (e.g. cloud, gateway etc.) in order to support object, speech gesture recognition. Based on conditions to proceed, instruction and additional information are displayed and may needs to be acknowledge or marked.
- Information back channel: HMD sensors (acoustic / optical) are continuously streaming information for analytics (e.g. acknowledgement, object and speech recognition etc.) purpose back to eco host

The above diagram depicts the main flow of events regarding speech recognition in a sequence diagram
- The machine operator's speech commands are recorded by the HMD and sent to Guidance component in the IoT cloud environment
- The speech data is processed by a third party microservice and the data is sent back to the Guidance component and in turn to the HMD
- Integration into the SecureIoT services is done with a probe recording speech audio data, processed speech data, and derived instructions

**Post-conditions & Alternative flows**

- Essentially same as pre-conditions
- Operator’s instruction distributed to machine directly or partly separated
- Manual operation and data interface usage.

**Use case diagram**
Data collection

- Data is acquired by IoT Gateway

Security gaps & risks

- Information distribution manipulation
- Information rerouting from device
- Manipulation of ID recognition and linkage between device and operator

Threats

- Tampering, Spoofing: An intruder manipulates incoming information and instructions lead to miss-/fail- handling of corresponding / addressed machine to operate
- Information Disclosure, Tampering: An intruder gathers data streams from HMD device to intercept real time traffic / acknowledgements etc. for manipulations of further distributions
- Information Disclosure: An intruder monitors location based information for profile generation and operator supervisions

Security level

SL1

5.1.2.2 Injection Molding

The smart factory of today represents a fully connected and flexible system where continuous streams of data provided by machines, sensors and IoT components are used for services like predictive maintenance and condition monitoring to optimize production flow. These services can be implemented on edge-, fog- or cloud level, depending on the specific application’s requirements, e.g. network latency.

In this example, we consider an injection molding machine that is connected to an IoT platform. It contains sensors that continuously monitor production data and provide it to other IoT components. These components perform pre-processing data analysis tasks and forward the results to an IoT cloud platform through an IoT gateway. The cloud platform then performs further data analysis functions to provide information on the injection molding process to the machine operator through a human machine interface. This data is sent from the IoT cloud through the gateway to the injection molding machine.

We assume that attacks predominantly involve remote network access (no physical access to the IoT device or infrastructure) by compromising vulnerable communication protocols, weak cryptography implementations, social engineering, and malware. Software attacks are relatively low cost to an attacker, although a highly targeted attack with company espionage background
may involve breaking a door lock to gain physical access to the system or bypassing Wi-Fi security safeguards (e.g. firewall). Aside from corporate espionage, other motivations for an attacker may be tampering or destroying machinery to interrupt production and manipulating machines in such a way, that the end product’s quality is sub-par, rendering the produced batch useless and unsellable. Usually, just a single compromised smart object among a set of interconnected IoT can provide unauthorized access to other smart objects.

An unauthorized user may gain access to the factory network and disclose the current production and sensor information. A conceivable attack scenario is the manipulation of sensor data in order to reduce the manufacturing performance. A more severe attack scenario is to gain direct machine control and thereby risk worker’s health or damage the machines.

An additional risk is the integration of foreign components into the machines. The machine park operator must rely on the trustworthiness of the manufacturer of IoT components. The implementation should be secure and not be manipulated (un-)intentionally.

In this scenario, we outline four cases. In UC2.1, data from machines, sensors and IoT components are pre-processed on edge-level and subsequently, Industrial Analytics services are run at edge- or cloud-level. In UC2.2, an IoT device is compromised and allows an intruder to monitor network traffic. In UC2.3, an intruder can additionally modify data of single sensors or IoT devices and thereby influence machine behavior. In UC2.4, an intruder can access a large number of IoT components and sensors and use them to attack other network components.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>52</td>
<td>Real-time machine-, sensor-, and IoT data are used for Industrial Analytics functions to optimize injection molding production flow on the shop floor and provide feedback to the machine operator.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use cases involved (*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use case 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>Description/scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC2.1</td>
<td>Continuous data from machines, sensors and IoT components are pre-processed on edge-level and subsequently, Industrial Analytics services are run at edge- or cloud-level</td>
</tr>
</tbody>
</table>
**Use case 1.2 (abuse)**

<table>
<thead>
<tr>
<th>id</th>
<th>UC2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description/scope</td>
<td>An IoT device is compromised and allows an intruder to monitor network traffic</td>
</tr>
</tbody>
</table>

**Use case 1.3 (abuse)**

<table>
<thead>
<tr>
<th>id</th>
<th>UC2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description/scope</td>
<td>An intruder can additionally modify data from a single sensor or IoT device and thereby influence machine behavior</td>
</tr>
</tbody>
</table>

**Use case 1.4 (abuse)**

<table>
<thead>
<tr>
<th>id</th>
<th>UC2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description/scope</td>
<td>An intruder can access a large number of IoT components and sensors and use them to attack other network components</td>
</tr>
</tbody>
</table>

---

**Table 5: Use case UC2.1**

<table>
<thead>
<tr>
<th>Scenario / use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
</tbody>
</table>

**Actors involved**

- Machine operator

**Systems involved**

- Machine
- Sensors
- IoT gateway and cloud components
Interfaces
- IT network
- OT network

Integration of SecureIoT services
- Data acquisition application on IoT Coupler/Gateway/Controller on edge-level
- Industrial Analytics services run on a platform in the cloud

Pre-conditions
- Machine is in production
- Signals are being acquired
- Network connection between systems is established

Main flow of events
- Real-time data of machine, sensors and IoT components are acquired and pre-processed on edge-level
- Basic Industrial Analytics functions are continuously run at edge-level, e.g. on the System-on-Chip (SoC) of the IoT Coupler/Gateway/Controller or an IPC
- The pre-processed data are sent to a platform-based service in the cloud
- Advanced Industrial Analytics functions are continuously run in the cloud
- Results of the edge- and cloud-based Industrial Analytics functions are displayed to the machine operator on a HMI display
- The diagram below shows the main flow of events as a sequence diagram, with the Guidance component representing the IoT platform on both cloud and edge level
- The integration to the SecureIoT services is performed by a Diagnostics Probe that records diagnostic data and processed diagnostic data.

Post-conditions & Alternative flows

- Essentially same as pre-conditions
- Machine operator is given feedback once first analytics are complete
Use case diagram

Data collection
- Data is acquired by IoT Gateway

Security gaps & risks
- Software implementation in IoT component and network infrastructure
- Hardware implementation, e.g. Spectre, Meltdown...
- Careless employees, e.g. introducing malware via unsecure media

Threats
- Information Disclosure: UC2.2 - An IoT device is compromised and allows an intruder to monitor network traffic
- Tampering, Denial of Service: UC2.3 - An intruder can additionally modify data of single sensors or IoT devices and thereby influence machine behavior; UC1.4 - An intruder can access many IoT components and sensors and use them to attack other network components
Most parts of UCs 2.2-2.4 are very similar in content to UC 2.1. In UC 2.2, an IoT device (or another network infrastructure device) is compromised and allows an intruder to monitor network traffic in a read-only way. In UC 2.3, the intruder can additionally modify data of single sensors or IoT devices and thereby influence machine behavior. This can potentially lead to dangerous behavior for humans (e.g. safety functions disabled) or equipment (like in the case of Stuxnet). In UC 2.4, an intruder has compromised many IoT components or sensors and uses them to attack other network components in the shop floor, e.g. DoS attacks.

### 5.1.2.3 Product Configuration

Industry 4.0 aims to industrialize the product manufacturing process and achieve sustainably improved productivity through product digitalization, standardization, and automation. To this end, the digitization of individual components enables virtual engineering, which allows the planning, configuration, and testing of a product, to streamline the production process. Consider an industrial control cabinet for an example. In this case, virtual engineering steps are as follows:

- Preparation of the electrical circuit diagram / electrical design engineer / E-CAD system
- Selection of products for implementation of the manufacturer's electrical schematic / electrical constructor / configurator of the manufacturer's electrical schematic / electrical constructor.
- Placement of Products / Electrical Constructor / Switch Cabinet Manufacturer's Configurator.
- Technical testing of the construction (standards; e.g. dimensioning of the cooling of the switch cabinet) / electrical designer / certification tool of the certifier.
- Preparation of the production documentation (electrical / mechanical) / electrical constructor / tool of the switch cabinet manufacturer.
- Start of production.

In this example, different stakeholders are part of the production, since the components do not originate from single supplier. Enabling the collaboration between these stakeholders and incorporating existing experience using artificial intelligence can further improve product quality, as well as reduce implementation time. Collaborative industrial IoT enables the automated exchange of product configuration data.
The exemplary cabinet control use case can be generalized to the configuration of arbitrary products. An engineer uses the product configuration software to design a product consisting of diverse parts from different suppliers, the specifications are published under a standardized (eclass) format in databases. Then, the construction is tested within the virtual engineering environment. If all tests are ok, the production documentation can be prepared, and production is started. The production machine receives the configuration data and produces the product, according to its production capabilities.

Table 6: Usage Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S3</td>
<td>Configuration of diversely composited products using a virtual engineering solution. Parts’ specifications are present in a database provided by suppliers to be used by the engineer. The configuration data of the product is sent to the production via the cloud platform, which begins with production autonomously. The cloud platform may do processing before the configuration data is sent to the production machines.</td>
</tr>
</tbody>
</table>

Use cases involved (*):

<table>
<thead>
<tr>
<th>Use case 3.1</th>
<th>id</th>
<th>Description/scope</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC3.1</td>
<td>An engineer plans a product configuration, which is then tested virtually. After successful testing, the configuration data is send to the cloud platform.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 3.2</th>
<th>id</th>
<th>Description/scope</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC3.2</td>
<td>The cloud platform processes and transfers the configuration data to the production machines, which produce the configured product according to their capabilities.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 3.3 (abuse)</th>
<th>id</th>
<th>Description/scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description/scope</td>
<td>An attacker intercepts configuration specification to steal intellectual property from the concerned parties.</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Use case 3.4 (abuse)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>UC3.4</td>
<td></td>
</tr>
<tr>
<td><strong>Description/scope</strong></td>
<td>An attacker tampers with the configuration data so that the resulting product is faulty.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7: Use case UC3.1**

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
<td><strong>S3/UC3.1</strong></td>
</tr>
</tbody>
</table>

**Actors involved**

- Engineer

**Systems involved**

- Configurator Software

**Interfaces**

- IT network

**Integration of SecureIoT services**

- Configuration monitoring probe

**Pre-conditions**

- The engineer has an untested and unfinished version of the configuration

**Main flow of events**

- The engineer finishes the configuration
- The engineer starts the testing of the configuration
- The configuration test is executed
- If the test is passed successfully, the configuration is sent to the cloud platform
- In UC3.2, The cloud platform processes the configuration and sends it to production machines

The above diagram shows the flow of events as a sequence diagram, with the Extractor representing the IoT cloud environment, which extracts instructions for the production machine from the product configuration and forwards it to the machine.

- The integration with SecureIoT services is performed by a configuration probe, which monitors the product configuration data and the extracted configuration.

Alternative Flows

- If the test is unsuccessful, the engineer is notified and not data is sent

Use case diagram
Data collection

- Product configuration file

Security gaps & risks

- Configuration file confidentiality and integrity

Threats

- Information Disclosure: Attacker may intercept the configuration file
- Tampering: Attacker may alter the configuration file
### Table 8: Use case UC3.2

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
<td><strong>S3/UC3.2</strong> The cloud platform processes and transfers the configuration data to the production machines, which produce the configured product according to their capabilities.</td>
</tr>
</tbody>
</table>

**Actors involved**

**Systems involved**

- IoT Cloud Components
- Production machines (not in scope)

**Interfaces**

- IT Network
- OT Network

**Integration of SecureIoT services**

- Configuration monitoring probe

**Pre-conditions**

- The configuration data has been successfully sent to the IoT platform

**Main flow of events**

- Cf. UC3.1
- The received configuration data is processed by extracting the information relevant for the production machines
- The extracted data is sent to the production machines

**Post-conditions & Alternative flows**

-  

**Use case diagram**
Cf. UC3.1

<table>
<thead>
<tr>
<th>Data collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Configuration File</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security gaps &amp; risks</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Configuration file confidentiality and integrity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Attacker may intercept the configuration file</td>
</tr>
<tr>
<td>- Attacker may alter the configuration file</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL1</td>
</tr>
</tbody>
</table>
5.2 Connected Car and Autonomous Driving

5.2.1 Description

As it was introduced in section 4 of deliverable D2.1 Reference Scenarios and Use Cases [1], the evolution of the traditional concept of a vehicle towards a digital autonomous system has already started. In a short-term future, cars will be able to exchange huge amounts of data with other digital entities (V2Internet, V2I and V2V use-cases) and to take real-time decisions in an autonomous way that may have consequences in critical aspects like the citizens personal security, public infrastructures damages or traffic congestion.

This chapter of deliverable D6.1 provides a detailed description of the usage scenarios within Connected Car and Autonomous Driving domain that will be developed within task T6.4. Therefore, it presents the key functionalities and requirements in the form of use-cases that must be implemented to enable the envisaged functionalities to be delivered from the application point of view:

- Usage based insurance analysis: assessment of driver behavior and calculus of risks.
- Warnings on traffic and road conditions: analysis of data coming from multiple vehicles to understand traffic conditions in different locations.

The use-cases also consider the potential abuse-cases that are relevant to these applications, relying on subsection 4.2 of D2.1 [1] as baseline and the specific technologies and components to be used.

The final aim will be the evaluation and assessment in a real and challenging application of the potentials benefits of SecureIoT platform, models and services in order to improve the security of two real V2Internet applications. In this sense, it also covers the specification of aspects needed to perform this assessment, e.g. the identification of the involved IoT assets, the available datasets and the methodology to be followed.

5.2.1.1 Actors

The specification of the use-cases (and abuse-cases) requires identifying which are the main actors (humans and systems) that interact within them and which are the functionalities they need.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
<th>Baseline technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard unit (OBU)</td>
<td>IoT device embedded in the vehicle. captures information from the different internal buses. It may also apply some edge-processing algorithms in order to aggregate and reduce the information, which must be uploaded to the IoT</td>
<td>IDIADA IDAPT platform (subsection 5.2.3.1)</td>
</tr>
</tbody>
</table>
platform. Finally, it facilitates the transport of data relating to vehicle usage to the cloud.

In addition, it may receive potential commands from the exterior and map them into the corresponding internal bus actions. In V2I and V2V use-cases, it will be also in charge of the exchange of data with the rest of the smart-objects involved in the IoT deployment, of executing advanced intelligent functionalities (e.g. AI algorithms) and taking smart decisions in real-time considering the collected information. This latter aspect is out of the scope of SecureIoT use-cases.

An example of the typical functional elements of the vehicle OBU is provided in Figure 5 below.

<table>
<thead>
<tr>
<th>IoT cloud platform</th>
<th>It receives, validates and stores information from the vehicles. It may also combine it with other data sources in order to unveil a bigger potential for the creation of high-value-added services (e.g. weather stations, smartphones, smart city sensors...).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service provider backend</td>
<td>It exploits the IoT Cloud Platform functionalities and APIs to gain access to the vehicle data and to feed its algorithms and services.</td>
</tr>
</tbody>
</table>

Each one of these actors may be composed of multiple internal components that will be further described in the following subsections.

Figure 4 provides a high-level representation of these actors within a V2Internet environment. As it can be seen, the vehicle (OBU device) plays the role of a typical IoT – edge devices, while the IoT platform allows persisting and aggregating data for the final creation of advanced services.
5.2.1.2 **Logical view for connected vehicle scenarios**

While the use-cases diagrams that will be described in subsection 5.2.2 below, specify the key functionalities and main requirements provided by each one of the components of the system, the logical view is the basis to understand its architecture and organization. The main objects (i.e., subsystems, packages, classes and components) to realize the scenarios’ use-cases are presented in Figure 6(edge), Figure 7 (cloud) and Figure 8 (application).
5.2.1.2.1 Edge layer

Figure 6: Logical view for edge layer (vehicle)
5.2.1.2.2 Cloud layer

Figure 7: Logical view for cloud layer (FIWARE – IoT platform)

5.2.1.2.3 Application layer

Figure 8: Logical view for application layer (AI models for scenarios)
5.2.2 **Use cases (abuse cases) involved**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S3.1 – Usage based insurance</strong></td>
<td>S3.1</td>
<td>This application involves analysing vehicle data to assess driver behaviour and hence determine the driver risks in order to tailor insurance premiums for customers using this service.</td>
</tr>
<tr>
<td><strong>S3.2 – Warnings on traffic and road conditions</strong></td>
<td>S3.2</td>
<td>This application involves analysing location-based and vehicle data coming from multiple vehicles to understand the traffic conditions in different locations.</td>
</tr>
</tbody>
</table>

### Use cases involved (*)

<table>
<thead>
<tr>
<th>Use case 1</th>
<th>Service initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.1</td>
</tr>
</tbody>
</table>

**Description/scope**
The vehicle owner (applicant) initializes the service using a web-based service hosted by the IoT platform. The service provides a means of configuring and defining additional parameters to customize the process, and acts as a front-end to the applicant to provide consent for the service.

<table>
<thead>
<tr>
<th>Use case 2</th>
<th>Edge data acquisition and transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.2</td>
</tr>
</tbody>
</table>

**Description/scope**
The OBU obtains raw vehicle data from multiple sources (Multi-CAN, GPS, IMU, etc.) and puts context to that information (i.e., how frequent the data is acquired). Collected data is then aggregated, stored locally and uploaded to the IoT Cloud platform through cellular or WI-FI communication interfaces (considering the most appropriate or efficient transmission strategy). In addition, some local processing algorithms may be applied in order to comply with privacy preservation regulations.

| Use case 3 | Cloud data processing, storage and streaming |
### Use case 4

**Vehicle data aggregation**

**id** UC3.4

**Description/scope**

To reduce the complexity of the process and the computational resources required to analyse raw vehicle data stored (or streamed) in the IoT Cloud Platform will be aggregated considering two axes:

- Raw vehicle data may be acquired at the edge by the OBU using heterogeneous time intervals. At cloud-application level, the raw vehicle data will be aggregated using a normalized duration in higher-level segments of information.
- Geographically using a geo-histogram representation based on a Mercato projection zoom level of 14-15, which will provide a resolution of 2.4-1.2 km, enough to discriminate with enough level of accuracy traffic conditions within a city.

Examples of this type of vehicle data representations may follow the format proposed in [http://www.automat-project.eu/sites/default/files/automat/public/content-files/articles/Open_CVIM_specification_v120.pdf](http://www.automat-project.eu/sites/default/files/automat/public/content-files/articles/Open_CVIM_specification_v120.pdf)

### Use case 5

**Driving profile classification: machine learning**

**id** UC3.5

**Description/scope**

The classification of the driving profile is the core element of the *Usage based insurance* scenario and will be based on data analytics techniques (e.g. supervised machine learning). This use-case covers the process of setting-up the corresponding model (e.g., data cleaning, dimensions reduction, aggregation, labelling and features extraction), training (which includes tuning and error estimation) with test data and finally its application to new data (batch and streaming) to classify the drivers’ behaviour (safe driver, risky driver). This use-case is only relevant for S3.1 scenario.
<table>
<thead>
<tr>
<th>Use case 6</th>
<th>Traffic status classification: machine learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.6</td>
</tr>
<tr>
<td>Description/scope</td>
<td>A machine-learning model is initially proposed to obtain insights regarding the status of the traffic. Again, this use-case will span through the steps of pre-processing test data, training the models, error assessment, tuning and final deployment in production. This use-case is only relevant for S3.2 scenario.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 7</th>
<th>Data visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.7</td>
</tr>
</tbody>
</table>
| Description/scope | A web-based service will illustrate the results of:  
   a) S3.1: The driving profile classification model, together with some additional statistics regarding the vehicle usage (i.e., average km/day, types of roads, etc.).  
   b) S3.2: The results of the traffic status classification model using a live map. |

<table>
<thead>
<tr>
<th>Use case 8 (abuse)</th>
<th>Vehicle impersonation (Man in the middle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.8</td>
</tr>
<tr>
<td>Description/scope</td>
<td>In this abuse-case, a malicious actor impersonates the OBU of a vehicle under assessment, having the ability to upload fake vehicle data to the IoT Cloud Platform. This abuse-case may result in the assignment of an incorrect profile to the victim driver.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 9 (abuse)</th>
<th>Tampering of data-in-transit</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.9</td>
</tr>
<tr>
<td>Description/scope</td>
<td>Data is intercepted and deliberately manipulated, destroyed, or forwarded during transmission from the edge OBU device to the IoT Cloud Platform.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 10 (abuse)</th>
<th>Tampering of data-at-rest</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>UC3.10</td>
</tr>
<tr>
<td>Description/scope</td>
<td>Data stored temporarily by the OBU, or by the IoT Cloud Platform in different data-store components may be modified by an unauthorized actor.</td>
</tr>
<tr>
<td>Use case 11 (abuse)</td>
<td>IoT cloud platform impersonation (Man in the middle)</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>id</td>
<td><strong>UC3.11</strong></td>
</tr>
<tr>
<td>Description/scope</td>
<td>The impersonation of the IoT Cloud Platform may result in a significant security breach with severe consequences in terms of privacy protections. An unauthorized malicious actor may gain access to sensitive personal information of a vast number of drivers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 12 (abuse)</th>
<th>Denial of Service attack to IoT Cloud Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td><strong>UC3.12</strong></td>
</tr>
<tr>
<td>Description/scope</td>
<td>If the IoT Cloud Platform suffers a DoS attack, its service availability could be affected and therefore legitimate OBUs may not be able to upload their latest data. This abuse-case may hinder the driving profile estimation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 13 (abuse)</th>
<th>Malware infection during Over-The-Air (OTA) software update</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td><strong>UC3.13</strong></td>
</tr>
<tr>
<td>Description/scope</td>
<td>As it was explained in D2.1 and D2.2, the capacity to dynamically update embedded firmware/software is a powerful mechanism to increase the reliability and security of a connected or autonomous vehicle during its complete lifetime. Nevertheless, it also poses one of the biggest security and privacy risks since it may give full-access to the OBU edge device, which is connected to the vehicle internal networks.</td>
</tr>
<tr>
<td>Scenario / use case</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>---</td>
</tr>
<tr>
<td>id</td>
<td><em>UC3.1</em></td>
</tr>
</tbody>
</table>

### Actors involved
- Vehicle
- Driver (Applicant)
- OEM / Vehicle manufacturer
- IoT platform provider
- Connected vehicle service provider

### Systems involved
- Vehicle OBU (i.e. IDAPT platform)
- IoT Cloud Platform (i.e. FIWARE)
  - Backend device management component
- Connected vehicle management application
- Connected vehicle service

### Interfaces
- 4G/Wi-Fi communication protocol between Vehicle OBU and IoT Cloud Platform.
- Secure communication channels: HTTPS over TLS / SSL.
- RESTful API / Message broker between Vehicle OBU and IoT Cloud Platform
- RESTful API / Message broker between IoT Cloud Platform and Connected vehicle service / management application.

### Integration of SecureIoT services
- SecureIoT data probes to monitor information flows between Vehicle OBU and IoT Cloud Platform.
- SECaaS services will detect risks and assess compliance.

### Pre-conditions
- Vehicle OBU connected to IoT Cloud Platform (via 4G or Wi-Fi).

### Main flow of events
- Driver provisions the Vehicle in the IoT Cloud Platform through the connected vehicle application, including information to uniquely identify the car (i.e., VIN, manufacturer, model, etc) and to establish / check connection (i.e., URL, port).
- Driver accepts connected vehicle management application conditions and terms.
- The IoT Cloud Platform (backend device management) establishes a connection with the Vehicle OBU and extracts information regarding the firmware / software version.
- The connected vehicle management application shows to the application which specific services are available for the vehicle.
- The applicant accepts the conditions and terms of a specific connected vehicle service: which vehicle information will be collected, how it will be processed, how long it will be retained, etc.
- The connected vehicle service through the IoT Cloud Platform (backend device management) informs the Vehicle OBU about the new functionalities that must be deployed at the Edge level in order to provide the new service.

**Post-conditions & Alternative flows**

- OEM or carmaker provisions the Vehicle in the IoT Cloud Platform, including information to uniquely identify the car (i.e., VIN, manufacturer, model, etc) and to establish / check connection (i.e., URL, port).
- The connected vehicle service through the IoT Cloud Platform sends an OTA update to deploy an updated firmware in the Vehicle OBU.

**Use case diagram**

The diagram illustrates the steps involved in the use case, including:
- Preconditions
- Main flow of events
- Data collection

**Data collection**
- Vehicle OBU (REST Interface)
**Vehicle OBU (Context data)**
- Performance data, such as CPU usage.

**Traffic / Network Data**
- PCAP logs, such as Wireshark or “tcpdump”.

**IoT Cloud Platform (REST Interface)**
- REST Interface of Context Broker.

**IoT Cloud Platform (Context Data)**
- Performance Data, such as CPU usage.

### Security gaps & risks
- Vulnerabilities in software implementation in IoT platform components and vehicle OBU.
- Bad configuration of communication protocols and networks.

### Threats
- Phishing, Abuse of privileges by staff (insider attack), DDoS, Eavesdropping (not protected communication channel)

### Security level
- SL3.
Table 12: Use case UC3.2

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>id</th>
<th>UC3.2</th>
</tr>
</thead>
</table>

A. Actors involved
- Vehicle
- IoT platform provider

B. Systems involved
- Vehicle OBU (i.e. IDAPT platform)
- IoT Cloud Platform (i.e. FIWARE)
  - Backend device management component
  - Context broker
  - PEP Proxy

C. Interfaces
- 4G/Wi-Fi communication protocol between Vehicle OBU and IoT Cloud Platform (IoT agent).
- Secure communication channels: HTTPS over TLS / SSL.
- RESTful API / Message broker between Vehicle OBU and IoT Cloud Platform (Backend device management component).
- Vehicle Data Interface (CAN, GPS/IMU).

D. Integration of SecureIoT services
- SecureIoT data probes to monitor information flows between Vehicle OBU and IoT Cloud Platform.
- SECaaS services will detect risks and assess compliance.

E. Pre-conditions
- UC3.1

F. Main flow of events
- The Vehicle OBU obtains periodically raw data from multiple sources (Multi-CAN, GPS, IMU, etc).
- The Vehicle OBU encodes the information using a proprietary information model.
- The Vehicle OBU attaches a timestamp to each piece of data.
- The Vehicle OBU aggregates data using time-series based format.
- The Vehicle OBU stores locally the vehicle information.
- If 4G or Wi-Fi connection is available, the Vehicle OBU streams data to the Backend device management component of the IoT Cloud Platform.
- If there is no wireless connection, the Vehicle OBU uses buffers to persist data. Buffers will be emptied using FIFO strategy as soon connectivity is recovered.
- The PEP or API Proxy of the IoT Cloud Platform intercepts the request sent by the OBU in order to enforce authentication and authorization.

### Post-conditions & Alternative flows

- The Vehicle OBU transmits the information directly to the Context broker if using RESTful APIs and ETSI NGSI-LD information model is possible.
- The PEP or API Proxy of the IoT Cloud Platform does not find an appropriate access token and rejects the request sent by the OBU.

### Use case diagram

The diagram illustrates the flow of events, conditions, and alternative paths in the use case scenario. Key steps include:

1. The Vehicle OBU periodically obtains raw data from vehicle sources (CAN) and other sensors (GPS/IMU).
2. Vehicle OBU interprets data using a proprietary information model.
3. A collection or "snapshot" of data is aggregated and assigned a timestamp to provide sequential context.
4. A series of data ("snapshots") are stored locally on the vehicle OBU.
5. If a connection (Wi-Fi/4G) to the IoT Cloud Platform is available, the Vehicle OBU streams data to the IoT Cloud Platform.
6. If no connection is available, the Vehicle OBU uses buffers to persist data. Buffers will be emptied using a FIFO strategy when a connection is available.
Data collection

<table>
<thead>
<tr>
<th>- Vehicle OBU (REST Interface)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- CAN Data to identify vehicle such as VIN, manufacturer.</td>
</tr>
<tr>
<td>- GPS/IMU data to provide movement/location data/time data.</td>
</tr>
<tr>
<td>- OBU Application Data, such as Firmware/Software version.</td>
</tr>
<tr>
<td>- Linux CAN Tools (canutils).</td>
</tr>
<tr>
<td>- ROS logging (rosbag).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>- Vehicle OBU (Context data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Performance data, such as CPU usage.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>- Traffic / Network Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>- PCAP logs, such as Wireshark or “tcpdump”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>- IoT Cloud Platform (REST Interface)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- REST Interface of Context Broker.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>- IoT Cloud Platform (Context Data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Performance Data, such as CPU usage.</td>
</tr>
</tbody>
</table>

Security gaps & risks

- Vulnerabilities in software implementation in IoT platform components and vehicle OBU.
- Bad configuration of communication protocols and networks.

Threats

Vulnerabilities in databases, unprotected encryption keys, extraction of vehicle data/code.

Security level

SL4.
### Table 13: Use case UC3.3

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>UC3.3</strong></td>
</tr>
</tbody>
</table>

**Actors involved**
- IoT platform provider
- Connected vehicle service provider

**Systems involved**
- IoT Cloud Platform (i.e. FIWARE)
  - Backend device management component
  - Context broker
  - Cygnus
  - Storage system
- Connected vehicle service

**Interfaces**
- RESTful API between IoT agent and Backend device management component.
- RESTful API between Context broker and Cygnus.
- RESTful API between Context broker and Connected vehicle service.
- Proprietary interface between Cygnus and Storage system.
- Vehicle Data Interface (CAN, GPS/IMU).

**Integration of SecureIoT services**
- SecureIoT data probes to monitor information flows between Vehicle OBU and IoT Cloud Platform.
- SECaaS services will detect risks and assess compliance.

**Pre-conditions**
- UC3.1
- UC3.2

**Main flow of events**
- Backend device component decodes data from vehicle information model.
- Backend device component decodes maps and normalized data to ETSI NGSI-LD data model.
- Backend device component validates and cleans data.
- Backend device component sends an update request to the Context broker.
- Context broker sends notification of new data to Cygnus (batch processing).
- Cygnus persists new data in the corresponding data storage component.
- Context broker sends notification of new data to connected vehicle service (stream processing).

**Post-conditions & Alternative flows**

**Use case diagram**

**Data collection**

- Vehicle OBU (REST Interface)
  - CAN Data to identify vehicle such as VIN, manufacturer
  - CAN Data for vehicle operation (e.g. vehicle speed).
  - GPS/IMU data to provide movement/location data/time data.
  - OBU Application Data, such as Firmware/Software version.
  - Linux CAN Tools (canutils).
- ROS logging (rosbag).
- Vehicle OBU (Context data)
  - Performance data, such as CPU usage.
- Traffic / Network Data
  - PCAP logs, such as Wireshark or “tcpdump”.
- IoT Cloud Platform (REST Interface)
  - REST Interface of Context Broker.
- IoT Cloud Platform (Context Data)
  - Performance Data, such as CPU usage.

**Security gaps & risks**

- Vulnerabilities in software implementation in IoT platform components.
- Bad configuration of communication protocols and networks.

**Threats**

Vulnerabilities in databases, unprotected encryption keys, unauthorized access to files or data, accept data from untrusted source.

**Security level**

SL2.
### Table 14: Use case UC3.4

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC3.4</td>
</tr>
</tbody>
</table>

#### Actors involved
- IoT Cloud Platform provider
- Connected vehicle service provider

#### Systems involved
- IoT Cloud Platform (i.e. FIWARE)
  - Storage system
- Connected vehicle service
  - Pre-processing / aggregation component

#### Interfaces
- 4G/Wi-Fi communication protocol between Vehicle OBU and IoT Cloud Platform.
- Secure communication channels: HTTPS over TLS / SSL.
- RESTful API / Message broker between Vehicle OBU and IoT Cloud Platform.
- RESTful API / Message broker between IoT Cloud Platform and Connected vehicle service / management application.
- Vehicle Data Interface (CAN, GPS/IMU).

#### Integration of SecureIoT services
- SecureIoT data probes to monitor information flows between IoT Cloud Platform components.
- SECaaS services will detect risks and assess compliance.

#### Pre-conditions
- UC3.1
- UC3.2
- UC3.3

#### Main flow of events
- The connected vehicle service applied a normalized time interval to aggregate vehicle data.
- The connected vehicle service aggregates vehicle data considering datapackages of fixed duration.
- The connected vehicle service determines minimum and maximum values for latitude and longitude of each datapackage.
- The connected vehicle service calculates a geo-histogram using Mercato projection zoom level of 14.-15.
- The results are persisted again in the Storage component of the IoT Cloud Platform.
- The results are streamed to the machine learning model to be used in the corresponding connected vehicle service data analysis process.

Post-conditions & Alternative flows

Use case diagram

Data collection
- **Vehicle OBU (REST Interface)**
  - CAN Data to identify vehicle such as VIN, manufacturer.
  - CAN Data for vehicle operation (e.g. vehicle speed).
  - GPS/IMU data to provide movement/location data/time data.
  - OBU Application Data, such as Firmware/Software version.
  - Linux CAN Tools (canutils).
  - ROS logging (rosbag).
- **Vehicle OBU (Context data)**
  - Performance data, such as CPU usage.
- **Traffic / Network Data**
  - PCAP logs, such as Wireshark or “tcpdump”.
- **IoT Cloud Platform (REST Interface)**
  - REST Interface of Context Broker.
- **IoT Cloud Platform (Context Data)**
  - Performance Data, such as CPU usage.

### Security gaps & risks
- Vulnerabilities in software implementation in IoT platform components.
- Bad configuration of communication protocols and networks.

### Threats
Vulnerabilities in databases, Unprotected encryption keys, Unauthorized access to files or data, accept data from untrusted source.

### Security level
SL2.
### Table 15: Use cases UC3.5 / 3.6

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
<td><strong>UC3.5 / UC3.6</strong></td>
</tr>
</tbody>
</table>

#### Actors involved

- IoT Cloud Platform provider
- Connected vehicle service provider
- Service developer

#### Systems involved

- IoT Cloud Platform (i.e. FIWARE)
  - Storage system
- Connected vehicle service
  - Machine learning model

#### Interfaces

- 4G/Wi-Fi communication protocol between Vehicle OBU and IoT Cloud Platform.
- Secure communication channels: HTTPS over TLS / SSL.
- RESTful API / Message broker between Vehicle OBU and IoT Cloud Platform.
- RESTful API / Message broker between IoT Cloud Platform and Connected vehicle service / management application.
- Vehicle Data Interface (CAN, GPS/IMU).

#### Integration of SecureIoT services

- SecureIoT data probes to monitor information flows between IoT Cloud Platform components and applications.
- SECaaS services will detect risks and assess compliance.

#### Pre-conditions

- UC3.1
- UC3.2
- UC3.3
- UC3.4

#### Main flow of events

- Service developer extracts batch data from the Storage system.
- Service developers pre-process the batch data (i.e., cleansing, dimensions reduction, labelling and features extraction)
- Service developer creates a test dataset and a validation dataset.
- Service developers trains and tunes a machine learning data model, obtaining also an error estimation
- New data is streamed to the model in order to apply classification: driving profile in S3.1 or traffic status in S3.2.
- The results are stored in the storage system of the IoT Cloud Platform.
- The results are streamed to the presentation and visualization layers of the connected vehicle service.

**Post-conditions & Alternative flows**

**Use case diagram**

**Data collection**

- Vehicle OBU (REST Interface)
  - CAN Data to identify vehicle such as VIN, manufacturer.
  - CAN Data for vehicle operation (e.g. vehicle speed).
  - GPS/IMU data to provide movement/location data/time data.
- OBU Application Data, such as Firmware/Software version.
- Linux CAN Tools (canutils).
- ROS logging (rosbag).

- Vehicle OBU (Context data)
  - Performance data, such as CPU usage.

- Traffic / Network Data
  - PCAP logs, such as Wireshark or “tcpdump”.

- IoT Cloud Platform (REST Interface)
  - REST Interface of Context Broker.

- IoT Cloud Platform (Context Data)
  - Performance Data, such as CPU usage.

### Security gaps & risks

- Vulnerabilities in software implementation in IoT platform components.
- Bad configuration of communication protocols and networks.

### Threats

Vulnerabilities in databases, unauthorized access to files or data, accept data from untrusted source, integrity of the data corrupted.

### Security level

SL2.
### Table 16: Use case UC3.7

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UC3.7</td>
</tr>
</tbody>
</table>

#### Actors involved

- Driver (Applicant)
- IoT Cloud Platform provider
- Connected vehicle service provider

#### Systems involved

- IoT Cloud Platform (i.e. FIWARE)
  - Storage system
- Connected vehicle service
  - Pre-processing / aggregation component

#### Interfaces

- 4G/Wi-Fi communication protocol between Vehicle OBU and IoT Cloud Platform.
- Secure communication channels: HTTPS over TLS / SSL.
- RESTful API / Message broker between Vehicle OBU and IoT Cloud Platform.
- RESTful API / Message broker between IoT Cloud Platform and Connected vehicle service / management application.
- Vehicle Data Interface (CAN, GPS/IMU).
- Device for illustrating service (e.g. Smart Phone / Web browser access)

#### Integration of SecureIoT services

- SecureIoT data probes to monitor information flows between applications and users.
- SECaaS services will detect risks and assess compliance.

#### Pre-conditions

- UC3.1
- UC3.2
- UC3.3
- UC3.4
- UC3.5/ UC3.6

#### Main flow of events

**S3.1 Usage based insurance**
- Applicant login in the connected vehicle service.
- Applicant manages connected vehicles (i.e., register (UC3.1), update and delete)
- Applicant manages connected services (i.e., subscribe (UC3.1), update and delete)
- Applicant visualizes collected raw vehicle information
- Applicant visualizes statistical information about vehicle usage
- Application visualizes driving profile classification results

S3.2 Warnings on traffic and road conditions
- Connected vehicle service provider login
- Connected vehicle service provider visualizes statistics about collected traffic information
  (i.e. geographical heatmap illustration amount of data, time histograms, etc)
- Connected vehicle service provider introduces filters (i.e., geographical coordinates, dates, etc)
- Connected vehicle service provider visualizes historic traffic information
- Connected vehicle service provider visualizes live traffic information

Post-conditions & Alternative flows

Use case diagram
Data collection

- Vehicle OBU (REST Interface)
  - CAN Data to identify vehicle such as VIN, manufacturer.
  - CAN Data for vehicle operation (e.g. vehicle speed).
  - GPS/IMU data to provide movement/location data/time data.
  - OBU Application Data, such as Firmware/Software version.
  - Linux CAN Tools (canutils).
  - ROS logging (rosbag).

- Vehicle OBU (Context data)
  - Performance data, such as CPU usage.

- Traffic / Network Data
  - PCAP logs, such as Wireshark or “tcpdump”.

- IoT Cloud Platform (REST Interface)
  - REST Interface of Context Broker.
  - Driver performance rating (S3.1)
  - Live map data (S3.2)

- IoT Cloud Platform (Context Data)
  - Performance Data, such as CPU usage.
Security gaps & risks

- Vulnerabilities in software implementation in IoT platform components.
- Bad configuration of communication protocols and networks.

Threats

Vulnerabilities in databases, unauthorized access to files or data, accept data from untrusted source, integrity of the data corrupted, DDoS, Virus on user’s device.

Security level

SL2.

5.2.3 Overview of IoT Assets

The following IoT Assets will be used to implement the main actors described in 5.2.1.1 and which interact within the previously described use-cases.

<table>
<thead>
<tr>
<th>IoT Asset</th>
<th>Actor</th>
<th>Category</th>
<th>IoT layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDAPT platform</td>
<td>Vehicle OBU</td>
<td>Smart object</td>
<td>Edge</td>
</tr>
<tr>
<td>FIWARE Orion Context Broker</td>
<td>IoT Cloud Platform</td>
<td>Context data management</td>
<td>Cloud</td>
</tr>
<tr>
<td>FIWARE IDAS</td>
<td>IoT Cloud Platform</td>
<td>IoT device management</td>
<td>Cloud</td>
</tr>
<tr>
<td>FIWARE PEP Proxy</td>
<td>IoT Cloud Platform</td>
<td>API Proxy / API GW</td>
<td>Cloud</td>
</tr>
<tr>
<td>FIWARE KeyRock</td>
<td>IoT Cloud Platform</td>
<td>Identity management</td>
<td>Cloud</td>
</tr>
<tr>
<td>FIWARE Cygnus</td>
<td>IoT Cloud Platform</td>
<td>Data streaming</td>
<td>Cloud</td>
</tr>
</tbody>
</table>

5.2.3.1 IDIADA IDAPT OBU

IDAPT is a research and development tool designed by Applus IDIADA that brings together traditional automotive technologies, emerging connected vehicle technologies and high powered computing into one automotive-grade development ECU. For SecureIoT, IDAPT will be installed into a vehicle and represent an automotive telematics unit capable of capturing data vehicle data, such as the vehicle network (CAN), vehicle location/movement data (GPS/IMU) and V2X Co-
operative Awareness Messaging (CAM). IDAPT will interact with FIWARE via a cellular 4G modem, or Wi-Fi.

There are 3 core SoCs (system on chip) in the IDAPT unit:

- NVIDIA Jetson TX2 (Ubuntu 16.04, modified NVIDIA Kernel)
  - CPU – Dual Denver2/2 + Quad ARM A57/2
  - GPU – NVIDIA Pascal, 256 CUDA cores
  - Wi-Fi 802.11a/b/g/n
- Infineon Aurix TC233LP (Automotive Safety Processor)
- Autotalks (V2X – DSRC/802.11p)
- Additional technologies in IDAPT are:
  - NVS Technologies NV08C-RTK (Supporting GPS module)
  - Invensense IGM-20602 (Supporting IMU module)
  - Ublox Toby L210 (Supporting 4G Cellular modem)

As IDAPT’s development environment is Ubuntu, the use case scenario applications benefit from existing Ubuntu software libraries and tools such as GPSD and SocketCAN. Additionally, NVIDIA also provide specific tools for their hardware such as ‘CUDA’ for parallel processing. Use case scenario Applications will be implemented in C/C++ but also utilizing additional local (IPC) and external (internet) communication support tools.

- The connected car use case scenario applications will be built on the NVIDIA SoC and the scope will be limited to the NVIDIA SoC for the project.

5.2.3.2 FIWARE platform

FIWARE is an open-source software platform that can be customized and deployed to create smart applications leveraging IoT, cloud or Big Data technologies in different domains. The FIWARE foundation was created in 2016 to boost its improvement and adoption in multiple verticals [7].
FIWARE core component is the Context Broker that implements an Application Programming Interface (API) to manage Context Information Management. This API is fully specified as part of the standard ETSI NGSI-LD and abstracts the whole high-level layers of the platform and the applications from domain specific protocols and data models [8].

FIWARE also provides a great flexibility since the functionalities of its main components are publicly specified in the form of Generic Enablers (GEs) [9]. At least, an open-source implementation of FIWARE GEs is also available. Thus, developers are able to assemble their own FIWARE flavors according to their needs, selecting and configuring only the required components and having the possibility of implementing additional ones that exploit GEs open interfaces.

This approach will be followed for SecureIoT connected vehicle scenarios:

Initially, the usage of the most relevant FIWARE IoT components is considered to collect, harmonize and store vehicle data at cloud level, i.e., NGSI Backend device management or IDAS [12], NGSI Context Broker [10], Cygnus [11] and data storages. Security will be enforced relying on KeyRock [13] for identity management and a PEP Proxy GE [14]. Finally, specific algorithms and models leveraging on modern AI technologies will be developed during the project to achieve the functionalities of the two applications.

5.3 Socially Assistive Robots and IoT Applications

5.3.1 Description

Nowadays, most robots and robotics platforms [3] tend to deliver functionalities independently of other internet-connected devices [15]. Hence, i.e. despite their pertinence and possible integration, IoT platforms and robots operate in isolation [16], which is a lost opportunity for a host of innovative functionalities. Security concerns are among the main barriers and concerns
against the integration of IoT with smart semi-autonomous objects such as robots. In the scope of the SecureIoT socially assistive robots use case, we will demonstrate the secure integration of LuxAI’s QT robot(s) in an environment provided by iSprint’s CloudCare2U (CC2U) IoT healthcare platform, which holds the promise to enhance the functionalities offered by both QT and CC2U, while offering a host of business opportunities for both companies (LuxAI, iSPRINT). CC2U integrates a wide range of sensors and wearable devices (e.g., fitbit, motion trackers) in order to accurately detect the users’ context and provide relevant personalized services. CC2U includes already a virtual robot as an output interface. As part of this use case, QT will become CC2U’s interface for interaction with end-users.

The envisaged integration of QT with CC2U will focus on the delivery of personalized ambient assisted living functionalities, fully in-line with the business strategies of both partners. In particular, QT will be used to deliver personalized rehabilitation and coaching exercises, as part of wider (rehabilitation or coaching) programs, managed and delivered through CC2U. In order to support these applications, a dense IoT network enabling continuous interaction between IoT devices managed by CC2U, QT robots, human users and the environment will be established. The integration between QT and CC2U can be directly realized, as CC2U has already a virtual robot interface for interaction with end-users, which will be replaced by QT. The integration challenge will however lie on keeping track of the state of QT and the environment, as well as on implementing distributed task assignment strategies, such as the Consensus-Based Bundle Algorithm (CBBA), which enable the distribution of application logic across different smart objects.

Table 18 presents the definitions related to the Socially Assistive Robots and IoT Applications use case.

**Table 18: Usage Scenarios**

<table>
<thead>
<tr>
<th>users</th>
<th>Inhabitants of the AAL environment (i.e. end users of the system)</th>
</tr>
</thead>
<tbody>
<tr>
<td>system (i.e. smart home)</td>
<td>The smart home consisting of the CC2U IoT network, cloud platform and interfaces as well as the QTrobot and its interfaces.</td>
</tr>
<tr>
<td>games</td>
<td>Interactive applications for therapy, rehabilitation, fitness or education purposes</td>
</tr>
<tr>
<td>professionals</td>
<td>Healthcare professionals which devise / develop/ select and recommend / personalize the games for users.</td>
</tr>
<tr>
<td>QT</td>
<td>QTrobot</td>
</tr>
<tr>
<td>QT U-App</td>
<td>Tablet app of QTrobot for users</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>QT P-App</td>
<td>Tablet app of QTrobot for professionals</td>
</tr>
</tbody>
</table>

5.3.1.1 **Attacker Motivational Rationale**
- Data Theft: Blackmailing, Espionage, etc.
- Destruction of Environment: physical damage to the robots environment and psychological damage to humans. (Vandalism, Terrorism, etc.)

5.3.1.2 **Benchmarking (Validation Methodology) [Mainly Overarching]**
- Create list of threats, patterns of attack and scenarios
  - Attacks on sensor data: e.g. active and passive side-channel attacks.
  - Attacks on Hardware: e.g. physical manipulation.
  - Attacks on software, infrastructure and OS: e.g. malware, DoS, DDoS, Phishing
- Simulate/emulate scenarios.
- Generate expected outcome, based on CC2U best practices and Know-How (without SecureIoT framework/architecture support.)
- Gather behavioral data in same use cases as in c) using the SecureIoT framework.
- Compare expected outcomes c) with gathered data d).

5.3.1.3 **Metrics & KPIs in both ways of integration:**
- Percentage of attacks correctly recognized taking into account the time, so that countermeasures (turning into fail-safe mode, sending alert, etc) are taken in a timely manner. The number of false positives, and false negatives are also a common metrics when trying to minimize one of them, in order to minimize one over another.
- Delay in decisions making for coaching (in general impact on the system performance including on data communication, processing and application responsiveness)
- Rate of correct CC2U coaching actions in the case of specific attacks to Things or devices
- Cost of an attack in each case – based on risk assessment
- Impact on user metrics (e.g. technology acceptance)
- Impact on system developer, integrator and deployer (developer friendliness factors such as learning curve, development time, software adaptability and re-usability, development and audit effort regarding compliance to privacy and security standards)
- Impact on system integration.
### 5.3.1.4 Actors

The specification of the use-cases requires identifying the main actors (humans and systems) that interact within them and the functionalities they need.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Description</th>
<th>Baseline technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTrobot onboard computer</td>
<td>This is the main control board of the QT robot which includes two standard WIFI interfaces, multiple USB interfaces to connect with different robot peripherals such as camera, multi-array microphone, speaker and robot motors and sensors.</td>
<td>Intel x86 multicore PC board running standard Linux operating system</td>
</tr>
<tr>
<td>QTrobot Tablet</td>
<td>An android-based tablet which connects to QTrobot cloud platform and QTrobot. The tablet is used as user and professional’s interfaces to develop therapy curriculum/applications (graphical programming) and for interaction during application execution.</td>
<td>Android-based tablet</td>
</tr>
<tr>
<td>QTrobot cloud</td>
<td>QTrobot service provider backend exploits the cloud platform web services and RESTful APIs to gain access to the robot data, application deployment and etc.</td>
<td>-</td>
</tr>
<tr>
<td>QTrobot ROS interface</td>
<td>ROS is used in different levels of QTrobot application development such as interfacing with QTrobot Tablet</td>
<td>Robot Operating System (ROS)</td>
</tr>
<tr>
<td>CC2U home gateway</td>
<td>Home component of the CC2U system which includes Ethernet, Bluetooth and Wi-Fi interfaces, multiple USB interfaces to connect with different peripherals such as wearable sync dongles and sensor boards. It collects data from the home sensors, processes it and sends it to the CC2U cloud.</td>
<td>Intel x86 PC running Ubuntu (Linux-based) operating system</td>
</tr>
<tr>
<td>CC2U cloud</td>
<td>A server-based environment that collects data from all CC2U User exposed devices, processes it and feeds it to the user through the web-based UI or to other 3rd party platforms through RESTful API.</td>
<td>-</td>
</tr>
</tbody>
</table>
Figure 11 shows an overview of the system and involved actors in the socially assistive robots and IoT applications scenarios and Figure 12 and Figure 13 respectively represents the simplified software stack diagram of QTrobot system architecture and that of CC2U.

**Figure 11:** Overview of the system and involved actors in the Socially Assistive Robots and IoT Applications scenarios

**Figure 12:** QTrobot software stack diagram
5.3.2 Scenarios and use cases description

The socially assistive robots and IoT applications are categorized in four different scenarios each consists of multiple use cases. The scenarios have been constructed in an extensible manner to validate different SECAS services by involving more actors, services and covering different security requirements. Below we summarize these scenarios and their relevance to SECAS services and SecureIoT Platform Architecture. In the next sections, we describe each scenario and its use cases in detail.

- **SAR-S1 (Cognitive & physical games scenario)** - This scenario mostly focuses on the initial integration of QTrobot edge and CC2U gateway/cloud platform and provides the test bed to validate the integration of these two systems with the first release of SECAS services. It covers some aspects of IoT Risk Assessment and Mitigation along with IoT Developers Support. Moreover, at this stage the necessary infrastructure will be provided to adopt SECAS system and application level data collections probes.

- **SAR-S2 and SAR-S3 (Monitoring & checkups scenario, Daily Calendar and System Admin scenario)** - These two scenarios stretch out the SAR-S1 in multiple directions.
  - Firstly, more subsystems and actors such as QTrobot Tablet, QTrobot cloud platform, patients and professionals will be involved. Having more actors allows us to cover more aspects of SECAS IoT Risk Assessment and Mitigation service at different points of edge and cloud. Moreover, other aspects of SECAS IoT
Developers Support service such as access policies, privileges and new IoT platform support (i.e. android os) will be validated.

- Secondly, the simulated and real user data will be collected and analyzed in CC2U and QTrobot cloud platforms. This implies involving SECAS IoT Compliance Auditing services such as Identification of potentially sensitive information exchanged between QTrobot and CC2U cloud platform (i.e. GDPR) and logging at different levels.
- Finally, at different edge and cloud points, these scenarios will be integrated with others SecureIoT Platform subsystems such as data routing and actuation from Data Collection and Actuation Layer and with Security Knowledge Base from Security Intelligence Layer.

- **SAR-S4 (Companionship – User Friendly Interface to Digital Life scenario)** - The remaining aspects of SECaaS services will be validated at this stage. By extending the involved actors to a third-party platform (e.g. Amazon cloud) and implementing the complete set of SECaaS services, the complete secure IoT platform will be validated during the final scenario. Moreover, involving a third-party platform opens new challenges to understand and validate how SECaaS services can be seamlessly integrated with other existing platform.

The following table summarize the main coverage of SECaaS services by each scenario.

<table>
<thead>
<tr>
<th>Scenario ID</th>
<th>Scenario Name</th>
<th>Main SECaaS services (with order of coverage)</th>
</tr>
</thead>
</table>
| SAR1        | Cognitive & physical games | 1. IoT Risk Assessment and Mitigation  
2. IoT Developers Support |
| SAR2        | Monitoring & checkups | 1. IoT Compliance Auditing  
2. IoT Risk Assessment and Mitigation  
3. IoT Developers Support |
| SAR3        | Daily Calendar and System Admin | 1. Legal and regulatory implications  
2. IoT Compliance Auditing |
| SAR4        | Companionship | 1. IoT Developers Support  
2. Legal and regulatory implications |

**5.3.2.1 SAR-S1: Cognitive & physical games scenario**

In this scenario, we integrate QTrobot edge with CC2U gateway and cloud platform where some advertisement and coaching messages from CC2U platform are received and read/played by QTrobot. The communicated messages to the robot include text and audio files. Moreover,
invoked by CC2U platform, QTrobot plays a cognitive (memory) game with user, which involves robot joints movements. The game will be developed by a professional and deployed into the QTrobot prior to the experiment. All above-mentioned use cases will be validated in two different setups based on the integration paradigm and interface between robot and CC2U:

- **Loose Integration**: QTrobot will be connected to the CC2U cloud platform via a separate home gateway.
- **Tight Integration**: QTrobot will act as home gateway, given the similarities between the QTrobot onboard computer and the CC2U home gateway, and directly connect to CC2U cloud platform.

This will allow us to understand better the challenge of integrating SECAS services at different communication points such as IoT developer support in case of architectural and deployment changes of IoT devices and platforms.

### Table 19: Cognitive & physical games scenario (SAR-S1) and relevant use cases

<table>
<thead>
<tr>
<th>Scenario - Cognitive and Physical Games</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use cases involved (*)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use case 1</strong></td>
<td>Text and audio coaching message (loosely integrated)</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td><strong>SAR-UC1.1</strong></td>
</tr>
<tr>
<td><strong>Description/scope</strong></td>
<td>QTrobot receives and renders text and audio coaching messages from CC2U home gateway in a loosely integrated architecture.</td>
</tr>
<tr>
<td><strong>Use case 2</strong></td>
<td>Cognitive and physical game (loosely integrated)</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td><strong>SAR-UC1.2</strong></td>
</tr>
<tr>
<td><strong>Description/scope</strong></td>
<td>QTrobot plays a cognitive game (e.g. gesture memory game) with user. The result of the game is communicated back to CC2U. The invocation of the game and the relevant information (e.g. difficulty level) is received from CC2U home gateway.</td>
</tr>
<tr>
<td><strong>Use case 3</strong></td>
<td>Text and audio coaching message (tightly integrated)</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td><strong>SAR-UC1.3</strong></td>
</tr>
<tr>
<td>Description/scope</td>
<td>QTrobot directly communicates with CC2U cloud platform and reads text and audio coaching messages in a tightly integrated architecture.</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Use case 4</strong></td>
<td>Cognitive and physical game (tightly integrated)</td>
</tr>
<tr>
<td>id</td>
<td>SAR-UC1.4</td>
</tr>
<tr>
<td>Description/scope</td>
<td>QTrobot plays cognitive game (e.g. gesture memory game) with the user and communicates back the result of the game directly to CC2U cloud platform in a tightly integrated architecture.</td>
</tr>
</tbody>
</table>

### 5.3.2.2 **SAR-S2: Monitoring & checkups scenario**

This scenario extends the SAR-S1 by introducing more platforms and actors to the existing system. Mainly, the QTrobot will communicate with different actors such as QTrobot Tablet and CC2U collected sensory data and interact with patient to collect health information such as vital signs, sleep measures, health questionnaire and cognitive game results. These data are then collected by the CC2U and QTrobot cloud platform for visualization, reporting and monitoring. For this scenario, we propose to use the tight integration paradigm for QTrobot-CC2U where QTrobot becomes a touch point for CC2U health data collection. The validation data for this scenario fall into three categories:

- **CC2U simulated health data**: data on vital signs, sleep measures and activities of the user comes from CC2U simulator.
- **QTrobot games and health questionnaires results**: game status/results and user emotion data are collected from LuxAI team members during the in-house experiment validation.
- **Collected data from actual users**: real data and feedback collected by questionnaires from habitants of elderly care centers.

The aim of this scenario is to mostly validate SECAS IoT Compliance Auditing services such as Identification of potential sensitive information (i.e. GDPR) along with SECAS IoT Developers Support for new IoT platform (i.e. Android OS). Moreover, the scenario also covers more aspects of SECAS IoT Risk Assessment and Mitigation service at different points of edge and cloud.
### Table 20: Monitoring & checkups scenario (SAR-S2) and relevant use cases

<table>
<thead>
<tr>
<th>Scenario - Monitoring and checkups</th>
<th>id</th>
<th>SAR-S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td>Multiple health-related and user acceptance information are collected by different touch points (QTrobot, Tablet, Questionnaires and Simulated health data) and are made available for monitoring, visualization and reporting in CC2U and QTrobot cloud platform for patients, professionals and therapist.</td>
</tr>
<tr>
<td>Use cases involved (*)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use case 1</td>
<td>id</td>
<td>SAR-UC2.1</td>
</tr>
<tr>
<td>Description/scope</td>
<td></td>
<td>Simulated data on vital signs, sleep measures and user activities are collected by QTrobot and aggregated into CC2U cloud platform.</td>
</tr>
<tr>
<td>Use case 2</td>
<td>id</td>
<td>SAR-UC2.2</td>
</tr>
<tr>
<td>Description/scope</td>
<td></td>
<td>Cognitive game results, user's performance and emotion are continuously monitored, collected and aggregated into CC2U and QTrobot cloud platform.</td>
</tr>
<tr>
<td>Use case 3</td>
<td>id</td>
<td>SAR-UC2.3</td>
</tr>
<tr>
<td>Description/scope</td>
<td></td>
<td>Video exercise results, user's emotion and performance are continuously monitored, collected and aggregated into CC2U and QTrobot cloud platform.</td>
</tr>
<tr>
<td>Use case 4</td>
<td>id</td>
<td>SAR-UC2.4</td>
</tr>
<tr>
<td>Description/scope</td>
<td></td>
<td>A set of predefined health related questions from CC2U cloud platform are received and displayed to the user using QTrobot Tablet and the results are stored on the CC2U and QTrobot cloud platform.</td>
</tr>
</tbody>
</table>
5.3.2.3 SAR-S3: Daily calendar and system admin scenario

This scenario extends the SAR-S2 where a professional/therapist can access to the relevant cloud platform to visualize the health monitoring results and update the contents (e.g. cognitive games, medical reminder, other contents) and health questionnaires. Moreover, users (patients) can also access to their own personal calendar to setup their own reminders. Within this scenario, different access policy levels (users, professional, admins) and privileges as part of SECAS IoT Developers Support will be validated.

Different regulation and access policies should be formulated to properly cover the following situations:

1. Professional accesses patient health data
2. Patient accesses his/her own health and personal data
3. Robot accesses patient health and personal data

Table 21: Daily Calendar and System Admin scenario (SAR-S3) and relevant use cases

<table>
<thead>
<tr>
<th>Scenario - Daily Calendar and System Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

Use cases involved (*):

<table>
<thead>
<tr>
<th>Use case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
</tbody>
</table>

### 5.3.2.4 SAR-S4: Companionship – User Friendly Interface to Digital Life scenario

The companionship scenario mostly focuses on IoT Developers Support by involving a third-party platform (e.g. Amazon AWS) to understand and validate how SECAS services can be seamlessly integrated with other existing platform.

#### Table 22: Companionship – User Friendly Interface to Digital Life scenario (SAR-S4) and relevant use cases

<table>
<thead>
<tr>
<th>Scenario - Companionship – User Friendly Interface to Digital Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Use cases involved (*)</td>
</tr>
<tr>
<td>Use case 1</td>
</tr>
<tr>
<td>id</td>
</tr>
<tr>
<td>Description/scope</td>
</tr>
</tbody>
</table>
5.3.3 **Use cases specifications**
In the following tables, we describe each use case in detail.

### Table 23: Use cases UC1.1

<table>
<thead>
<tr>
<th>Scenario / use case - Text and audio coaching message (loosely integrated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
</tbody>
</table>

**Actors involved**
- QTrobot
- CC2U home gateway
- User

**Systems involved**
- QTrobot - CC2U gateway bridge

**Interfaces**

**User Interfaces**
- QTrobot Wifi
- CC2U home gateway Wifi
- RESTful API / Message broker between QTrobot and CC2U home gateway

**Integration of SecureIoT services**
Some of the SECAS IoT Risk Assessment and Mitigation along with IoT Developers Support services will be integrated to establish a secure communication between QTrobot and CC2U home gateway. Moreover, at this stage the necessary infrastructure will be provided to adopt SECAS system and application level data collections (probes).

**Pre-conditions**
- QTrobot and CC2U home gateway are connected to the same Wifi network
- A list of predefined text/audio message are loaded into CC2U home gateway

**Main flow of events**
- A list of predefined message are loaded into CC2U home gateway
- Messages are transmitted to QTrobot in specific order
- QTrobot read/play the messages in the order of arrival
- User checks for the correctness of the message content and order
Post-conditions & Alternative flows

Post-Condition: The messages are read/played correctly by QTrobot and validated by user.

Alternative flows:
- The activity is not started and no messages are received
- Messages received in wrong order
- Messages received with unacceptable delay
- Wrong messages are received

Use case diagram

Data collection
- Text/audio messages passed throw CC2U Gateway - QTrobot bridge
- Internal ROS messages between QTrobot message broker / Audio interface

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

Threats
- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
### Man-in-the-Middle, Injection, Configuration Manipulation

<table>
<thead>
<tr>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL2</td>
</tr>
</tbody>
</table>

### Table 24: Use cases UC1.2

<table>
<thead>
<tr>
<th>Scenario / use case - Cognitive and physical game (loosely integrated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
</tbody>
</table>

#### Actors involved

- QTrobot
- CC2U home gateway
- User

#### Systems involved

- QTrobot - CC2U gateway bridge

#### Interfaces

**User Interfaces**

- QTrobot Wifi
- CC2U home gateway Wifi
- RESTful API / Message broker between QTrobot and CC2U home gateway

#### Integration of SecureIoT services

Some of the SECAS IoT Risk Assessment and Mitigation along with IoT Developers Support services will be integrated to establish a secure communication between QTrobot and CC2U home gateway. Moreover, at this stage the necessary infrastructure will be provided to adopt SECAS system and application level data collections (probes), security policy template and knowledge base.

#### Pre-conditions

- QTrobot and CC2U home gateway are connected to the same Wifi network
- Information regarding the cognitive games such as, duration, level and expected result are stored in CC2U home gateway.

#### Main flow of events
- A user prepares information regarding the cognitive games such as, duration, difficulty level and expected result.
- User loads the information into CC2U home gateway and starts the application
- CC2U home gateway remotely starts the game
- QTrobot performs the game in front of the user
- QTrobot sends confirmation messages to CC2U home gateway regarding the finishing of the game and the game level achievement.

Post-conditions & Alternative flows

Post-Condition: The game should be performed correctly and the status and the result of the game should be correctly received by CC2U home gateway.

Alternative flows:
- The activity is not started and no start messages are received
- Wrong information for launching the game are received
- Wrong or no status feedback are received from QTrobot

Use case diagram

Data collection

- Information passed through CC2U Gateway - QTrobot bridge
### Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

### Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

### Security level

SL2

**Table 25: Use case UC1.3**

<table>
<thead>
<tr>
<th>Scenario / use case - Text and audio coaching message (tightly integrated)</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAR-S1/UC1.3</td>
</tr>
</tbody>
</table>

**Actors involved**

- QTrobot
- CC2U Cloud platform
- User

**Systems involved**

- QTrobot - CC2U cloud platform SDK
### User Interfaces
- QTrobot Wifi
- RESTful API / Message broker between QTrobot and CC2U cloud platform

### Integration of SecureIoT services
Some of the SECAS IoT Risk Assessment and Mitigation along with IoT Developers Support services will be integrated to establish a secure communication between QTrobot and CC2U home gateway. Moreover, at this stage the necessary infrastructure will be provided to adopt SECAS system and application level data collections (probes).

### Pre-conditions
- QTrobot is connected to CC2U cloud platform via internet
- A list of predefined text/audio message are loaded into CC2U cloud platform

### Main flow of events
- A user prepare a list of text/audio message to be read/play by robot
- User load the message list into CC2U cloud and start the application
- Messages are transmitted to QTrobot in specific order
- QTrobot read/play the messages in the order of arrival
- User checks for the correctness of the message content and order

### Post-conditions & Alternative flows
Post-Condition: The messages are read/played correctly by QTrobot and validated by user.

Alternative flows:
- The activity is not started and no messages are received
- Messages received in wrong order
- Messages received with unacceptable delay
- Wrong messages are received
Use case diagram

Data collection

- Text/audio messages passed directly to QTrobot
- Internal ROS messages between QTrobot message broker / Audio interface

Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

Security level

SL2
### Table 26: Use case UC1.4

<table>
<thead>
<tr>
<th>Scenario / use case - Cognitive and physical game (tightly integrated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
</tbody>
</table>

### Actors involved

- QTrobot
- CC2U Cloud platform
- User

### Systems involved

- QTrobot - CC2U Cloud platform SDK

### Interfaces

**User Interfaces**

- QTrobot Wifi
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

### Integration of SecureIoT services

Some of the SECAS IoT Risk Assessment and Mitigation along with IoT Developers Support services will be integrated to establish a secure communication between QTrobot and CC2U home gateway. Moreover, at this stage the necessary infrastructure will be provided to adopt SECAS system and application level data collections (probes), security policy template and knowledge base.

### Pre-conditions

- QTrobot is connected to CC2U cloud platform via internet
- Information regarding the cognitive games such as, duration, level and expected result are stored in CC2U cloud platform.

### Main flow of events

- A user prepares information regarding the cognitive games such as, duration, difficulty level and expected result.
- User loads the information into CC2U cloud platform and starts the application
- CC2U cloud remotely starts the game
- QTrobot performs the game in front of the user
- QTrobot send confirmation messages to CC2U cloud platform regarding the finishing of the game and the game level achievement.
Post-conditions & Alternative flows

Post-Condition: The game should be performed correctly and the status and the result of the game should be correctly received by CC2U cloud platform.

Alternative flows:
- The activity is not started and no start messages are received
- Wrong information for launching the game are received
- Wrong or no status feedback are received from QTrobot

Use case diagram

Data collection
- Information passed throw CC2U Gateway - QTrobot bridge
- Internal ROS messages between QTrobot message broker / robot motor controller
- QTrobot internal state including robot actuator

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

### Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

### Security level

SL2

---

#### Table 27: Use cases UC2.1

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>Health-related data collection (simulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SAR-S2/UC2.1</td>
</tr>
</tbody>
</table>

#### Actors involved

- QTrobot
- CC2U Cloud platform
- CC2U simulator

#### Systems involved

- QTrobot - CC2U Cloud platform SDK
- Qtrobot - CC2U simulator platform SDK

#### Interfaces

**User Interfaces**

- QTrobot Wifi
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

**Integration of SecureIoT services**
More aspects of SECAS IoT Risk Assessment and Mitigation service at different points of edge and cloud will be integrated to also validate the performance of the system with/without SECAS service. Moreover, specific aspects of SECAS IoT Developers Support service such as access policies will be validated.

Pre-conditions

- QTrobot is connected to CC2U cloud platform via internet
- CC2U simulator is initialized and simulated health data are accessible by QTrobot

Main flow of events

- CC2U simulator generates health-related data and stores it in local database
- CC2U simulates and at the same time publishes the data for QTrobot
- QTrobot collects all data
- QTrobot send collected data to CC2U cloud platform
- CC2U cloud platform stores the data for future analysis

Post-conditions & Alternative flows

Post-Condition: The health generated data from CC2U simulator should be sent by acceptable (to be defined during the experiment) rate to QTrobot using integrated SECAS services. The data collected in CC2U cloud must be the same to the one stored locally by CC2U simulator.

Alternative flows:
- The activity is not started, and no data are received to QTrobot
- Data received to QTrobot with big delay or corrupted
- Wrong data collected on the CC2U cloud
Data collection

- Information passed through CC2U Simulator to QTrobot
- Information passed through QTrobot and CC2U cloud platform

Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity's performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation
Table 28: Use cases UC2.2

Scenario / use case - Cognitive game result data collection

<table>
<thead>
<tr>
<th>id</th>
<th>SAR-S2/UC2.2</th>
</tr>
</thead>
</table>

Actors involved

- User
- QTrobot
- CC2U Cloud platform
- QTrobot Cloud platform

Systems involved

- QTrobot - CC2U Cloud platform SDK
- QTrobot actuator controller
- QTrobot emotion/gesture recognition

Interfaces

- QTrobot Wifi
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

Integration of SecureIoT services

This use case implies involving SECAS IoT Compliance Auditing services such as Identification of potential sensitive information exchanged between QTrobot and CC2U cloud platform (i.e. GDPR) and logging at different levels.

Pre-conditions

- QTrobot is connected to QT and CC2U cloud platform via internet
- The cognitive game is initialized on the robot
- User is presented in front of the robot

Main flow of events

- QTrobot detect the presence of user and starts the game
- QTrobot performs the game in front of the user
- User plays the cognitive game
- QTrobot collects user performance such as user’s emotion and game result
- QTrobot detects the end of the game
- QTrobot sends collectd data to CC2U and QT cloud platform

Post-conditions & Alternative flows

Post-Condition: The game should be performed correctly, and the status and the result of the game should be correctly received by CC2U and QT cloud platform.

Alternative flows:
- The activity is not started, and no start messages are received
- Wrong or no status feedback are received by CC2U cloud from QTrobot
- Wrong or no status feedback are received by QT cloud from QTrobot

Use case diagram

Data collection
- Information passed through QTrobot to QT cloud platform
- Information passed through QTrobot to CC2U cloud platform
- QTrobot internal state including robot actuator
- User facial emotions
- User recognised face photo

### Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the CC2U cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

### Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

### Security level

SL2

---

**Table 29: Use cases UC2.3**

<table>
<thead>
<tr>
<th>Scenario / use case - Virtual coaching result collection</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAR-S2/UC2.3</td>
</tr>
</tbody>
</table>

**Actors involved**

- User
- QTrobot
- QTrobot tablet
- CC2U Cloud platform
- QTrobot Cloud platform
### Systems involved

- CC2U video coaching system
- QTrobot - CC2U Cloud platform SDK
- QTrobot tablet App
- QTrobot emotion detection

### Interfaces

#### User Interfaces
- QTrobot Wifi
- QTrobot second Wifi connected to the tablet
- ROS interface for QTrobot tablet
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

### Integration of SecureIoT services

This use case implies involving SECAS IoT Compliance Auditing services such as Identification of potential sensitive information exchanged between QTrobot and CC2U cloud platform (i.e. GDPR) and logging at different levels along with SECAS IoT Developers Support for new IoT platform (i.e. android os).

### Pre-conditions

- QTrobot is connected to QT and CC2U cloud platform via internet
- QTrobot tablet is connected to the robot via robot second wifi
- The virtual coaching is initialized on the tablet
- User is presented in front of the robot

### Main flow of events

- QTrobot plays the virtual coaching video on the tablet to the user
- User plays and follow the video coaching exercise
- QTrobot collects user performance such as user’s emotion
- QTrobot detects the end of video coaching session
- QTrobot sends collected data to CC2U and QT cloud platform

### Post-conditions & Alternative flows

**Post-Condition:** The collected result of video coaching session should be correctly received by CC2U and QT cloud platform.

**Alternative flows:**
- The activity is not started on the tablet and no start messages are received
- Wrong or no status feedback are received by CC2U cloud from QTrobot
Wrong or no status feedback are received by QT cloud from QTrobot

Use case diagram

- User
- QTrobot tablet interface
  - perform coaching exercise
  - monitor and collect user performance
- CC2U gateway SDK
  - store data
  - send data
- QTrobot Cloud API
  - store data
  - send data

Data collection
- Information passed through QTrobot to QT cloud platform
- Information passed through QTrobot to CC2U cloud platform
- Information passed through QTrobot and tablet
- User facial emotions
- User recognised face photo

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

### Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

### Security level

SL2

### Table 30: Use cases UC2.4

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>Health questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SAR-S2/UC2.4</td>
</tr>
</tbody>
</table>

### Actors involved

- User
- QTrobot
- QTrobot tablet
- CC2U Cloud platform
- QTrobot Cloud platform

### Systems involved

- QTrobot - CC2U Cloud platform SDK
- QTrobot tablet App
- QTrobot emotion detection

### Interfaces
User Interfaces
- QTrobot Wifi
- QTrobot second Wifi connected to the tablet
- ROS interface for QTrobot tablet
- RESTful API /Message broker between QTrobot and CC2U Cloud platform

Integration of SecureIoT services

This use case implies involving SECAS IoT Compliance Auditing services such as Identification of potential sensitive information exchanged between QTrobot and CC2U cloud platform (i.e. GDPR) and logging at different levels along with SECAS IoT Developers Support for new IoT platform (i.e. android os).

Pre-conditions
- The questionnaire is prepared by professional
- QTrobot is connected to QT and CC2U cloud platform via internet
- QTrobot tablet is connected to the robot via robot second wifi
- The questionnaire is initialized on the tablet
- User is presented in front of the robot

Main flow of events
- QTrobot shows the questionnaire on the tablet to the user
- User answer the health questions
- QTrobot tablet collect user answer
- QTrobot receive the user’s answers from tablet
- QTrobot send collectd data to CC2U and QT cloud platform

Post-conditions & Alternative flows

Post-Condition: The user should answer the questions and the status and the results should be correctly collected on the QTrobot and CC2U cloud platform.

Alternative flows:
- The activity is not started on the tablet and no start messages are received
- No answer collectd on the tablet
- No answer received by QTrobot from tablet
- Wrong or no status feedback are received by CC2U cloud from QTrobot
- Wrong or no status feedback are received by QT cloud from QTrobot

Use case diagram
Data collection

- Information passed through QTrobot to QT cloud platform
- Information passed through QTrobot to CC2U cloud platform
- Information passed through QTrobot and tablet

Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

**Threats**

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

**Security level**

SL2

**Table 31: Use cases UC3.1**

<table>
<thead>
<tr>
<th>Scenario / use case</th>
<th>Medical reminder</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SAR-S3/UC3.1</td>
</tr>
</tbody>
</table>

**Actors involved**

- User (professional)
- CC2U/QTrobot Cloud platform

**Systems involved**

- Patient web portal

**Interfaces**

**User Interfaces**

- QTrobot Wifi
- CC2U portal
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

**Integration of SecureIoT services**

This use case covers different access policy levels (i.e. professional) and privileges as part of SECAS IoT Developers Support and Compliance Auditing services.

**Pre-conditions**

- There is a patient agenda application on the CC2U/QTrobot cloud
- Agenda is shared for both medical and personal usage
- Agenda can have multiple access policy

Main flow of events

- A professional access the patient agenda via web interface
- Professional visualize, analyze the data and set different reminder for patient
- Reminders are stored on the patient agenda from cloud platform

Post-conditions & Alternative flows

Post-Condition: The medical reminders are stored properly into the patient agenda from cloud

Alternative flows:
- Professional does not have proper privilege to set the reminder
- Professional have extra privilege to see/modify the patient personal reminders
- An unauthorized entity can access and modify patient agenda

Use case diagram
Data collection

- login and authentication data
- other data exchanged between web interface and agenda cloud application

Security gaps & risks

- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

Security level

SL2

<table>
<thead>
<tr>
<th>Table 32: Use cases UC3.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario / use case</strong></td>
</tr>
<tr>
<td>id</td>
</tr>
</tbody>
</table>

**Actors involved**

- User (patient)
- CC2U/QTrobot Cloud platform

**Systems involved**

- Patient web portal

**Interfaces**
### User Interfaces
- QTrobot Wifi
- CC2U portal
- RESTful API / Message broker between QTrobot and CC2U Cloud platform

### Integration of SecureIoT services
This use case covers different access policy levels (i.e. user) and privileges as part of SECAS IoT Developers Support and Compliance Auditing services.

### Pre-conditions
- There is a patient agenda application on the CC2U/QTrobot cloud
- Agenda is shared for both medical and personal usage
- Agenda can have multiple access policy

### Main flow of events
- user (patient) access his personal agenda via web interface
- user add a reminder for himself
- Reminders are stored on the patient agenda from cloud platform

### Post-conditions & Alternative flows
Post-Condition: The personal reminders are stored properly into the patient agenda from cloud

Alternative flows:
- User does not have proper privilege to set the reminder
- User have extra privilege to see/modify the medical related reminders
- An unauthorized entity can access and modify patient agenda

### Use case diagram
Data collection
- login and authentication data
- other data exchanged between web interface and agenda cloud application

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

Threats
- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

Security level
Table 33: Use cases UC3.3

<table>
<thead>
<tr>
<th>Scenario / use case - Personal reminder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SAR-S3/UC3.3</td>
</tr>
</tbody>
</table>

**Actors involved**

- User (patient)
- QTrobot
- CC2U/QTrobot Cloud platform

**Systems involved**

- User
- QTrobot
- Patient web portal

**Interfaces**

- QTrobot Wifi
- RESTful API / Message broker between robot and CC2U/QTrobot Cloud platform

**Integration of SecureIoT services**

This use case covers different access policy levels (i.e. robot) and privileges as part of SECAS IoT Developers Support and Compliance Auditing services.

**Pre-conditions**

- There is a patient agenda application on the CC2U/QTrobot cloud
- Agenda is shared for both medical and personal usage
- Agenda can have multiple access policy
- Agenda is already filled with some personal and medical reminders

**Main flow of events**

- Cloud agenda application send reminder information to QTrobot via push notification
- QTrobot process the reminder and react correspondingly:
  - cognitive game and coaching video will be performed and play back for patient
  - appointments, time to rest, and similar reminder will be read for user
- QTrobot monitor the performing activity and update the agenda with the relevant result
Post-conditions & Alternative flows

Post-Condition: The reminders should be properly processed (as it is explained above) and the result should be correctly stored on the cloud.

Alternative flows:
- Robot does not have proper privilege to access the reminders
- Robot has extra privilege to modify medical and personal reminders
- An unauthorized entity can access the reminders and push the result

Use case diagram

Data collection
- login and authentication data
- the result of reminder activities
- other data exchanged between web interface and agenda cloud application

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

### Threats

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

### Security level

| SL2 |

---

**Table 34: Use cases UC4.1**

<table>
<thead>
<tr>
<th>Scenario / use case - Digital assistant id</th>
<th>SAR-S4/UC4.1</th>
</tr>
</thead>
</table>

### Actors involved

- User (patient)
- QTrobot
- CC2U Cloud platform
- Amazon AWS

### Systems involved

- User
- QTrobot
- Amazon Alexa SDK

### Interfaces

- QTrobot Wifi
- RESTful API / Message broker between robot and CC2U Cloud platform
- RESTful API between QTrobot and Amazon AWS
Integration of SecureIoT services

This use case covers Legal and regulatory implications and IoT Developers Support for new platform.

Pre-conditions

- QTrobot is connected to Amazon AWS (Alexa services)
- QTrobot is connected to CC2U Cloud
- Proper user account for AWS is created
- Required alexa skills for this scenario have been collected/developed on AWS
- A set of functionalities/skills from CC2U and AWS are known by robot to respond to a set of known questions

Main flow of events

- User initiates interaction with QTrobot with specific wake word (e.g. ‘Hey QT’)
- Wake word is detecting by QTrobot
- User is recognized (voice and/or facial recognition) and voice interaction with user starts
- User asks question regarding weather, news, his health profile, etc.
- The question is understood by QTrobot
- Proper query is sent to CC2U (patient health portal) or AWS depending on the question
- The response is collected and given to the user via voice message

Post-conditions & Alternative flows

Post-Condition: User’s questions are understood and properly queried from corresponding cloud backend (CC2U/AWS) and the correct response is given

Alternative flows:
- User is not recognised
- User is not valid
- User question is wrongly routed to the cloud backend raising different access polices and data protection issues

Use case diagram
Data collection
- User voice and face image
- Data logging at different levels
- User questions and activities

Security gaps & risks
- **Loss of Communication**: an activity is not started, activity results are not uploaded, or activity’s performance may be hindered due to loss of communication between QT and the cloud.
- **Eavesdropping**: users’ privacy is breached due to sniffing of the communication between QT and the cloud or QT and QT U-App.
- **Communication hijacking**: the proposed activity by the system is altered in the benefit of hackers including stealing information and emotional or physical damage to users.

**Threats**

- DoS, DDoS and Jamming,
- Sniffing, Fingerprinting
- Man-in-the-Middle, Injection, Configuration Manipulation

**Security level**

SL2
6 Validation and testing

According to [6] system security verification and validation requires sufficient testing needs to be conducted to verify that functional security requirements have been satisfied. This testing should be performed in an operational environment, after the system has been integrated with other enterprise infrastructure components.

Ideally, this testing will occur throughout the development lifecycle as well as the implementation/integration, deployment, and operations ones.

The IoT security lifecycle involves both Verification and Validation (V&V). Concerning verification, this process provides the assurances that the system operates according to a set of requirements that appropriately meet stakeholder needs. Concerning validation, this is the assurance that an IoT system product, service, or system meets the needs of the customer and other identified stakeholders—in an IoT system, this means that the system definition and design is sufficient to safeguard against threats. Verification is the evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or market-imposed constraint. For IoT systems, this means that the security services and capabilities were implemented according to the design.

One approach to verifying functional security requirements is the use of simulators and emulators. In this way, one can generate the expected events, while at the same time another can generate attacks, based on the most common types.

6.1 Validation and testing for Multi-Vendor Industrie 4.0 (MVI)

To validate the use cases contained within the use cases considering Industry 4.0, a simulation solution is used. Using the simulation, the industrial machinery, sensors, and IoT devices from the use cases can be connected to the SecureIoT platform to allow for an assessment of its functionality.

6.1.1 Methodology

6.1.1.1 Release plan

<table>
<thead>
<tr>
<th>Release</th>
<th>Start</th>
<th>Ends</th>
<th>Doc</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Determine Domain Model, Functional Model and Information</td>
</tr>
</tbody>
</table>
## D6.1.2 Datasets for Development and Testing

To test the security services provided by the SecureIoT platform, datasets are required that allow for testing without a full implementation and integration of the usage scenarios and the SecureIoT platform. There are three datasets described in the following, one for each of the usage scenarios contained within the Industry4.0 cluster.

- **Injection Molding Machine**: Injection molding status and diagnostic data EUROMAP63 format
- **Product Configurator**: Product configuration file, AutomationML format
- **Head Mounted Display**: Speech data, Location data, Visual Data

## 6.1.1.3 Validation Use Case Scenarios

The validation involves all three usage scenarios described in the Industry 4.0 cluster. Therefore, all physical components generating the data to be monitored are virtualized in the environment provided by P@ssport.

### 6.1.1.3.1 Injection Molding Machine

In a first step, historical sensor data from the injection molding machine is collected and used for validating the SECaaS services.

Since the injection molding machine is an active production tool, it may not be available for validation purposes in the context of SecureIoT. Therefore, a model of its control and sensor elements is created within the virtualized environment. This allows for passing data generated...
by the virtualized model through the IoT platform and in turn the SECaaS services provided by the SecureIoT platform. In a first step, historical sensor data from the injection molding

6.1.1.3.2 Head Mounted Display
Similarly, to the injection molding machine, historical data from the head mounted display is used for initial validation purposes. To this end, the data recorded by an HMD in the real world (speech, location data, video data) is fed into the simulation.

A more mature validation may be performed with an actual HMD connected to the virtualized environment and its usual IoT platform.

6.1.1.3.3 Product Configurator
In the product configuration use case, the validation is performed using an existing configuration. Since the use case describes the product configuration to come from an outbound service, the configuration file is sent into the virtualized environment to be processed by the extractor. Then, the part specification extracted from the product configuration file is sent to the virtualized production machine through the IoT platform.

6.1.1.4 Requirements and Validation of SECaaS Services

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>Monitoring of data streams from Injection Molding Machine and Head Mounted Display</td>
<td>R5.1.2</td>
<td>Inconsistent/malicious data streams are injected in the virtualized environment.</td>
<td>Precision &amp; recall (Coherence of risk assessment, usefulness of proposed mitigation). Efficiency (Timely response).</td>
</tr>
<tr>
<td>1.1.2</td>
<td>Validation of consistency of files passed between components (esp. product configuration and part specification files)</td>
<td>R5.1.2</td>
<td>The files are altered in the virtualized environment and then passed on.</td>
<td>Effectivity (Coherence of risk assessment, usefulness of proposed mitigation).</td>
</tr>
<tr>
<td>1.1.3</td>
<td>Consistency of guidance information received by the HMD</td>
<td>R5.1.2</td>
<td>Inconsistent/non-sensical guidance messages are injected.</td>
<td>Effectivity (Coherence of risk assessment, usefulness of proposed mitigation).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1</td>
<td>Integration with IoT Platform</td>
<td>R5.3.4</td>
<td>Functional verification</td>
<td>(TBD)</td>
</tr>
</tbody>
</table>
Table 38: Industry 4.0 Requirements to IoT Security Knowledge Base

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1</td>
<td>Integration with IoT Platform</td>
<td>R5.3.4</td>
<td>Functional verification</td>
<td>(TBD)</td>
</tr>
</tbody>
</table>

6.2 Validation and testing for IoT Applications and Connected Car and Autonomous Driving (CCAD)

6.2.1 Methodology

6.2.1.1 Release plan
This chapter defines a detailed implementation and verification plan that will finish not only with the verification and validation of the Connected Car and Autonomous Driving applications' functionalities, but also with the assessment of SecureIoT platform and services. It also illustrates which specific aspects will be developed and integrated for the three prototypes 'versions to be released during T6.4: D6.9 (M15), D6.10 (M24) and D6.11 (M33).

As a summary, the following goals have been defined for these deliverables:

Table 39: Release plan for Connected and Autonomous Driving scenarios (T6.4)

<table>
<thead>
<tr>
<th>Release</th>
<th>Start</th>
<th>Ends</th>
<th>Doc</th>
<th>Goals</th>
</tr>
</thead>
</table>
                       - Specific data collection probes (WP3)  
                       - Implementation of scenario #1  
                       - Integration of initial version of security implementations (WP4).  
                       - Integration of initial version of SECaaS (WP5) |
                       - Injection of attacks.  
                       - Integration of updated WP3-5 prototypes  
                       - Functional validation of scenarios’ applications with simulated data. |
In the case of the first release, a more detailed scheduling has been also planned as Table 40 shows, defining sprints of one month of duration for the implementation and integration of the prototypes.

Table 40: Sprint plan for Release#1 of Connected and Autonomous Driving scenarios (T6.4)

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Timeline</th>
<th>Goals</th>
</tr>
</thead>
</table>
| Sprint#1 | M9 | - Deployment, configuration and testing of FIWARE IoT Cloud platform components  
  o Backend device management  
  o Orion Context Broker  
  o Cygnus  
  o Data-storage components  
  o PEP Proxy  
  o IDM KeyRock  
- Set-up and connection to FIWARE IoT Cloud platform of IDAPT test units. |
| Sprint#2 | M10 | - Setting-up of simulation tools for the generation of development and testing datasets.  
- Interconnection of simulation tools with scenarios’ infrastructure through IDAPT platform.  
- Generation and persistence of development and testing datasets using FIWARE IoT Cloud platform components.  
- D6.9 ToC |
| Sprint#3 | M11 | - Batch data analysis, pre-processing and feature engineering.  
- Development (and training) of machine learning models for scenario#1. |
| Sprint#4 | M12 | - Deployment of machine learning model for scenario#1 to work with streaming data.  
- Development of scenario#1 user interfaces.  
- Implementation and deployment of data collection probes.  
  o IDAPT unit  
  o FIWARE IoT Cloud platform components |
6.2.1.2 Datasets for development and testing

Testing, verification and validation for Connected Car and Autonomous Driving based applications is challenging due to the following reasons:

- Test information coming from a relevant number of vehicles is needed for algorithms relying on Big Data.
- Test information must be realistic to be useful. For instance, data for many vehicle signals must be correlated (speed vs rpm, speed vs acceleration, etc), data must correspond to real trips, data must consider traffic control mechanisms and limitations or the influence or other vehicles.
- Test information must be collected with high-frequency and at multiple levels to enable isolated development of the different microservices or components that are part of the system. For instance: at vehicle internal CAN buses and onboard vehicle unit.
- Personal data protection regulations must be adhered to.
- Most of today’s cars do not include advanced mechanisms to collect and transmit high-quality data.
- Internal vehicle communications typically rely on OEMs’ proprietary protocols and technologies.
- Injecting attacks in real vehicles may suppose risks in terms of material and personal damages.

To the best of our knowledge, a large-scale open dataset which includes rich and high-quality information about vehicle signals does not exist and perhaps its creation is impractical given the nature of the industry. This is mostly due to vehicle manufacturers having proprietary methods and protocols for vehicle signals on their vehicle networks. Such a dataset may only be relevant for one vehicle model year of one vehicle variant from one vehicle manufacturer. While some vehicle subsystems could be reused throughout different vehicles variants and model years, a subset of the data-set may only be relevant in that instance. The creation of such dataset is also unfeasible due to the requiring a dedicated test fleet, which would imply huge costs in terms of vehicles, personal and IT infrastructure. Thus, within SecureIoT T6.4, development, testing and verification will be based on:

| Sprint#5 | M13 | - Interconnection of data collection probes to WP3 platform. From this moment, streaming and batch data shall be available through SecureIoT platform.
|         |     | - Integration of WP4 security implementations
|         |     | - Integration of WP5 SECaaS
| Sprint#6 | M14 | - Technical verification and conclusions of integrated prototypes.
|         |     | - Final version of D6.9 |
• Vehicle data simulators: PreScan [25] and SUMO [26]. This approach is quite common in the development of vehicle-based application. For instance, in [27], Rivoirard. L and all state: “The performance of VANETs (Vehicular Ad Hoc Network) can be evaluated by means of real-world testings on the ground or through modelling and simulation. For the aforementioned reasons, the first solution may be costly and will not carry out the whole field situations: it is difficult to equip a significant number of vehicles and to experiment the performance of network communication in every different traffic conditions. The second solution, namely modelling, is easier to realize by means of computers and may include many complex models. However, the accuracy depends notably on the availability of realistic mobility models.”

• Real vehicle data collected at small scale: vehicle signals captured with high-frequency for a single vehicle in a controlled environment (i.e., IDIADA test track) during a limited duration (~20minutes).

6.2.1.3 Validation Use Case Scenarios

6.2.1.3.1 Validation Use Case Scenario – Usage Based Insurance
For the usage-based insurance scenario, a driver classification profile will be developed, initially utilizing the vehicle data simulators and extended to use data from a physical vehicle to determine how a driver performs with a set of obstacles for a drive-cycle.

Driver classification will be determined based on how a driver handles obstacles such as junctions, traffic lights, roundabouts and adhering to defined speed limits.

The data simulator utilizes a steering wheel and pedals (accelerator, brake) as input to influence how the simulated vehicle performs in a simulation environment. A small route (based on a real location) will be implemented to provide context for the simulated data.

As development matures, the system will be integrated into a real vehicle to give a more realistic result. A formal validation will consist of one physical vehicle driving along a pre-defined route on the IDIADA providing ground in Barcelona.

6.2.1.3.1.1 Validation Use Case Scenario – Warnings on traffic and road conditions
Initially, the development, testing and validation of this scenario from a functional point of view will be based on SUMO [26]. To run a realistic traffic simulation with this tool the following process must be followed:
As it is explained in [28], “in SUMO a street network consists of node (junctions) and edges (streets connecting the junctions)”. It is possible to create the representation of a city road plan manually with a XML file, but it’s much more practical to import it from an already existing source like Open Street Map [29].

Thus, the most crucial part is the generation of the traffic. Again, we can describe trips (starting / end points and departure time) manually but this would not enable an accurate scenario to be built efficiently. SUMO also supports randomizing traffic demand or importing information from different options [30], e.g., Origin-Destination-Matrices (OD-matrices) provided by traffic authorities, observations made by induction loops or even population statistics. While the former option would be faster, and it may be useful for the sake of end to end testing and validation, the probabilities of obtaining reasonable results are limited. In our tests, in most cases simulations end with the fatal congestion of the whole city after a very short time if a relevant vehicle density is introduced. The latter option seems to be the best so that valuable data can be used to train a machine learning model with the capacity of classifying (or even predicting) interesting traffic situations and conditions.

For the first steps of the scenario development, we will rely on three of the most relevant SUMO scenarios that are already available, and which are mapped to real European cities:

- Bologna (Italy): one-hour period (morning peak).
- Luxembourg: 24 hours period, area of 156 km², almost 300,000 inserted vehicles.
- Cologne (Germany): 24 hours period, around 1.5 million trips.
As it can be seen, all the simulation scenarios cover situations that are of great interest for the kind of application developed within this specific use-case: big European cities with massive amounts of cars, causing traffic jams (especially at peak hours) and crashes.

### 6.2.1.4 Requirements and validation of SECaaS services

Within SecureIoT, the fundamental goal of WP6 use-cases is not the implementation of application domain specific functionalities or services leveraging on advanced and complex IoT technologies and platforms. Rather, it is the technical verification and the functional and business validation of the platform, safety mechanisms and services developed by the rest of work-packages that is of paramount importance.

Reaching a common floor between research tasks (WP2-5) and demonstration use-cases is of special importance in order to conduct an appropriate integration and validation, with the final horizon of extracting KPIs that demonstrate the potential of the project achievements. In this sense, for the Connected Car and Autonomous Driving use-cases, proposes some examples of required functionalities that solve the specific problems of this domain, explaining also how they are connected to D2.2 requirements (subsection 2.2), the process followed for their verification and hints of the potential KPIs that could be derived.

#### Table 41: Car and Autonomous Driving use-cases requirement to Risk Assessment and Mitigation Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1</td>
<td>Capability to monitor irregular messaging/activity on the vehicle CAN bus.</td>
<td>R5.1.2 R5.1.7</td>
<td>Another HW tool (Raspberry PI) is connected to the vehicle network and injects irregular and unexpected messages.</td>
<td>Precision &amp; recall [34]. Time needed to detect attack. Coherent risk assessment. Proposition of useful mitigation actions.</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Validation of end to end transmitted information, i.e., location-based information must be consistent at different data probes (i.e., OBU, network traffic, IoT cloud platform)</td>
<td>R5.1.2 R5.1.7</td>
<td>A spoofing device performs data tampering indicating that the victim vehicle is several miles away.</td>
<td></td>
</tr>
<tr>
<td>2.1.3</td>
<td>Identification of abnormal network traffic rates coming from a specific</td>
<td>R5.1.2 R5.1.7</td>
<td>A malicious device will be simulated using an IDAPT</td>
<td></td>
</tr>
</tbody>
</table>
vehicle. This situation could signalize the injection of fake information.

**2.1.4 Identification of abnormal hardware resources usage at OBU level**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>R5.1.2</td>
<td>A DDoS attack will be simulated in a test IDAPT unit.</td>
<td></td>
</tr>
</tbody>
</table>

**2.1.5 Risk assessment identifies typical security controls, i.e., use of HTTPS or SSL/TLS encryption in the link between OBU and Cloud platform.**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>R5.1.3</td>
<td>Several tests will be done using the RAE service, enabling/disabling different security control mechanisms.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 42: Car and Autonomous Driving use-cases requirement to Compliance Auditing Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>Detect existence of fine-grain logs at the different layers of the system to demonstrate accountability (GDPR related)</td>
<td>R5.2.1</td>
<td>Assessment by IDIADA / ATO teams.</td>
<td>Compliance auditing of GDPR related controls (accountability)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R5.2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2.2</td>
<td>Identification of potential sensitive information exchanged between vehicle OBU and IoT Cloud Platform (GDPR related)</td>
<td>R5.2.2</td>
<td>Assessment by IDIADA / ATO teams.</td>
<td>Compliance auditing of GDPR related controls (privacy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R5.2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2.3</td>
<td>Detect existence of detailed (per-container, per-device, per-component) granular access control for users,</td>
<td>R5.2.1</td>
<td>Assessment by IDIADA / ATO teams.</td>
<td>Compliance auditing of GDPR related controls (authentication &amp; authorization)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R5.2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R5.2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Required functionality</td>
<td>D2.2 reqs</td>
<td>Verification mechanism</td>
<td>KPIs</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Integration with FIWARE IDM KeyRock component.</td>
<td>R5.3.4</td>
<td>Functional verification as part of use-cases</td>
<td>IoT Cloud Platform supported</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Support for ARM based architectures (OBU component)</td>
<td>R5.3.3</td>
<td></td>
<td>Smart object supported</td>
</tr>
</tbody>
</table>

Table 43: Car and Autonomous Driving use-cases requirement to Programming Support Services
2.3.3 Support for distributed access control management: different access control may be applied, one for the OBU and at the IoT Cloud platform.

2.3.4 Support for Python and Node.JS programming languages.

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1</td>
<td>Include relevant information for IDAPT smart object and FIWARE main components.</td>
<td>R5.4.1</td>
<td>Assessment based on the know-how and experience of ATOS and IDIADA teams.</td>
<td>-</td>
</tr>
</tbody>
</table>

### 6.3 Validation and testing for Socially Assistive Robots and IoT Applications (SAR)

#### 6.3.1 Methodology

The socially assistive robots and IoT applications (SAR) scenarios are based on joint integration of CloudCare2U (CC2U) and LuxAI QTrobot. The scenarios consist of different use-cases, each involving multiple subsystems at edge and cloud level to cover all aspects of SECaaS services as it has been described previously, validation of the scenarios can be classified as following categories:

- **Validation of the soundness of integrated scenarios**: This activity verifies the soundness of each scenario with and without the implementation of SECaaS services to ensure that implementation and integration are correct. These are the in-lab validations, which will be carried out by the team experts, and using some automated testing frameworks such as Robot Testing Framework (RTF) [35].
- **Validation of SECaaS services and security requirements**: This activity validates scenario with the relevant integrated SECaaS services. This will be done by properly developing abuse cases; each embraces a set of security gaps and issues, which are covered by corresponding SECaaS service.

- **Validation of business value and user acceptance**: This activity mostly validate the usability and ease of use of healthcare applications for users and the professionals working in elderly and other healthcare centers, including impact of the integration of SecureIoT security services on the system performance and user acceptance. Moreover, the security challenges in implementation of such technology in elderly and health care houses will be investigated.

### 6.3.1.1 Release plan

This section defines a detailed implementation and verification plan that will verify and validate the Socially Assistive Robots and IoT applications’ functionalities, but also with the assessment of SecureIoT platform and services for SAR. It also illustrates which specific aspects will be developed and integrated for the three prototypes ’versions to be released during T6.4: D6.9 (M15), D6.10 (M24) and D6.11 (M33).

As a summary, the following goals have been defined for these deliverables:

<table>
<thead>
<tr>
<th>Release #1</th>
<th>Start</th>
<th>Ends</th>
<th>Doc</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Specific data collection probes (WP3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Implementation of scenario SAR-S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Integration of initial version of security implementations (WP4).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Integration of initial version of SECaaS (WP5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Release</th>
<th>Start</th>
<th>Ends</th>
<th>Doc</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Integration of updated WP3-5 prototypes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Functional validation of scenarios’ applications with simulated data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Initial technical assessment of WP3-5 prototypes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Release</th>
<th>Start</th>
<th>Ends</th>
<th>Doc</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Integration of updated WP3-5 prototypes</td>
</tr>
</tbody>
</table>
In the case of the first release, a more detailed scheduling has been also planned as Table 46 shows, defining sprints of one month of duration for the implementation and integration of the prototypes.

**Table 46: Sprint plan for Release#1 of Socially Assistive Robots and IoT Applications scenarios (T6.4)**

<table>
<thead>
<tr>
<th>Sprint</th>
<th>Timeline</th>
<th>Goals</th>
</tr>
</thead>
</table>
| Sprint#1 | M9       | - Deployment, configuration and testing of QT/CC2U integrated platform  
- Set-up connections from CC2U platform to QT robot. |
| Sprint#2 | M10      | - Setting-up of CC2U simulator for the generation of development and testing datasets.  
- Set-up connections from QT robot to CC2U platform  
- D6.9 ToC |
| Sprint#3 | M11      | - Data analysis, pre-processing and feature engineering.  
- Development of scenario SAR-S1 user interfaces. |
| Sprint#4 | M12      | - Development of scenario SAR-S1 application.  
- Implementation and deployment of data collection probes.  
- Prototypes implementation description in D6.9 |
| Sprint#5 | M13      | - Interconnection of data collection probes to WP3 platform.  
- Integration of WP4 security implementations  
- Integration of WP5 SECaaS |
| Sprint#6 | M14      | - Technical verification and conclusions of integrated prototypes.  
- Final version of D6.9 |

### 6.3.1.2 Datasets for development and testing

Testing, verification and validation for Socially Assistive Robots and IoT based applications is challenging due to the following reasons:

- Test information coming from a relevant number of homes and patients/caregivers is needed for algorithms relying on Big Data.
- Test information must be realistic to be useful. For instance, data for many patients can represent chronic increases or decreases depending on the patient medical history. However, there is no public access to the individual’s health records.
- Personal data protection regulations must be adhered to.
- Some medical devices rely on OEMs’ proprietary protocols and technologies.
- Injecting attacks in real medical devices may suppose risks in terms of material and personal damages.

While there is a plethora of large-scale open dataset related to IoT and sensors, we did not find any datasets, which includes rich and high-quality information about data from patient homes (except ambient parameters – temperature, humidity, but this is mostly for outdoor). This is mostly due to the sensitive nature of data collected, which could be interpreted, loosely, as personal. Thus, within SecureIoT T6.4, development, testing and verification will be based on:

- CC2U Simulator: the simulator is an evolution of the eWALL simulator, described in [35]. It is essentially a dual simulator, simulating on the one hand the life of the care recipient and on the other hand the sensor and signal processing entities that generate the metadata from the home and the body of the care recipient
- Collected data from test users: data, especially regarding sensory data generated from the QTrobot sensors will be generated from the interaction of the system by the internal members of the team such as developers and non-technical team members. The data can then be re-played, simulating the actual recorded interaction, to repeat a scenario several times for test and development purposes.

6.3.1.2.1 CC2U Simulator

For the needs of the validation of the usage scenario, a simulator from CC2U will be utilized to generate the traffic of a day-by-day case. By simulating the life of care recipients, we facilitate the evolution of any higher-level components that might be developed, and we can have a plethora of sensor data, without having to install any physical sensors. Therefore, we can achieve multiple repositories of caring home metadata, the only difference being that they are simulated rather than stemming from actual sensors.

The simulator provides virtual home care environments, comprising a generator for the activities of the care recipient in the home and outside, and for the measurements and the metadata from the systems observing the care recipients. The virtual environment has already been tested extensively, and the results have been already validated in the eWALL FP7 project.

The higher-level components of CC2U utilise metadata from the caring homes in order to reason about the context of the care recipients and to provide personalised services to them. To do so, these components need a large volume of metadata, in terms of:

- Timespan covered
- Diversity of sensors and processing algorithms
- Diversity of users

Currently the caring home sensing functionality is still evolving. Moreover, its deployments are few and partial in terms of what each site supports. In addition, the timespan the deployments are functional is sparse: the sensing environments are relatively new, and they are not always operational. A shadow user is a simulation, driven by models for the home, the weather, the sensing environment and the behaviour of the user in it. The home model is fixed at initialisation.
time, the weather is obtained from an external service, and the sensing environment model comprises conditional probabilities of the measurements given the user state. Finally, the user behaviour model is described by state transition probabilities that vary based on four user-driving forces, as discussed at the user state section.

The simulator returns all the metadata expected from an actual home, obtained by processing the measurements of the sensors. The metadata are organised into the same CouchDB databases, in the same JSON format found in actual deployments. In addition, it returns the user state that resulted to these metadata. The goal of the higher-level components is to infer this state from the given metadata.

6.3.1.3 Validation Scenarios

6.3.1.3.1 Validation Scenario – Simulation and In-House Testing of QTrobot-CC2U Integration

This validation scenario is a broader one and can be applied to any of the Use Case scenarios described in 9.2 that involves data communication between QT and CC2U, especially for UC1.2, 3.1 and 3.2. Initially we will use the CC2U simulator to generate user behaviour data, for the CC2U platform or the games embedded into CC2U web-platform. Based on this data generated by CC2U some actions will be generated, actions that will be communicated to the user by QT robot. The robot will collect user inputs and will send them back to CC2U triggering new data and new actions to be sent to QT and to be carried out by the user.

This scenario will be validated if:

1. All the actions generated by CC2U are in line with the raw data (correct processing of the raw data)
2. All the actions generated from raw data by CC2U cloud and sent to QT, are received by QT (validating the CC2U->QT communication link)
3. QT sends the correct data to the user (validating the QT->User communication link)
4. QT gathers all the user input with regards to a specific action (validating the User->QT communication link)
5. CC2U cloud receives all the user input as sent by QT (validating the QT->CC2U communication link)
6. CC2U processes the new data and generates valid new actions

6.3.1.3.2 Validation Scenario – Simulation and In-House Testing of 3rd Party System Integration

The aim is to validate both the connection between QT and CC2U and between QT and 3rd Party services. The scenario covers multiple use case like UC1.4, 2.1, 2.3, 2.4, 4.1. In this scenario, QT triggers different actions that have to be delivered by third-party applications such as Amazon Alexa or by other stakeholders like the user physician. Hence, in this type of scenario the system must provide accurate raw data so that these data will trigger the most suitable invoke of other
services. Data is generated through CC2U simulator along with data gathered from in-house test users.

This scenario will be validated if:

1. QT triggers are correctly generated from raw data (correct processing of the raw data and correct identification of suitable triggers)
2. The trigger invokes the correct 3rd Party service or send a message to the correct stakeholder (validating the trigger->action link)
3. The 3rd Party service or stakeholders receives all the data that they need in order to complete the action (validating the QT->3rdParty/stakeholder communication link)
4. The 3rd Parties services/stakeholders are providing the correct feedback as what the user would expect (validating the 3rdParty understanding and implementation of user requirements)
5. QT receives the feedback (validating the 3rdParty/stakeholder>QT communication link)
6. QT sends the feedback to the user (validating the QT/User communication link)

6.3.1.3.3 Validation Scenario – Usability and User Acceptance in Real Environment

This validation scenario focuses on the validation of the Socially Assistive Robots and IoT Applications (SAR) scenarios from the end user perspectives including both the patients and the therapists. We will investigate factors such as the usability and ease of use of the system and the acceptance of the users. We will also investigate the impact of the integration of SECaaS services on the users’ attitude and acceptance as well as on the system performance metrics from the users’ perspective.

6.3.1.3.3.1 Criteria of participation:

Elderlies between the age of 65 to 90, living in a residence care house for geriatric, who have no known diagnosis of any of the degenerative brain disorders, such as Alzheimer’s disease and are not a known case of any severe medical condition. The participants are considered as healthy seniors who have the ability to read the research explanation document and can give a conscious consent to voluntarily participate in the pilot project. The participants will be excluded from the experiment if they are not able to give a conscious consent on their willingness to participate in the experiment as well as if they have severe auditory or visual impairment or if they are not able to communicate verbally and respond to the questions.

6.3.1.3.3.2 Aims of the experiment

This study should allow us to understand:

- Procedure of implementation of an ambient assisted living in an elderly care house
- Security challenges in implementation of such technology in elderly care houses
• Security expectations of healthcare professional and decision makers in an elderly care house
• Attitude of healthcare professionals working in elderly care toward using a socially assistive robot
• Usability and ease of use of such robot for healthcare professionals working in elderly care, including impact of the integration of SecureIoT security services on the system performance and user acceptance
• Acceptance barrier for healthcare professionals working in elderly care
• Attitude of seniors living in an elderly care house toward using a robot as a companion, coach and instructor and impact of implemented security services
• Efficiency of using a social robot in improving the mood and quality of life of senior people living in an elderly care house and its ability to promote healthcare activities in seniors

6.3.1.3.3 General information about the experiment
The total duration of the study is 6 months, with a total of 5 to 10 elderly care centers, including 20 to 40 seniors living in an elderly care house and 5 to 10 healthcare professionals working in the same elderly care house.

During the session we will evaluate the baseline attitude of elderly professionals toward using a robot using standard technology acceptance questionnaires such as TAM1, TAM2, UTAUT and TPB. Then we will have a 10 to 30 minutes session for each senior person, focusing on robot applications related to cognitive games and rehabilitation physical activities. After the hands-on experiment, we will reevaluate the attitude and acceptance rate and we will compare the pre-test and post-test results to evaluate the acceptance and attitude of users.
6.3.1.4 Requirements and validation of SECaaS services

Within SecureIoT, the fundamental goal of WP6 use-cases is the technical verification and the functional and business validation of the platform, safety mechanisms and services developed by the rest of work-packages.

Reaching a common floor between research tasks (WP2-5) and demonstration use-cases is of special importance in order to conduct an appropriate integration and validation, with the final horizon of extracting KPIs that demonstrate the potential of the project achievements. In this sense, for the Socially Assistive Robots and IoT Applications use-cases, we propose some examples of required functionalities that solve the specific problems of this domain, explaining also how they are connected to D2.2 requirements (subsection 2.2), and the process followed for their verification and hints of the potential KPIs that could be derived.

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1</td>
<td>Capability to monitor irregular messaging/activity sent</td>
<td>R5.1.2</td>
<td>A WiFi enabled device is connected to the wireless network</td>
<td>Precision &amp; recall [34].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R5.1.7</td>
<td></td>
<td>Time needed to detect attack.</td>
</tr>
</tbody>
</table>

Table 47: Socially Assistive Robots and IoT Applications use-cases requirement to Risk Assessment and Mitigation Service
<p>| 2.1.2 | Validation of end to end transmitted information, i.e., patient information must be consistent at different data probes (i.e., QT robot, sensor, network traffic, CC2U cloud platform) | R5.1.2 R5.1.7 | A spoofing device performs patient data tampering making patient seem chronically ill when he is not or vice versa. | Coherent risk assessment. Proposition of useful mitigation actions. |
| 2.1.3 | Identification of abnormal network traffic rates coming from a specific robot. This situation could signalize, the injection of fake information, or leakage of information. | R5.1.2 R5.1.7 | Malicious code increasing (abnormal) network traffic is deployed in the robot. | |
| 2.1.4 | Identification of abnormal hardware resources usage at QT robot (i.e. side channels) | R5.1.2 R5.1.7 | Several probes monitoring resource allocation and usage. | |
| 2.1.5 | Risk assessment identifies typical security controls, i.e., use of HTTPS or SSL/TLS encryption in the link between the robot and Cloud platform. | R5.1.3 R5.1.6 R5.1.7 | Several tests will be done the RAE service enabling/disabling different security control mechanisms. | Coherent risk assessment. Proposition of useful mitigation actions. Performance. |
| 2.1.6 | Capability to recover from communication loss between the robot and Cloud Platform | R5.1.6 R5.1.7 | A DoS/DDoS attack will be simulated (e.g. WiFi enabled device injecting too much traffic, jamming) | Delays in systems decision making (responsiveness) . Impact on user metrics. |</p>
<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>Detect existence of fine-grain logs at the different layers of the system to demonstrate accountability (GDPR related)</td>
<td>R5.2.1, R5.2.4, R5.2.5</td>
<td>Internal assessment (iSPRINT/LuxAI) + Partner support</td>
<td>Compliance auditing of GDPR related controls (accountability)</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Identification of potential sensitive information exchanged between QT robot and CC2U Cloud Platform (GDPR related)</td>
<td>R5.2.2, R5.2.4</td>
<td>Internal assessment (iSPRINT/LuxAI) + Partner support</td>
<td>Compliance auditing of GDPR related controls (privacy)</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Detect existence of detailed (per-device, per-component) granular access control for users, processes, etc. (GDPR related)</td>
<td>R5.2.1, R5.2.3, R5.2.4, R5.2.5</td>
<td>Internal assessment (iSPRINT/LuxAI) + Partner support</td>
<td>Compliance auditing of GDPR related controls (authentication &amp; authorization)</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Verifying that communications between QT and CC2U Cloud platform are encrypted and rely on mutual authentications (NIST / ENISA related)</td>
<td>R5.2.1, R5.2.4</td>
<td>Several tests will be done using the Compliance auditing service, enabling/disabling different security control mechanisms.</td>
<td>Compliance auditing of ENISA related controls (privacy)</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Software modules must not be able to escalate its privileges (GSMA related).</td>
<td>R5.2.2, R5.2.3</td>
<td>CC2U or QT Cloud software running on an cloud platform will be updated using OTA mechanism. The new version will simulate a malicious version that tries to inject commands in the web-service communication and to obtain statistics</td>
<td>Compliance auditing of GSMA related controls</td>
</tr>
</tbody>
</table>
The robot will also be affected by an OTA update (update QT base software/middleware).

### Table 49: Socially Assistive Robots and IoT Applications use-cases requirement to Programming Support Services

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.1</td>
<td>Support for typical cloud infrastructures</td>
<td></td>
<td>Functional verification as part of use-cases</td>
<td>CC2U Cloud Platform supported</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Support for android-based Platform (QTrobot tablet)</td>
<td>R5.3.3 R5.3.4</td>
<td></td>
<td>Number of f supported platforms</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Support for distributed access control management: different access control may be applied, one for QT robot and at the CC2U Cloud platform.</td>
<td>R5.3.1</td>
<td></td>
<td>Multiplatform interactions demonstrated</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Support for Python and Node.JS programming languages.</td>
<td>R5.3.5</td>
<td></td>
<td>Number of mainstream programming languages supported</td>
</tr>
</tbody>
</table>

### Table 50: Socially Assistive Robots and IoT Applications use-cases requirement to IoT Security Knowledge Base

<table>
<thead>
<tr>
<th>ID</th>
<th>Required functionality</th>
<th>D2.2 reqs</th>
<th>Verification mechanism</th>
<th>KPIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1</td>
<td>Include relevant information about QTrobot components and CC2U supported sensors and devices.</td>
<td>R5.4.1</td>
<td>Internal assessment (iSPRINT/LuxAI) + Partner support.</td>
<td>Coverage of the main components</td>
</tr>
</tbody>
</table>
7 Integration Guidelines

This section contains an initial set of requirements that shall be satisfied by the different tiers of SecureIoT platform in order to ensure a smooth and seamless integration with relevant IoT scenarios and use-cases.

7.1 Data Collection and Actuation Layer

The main characteristics of IoT systems identified by ISO/IEC 30141 have been used as baseline to the specification of the integration guidelines with respect to the data collection and actuation layer.

- **Auto-configuration**
  Data collection probes should include default settings to automatically work against SecureIoT infrastructure. Appropriate security and authentication mechanisms will be included.

- **Management**
  SecureIoT platform shall include visualization tools, dashboards and APIs to allow the management of deployed probes and the customization of their configuration. Some of the features controlled by the management dashboards shall be: type of data to be monitored, intelligent collection mechanisms, access control, security mechanisms to be applied, etc.

- **Network communication**
  The exchange of collected data with SecureIoT cloud platform shall rely on appropriate, reliable and robust communication protocols, capable of supporting the specific requirements of each use-case.

- **Real-time capabilities**
  SecureIoT probes must be able to subscribe to message brokers or events at the different levels of the IoT stack.
  Real-time information will be routed to the upper layers of the system through a streaming infrastructure.

- **Context-awareness**
  Data probes shall be able to modify dynamically its behavior and performance considering the context, e.g., different quality of service levels may be applied to different types of data.

- **Timeliness**
  Data probes shall be able to collect data with multiple sampling rates in a proactive way or passively by subscription.
  Safety-critical environment may impose additional requirements in terms of latency.

- **Composability / compatibility**
  Data probes may be able interoperable (i.e. capacity to monitor) the most common types of IoT related platforms (e.g. FIWARE, MindSphere), components (e.g., MongoDB, MQTT,
AMQP), networks (HTTP, CoAP) and hardware platforms (e.g. Arduino, Raspberry Pi, STM32).

- **Discoverability**
  SecureIoT shall include mechanisms for automatic provisioning of deployed data probes.

- **Shareability**
  Depending on the nature of the system, component or network to be monitored, data probes may collect information from multiple entities. For instance, probes deployed at an IoT gateway may gather information from connected sensor and smart objects.

- **Unique identification**
  For traceability purposes, all data probes must have a unique identifier.

- **Confidentiality**
  Mechanisms to protect data at rest and in transit must be implemented, e.g., authentication, authorization, encryption, etc.

- **Integrity**
  Mechanisms to ensure that information is not altered must be implemented in data probes.

- **Protection of personally identifiable information**
  In general, data probes must apply anonymization mechanisms to remove sensitive information from collected data.

- **Regulation compliance**
  Probes must respect the statements of relevant general regulations (e.g. GDPR) and scenario-specific ones.

- **Scalability**
  Data probes at cloud level must be able to scale horizontally in order to attend increasing volumes of data.

- **Deployment**
  Data probes must support deployable using Docker containers. Images for different architectures will be provided.

### 7.2 Security Intelligence layer

The security intelligence layer is wedged between the data collection and actuation layer and the SECaaS layer. As such, it should be designed to be easily integrated with both, and both (the other 2 layers) should be designed in such a way to allow smooth integration with this layer.

The following logical resources from the security intelligence layer can be manipulated using API methods:

- IoT Security Templates
- Probes Registry
- Security Policies
- Template Executions (Rules)
The Security Intelligence API will provide custom methods for querying stats about various entities registered in it. A swagger definition will be provided.

7.2.1 Typical Action Types / Verbs
- **CRUD operations** - GET, POST, PUT and DELETE requests on most logical resources. These are also known as verbs. Please note that not all resources support all the operations because of both technical and business reasons.
- **Requesting Data** - There are multiple parameters that, used in conjunction with read (GET) operations, will help query, filter, sort, select, and expand specific information.
- Custom actions and actions which are not tied to a logical resource:
  - Stats methods provide aggregated information on different entities;
  - Account methods provide authentication methods to Intelligence Layer;

The Probes Registry resource is used by the Intelligence Layer to communicate with the IoT systems.

7.2.2 Specific API examples

7.2.2.1 IoT Security Templates
GET IoT Security Templates
- Retrieve list of IoT Security Templates
- Retrieve list of IoT Security Templates for specific platform
- Retrieve list of IoT Security Templates for specific device
- Retrieve list of IoT Security Templates by Status
- Retrieve IoT Security Template by name/id

POST IoT Security Templates
- Add IoT Security Template
- Add IoT Security Template for specific platform
- Add IoT Security Template for specific device

PUT IoT Security Templates
- Update IoT Security Template for specific platform
- Update IoT Security Template for specific device
- Update Status of IoT Security Template (active/obsolete)

DELETE
7.2.2.2 Probes Registry

GET Probes

- Retrieve list of Probes
- Retrieve list of Probes for specific platform
- Retrieve list of Probes for specific device
- Retrieve list of Probes by Status
- Retrieve Probe by name/id

POST Probes

- Add Probe
- Add Probe for specific platform
- Add Probe for specific device

PUT Probes

- Update Probe for specific platform
- Update Probe for specific device
- Update Status of Probe (active/inactive)

DELETE

- Delete Probe for specific platform
- Delete Probe for specific device
- Delete Probe by name/id

7.2.2.3 Security Policies

GET Security Policy

- Retrieve list of Security Policies
- Retrieve list of Security Policies for specific platform
- Retrieve list of Security Policies for specific device
- Retrieve list of Security Policies by Status
- Retrieve Security Policy by name/id

POST Security Policy

- Add Security Policy
• Add Security Policy for specific platform
• Add Security Policy for specific device

**PUT Security Policy**

• Update Security Policy for specific platform
• Update Security Policy for specific device
• Update Status of Security Policy (enforced/not enforced)

**DELETE**

• Delete Security Policy for specific platform
• Delete Security Policy for specific device
• Delete Security Policy by name/id

### 7.2.2.4 Template Executions (Rules) / IoT Security Knowledge Base

**GET Rules**

• Retrieve list of Rules
• Retrieve list of Rules for specific platform
• Retrieve list of Rules for specific device
• Retrieve list of Rules by Status
• Retrieve list of Rules by start/end date
• Retrieve Rule by name/id

**POST Rules**

• Add Rule
• Add Rule for specific platform
• Add Rule for specific device

**PUT Rules**

• Update Rule for specific platform
• Update Rule for specific device
• Update Status of Rule (used/not used)

**DELETE**

• Delete Rule for specific platform
• Delete Rule for specific device
• Delete Rule by name/id
7.2.2.5 **IoT Security Analytics**

**GET Analytics**

- Retrieve list of Analytics
- Retrieve list of Analytics for specific platform
- Retrieve list of Analytics for specific device
- Retrieve list of Analytics for different filters (start/end date, list of devices, platforms etc.)
- Retrieve Report by type
- Retrieve Report for specific platform
- Retrieve Report for specific device
- Retrieve Report for different filters (start/end date, list of devices, platforms etc.)

7.2.2.6 **Configuration**

**GET Config**

- Retrieve list of configurations
- Retrieve configuration for specific platform
- Retrieve configuration for specific device
- Retrieve configuration by type
- Retrieve configuration for different filters

**PUT Config**

- Update config for specific device
- Update config for specific platform
- Update config by type

7.3 **SECaaS layer**

Guidelines related to SECaaS layer will be provided in deliverable D6.2.

7.4 **General guidelines for 3rd party vendor integration**

By allowing products to easily integrate with SecureIoT one can add considerable value to one’s product and improve the user’s security experience. As a general guideline for any 3rd party vendors that want to connect with SecureIoT:

1. Build extensive SecureIoT know-how before starting the testing phase so that the integration problems won’t be mistaken as SecureIoT issues.
2. Always verify the communication between the 3rd party vendor product and SecureIoT
3. Always check if the functionalities remain intact following the end-to-end integration process or after a system upgrade (either 3rd party or SecureIoT)

To properly integrate a system with SecureIoT the following methodology [37] should be followed:

1. **The proper test infrastructure**
   Based on the number of interfaces a user can access 3rd party interface (mobile, web, desktop) every one of them should be tested with the SecureIoT platform. Also, other variables, should be anticipated and tested for, such as feature functionality across browsers, supported languages, and platform support for third-party integrations.

2. **Know-how about SecureIoT platform (API, functionalities that it provides)**
   Detailed know-how about SecureIoT platform allows testers to anticipate latent defects related to the integration process, API integrations, and more.

3. **Cross-platform check**
   A number of platforms will be used to access the 3rd party application, and compatibility testing should be performed to ensure a smooth, intuitive user experience regardless of OS/browser configuration.

4. **Perform volume (soak) testing to maintain data integrity**
   Typically, heavy data is uploaded and transferred via the SecureIoT integration. In some cases, a defect in the application can prevent a part of the data from uploading. Soak testing has to be performed to capture these defects before moving the integration to production.

5. **Maintain the security of 3rd party users and their data.**
   Though security is integral in all applications, it is especially relevant for integration with third parties like, SecureIoT. Neglecting to test what effect will access of SecureIoT have on your user base can have many negative consequences.

6. **Verify that permissions and restrictions are accurate.**
   Some of the systems integrated with SecureIoT will have countless users, each with different permission and restriction settings. Possible user scenarios, such as access with read-only/write-only privileges have to be tested and checked to see if they have access to the data provided by SecureIoT.

7. **Become the end-user**
   This is often not as easy as it sounds and is forgotten during the integration releases. An effective way to catch defects is to test the integration as if the tester were an actual user of the final system.
8 Conclusions

In the previous sections of deliverable 6.1, the first version of the Detailed Specifications of Usage Scenarios and Planning of Validation Activities was presented. The document follows the methodology that has been introduced in the project with the specification of the iterations and the related planning. The usage scenarios of SecureIoT, namely Multi-Vendor Industrie 4.0; Connected Car and Autonomous Driving; Socially Assistive Robots and IoT Applications were described, with information about the attacker motivational rationale, benchmarking, presentation of metrics & KPIs in both ways of integration, the use/abuse cases involved and its validation. Further to the specifications of the use cases, the validation methodology was also presented and linked with the requirements defined under D2.2 [2] with the validation mechanisms and the KPIs. Moreover, the deliverable presents generic integration guidelines for any system to be used, beyond the three usage scenarios.

This version of the specifications and validation activities, consider the status of development of the project and D6.1 output (see sections 6 and 7) will be used by the project WPs to elaborate the design and development of the systems’ components. In the next version of this document (D6.2 in M18), a final iteration will be presented that will consider all updates from WP3, WP4 and WP5, to result in the final specifications and validation planning.
9 References

[1] SecureIoT deliverable D2.1
[2] SecureIoT deliverable D2.2
[17] The eWALL project Deliverable D2.7: Final user and system requirements and architecture. March 2015.
[25] https://tass.plm.automation.siemens.com/prescan
[29] https://www.openstreetmap.org/
[34] https://en.wikipedia.org/wiki/Precision_and_recall
[37] 7 Essential Tips for Functional Testing of Collaboration Applications
https://blog.qasource.com/7-essential-tips-for-functional-testing-of-collaboration-applications/