



SEMIoTICS

Deliverable D2.2

SEMIoTICS usage scenarios and requirements

Deliverable release date	Initial 08.06.2018, revised 25.11.2019
Authors	<ol style="list-style-type: none"> 1. Ermin Sakic, Darko Anicic, Arne Bröring, Vivek Kulkarni (SAG) 2. Christos Tzagkarakis, George Hatzivasilis FORTH, 3. Jordi Serra, Luis Sanabria-Russo (CTTC), 4. George Spanoudakis (STS), 5. Domenico Presenza, Keven Kearney (ENG), 6. Danilo Pau, Mirko Falchetto (ST-I), 7. Tobias Marktscheffel, Joachim Posegga (UP) 8. Urszula Rak, Sylwester Zieliński, Karolina Walędzik, Łukasz Ciechomski (BS) 9. Kostas Ramantas, Prodromos Vasileios (IQU)
Responsible person	Vivek Kulkarni (SAG)
Reviewed by	Łukasz Ciechomski (BS), Mirko Falchetto (ST-I), Karolina Waleczik (BS), Christos Tzagkarakis (FORTH), Jordi Serra (CTTC), Arne Bröring (SAG), Urszula Rak (BS), Darko Anicic (SAG), George Hatzivasilis (FORTH), Ermin Sakic (SAG), Keven Kearney (ENG), Volkmar Doericht (SAG), Luis Sanabria-Russo (CTTC), Tobias Marktscheffel (UP), Prodromos Vasileios (IQU), George Spanoudakis (STS), Philip Wright (ENG)
Approved by	<p>PTC Members (Vivek Kulkarni, Nikolaos Petroulakis, Ermin Sakic, Mirko Falchetto, Domenico Presenza, Verikoukis Christos)</p> <p>PCC Members (Vivek Kulkarni, Ioannis Askoxylakis, Verikoukis Christos, Georgios Spanoudakis, Domenico Presenza, Danilo Pau, Joachim Posegga, Darek Dober, Kostas Ramantas, Ulrich Hansen)</p>
Status of the Document	Final
Version	1.0 revised
Dissemination level	Public

Table of Contents

1.	Introduction	5
1.1.	PERT chart of SEMIoTICS.....	6
2.	SEMIoTICS Use Cases	7
2.1.	Use Case 1: Local smart behavior in a wind turbine to provide value added services.....	8
2.2.	Use Case 2: Socially Assistive Robotic Solution for Ambient assisted living (SARA)	19
2.3.	Use Case 3: Artificial Intelligent Embedded Sensing Platform	39
3.	Non-SEMIoTICS Use Cases.....	49
3.1.	Use Case 4: SPDI pattern-based management of smart building infrastructure	50
3.2.	Use Case 5: Leveraging SDN/NFV for minimizing energy consumption in multiple smart buildings	58
3.3.	Use Case 6: IoT platform interoperability in case of a power plants.....	69
3.4.	Use Case 7: Adaptive monitoring of the smart micro-grid in a cluster of buildings	80
3.5.	Use Case 8: Machine learning and edge analytics for smart cities	87
3.6.	Use Case 9: Security and privacy enhanced smart wearables.....	94
3.7.	Use Case 10: Semantics in IIoT engineering and orchestration in industry automation systems	98
4.	Requirements classification for SEMIoTICS architecture	107
4.1.	Classification of requirements	107
5.	Summary.....	126

List of Use Case Authors and Reviewers

Nr.	Authors	Consortium internal reviewers
Use Case 1	Ermin Sakic (Siemens), Dr. Darko Anicic (Siemens), Dr. Arne Bröring (Siemens), Vivek Kulkarni (Siemens)	Łukasz Ciechomski (BlueSoft), Mirko Falchetto (ST-I)
Use Case 2	Domenico Presenza (ENG) Keven Kearney (ENG)	Karolina Waledzik (BlueSoft), Christos Tzagkarakis (FORTH)
Use Case 3	Danilo Pau (ST-I) Mirko Falchetto (ST-I)	Dr. Jordi Serra (CTTC), Dr. Arne Bröring (Siemens)
Use Case 4	Dr. George Hatzivasilis (FORTH)	Urszula Rak (BlueSoft), Dr. Darko Anicic (Siemens)
Use Case 5	Dr. Jordi Serra (CTTC) Dr. Luis Sanabria-Russo (CTTC) Dr. Kostas Ramantas (IQUADRAT)	Dr. George Hatzivasilis (FORTH), Ermin Sakic (Siemens)
Use Case 6	Urszula Rak (BlueSoft), Sylwester Zieliński (Bluesoft), Karolina Walędzik (Bluesoft), Łukasz Ciechomski (Blusoft)	Keven Kearney (ENG), Volkmar Doericht (Siemens), Mirko Falchetto (ST-I)
Use Case 7	Prodromos Vasileios (IQUADRAT)	Dr. Luis Sanabria-Russo (CTTC), Dr. George Hatzivasilis (FORTH)
Use Case 8	Christos Tzagkarakis (FORTH)	Tobias Marktscheffel (Uni Passau), Prodromos Vasileios (IQUADRAT)
Use Case 9	Tobias Marktscheffel (Uni Passau), Prof. Joachim Posegga (Uni Passau)	Dr. George Spanoudakis (Sphynx), Philip Wright (ENG)
Use Case 10	Dr. Darko Anicic (Siemens), Dr. Arne Bröring (Siemens)	Dr. George Spanoudakis (Sphynx), Prodromos Vasileios (IQUADRAT), Tobias Marktscheffel (Uni Passau)

EXECUTIVE SUMMARY

Global networks like Internet of Things (IoT) create an enormous potential for new generations of IoT applications, by leveraging synergies arising from the convergence of consumer, business and industrial Internet which eventually creates an open & global network connecting people, data, and “things”. SEMIoTICS relates to smart objects, IoT applications, and existing IoT platforms, and how do they map onto a generic deployment infrastructure consisting of private and public clouds, networks, and field devices. *SEMIoTICS* will target three IoT application scenarios: two verticals in the areas of energy and health care and one horizontal in the areas of intelligent sensing. These scenarios have been selected since they involve: (a) different and heterogeneous types of smart objects (i.e., sensors, smart devices, actuators) and software components; (b) different vertical and horizontal IoT platforms; and (c) different types of security, privacy, dependability and interoperability requirements.

This document reveals the foreseen usage of secure and dependable actuation and semi-autonomic behaviour in IoT/IIoT applications and their corresponding requirements in 5 specific domains - Energy, Healthcare, Smart City, Smart Building and Industry Automation. In total, 10 use cases (3 SEMIoTICS and 7 Non-SEMIoTICS) in these domains are elaborated including use case specific requirements. The main output of this deliverable includes important requirements to be detailed in Task 2.3 for SEMIoTICS demonstration scenarios. The TRL level 4 is foreseen by the project for those 3 Lab trial scenarios. These requirements are classified based on certain defined classes and relevant scenarios which will be demonstrated in Work Package 5 (WP5). This deliverable gives input to the following tasks in WP2 and the requirements given in this deliverable will be analyzed further in SEMIoTICS architecture definition for smart sensing and smart actuation.

1. Introduction

This deliverable provides an analysis of smart sensing and actuation use cases from various industrial domains and provides the basis for the following activities for requirements analysis and architecture definition in WP2 of SEMIoTICS.

The idea behind the different use cases in this deliverable is to find out the existing use of smart sensing, smart actuation and semi-autonomic behaviour in IoT/Industrial IoT (IIoT) domains where these technologies are poised to be well established and to foresee usage in new domains where these technologies may be used and exploited further after the project. The additional use cases (Non-SEMIoTICS use cases) are coming from the partner competencies in the project as mentioned in the Description of Action (DoA). In total, there are 5 broad categories of use cases in this deliverable, namely:

- Energy (use cases 1 & 6)
- Healthcare (use case 2)
- Smart City (use cases 3, 7 & 8)
- Smart Building (use cases 4 & 5)
- Industry Automation (use case 10)

From the 10 use-cases, SEMIoTICS will perform the work to demonstrate first 3 (UC1-UC3) of the 10 use cases. There is further use case (9 – Security and Privacy) which belongs to all these five categories due to its generic nature of application. Chapter 2 of this document describes in-depth details about 3 SEMIoTICS use cases and their respective requirements which will be demonstrated in Lab trials. Chapter 3 describes details about 7 Non-SEMIoTICS use cases, which will not be shown in any Lab trials of SEMIoTICS. Chapter 4 summarizes the common requirements from all these ten use cases along with the classification of those selected requirements. The TRL level 4-6 is foreseen by the project for those 3 Lab trial (UC1-UC3) scenarios. Finally, chapter 5 summarizes the outcome of this deliverable. The overall approach and methodology is described in the following Figure 1.

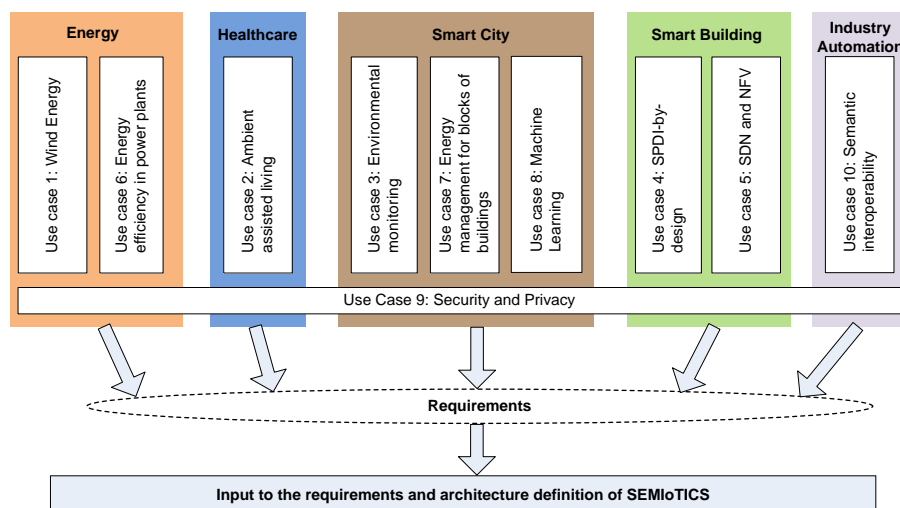
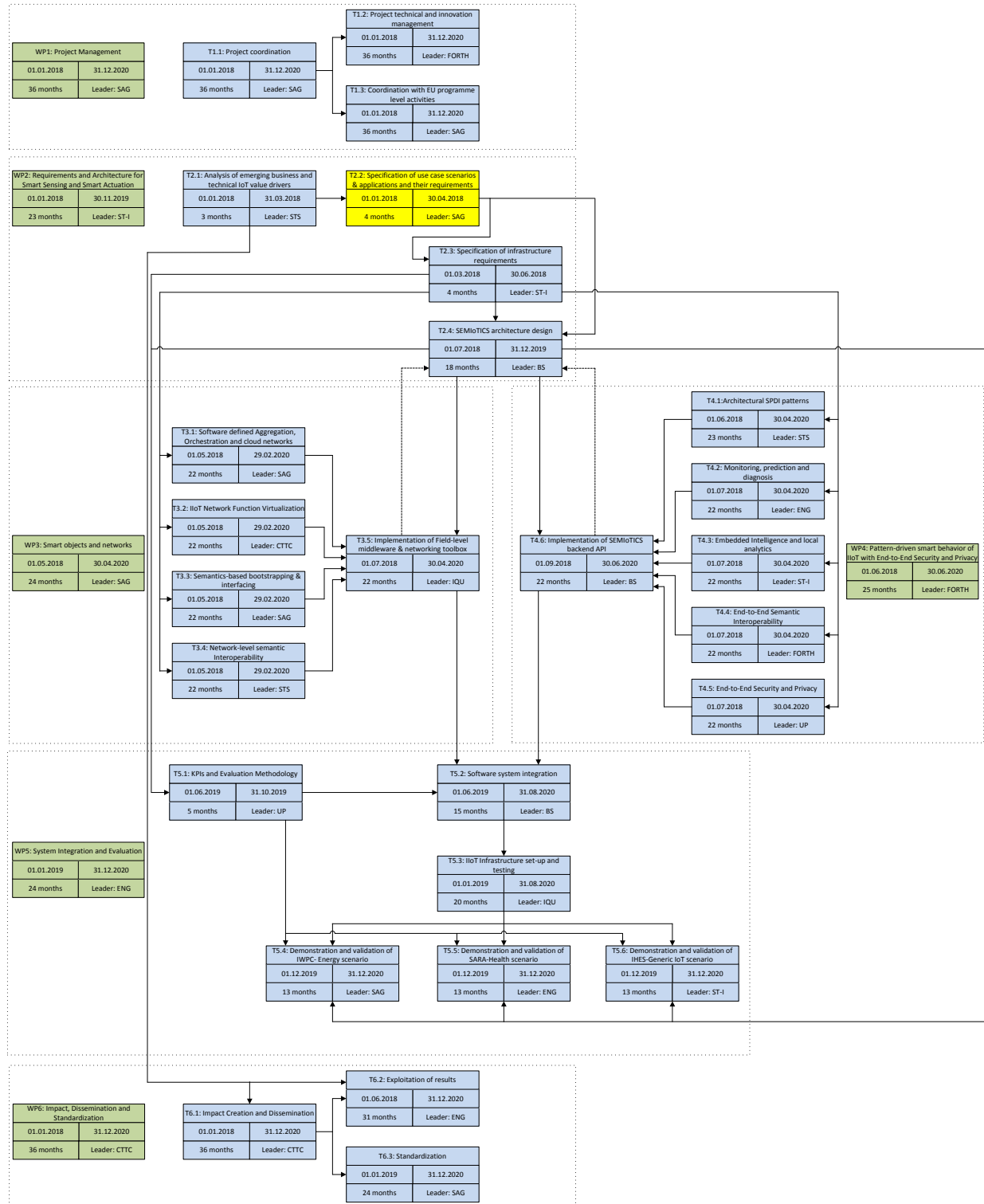


FIGURE 1 APPROACH AND METHODOLOGY

1.1. PERT chart of SEMIoTICS



Please note that the PERT chart is kept on task level for better readability.

2. SEMIoTICS Use Cases

This chapter contains 3 in-depth use cases belonging to SEMIoTICS. Each use case has the following uniform structure and the text marked in *blue* contains the respective use case specific information.

Sub-section	Description
1	Scope and Objectives of Use Case: <i>The scope defines the limits of the use case and the objectives mention the motive behind the use case</i>
2	Narrative of Use Case (Short & Complete description): <i>Short description is intended to summarize the main idea of use case and Complete description is a full narrative of the use case from an expert user's point of view, describing what occurs when, why, with what expectation, and under what conditions.</i>
3	Technical Details
	3.1 Diagrams of Use Case: <i>The diagrams show step-by-step interactions. e.g. Use case diagram, sequence diagram, activity diagram, etc.</i>
	3.2 Actors: <i>In this section, actors which are involved in the use case are listed and described. These can for instance include people, systems, applications, databases, devices, etc.</i>
	3.3 Triggering Event, Preconditions, Assumptions, Success criteria/expected outcome: <i>Triggering event describes what event(s) trigger(s) this use case (Actor/System/Information/Contract). Pre-condition(s) describe(s) what condition(s) should have been met before this use case happens. Assumption may be used to define further general assumptions for this use case. Finally, the success criteria or expected outcome of the use case, if foreseen is mentioned.</i>
	3.4 Information exchanged between actors: <i>It summarizes the information exchanged between two actors with high level concepts for SDN/NFV.</i>
	3.5 Requirements description: <i>Requirements coming from high level concepts and technical details written in the use case.</i>

2.1. Use Case 1: Local smart behavior in a wind turbine to provide value added services

The overall success criteria of UC1, is to demonstrate the coexistence of a highly integrated control system and an agile IIoT ecosystem based on the SEMIoTICS framework; which in return allows service providers to deploy new value-added services faster and provide effective access to data from both an existing control system and the newly deployed edge sensor network. The demonstrator of UC1 in the lab trials will address some of the common challenges in extending the capabilities of an existing control system in a brownfield environment in the Wind Energy domain. Control systems for industrial processes are in general terms heavily embedded of nature. Extending the capabilities of such a system typically involves engagement with component manufacturer or service integrator to fuse-in the new functionality, that must be tested to validate the new component does not void the integrity and the intended behavior of the control system. Most of these control systems expose certain high-level machine-readable interfaces for seamless integration with other systems, this is not enough to take full advantage of the value that can be generated.

The SEMIoTICS approach providing an end-to-end eco-system allowing service integrators and service operators to build and deploy new value-added services, interfacing directly with the existing assets, is unprecedented.

This SEMIoTICS use case 1 will be therefore shown in lab trial with Wind turbine model showing the real-world IIoT field devices and applications.

2.1.1. SCOPE AND OBJECTIVES OF USE CASE

The scope of this use case is limited to a wind turbine which consists of IIoT sensors and an IIoT Gateway connected to a 5G communication network in the wind park including internal Supervisory Control and Data Acquisition (SCADA) network (Core switches + SCADA servers, turbine switches + turbine servers) and external interfaces for remote services.

Objectives:

Current state of the art of Wind Turbine Controllers in a Wind Park control network is typically an embedded or highly integrated operating system, which follows a long process of thorough development, testing, pre-qualification, and finally deployment in the real world. As a result of this long process, introduction of new features, addition of new sensors, actuators and related advancements require several months or even years to be fully matured and operational in the field. To improve this situation and shorten this development cycle, the following advancements are focus of this use case:

- 1) **Local predictive analytics for structured data:** Operations and Management (O&M) personnel in the remote control center wants to know the inclination of all the steel towers on a number of specific wind farms, as these details will have to be shared with the customer to monitor the deformation and fatigue of the steel. To determine the inclination of a steel tower, a full cable-untwist procedure has to be activated. This happens, depending on wind conditions, 3-4 times a month. It is also possible to manually instruct the wind turbine to perform the unwind procedure. At the time of the unwinding procedure a high-frequency set of data is recorded. A relatively large amount of data is required to calculate the inclination. This data needs to be sent back to the remote control center to model and calculate the inclination. In SEMIoTICS, localized edge analytics will be applied, which will result in semi-autonomous IIoT behavior as only the container comprising the algorithm and the result of the inclination calculation is transferred between the wind turbine and the remote control center.

- 2) Local smart behavior and monitoring:** The sensing of unstructured data (e.g. video or audio) and acting locally to prevent any damage to the parts of the turbine in the long run is of key importance. Furthermore a federation, based on a semantic data model and capabilities, must be formed between the IIoT Gateway and the legacy control system or another IIoT Gateway, to enable the use of data exposed by the sensors connected to the legacy control system. This will allow a Cloud-App of the IoT ecosystem or the embedded localized compute capabilities of the IIoT Gateway to make use of data exposed by an existing sensor infrastructure, in an autonomous and local setup.
- 3) Multi-tenant capable network service establishment:** Wind Park network typically comprises of multiple stakeholders requiring access to a common network infrastructure. Providing reliable and Quality of Service (QoS)-constrained packet transmission for critical IoT services, in conjunction with non-real-time and bandwidth-heavy services necessitates service differentiation. Service slicing can be leveraged to distinguish different types of application and potentially tenants (stakeholders). Similarly, network slicing can enable the according network resource reservations in a manner where the potentially negative or malicious services are isolated from well-behaving critical services. Maintenance of the related service reservations is much easier in a centralized manner, where i.e. a single logical controller is in charge of a series of network resource reservations. Thus, a Software Defined Network (SDN)-like operation can support this use case. In SEMIoTICS, the IoT/Industrial IoT end-devices communicate their network requirements to the SDN controller, using the service specification advertisement messages, sent either directly to the northbound interface of the SDN controller, or proxied to the SDN controller using packet-in messages in SDN-capable switches.
- 4) Dependable and scalable SDN operation in both data-plane and control-plane:** Scalability and dependability is a very important factor in critical infrastructures such as wind parks. With SDN, centralized state management of the packet forwarding devices can allow for guaranteed QoS metrics in the communication between the IIoT gateway and the control systems in remote or private cloud. However, to cater for dependability of the IoT communication, the following data-plane and control-plane reliability issues must be considered:

 - a. Data-plane reliability concepts in SDN include packet duplication (redundant transmission) and fast-failover methods, where output ports of the data flows are reconfigured based on the current availability of links or end-to-end paths. SEMIoTICS will enable redundant configuration of data-flows so to provide for guaranteed availability of IoT flow transmissions independent of the source of failure. Depending on the failure model (single-failures or multiple-failures), a variable overhead in terms of utilized bandwidth resources must be accounted for the resource planning model in the SDN controller.
 - b. Current approaches to tackling the SDN controller's single point of failure in SDN entail controller state synchronization, which enables a distributed operation of the SDN controller instances. The state synchronization process is furthermore reliant on the assumption of the correct decision-making in the SDN controllers. Unavailable, unreliable (e.g. buggy) and malicious controller failures must, however, also be catered for. SEMIoTICS will introduce a control plane design that tolerates the unavailability and byzantine failures, i.e. distinguishing

and localizing the faulty distributed controller instances and appropriately reconfigures the control plane. To cater for scalability constraints related to handle a massive number of IoT devices, an intelligent controller-switch-client connection assignment must be introduced, in order to optimize the number of active controllers and the tolerable service request throughput. The assignment procedure must consider the overall controller capacities, and it must ideally minimize the controller and switch reconfiguration delays, so to minimize the request processing blocking times in the SDN controllers.

- 5) **Semantic-based engineering** – In our scenario, an engineer wants to add a new IIoT device (e.g., a sensor, actuator) or replace existing one with less effort. The device should be bootstrapped in a semi-automated fashion. Further on, it should expose its functionality over an IIoT Gateway via a machine-interpretable interface so that the engineer can develop a new service or application with that device. With the semantic approach in SEMIoTICS, the integration and configuration of a new IIoT device in an automation system of a Wind Park requires significantly less effort. For this purpose the IIoT device is equipped with a semantic description, which supports the process of device discovery, as well as commissioning, engineering and re-engineering of automation functions in an ecosystem of heterogeneous devices and services. The SEMIoTICS semantic models, which enable device descriptions, are based on standardized semantics such as OPC UA¹, and W3C Web of Things² (Thing Description), as well as on the IoT domain semantics created by iot.schema.org³. Thanks to the concept of semantic-based engineering, SEMIoTICS supports creation of IIoT applications in a cost-effective manner, such as for example, local predictive analytics for structured data and local smart behaviour and monitoring (see below).
- 6) **Semantics-based application creation** – Going beyond integration and configuration supported by the “semantic-based engineering” (described above), an application developer needs to be supported in connecting and composing services and devices to Edge-level or Cloud-level applications.
 - a. In order to facilitate quicker development of applications, typical types of compositions need to be selectable by the developer as templates (or “recipes”) that clearly describe how the logical links are established and what kind of devices are required in these compositions. For example, the process of deploying a firmware update or a common way of connecting sensors, such as a new anemometer, with an alarm notification system can be defined as templates. Another example is shown in Figure 3. Here, the local analytics are configured from a template composition. If the analysis of an audio signal identifies an anomaly, the data from an accelerometer and the anemometer are requested and it is double-checked if the weather conditions are rough. In case this is validated, an email is sent out.
 - b. As part of these definitions of compositions, the developer needs to be able to define QoS criteria on the network connections between the logical links. This requires a full integration of QoS network-related criteria with the semantic device and service descriptions. Hence, these QoS criteria need to be semantically described and interpretable by an SDN controller prior to the deployment of the application in order to check whether the communication infrastructure can meet the requirements of the application. When developing an application, the developer needs to be in position to orchestrate network resources in the same way as

¹ <https://opcfoundation.org/about/opc-technologies/opc-ua/>

² <https://www.w3.org/WoT/WG/>

³ Currently available from: <http://iotschema.org/>

resources exposed by field devices. Therefore, a semantic model for describing QoS network-related parameters and the SDN/Network Functions Virtualization (NFV) infrastructure is required, so that an automated evaluation of both is enabled. In the example of Figure 3, the application creation is attached with networking specific criteria from the high-level application point of view.

2.1.2. NARRATIVE OF USE CASE

2.1.2.1. SHORT DESCRIPTION

Monitoring and maintenance is critical to ensure a continuous power production and to avoid any malfunction. In the worst case, a lack of maintenance could lead to a turbine shutdown and destruction, cutting off the production of power. Storms, strong gusts of wind or a wind energy plant's boot-up phase can cause a tilt of the tower during operation. This inclination of the tower can shorten the overall plant lifetime when threshold values are exceeded. O&M personnel in the remote control center wants to know the inclination of all the steel towers on a number of specific wind farms, as these details will have to be shared with the customer to monitor the deformation and fatigue of the steel. As per current state of the art, high-frequency structured data need to be recorded and sent to the control center to calculate the inclination. This use case narrates the importance of measuring the absolute inclination angle locally inside the turbine in a reliable manner preventing limit values from being exceeded and thereby reducing the load on the system and without sending large amount of data back to the control center. Another example tackled by this use case is smart actuation by sensing unstructured video or audio data. When the turbine rotor is changing the direction in the line of wind to maximize energy production, then IIoT sensors can detect grease leakage or unintended noise. This detection/sensing of the unstructured data and acting locally to prevent any damage to the parts of the turbine in the long run will be of key importance.

2.1.2.2. COMPLETE DESCRIPTION

As of today Wind Turbine Controller in a Wind Park control network is typically an embedded or highly integrated operating system, which follows rigorously development and pre-qualification prior to deployment in the real world. As a result of this slow process, implementing new features, addition of new sensors, actuators and related advancements require several months or even years to be fully matured and operational in the field.

The Wind Turbine Controller controlling a respective turbine in a Wind Park operates at the edge of the turbine's network. Systems like these are increasingly able to leverage current cloud/backend-based resources to perform edge computing - if computing resources exist as needed along the path from a sensor to the cloud - and if these computing resources reduce the total amount of data to be sent to the cloud for storage, processing, and analysis. This use case tackles 2 sub use-cases: one with structured and the other with unstructured data. In both of the use cases, as per the current state of the art, all the data generated is sent to the remote control center for processing and analysis. This can sometimes be high-frequency, large amount of data which is required for analysis, calculation and for conclusion by taking the next actionable step. In both of the use cases, it is of paramount importance to semantically describe capabilities of Wind Turbine Controllers, underlying field devices, and their data so that edge analytics can be correctly applied.

1. Structured data: Taking local action on sensing and analysing structured data to determine the inclination of a steel tower.

To determine the inclination of a steel tower of a turbine, pre-requisite is to have a full cable-untwist procedure that has to be activated. This happens, depending on wind conditions, 3-4 times a month. It is also possible to manually instruct the wind turbine to perform the unwind procedure. This is the first step towards sending tower inclination details to the remote service center. At the time of the unwinding-procedure a high-frequency set of data is recorded. A relatively large amount of data is required to calculate the inclination. This dataset needs to be sent back to the remote control center to model and calculate the inclination. As the remote control center monitors thousands of wind turbines across the world, the sheer amount of data and central computing poses great difficulties in remote predictive maintenance. For remote control center, all the standard predictive maintenance procedures should be able to solve locally, thereby sending only the necessary data to the cloud/backend for analysis. In SEMIoTICS, localized edge analytics will be applied which will result in semiautonomous IIoT behavior as only the container comprising the algorithm and result of the inclination calculation is transferred between the wind turbine and the remote control center. The unnecessary data traffic between each turbine and remote control center is greatly reduced. Without the localized analytics functionality, all the high-frequency acceleration and nacelle position data should have transferred to remote control center resulting in suboptimal operation.

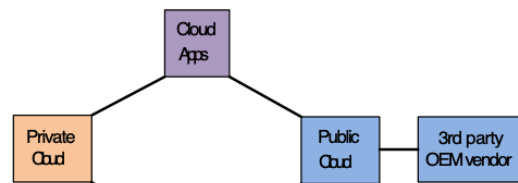
2. Unstructured data:

Within the turbine, there are many events which can be captured by IoT sensors such as Grease leakage detection during normal operation or unintended noise detection when the turbine rotor is changing the direction in the line of wind to maximize energy production. The sensing of this unstructured data and acting locally to prevent any damage to the parts of the turbine in the long run will be of key importance. In SEMIoTICS, this sensing of unstructured data by the IIoT gateway through sensors will be of importance and reacting locally via smart actuation to the captured data, which otherwise potentially can harm the rotating parts of the turbine if no immediate action is taken. This way localized analytics, as proposed in SEMIoTICS, will protect the critical infrastructure of renewable energy resources.

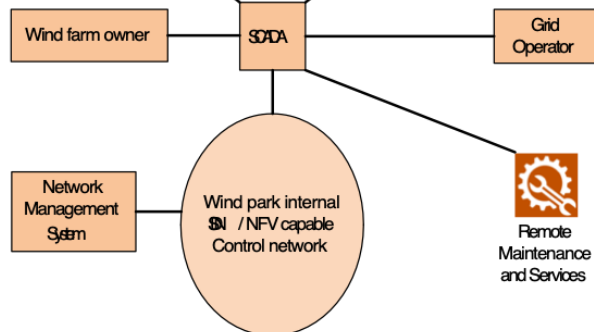
2.1.3. TECHNICAL DETAILS

2.1.3.1. DIAGRAMS OF USE CASE

Backend/ Cloud



Connectivity Network



Field devices Network

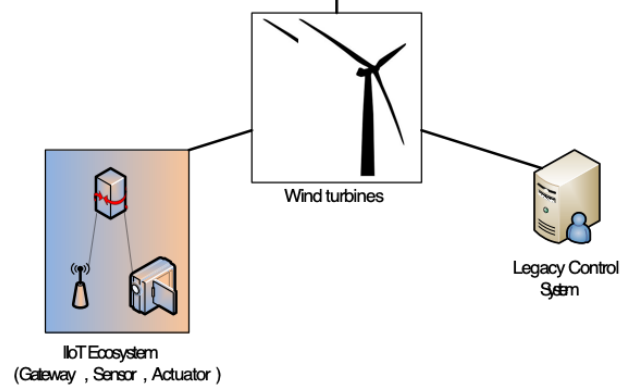


FIGURE 2: STAKEHOLDERS AT EACH LAYER IN WIND ENERGY use case

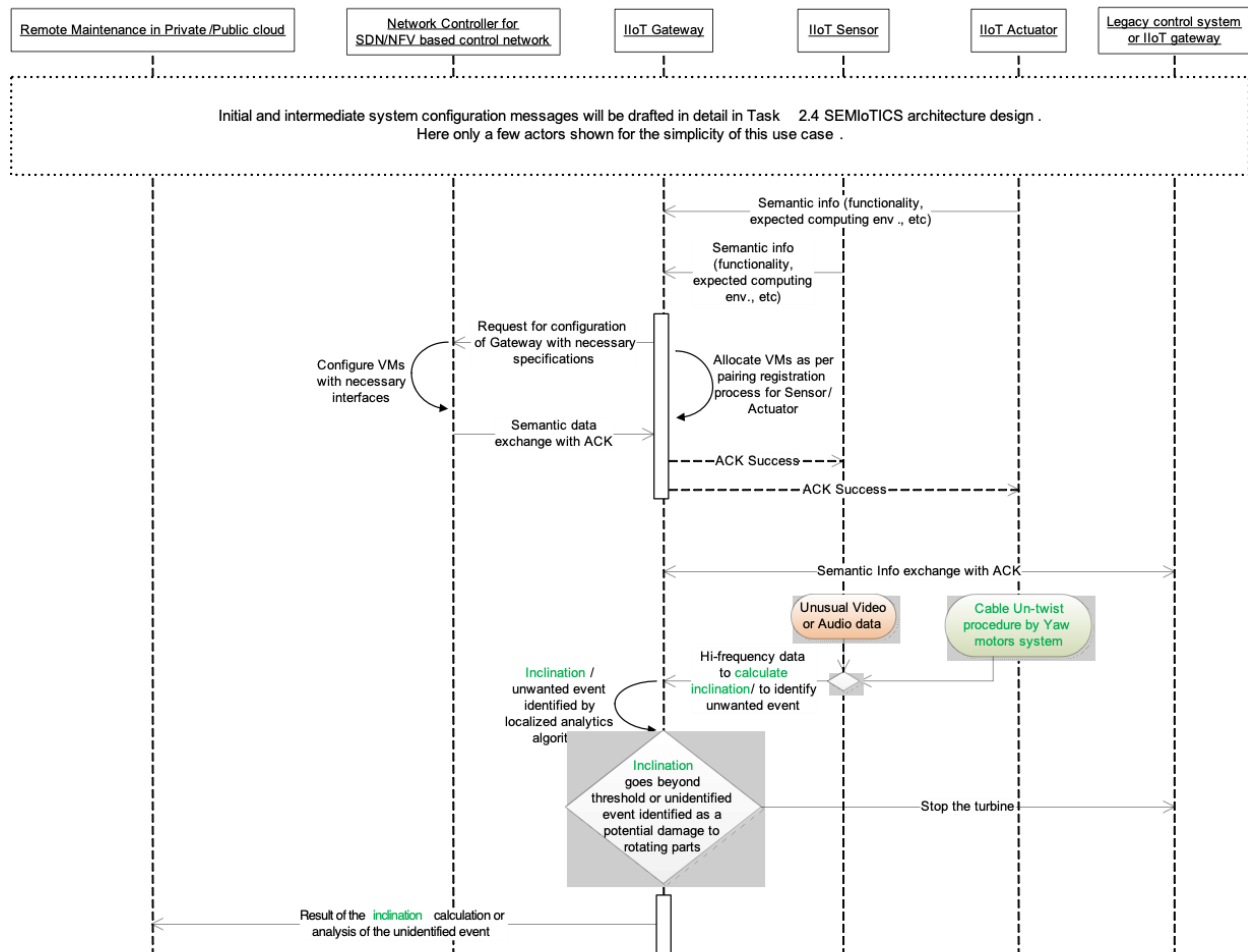


FIGURE 3: ACTIVITY SEQUENCE DIAGRAM

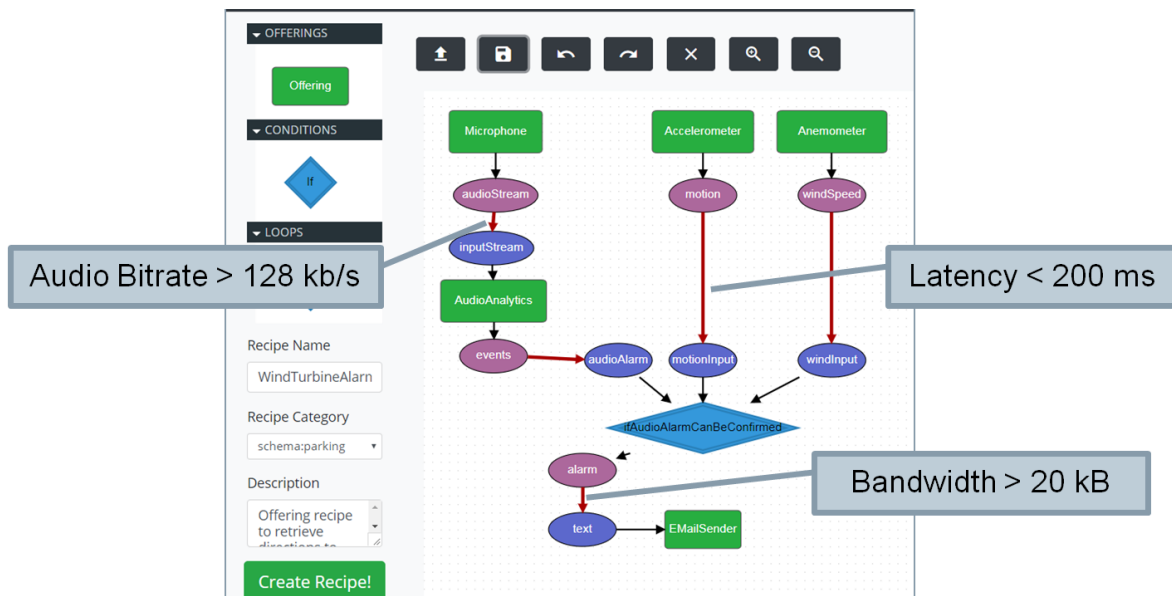


FIGURE 4: APPLICATION CREATION WITH NETWORKING QOS CRITERIA

2.1.3.2. ACTORS

Actors at the Field devices Network layer:

- 1) Wind turbines and associated control system: Current state of the art of Wind Turbine Controller in a Wind Park control network is typically an embedded or highly integrated operating system, which follows rigorously development and pre-qualification prior to deployment in the real world.
- 2) IIoT Gateway: IIoT gateway forms a central part in the IIoT ecosystem. It facilitates semantic bootstrapping of IIoT sensors and actuators. It is configured with different Virtual Machines (VMs) with requested computing environment by the Network controller.
- 3) IIoT Sensor: IIoT sensors collect continuous streams of data from the environment or the monitored system to be processed by IIoT gateway/Actuator or legacy control system.
- 4) IIoT Actuator: IIoT Actuator is acting on the instructions from pre-programmed IIoT gateway analytics algorithm or relayed sensor information from gateway to actuator.

Actors at the Connectivity Network layer:

- 1) SDN/NFV capable wind park control network: It is assumed in this use case that future industrial networks will be based on 5G SDN/NFV technologies. This follows the paradigm of programmable networks from 5GPPP initiative from EU H2020 programme.
- 2) Network management system: Initial configurations and policy in the 5G networks will be done via the network management system.
- 3) SCADA: The SCADA system is usually provided by the turbine supplier. SCADA connects the individual turbines, the sub-station and meteorological stations to a central computer. This computer and the associated communication system allow the wind park operator to supervise the behaviour of all the wind turbines and also the wind park as a whole. It will keep a record of all the activity on regular interval basis (e.g. 5-10 min.) and allows the operator to determine what corrective action, if any, needs to be taken. It also records energy output, availability and error signals, which will act as a basis for any warranty calculations and claims. The SCADA system also has to implement any requirements in the connection agreement to control reactive power production, to contribute to network voltage or frequency control, or to

limit power output in response to instructions from the Grid operator. It is also advisable in addition, if size of the wind park can justify the investment, to erect some permanent meteorological instrumentation on met masts. This equipment allows the performance of the wind farm to be carefully monitored and understood. If the wind farm is not performing according to its budget it will be important to determine whether this is due to poor mechanical performance or less-than-expected wind resource. In the absence of good quality wind data on the site it will not be possible to make this determination. Therefore large wind farms usually contain one or more permanent meteorological masts, which are installed at the same time as the wind farm.

- 4) Wind farm owner: The assets of wind farm are owned by wind farm owner or investor. This entity typically gives contract to wind park operator to operate Wind Park for a specific duration of time.
- 5) Grid operator: The wind park is typically connected to a neighboring grid of the grid operator. This operator takes care of the stability of the grid based on demand-response. When a certain amount of power is not needed in the grid, then the grid operator sends curtailment instructions to the SCADA network of the wind park in order to reduce the power generation.

Actors at the Backend/Cloud layer:

- 1) Private cloud: It is the server and applications infrastructure as well as associated services running on it owned by wind park operator/owner. It is a highly secure and private network infrastructure that is existing on-premises of the wind park owner.
- 2) Public cloud: The public cloud is defined as computing services offered by third-party providers over the public Internet, making them available to anyone who wants to use or purchase them. They may be free or sold on-demand, allowing customers to pay only per usage for the CPU cycles, storage, or bandwidth they consume. (Source: <https://azure.microsoft.com/en-us/overview/what-is-a-public-cloud/>)
- 3) Cloud Apps: Cloud application servers are typically located in a remote data center operated by a wind park operator. Data stored there is instantly available to authorized users for value added services offered by the application servers. Once the semantic federation based on data model and capabilities has been formed between the IIoT Gateway and the legacy control system or another IIoT Gateway, Cloud-Apps of the IIoT ecosystem will be able to use the data exposed by the sensors connects to the turbine.
- 4) 3rd party Original Equipment Manufacturer (OEM): All the OEM equipment that goes into the wind turbine or wind park belongs to this category. Typically these OEM vendors, request for access to their assets in all the wind turbines of a wind park during maintenance timeslot.

2.1.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The trigger for the first sub-use case is a full cable-untwist procedure that has been activated either manually (local/remote) or depending on wind conditions, 3-4 times a month. At the time of untwisting procedure a high-frequency set of sensor data is recorded locally in the IIoT Gateway Virtual Machine (VM). Localized analytics will be performed on the data gathered in the IIoT Gateway to calculate inclination. From deployment point of view, the IIoT devices (Sensors/Actuators) should expose their semantic information to the IIoT Gateway, which in turn

allocates necessary computing environment required for the minimum operation of the IIoT devices. A simple, yet robust registration process is required to pair an IIoT sensor with an IIoT Gateway. In this specific context, the IIoT Gateway will be used to extract features from the data on demand. The feature extraction must be based on a very flexible edge computing architecture including an application pipeline for initial filtering of data and buffer capabilities. Furthermore a federation, based on a semantic data model and capabilities, must be formed between the IIoT Gateway and the legacy control system or another IIoT Gateway, to enable the use of data exposed by the sensors connected to the legacy control system. This will allow a Cloud-App of the IIoT eco-system or the embedded localized compute capabilities of the IIoT Gateway to make use of data exposed by an existing sensor infrastructure, in a semi-autonomous and local setup for IIoTs. The legacy or existing IIoT gateway may be restricted or deliberately restricted to only perform the operations necessary to enable a safe operation of the wind turbine; hence it poses difficulties to upgrade the existing control system with new sensors, logic or process-intensive jobs, which in return may jeopardize the integrity of the wind turbine control system.

The success criteria in case of calculating inclination (structured data) is that if the calculated inclination in the IIoT Gateway goes beyond a given threshold value then the IIoT Gateway should send stop command to the legacy turbine controller through the semantic interface.

The trigger for the second sub-use case is that an unusual event has been identified in the turbine in the form of unstructured data (video or audio).

The success criteria in case of identifying unwanted events in the turbine (unstructured data – video/audio) is to evaluate if it poses danger to the rotating parts of the turbine. If the evaluation turns out that it is better to stop the rotating parts, then the IIoT Gateway should send the turbine a stop command to the legacy turbine controller through the semantic interface.

2.1.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- 1) IIoT Gateway ↔ IIoT Sensor ↔ IIoT Actuator: A simple, but robust registration process is required to pair an IIoT sensor with an IIoT Gateway. In this specific context, the task of IIoT Gateway will be to extract features from the data on demand.
- 2) IIoT Gateway ↔ Network Controller: IIoT gateway sends the request to configure itself as per combined requests from connected sensors and actuators. In return, network controller configures required VMs in IIoT gateway with necessary computing environment.
- 3) IIoT Gateway ↔ Remote Control Center: After the algorithm in IIoT gateway concludes that a certain action has to be taken, it sends the result of this action taken to the remote control center without sending high-frequency and large amount of data.
- 4) IIoT Gateway ↔ Legacy turbine control system: After the algorithm in IIoT gateway concludes that a certain action has to be taken by legacy turbine control system, it sends appropriate command (e.g. turbine stop) to take corrective action in order to avoid damage to the rotating parts.

2.1.3.5. REQUIREMENTS DESCRIPTION

BWC (Wind park operator) verified UC1 detailed description which was defined through an interview process with subject matter experts within the wind industry and the following requirements, which will be considered for the lab trials where it is not planned to involve further end-users (e.g. other

wind park operators). Furthermore, BWC as a consortium partner will support with the implementation and verification of UC1. Planned meetings with Advisory Committee with subject matter experts to verify those requirements and business aspects over the state of the art.

Req.-ID	Description
R1.1	Automatic establishment of networking setup to establish end-to-end connectivity between different stakeholders
R1.2	Automatic establishment of computing environment in IIoT Gateway necessary for the minimum operation of the IIoT devices through 5G network controller based on SDN/NFV
R1.3	Enabling the definition of network QoS on application-level and automated translation into SDN controller configurations.
R1.4	Network resource isolation for guaranteed Service properties – i.e. reliability, delay and bandwidth constraints.
R1.5	Fail-over and highly available network management in the face of either controller or data-plane failures.
R1.6	Decisions made by unreliable, i.e. faulty or malicious SDN controllers, SHALL be identified and excluded.
R1.7	Scalable operation of the SDN control to cater for a massive IoT device integration and large-scale request handling in the SDN controller(s) using a (near-) optimal IoT client – SDN controller assignment procedure.
R1.8	Semantic and robust bootstrapping/registration of IIoT sensors and actuators with IIoT gateway
R1.9	Semantic interaction between IIoT Gateway and legacy turbine control system
R1.10	Local analytical capability of IIoT Gateway to run machine learning algorithms (e.g. specific to 2 specific sub-use cases)
R1.11	Support of device composition and application creation through template approach.
R1.12	Standardized semantic models for semantic-based engineering and IIoT applications.

2.2. Use Case 2: Socially Assistive Robotic Solution for Ambient assisted living (SARA)

The Use Case 2 (UC2) described in the following section provides a benchmark for the assessment of the main key features of SEMIoTICS:

- **SPDI Patterns** : Developers of SARA need a means to accurately assess the impact of varying system configurations on the security, privacy, dependability and safety properties of the system: to verifiably ascertain, for example, if using a particular device for a particular purpose, or a particular configuration of devices, exposes the system to an unacceptable level of risk.
- **Semantic interoperability**: SARA must deal with a wide range of device semantics (different kinds of devices with different functions), data formats (syntactic representations), measurement unit conventions (for sensor readings), and communications protocols (e.g. Wi-Fi, ZigBee, Bluetooth). Various key aspects of SARA functionality, moreover, require that data from multiple sources is collated, aggregated and/or analyzed in a coherent collective fashion. The reliable detection of 'fall incidents', for example, may involve the continuous comparative evaluation of data from wearable IMU devices, RR handle-mounted pressure sensors and RA video cameras (among others)
- **Embedded intelligence and local analytics**: SARA will minimally require the continuous operation of a range of perceptual systems. This means multiple AI/ML processes working concurrently to extract different features from the sensory input in a timely fashion - i.e. without any significant network latency. We can thus identify a strong need for an IoT infrastructure that supports computationally intensive, distributed AI processing at the edge of the IoT network.
- **Network Management**: SARA has behaviors driven to a large extent by uncertain and unpredictable events (e.g. Fall event) in the environment – with a corresponding uncertainty and unpredictability in the generated computational load. Such variability prohibits the advance determination of optimal configurations of networking resources. Indeed, dealing with such dynamic run-time variability entails adopting correspondingly dynamic and flexible mechanisms for network management.

UC2 (SARA) uses human data only for testing purposes. Testing is done by research volunteers from the ENG R&D Lab. Based on the accepted maturity of the technology level (TRL 4-6) of SEMIoTICS in Lab Trials, UC2 owner (ENG) as an option involves of 2 or 3 (max) real patients under the supervision of medical Doctors. Data are maintained only for the time required for testing purposes. Data are store on servers managed by ENG and are not shared either within the consortium or outside the consortium.

2.2.1. SCOPE AND OBJECTIVES OF USE CASE

This use case employs the SEMIoTICS technologies to develop an Information and Communication Technology (ICT) solution aimed at sustained independence and preserved quality of life for elders with Mild Cognitive Impairment or mild Alzheimer's disease, with the overall goal of delaying institutionalization: supporting both 'aging in place' (individuals remain in the home of choice as long as possible) and 'community care' (long-term care for people who are mentally ill, elderly, or disabled provided within the community rather than in hospitals or institutions).

The solution, named **SARA** (**S**ocially **A**ssistive **R**obotic Solution for Mild Cognitive Impairment or mild **A**lzheimer's disease), will be developed as an expansion of Engineering's existing **AREAS®** Assisted Mobility Module, which is a component of Engineering's commercial AREAS® suite, an Enterprise Resource Planning (ERP) solution for the Health sector.

The current AMM solution focuses primarily on the risks of falls associated with physical decline and decreased mobility. Other pertinent risks – e.g. relating to cognitive decline and/or mismanagement of health monitoring and medication – that could lead to premature hospitalization, are not covered by the current solution, but will be addressed (to some degree) in SARA.

The general approach in SARA is, following the approach of Socially Assistive Robotics (SAR), to introduce an additional robotic component within the AMM. SARA will establish and maintain close (non-contact) interaction with the human user/patient, thus providing cognitive and physical assistance where possible, in order to sustain the individual's ability to autonomously perform tasks of daily living, and to achieve measurable progress in convalescence, rehabilitation and learning.

Examples of assistive tasks envisaged for the Robotic Assistant (RA) within SARA include:

- creating a bi-directional video stream between the end user and a doctor when it is detected that someone may be in distress;
- proposing physical activities such as going for a walk;
- proposing cognitive activities such as playing a game;
- reminding patients to take prescribed medication at allotted times;
- monitoring users' response times to spoken questions;
- proposing calling some relative or friend;
- escorting/accompanying users through their environment.

Several of the assistive tasks delegated to the RA entail the execution of computational tasks that can be particularly demanding in terms of computational intensity and consumption of networking resources. Examples include Simultaneous Localization And Mapping (SLAM) tasks involved in navigation (e.g. escorting tasks), bi-directional streaming of real-time audio-visual (and other) data, or the complex analysis of human activities and behaviours (e.g. facial expressions, verbal communication, gestures, gait and posture classification) undertaken as part of patient monitoring (e.g. during physical exercises aimed at the prevention of physical decline).

Similar computationally intensive tasks are performed by other AMM components – e.g. the Robotic Rollator (i.e. a walking frame using robotic technologies such as motion control technologies, sensing technologies, vision technologies, computational intelligence and so on) also performs SLAM computations for autonomous navigation. At the same time, the inter-connectivity of AMM components can also help reduce (e.g. by providing additional, disambiguating information) and/or distribute the computational load. In the context of health, the management of such distributed computational tasks and information sources – concerned largely with processing sensitive medical data - is subject (by law) to strict privacy and security requirements. The SARA use case thus provides broad scope for applying the SEMIoTICS technologies (e.g. semantic interoperability mechanisms, SPDI patterns, embedded intelligence).

The elicitation of the SARA requirements is done by the Engineering's Business Unit taking care of the evolution of the AREAS® E-Health Integrated Platform⁴. This elicitation process relies on the network of customers and partnership that Engineering has in the Healthcare market. The requirements for the SARA component are being collected through stakeholder interviews and focus groups.

The following sections present the requirements and specifications of SARA that are more relevant for the SEMIoTICS project. The contextualization of the SARA requirements in the SEMIoTICS project is being done by the Engineering's R&D team involved in the project.

2.2.2. NARRATIVE OF USE CASE

⁴ <https://www.eng.it/en/our-platforms-solutions/areas>

2.2.2.1. SHORT DESCRIPTION

Piero is 70 years old and lives in a 'smart' apartment in Rome, with his daughter who looks after his daily needs. Piero suffers mild Alzheimer's disease and has problems walking due to a recent hip replacement. As such, his General Practitioner (GP) registered him to receive an instance of the SARA solution.

The SARA solution supports both Piero and his daughter by performing various assistive tasks e.g. reminding to perform his prescribed cognitive rehabilitation exercises, take medicines, finding medicine boxes, providing physical support during walk, monitoring health.

One afternoon, while his daughter is out at the supermarket, SARA reminds Piero to perform one of his prescribed cognitive rehabilitation exercises and to take his medicine. Since Piero cannot find his medicines SARA drives Piero to the medicine box which is in the kitchen.

On the way SARA detects a change in his breathing, heart rate, and a change in his gait. All of which is sufficient to raise an alert towards Piero's daughter.

Thanks to SARA Piero's daughter can request a medical check for Piero while she is still at the supermarket.

A short while later Piero's daughter arrives home and, having received notification that a medic is available and waiting, initiates a new telepresence session with the medic - who has already accessed Piero's medical records and recent biometric data collected by SARA.

The medic relies on SARA to remotely guide Piero and his daughter through a few physical checks (e.g. taking his blood pressure). SARA allows Piero's daughter to easily hook up a heart & breathing monitor as requested by the medic. Hence the medic can remotely monitor the readings as Piero performs some simple physical exercises.

The medic confirming that all seems ok.

2.2.2.2. COMPLETE DESCRIPTION

[paragraph numbers are for later reference]

1	<p>Piero is 70 years old and lives in a 'smart' apartment in Rome, with his daughter who looks after his daily needs. Piero suffers mild Alzheimer's disease and has problems walking due to a recent hip replacement. As such, his General Practitioner (GP) registered him to receive the AREAS Mobility Service (AMS), and as part of this service he was duly given:</p> <ul style="list-style-type: none"> • A Body Area Network (BAN): comprising a wearable Inertial Measurement Unit (IMU) and mobile smartphone running a dedicated BAN app • A Robotic Rollator (RR): a semi-autonomous motorised wheeled walking frame for physical support in moving around; • A Robotic Assistant (RA): in this case Softbank's Pepper, a mid-sized humanoid robot.
2	<p>His 'smart' apartment is fitted various home automation devices - including a smart armchair and smart medicine box - all integrated into the SARA system.</p>
3	<p>One afternoon, while his daughter is out at the supermarket, Piero sits dozing in front of the television - which he has been watching for some time. The SARA system notices this and (following 'daily activity level' rules configured by Piero's GP) decides that Piero should instead perform one of his prescribed cognitive rehabilitation exercises.</p>
4	<p>Accordingly, the RA approaches Piero, calls his name a few times to get his attention, and suggests the exercise.</p>

5	Piero agrees, and they begin. For this particular exercise, the RA displays a sequence of pictures and symbols (on its chest-mounted tablet) which Piero has to correctly (verbally) identify.
6	With the exercise done, the RA reminds Piero that it is time for him to take his medicine.
7	Piero looks around but can't find his tablets. He asks the RA where they are, and the RA informs him that his 'smart' medicine box is in the kitchen.
8	Piero asks the RA to call his Rollator, which wakes, and autonomously steers itself (from its rest position in the corner of the room) to Piero. Using the RR for support, Piero stands, and walks to the kitchen - all the while being escorted and monitored by the RA.
9	On the way, however, Piero feels faint and staggers. The BAN detects a change in his breathing and heart rate, the RR notices a change in his gait and increased pressure on the handles, while the RA sees his expression change to a grimace. All of which is sufficient to raise an alert.
10	<p>The SARA system duly notifies Piero's daughter (as his assigned caregiver) of the situation. His daughter receives the alert, and initiates a telepresence session from her phone to the RA (to avoid having her father having to locate & answer his phone):</p> <ul style="list-style-type: none"> • The RA's chest tablet displays his daughter's face, and relays her voice, to Piero; • The RA's audio-visual input is relayed directly to his daughter's phone.
11	His daughter quickly ascertains that Piero is not in any immediate trouble, but she tells him that she'll be home soon, and just to make sure everything is ok, also sends (through SARA) to the GP a request for a medical check.
12	After the call, and feeling better, Piero continues to the kitchen. The smart apartment automatically turns off the lights in the lounge, on turns on those in the kitchen. The RA visually detects, and informs Piero, that his smart medicine box is over by the sink.
13	Piero sits down at the kitchen table and takes his pills, while the RR autonomously parks itself in the corner of the room out of the way.
14	A short while later his daughter arrives home and, having received notification that a medic is available and waiting, initiates a new telepresence session with the medic - who has already accessed Piero's medical records and recent BAN data.
15	After walking Piero and his daughter through a few physical checks (e.g. taking his blood pressure), the medic then asks Piero's daughter to hook up a heart & breathing monitor to the BAN and then remotely monitors the readings as Piero performs some simple RA guided physical exercises. In this case, the RA performs a series of pre-orchestrated movements, which Piero has to copy as best he can (with the RA giving performance feedback). The exam is soon over, with the medic confirming that all seems ok.

2.2.3. TECHNICAL DETAILS

2.2.3.1. DIAGRAMS OF USE CASE

This section contains the description of the use cases for the SARA scenario. Each use case is named and numbered for later reference, and contains:

- UML use case diagram
- Main success scenario

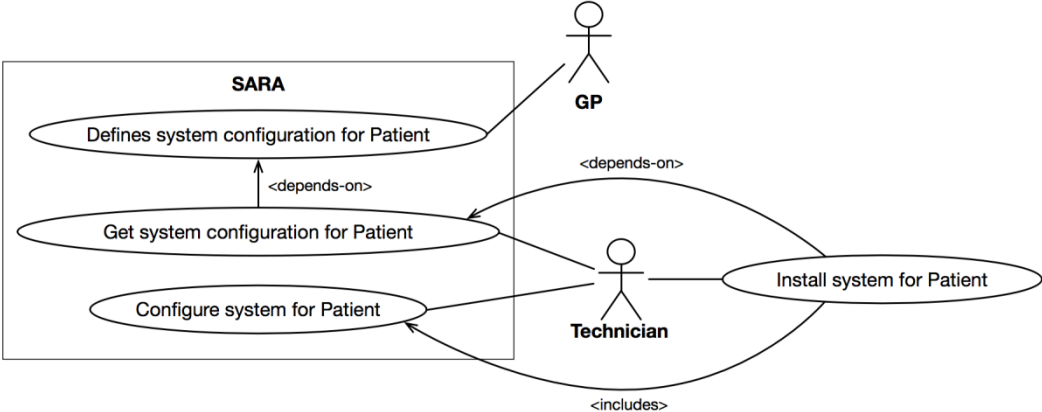
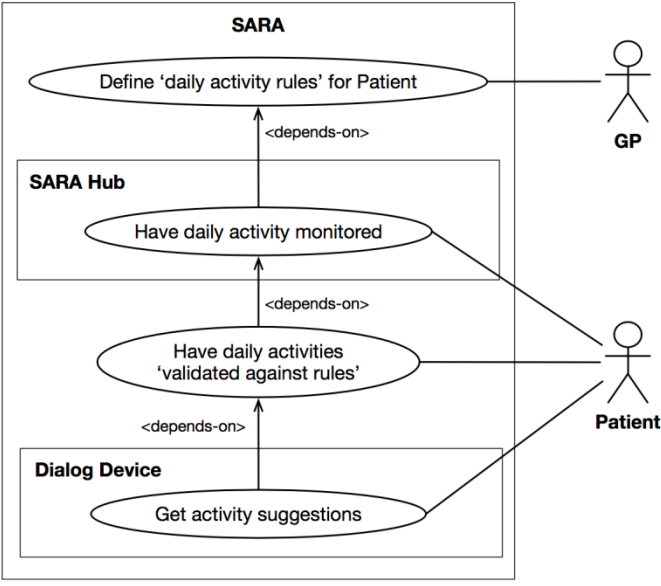
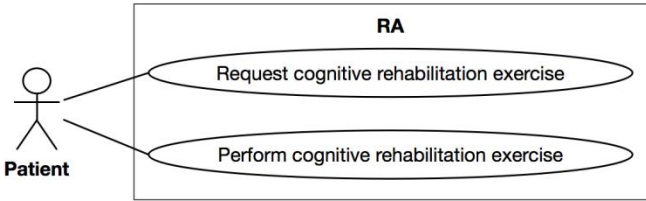
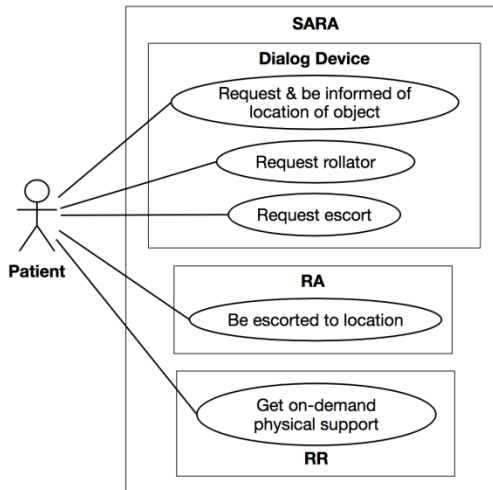
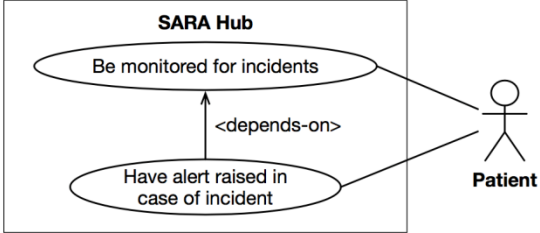
Name	UC2-1: System Configuration (narrative 1 & 2)
Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • GP defines and registers the system configuration for a given Patient - e.g. which specific SARA components to supply, and their (general) configuration parameters. • Technician retrieves the system configuration specified by the GP for a Patient and installs/configures the appropriate components – at which point the system is ready to be used by the Patient.
Name	UC 2-2: Activity Monitoring (narrative 3, 4 & 6)

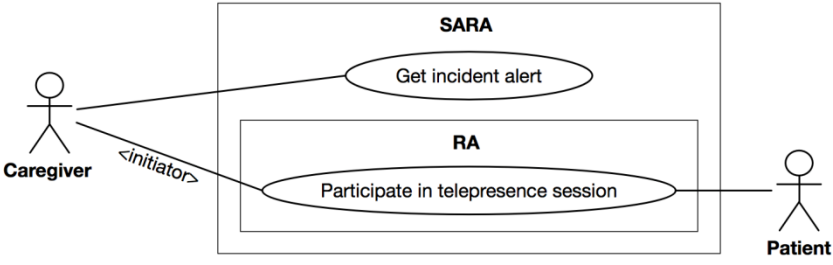
Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • GP defines & registers the 'daily activity rules' for a given Patient • System (via the SARA Hubs) continuously monitors the Patient's daily activities, which include key activities of daily living (e.g. walking, sleeping, eating, using the bathroom, ...), as well as specific treatment-related activities such as taking prescribed medication and performing cognitive/physical exercises. • System validates the Patient's monitored activities against the pre-scribed rules. • System (via BAN and/or RA), based on the validation (previous point), recommends activities to the Patient – including (among others): <ul style="list-style-type: none"> • reminders to take medicine • reminders to phone relatives (to maintain social contact) • suggestions to perform cognitive/physical exercises

Name	UC 2-3: Cognitive Rehabilitation (narrative 5)
Diagram	

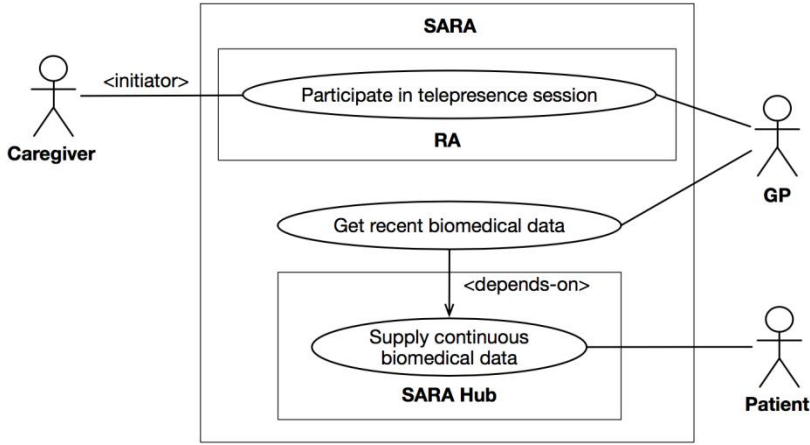
Main Success Scenario	<ul style="list-style-type: none"> • Patient 'requests' to start a cognitive rehabilitation exercise. <ul style="list-style-type: none"> • Note that in the narrative (5), the 'request' corresponds to Piero's acceptance (the phrase "Piero agrees") of the System's suggestion to perform the exercise. Hence 'requests' may be more or less explicit and be conveyed in diverse ways. • Patient, with the assistance of (and under the supervision of) the RA, performs the cognitive rehabilitation exercise. <ul style="list-style-type: none"> • Various exercises are available, involving different modes of Patient-RA interaction. In the narrative, the interaction is primarily through the RA's chest-mounted tablet.
------------------------------	---

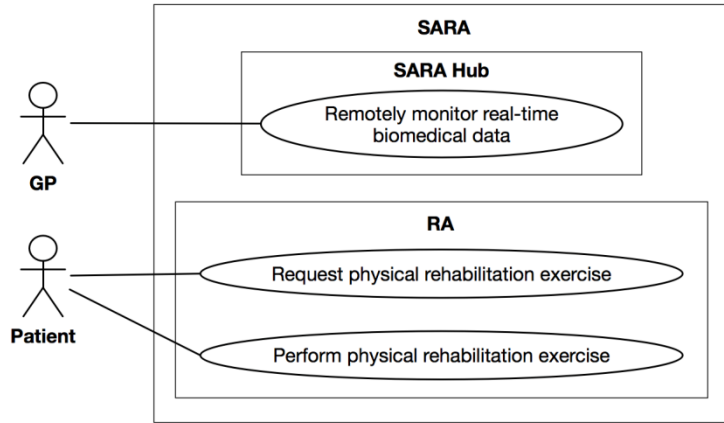
Name	UC 2-4: Patient Escort (narrative 7, 8, 12 & 13)
Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • Patient requests the location of an object. • System (via BAN/RA) informs the Patient of the location of the object. • Patient requests the use of the RR (for physical support). • RR (wakes if sleeping), steers itself to the Patient's location, and orients itself relative to the Patient ready for use. • Patient uses the RR for physical support while walking. • Patient requests to be escorted by the RA to a specified location. <ul style="list-style-type: none"> • Note that requests may be implicit – e.g. in the narrative: escorting is assumed to be the default behaviour of the RA. • RA escorts the Patient (with the RR) to a specified location. <ul style="list-style-type: none"> • The RA is completely self-guiding and must orchestrate its movements so as not to impede the Patient & RR. • The RR, while in use, is partially self-guiding: it steers itself towards the target location but overrides this behaviour if it detects that the Patient is guiding it somewhere else.

Name	UC 2-5: Incident Monitoring (narrative 9)
Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • System (via the Hubs, but also including high-level analysis of system-wide monitored events) continuously monitors Patient health indicators (e.g. vital signs, gait/posture, verbal utterances, ...) for anomalies and signs of distress. • System, in case of anomaly/stress (previous point) raises an alert.

Name	UC 2-6: Manage a Medical Alert (narrative 10)
Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • System sends an alert notification to the Patient's assigned Caregiver. • Caregiver receives the alert notification. • Caregiver (via the System & RA) initiates and engages in a telepresence session with the Patient. <ul style="list-style-type: none"> • 'telepresence' requires the real-time bi-directional transmission of high-quality audio-visual data.

Name	UC 2-7: Request Medical Assistance (narrative 11 & 14)
-------------	---

Diagram	
Main Success Scenario	<ul style="list-style-type: none"> • GP (via the System & RA) initiates and engages in a telepresence session with the Caregiver/Patient (see also UC 6 above). • GP accesses the ACS to retrieve the most recent logs of monitored bio-medical for the Patient. • System (via Hubs) continuously monitors and periodically logs bio-medical data for the Patient.

Name	UC 2-8: Remote Medical Check (narrative 15)
Diagram	

Main Success Scenario	<ul style="list-style-type: none"> GP remotely accesses the Patient's BAN (and/or RA & RR) to receive real-time streaming of monitored bio-medical data from the Patient. Patient 'requests' to start a physical rehabilitation exercise. <ul style="list-style-type: none"> In the narrative (15), the 'request' initially comes from the GP, with (we assume) implicit agreement from the Patient. Patient, with the assistance of (and under the supervision of) the Caregiver & RA, performs a physical rehabilitation exercise. <ul style="list-style-type: none"> Various exercises are available, involving different modes of Patient-RA interaction. In the narrative, the interaction is imitation, with the Patient copying the movements of the RA, with verbal feedback from the RA.
------------------------------	--

2.2.3.2. ACTORS

Stakeholder	Description
Patient (Care Receiver)	<p>Elders with Mild cognitive impairment (MCI)</p> <p>Person age > 60 or 65 with a slight but noticeable and measurable decline in cognitive abilities, including memory and thinking skills. A person with MCI is at an increased risk of developing Alzheimer's or another dementia. However, MCI does not always lead to dementia. In some individuals, MCI reverts to normal cognition or remains stable.</p> <p>Alzheimer Disease Patient</p> <p>Alzheimer's is a progressive disease that worsens over time, causing problems with memory, thinking and behaviour. It can be divided into three stages:</p> <ul style="list-style-type: none"> Mild: in this phase, the most noticeable deficit is memory loss, which mainly affects short-term memory (inability to remember recently learned facts and acquire new information). Moderate: in this stage the patient becomes unable to perform most common activities of daily living. Severe: during the final stage of AD, the person is completely dependent upon caregivers. <p>Objectives:</p> <p>To maintain their confidence & social activity and to reduce the risk of further impairment when performing daily activities.</p> <p>To maintain, or recover (following treatment), their ability to lead a normal life and to autonomously carry out the tasks of daily living.</p>
General Practitioner (GP)	<p>A medical professional.</p> <p>Objectives:</p> <p>To ensure Patients receive an AMS best suited to their idiosyncratic needs (satisfying the individual Patient's objectives). The GP is responsible for (continuously) assessing the Patient's needs, configuring an appropriate AMS and monitoring Patient progress.</p>

Caregiver	<p>Family (Informal) Caregiver Any relative, partner, friend or neighbour who has a significant personal relationship with, and provides a broad range of assistance for, an older person or an adult with a chronic or disabling condition. These individuals may be primary or secondary caregivers and live with, or separately from, the person receiving care.</p> <p>Objectives: To provide care to the Patient, while reducing their own risks of "caregiver burden" - i.e. psychological issues and/or physical health problems due to the demanding nature of, and emotional engagement involved in the task.</p> <p>Formal Caregiver A provider associated with a formal health service system, whether a paid worker or a volunteer.</p> <p>Objective: To provide care to the Patient, while reducing their own travel time (i.e. time spent at work, but not providing care), especially for interventions that are of short duration (e.g. monitoring of injuries, verification and follow-up with the family).</p>
Technician	<p>Technical IT systems expert.</p> <p>Objective: To ensure SARA technical components are correctly installed and configured (according to GP prescriptions) and functioning properly throughout the life-time of the service.</p>

SARA Technical Components

Component	Description
AREAS Cloud Services (ACS)	The components of the AREAS® suite made available as a service. This also includes generic services (e.g. Identity & Access Management, Storage) serving all components of the suite.

SARA Hub	<p>Body Area Network (BAN) A network of wearable sensors, identification tags and the like, carried or worn by the patient, and relying on the patient's personal mobile device as a hub for communication with other devices.</p> <p>Robotic Rollator (RR) A 'smart', wheeled walking frame providing physical support for the Patient, and offering Patient monitoring and autonomous navigation capabilities.</p> <p>Robotic Assistant (RA) A human-shaped mobile robot capable of:</p> <ul style="list-style-type: none"> recognizing people, including (to some degree) their behaviours, and the principal human emotions - all based on the interlocutor's verbal utterances (tone & words), facial expression, and body movements, moving around autonomously, naturalistic communication with his interlocutor through his body movements and his voice. <p>Smart Environment (SE) A building (e.g. home, hospital) equipped with devices continuously working to support elderly and disabled individuals in achieving an independent life at home. It includes a home gateway providing internet connectivity and computational power to run an ad hoc collection of services – in particular: providing information about the Patient's local environment (time, location, maps, weather conditions, etc.) in support of Patient monitoring and RR/RA navigation functions. This includes information from local IoT resources (e.g. beacons, tele-cameras), and external third-party sources (e.g. GPS, map-data – see below)</p>
AI Services	<p>Various components of the system, in particular the RR and RA, call on dedicated services for specialized artificial intelligence (AI) functions such as object & people/face recognition, natural language processing, or other computationally intensive tasks.</p>

Figure 5 shows an overview of the main subsystems of the SARA solution along with their key interactions:

- The AREAS Cloud services subsystem consists of the collection of SARA backend functionalities made available as services on the Internet.
- A set of software client interfaces allows General Practitioners, Technicians and Caregiver to access the management functionalities (i.e. patient enrolment, devices monitoring, preferences configuration).
- The SARA functionalities rely on the AREAS Business Framework to access both other services of the AREAS suite (e.g. Patient health record) and management services (e.g. Identity & Access Management, Storage) serving all components of the AREAS suite.

- The AREAS Cloud services rely on an IoT infrastructure to exploit the services offered by the collection of IoT devices part of the SARA solution. The interaction between the IoT infrastructure and the actual IoT devices is mediated by the four SARA hubs: Body Areas Network Gateway, Robotic Rollator, Robotic Assistant, Smart Environment Gateway.
- The SARA hubs rely on dedicated services for specialized artificial intelligence (AI) functions such as object & people/face recognition, natural language processing, or other computationally intensive tasks.
- Both patients and caregivers exploit SARA assistive services through the interaction with the various IoT devices connected to the SARA hubs.

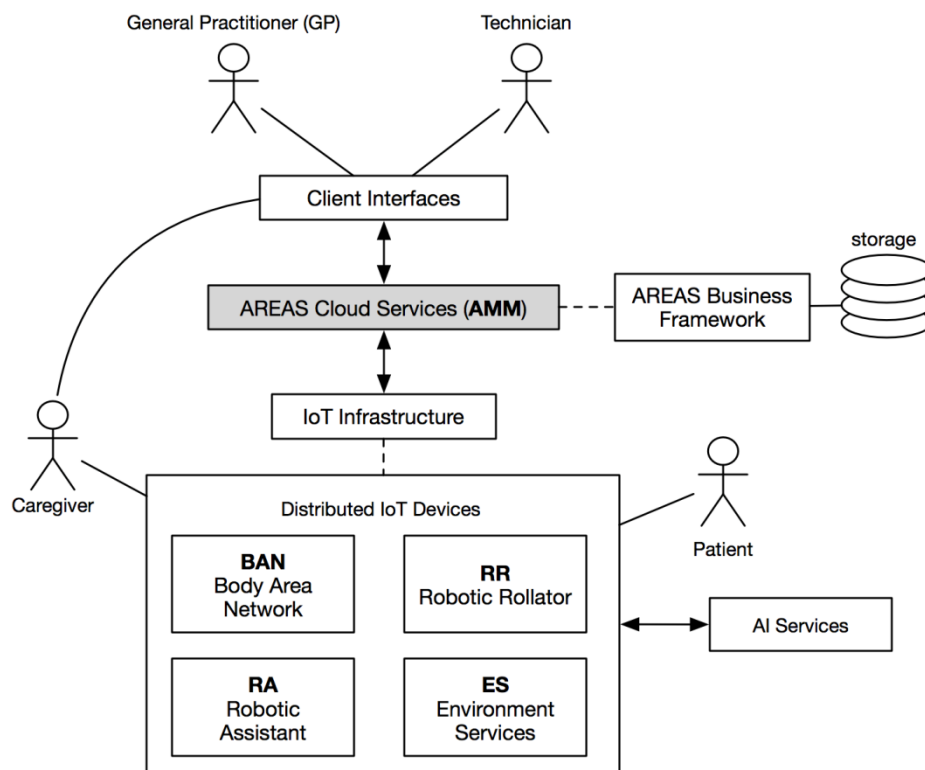


Figure 5: Overview of the SARA main subsystems

SARA System Components

Figure 6 shows an initial conceptualization of the SARA solution:

- The SARA solution results from the aggregation of three different kinds of components: the AREAS Cloud services, AI Services, and IoT Devices.
- The SARA solution includes three kinds of IoT devices: actuators, sensors and hubs.

- The SARA solution assumes the availability of proper connectivity between ARAEAS Cloud services, sensors, actuators and hubs.
- Within the SARA solution there are four kinds of hubs: Body Area Network (BAN), Robotic Assistant (RA), Robotic Rollator (RR), Smart Environment (SE).
- Dialog devices are devices supporting some form of Human-Computer Interaction: BAN and RA hubs are two different classes of Dialog devices.
- Robotic Devices are devices embedding some robotic technology. Mobotic Device are mobile robotic devices. The RA, RR and SE are all Robotic Devices. The RA and RR are also Mobotic Device.

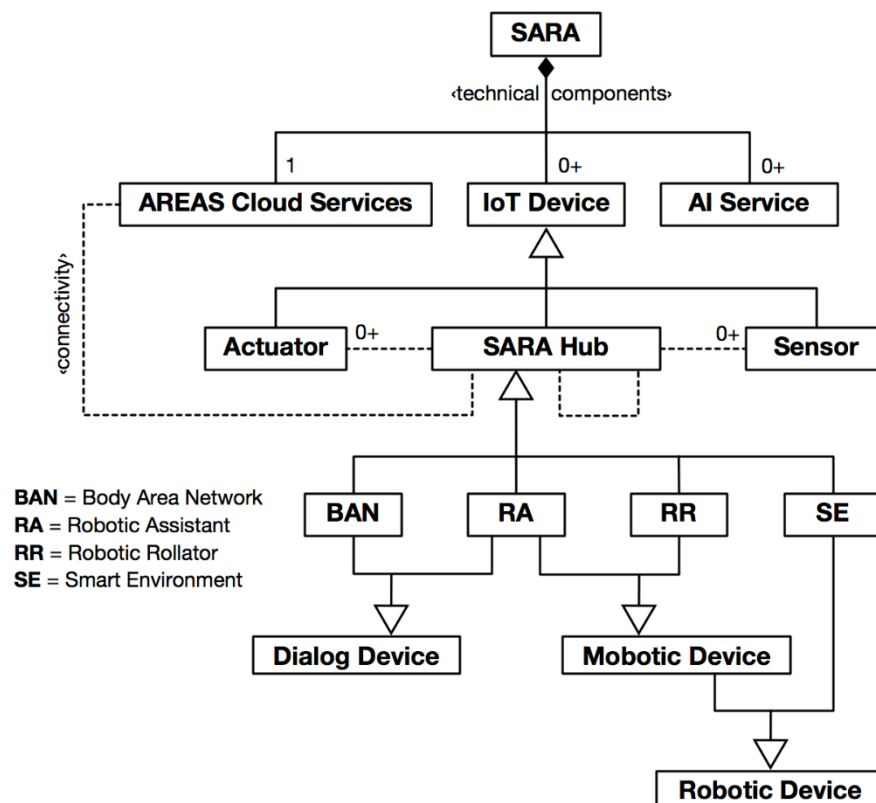


FIGURE 6: SARA SYSTEM COMPONENTS

Communications

The Figure 7 shows the main communication paths and protocols of the SARA solution:

- The devices within the BAN communicate using Bluetooth Low Energy (BLE) technology. The patient's smartphone is the hub of the BAN and is connected to the Internet via cellular connectivity (in the future 5G). The BAN hub can also communicate with the RR using either BLE or Wi-Fi connectivity.
- The devices within the RR communicate using a Controller Area Network bus (CAN-bus).
- Wi-Fi connectivity is provided by the Smart Environment hub and enables the communication between the four hubs (BAN, RR, RA, SE) of the SARA solution.

- The Smart Environment hub act as Internet gateway for the other hubs connected via Wi-Fi.
- The devices within the Smart Environment communicate using ZigBee technology.
- The SARA backend services and the SARA IoT field devices communicate using the Internet protocols (IPv6/IPv4). The Internet also enables the communication between the SARA client applications and SARA backend services.

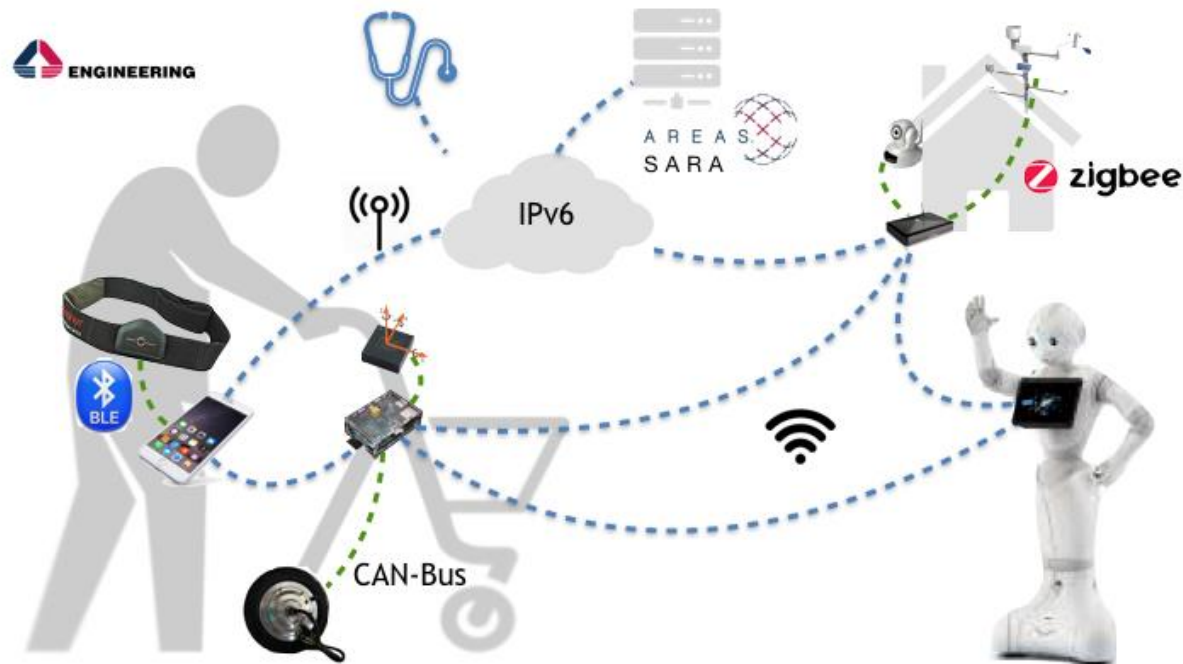


FIGURE 7: COMMUNICATIONS AND PROTOCOLS WITHIN SARA

2.2.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

This section lists generic privacy, security and safety concerns raised by the SARA use case.

- All (identifiable) Patient data is exchanged between system components using data encryption and HL7⁵ protocols.
- All (identifiable) Patient data is securely stored in the backend database (DB).
- For reasons of safety, backend DB should be redundant (if a malfunction occurs, the secondary DB starts working, avoiding data loss)
- All (identifiable) Patient data can be accessed only by authorized and authenticated individuals.
- GP has access to all Patient medical records.
- Technician only has access to technical system configuration information.
- Patient (or close relatives) must provide informed consent to use of the service prior to using the service.
- All monitored Patient data must be reliably tagged with the identity of the Patient - in order to avoid the case that data monitored from one Patient is incorrectly ascribed (at any point in the System's processing of that data) to another Patient.

⁵ <http://www.hl7.org/implement/standards/>

- The system (and all its components) must include failsafe safety measures to ensure that no harm can come to the Patient (or anyone else).
- The autonomous navigation mechanisms should be capable of navigation around the home without causing harm to or damaging individuals or items in the home.
- All audio-video stream transmission during telepresence must be encrypted and secure.
- For reasons of privacy, the RA must notify the Patient whenever a telepresence session begins & ends, also informing them of the identity of the remote operator.
- System components and communications technologies must neither interfere with, nor be negatively affected by, medical equipment.
- System components and communications technologies must be configured on reliable network to avoid service interruptions

2.2.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

The following diagram shows the flow of information between the actors of the SARA use case.

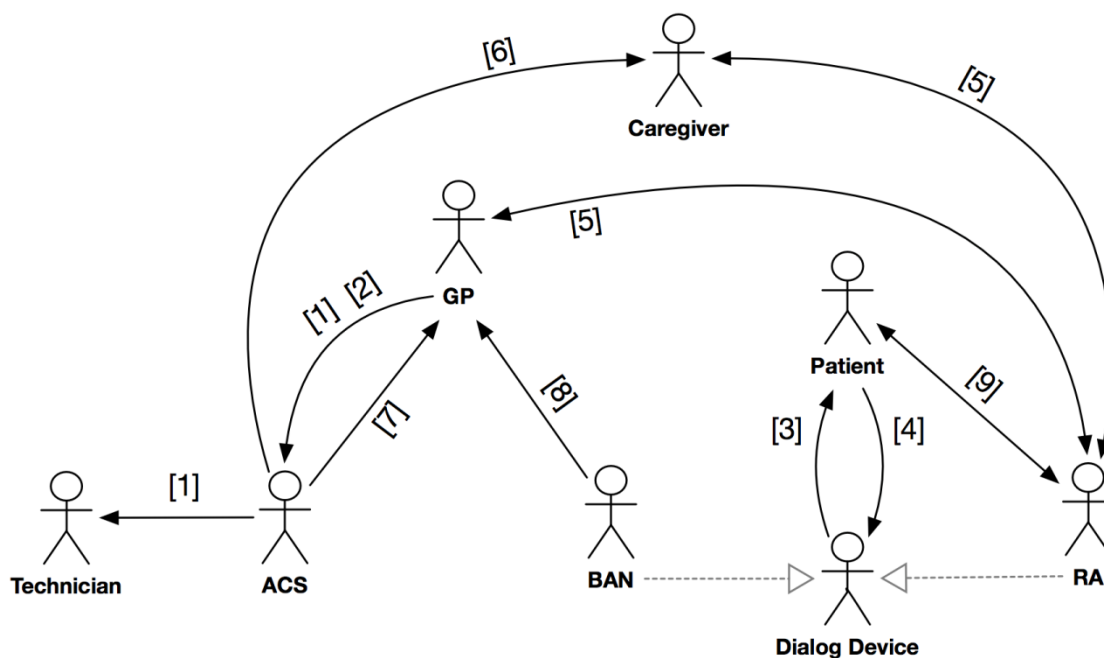


FIGURE 8: FLOW OF INFORMATION BETWEEN THE SARA ACTORS

#	Exchanged Information
1	Technical system configuration for Patient
2	'Daily activity rules' for Patient
3	Activity reminders & suggestions
4	Various requests for exercise, object locations, RR assistance and RA escort

5	Real-time audio-visual stream (for telepresence)
6	Incident alerts
7	Archived medical records & raw biometric data
8	Real-time raw BAN sensor data (various kinds)
9	Miscellaneous verbal messages

2.2.3.5. REQUIREMENTS DESCRIPTION

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", "MAY" are to be interpreted as described in RFC 2119.

Functional Requirements

Req.-ID	Description
R2.1	The SARA system MUST provide a User Interface (UI) for the GP to define system configurations for each Patient.
R2.2	The SARA system MUST provide a User Interface (UI) for the Technician to access system configuration data for Patients.
R2.3	The SARA system MAY provide a User Interface (e.g. on a remote client) to allow the Technician to configure the system for a Patient.
R2.4	The SARA system MUST provide a UI for the GP to define 'daily activity rules' for each Patient.
R2.5	The SARA system MUST (via the SARA Hubs) continuously monitor the Patient's daily activities.
R2.6	The SARA system MUST validate monitored Patient activities against the prescribed 'daily activity rules'.
R2.7	The SARA system MUST provide (via Dialog devices) timely activity suggestions to the Patient – e.g. for scheduled events (taking medicine, programmed exercises) and in case of 'violations' of the 'daily activity rules'.
R2.8	The RA MUST assist the Patient in performing a variety of cognitive rehabilitation exercises.
R2.9	The SARA system MUST recognize (via Dialog devices) and respond appropriately to Patient requests to perform cognitive exercises.

R2.10	The SARA Hubs MUST be able to detect, recognize, identify and determine the positions of significant objects (including people) in the environment.
R2.11	The SARA system MUST keep track of the changing positions of significant objects (including people) in the environment.
R2.12	<p>The SARA system MUST recognize (via Dialog devices) and respond appropriately to Patient requests to:</p> <ul style="list-style-type: none"> • be informed of locations of objects/people. • receive physical support from the RR. • be escorted to a location by the RA.
R2.13	The SARA system MUST (via SARA Hub devices) continuously monitor the Patient to detect, and raise an alert in case of, incidents (i.e. critical medical events such as falls, heart attacks, ...).
R2.14	The SARA system MUST reliably and rapidly notify assigned Caregivers of Patient incidents.
R2.15	The SARA system MUST (in conjunction with the RA) provide on-demand telepresence (bi-directional, real-time, high-quality, audio-video streaming) between the RA's tablet device (Patient side) and either the Caregiver's mobile phone (UC 6) or GP's desktop computer (UC 7).
R2.16	The SARA system MUST (for reasons of privacy) notify the Patient whenever a telepresence session has started/ended, and also inform the Patient of the identity of the remote operator.
R2.17	The SARA system MUST provide a UI for the GP to access historical logs of Patient bio-medical data from the ACS.
R2.18	The SARA system MUST periodically log (Hub monitored) Patient bio-medical data to the ACS.
R2.19	The SARA system MUST recognize (via Dialog devices) and respond appropriately to Patient requests for physical exercises.
R2.20	The SARA system MUST assist the Patient in performing a variety of physical rehabilitation exercises.

R2.21	The SARA system MUST provide (via the SARA Hubs) the GP remote access to real-time (Hub monitored) Patient biomedical data.
R2.22	The SARA system SHOULD support and track the dynamic addition/removal of system components (e.g. health monitors temporary added to the BAN).
R2.23	The SARA system SHOULD facilitate the integration of legacy and new IoT Devices (Hubs, Sensors & Actuators) - e.g. through standardized low-level API conformance.

Non-Functional Requirements

Req.-ID	Description
R2.24	<p>The SARA system should provide robust mechanisms to protect Patient-related data - i.e. to:</p> <ul style="list-style-type: none"> Authenticate the sources of Patient data – in particular: to avoid the case that data monitored from one Patient is incorrectly ascribed to another. Prevent unauthorized access to or manipulation of stored Patient data. Secure the transmission of Patient data. <p>Such data includes (but is not limited to):</p> <ul style="list-style-type: none"> System configuration data for the Patient (UC 1) Daily activity rules for the Patient (UC 2) Monitored daily activity data for the Patient (UC 2) Audio-visual streams transmitted during telepresence (UC 6 & 7)
R2.25	The communication of health-related Patient data between SARA components SHOULD employ/conform to HL7 protocols.
R2.26	The SARA system components and communications technologies MUST NOT interfere with, and MUST NOT be adversely affected by, medical equipment.
R2.27	The SARA system MUST perform with consistent quality (as assured through version control of source code and strict, systematic and high-coverage regression testing) in all target operating environments.
R2.28	The SARA system MUST fully comply with all relevant local government laws governing the privacy, security and storage of sensitive Patient health-related data.

R2.29	The SARA system MUST ensure that all health-critical communications (in particular: incident alerts) between components are prioritized in terms of receiving network resources and ensuring reliable transmission to destination.
R2.30	The SARA system MUST reliably and rapidly respond appropriately to health-critical events (in particular incident alerts).
R2.31	The SARA Mobotic devices SHOULD NOT (as far as this is preventable) cause any harm or damage to individuals and objects while navigating around the home.
R2.32	The SARA Mobotic devices MUST include emergency failsafe 'stop' mechanisms allowing them to be shut down in case of impending or actual risk to persons or objects.

2.3. Use Case 3: Artificial Intelligent Embedded Sensing Platform

2.3.1. SCOPE AND OBJECTIVES OF USE CASE

Nowadays pervasive IIoT and the need to satisfy their increasing demands for autonomy, energy-awareness and reliability have led embedded systems and users to move towards self-adaptive solutions providing IIoT units with self-adaptation, management and healing functionalities. A major ability for IIoT requires autonomously adapting their behaviors in response to changes affecting the mounted sensors (e.g., faults, ageing effects) or the environment (e.g., time-variant scenarios) or the applications the system is deployed in (e.g. Industry 4.0, Agriculture 2.0, Datacenters, Smart Homes and Cities, Predictive maintenance, etc.).

Given the complexity of the problem, self-adaptive IIoTs are highly fragmented with poor solutions suitable for embedded architectures. This use case introduces embedded solutions for detecting changes at the sensor level in units of IIoTs. It takes advantage of state of art solutions, to permit the units of an IIoTs to detect faults in the sensor or time-variance in the environment without requiring any-apriori information about the system under inspection or the changes.

The novelty of it resides in the joint use of learning mechanisms to build predictive models describing the sensor datastream over time; model-free change-detection mechanism inspecting changes in the acquired datastream and change-point methods to reduce the occurrence of false detections. This results into an IHES (Intelligent Heterogenous Embedded System) Platform.

It is then used to define a horizontal use case technology driven. It is meant to develop and prove some technology components in order to bring artificial intelligence and statistical approaches into resource constrained embedded sensors and associated micro controllers. It consists of a distributed embedded infrastructure for self-learning from raw data being self-adaptive, self-configurable and with self-managing capabilities needed for the evolution of IIoTs toward intelligent systems. It's horizontal because will provide high tech enablers at SEMIoTICS field device level to value added vertical use cases to consider such as typical IIoTs address.

The IHES platform is composed by 3 different entities: 1) a set of heterogenous sensors (e.g. environmental, inertial); 2) micro controller units (MCU) and 3) a coordinator gateway/router unit to connect the set of sensors and the MCU whereas the coordinator connects itself to the cloud. The MCU is an excellent low power programmable support for AI sensors and for connecting them to the gateway. The latter acts as a supervisor controlling sensing units and triggering their self-learning, self-adaptive, self-configuration and self-management capabilities with a global view of all of them. Furthermore, the gateway conveys compact meta-information from and to the cloud strictly related to key phase of self-learning.

This platform offers a sophisticated embedded system design methodology coupled with a technological implementation specifically designed for detecting changes at signal model level during the sensor acquisitions in these AI-powered sensing units.

The platform features a hierarchical, distributed, scalable system that will consist of several processing phases: 1) per single sensor acquisition of a physical phenomenon (e.g. temperature) followed by a learning phase to get a model out of the phenomenon 2) detection and validation of any change in the single sensor input signal dynamic vs. the model of it at run time and in an unsupervised way; 3) coordination of any re-learning phase of any of those single sensing units 4) any single sensor node will be coordinated in a selective way, depending on its correlation with any other single sensor composing the platform.

Objectives:

A technological implementation, at platform level, composed by hardware from STMicroelectronics of the proposed system with associated software components (such as predictive models, change detection and validation phases) will be developed.

AI will be introduced and deployed at edge field devices level to reduce the impact on the communication bandwidth of raw between the sensing units and the cloud as well as to increase responsiveness to any need of adaptation to the time varying nature of the application environment in which to deploy the platform.

This approach makes the system scalable and resilient to local faults. The IHES Sensing Units need also to achieve high event detection accuracy and low computational load and memory occupation in order to make it suitable for implementation on embedded micro controllers.

However, it is indeed its genericity, and the associated platform, that has been considered as an enabler for specific vertical applications that need to detect changes at single sensor level. This novel approach could be an inspiration for the developers who works at this task.

Traditional, non-AI, IoT sensing systems are nowadays widely available in the form of centralized cloud services. That means that raw sensor data needs to be carried through the network to a powerful server farm. AI is applied there. Several IoT cloud platforms allow connecting directly sensing nodes to the cloud service. Among the others the Microsoft Azure and Amazon AWS provides a powerful ecosystem for developers. Unfortunately, these systems are all connected-centric, centralized systems, where the nodes are simple raw data streamers and where typically actual AI is executed at remote location as part of the cloud service offering as said in a centralized computing platform. This approach poses unsolved challenges on data integrity, data security, privacy, system scalability and responsiveness, high system latencies, as a direct consequence of the fact that data are analyzed and classified in the remote cloud-centric infrastructure.

As an example, consider a centralized service of voice recognition under these assumptions: 1) Average person's daily utterances as 16,000 words⁶; 2) an average speech rate of 163 words per minute⁷; then we have 98 minutes of speech per day. Multiplying that number by 128 kbps communication data rate to the cloud, the result would be 94 MB of voice data per person per day, 1 million persons (≈ 94 TB), 10 million (≈ 1 PB) and 100 million (≈ 9 PB). Also the speech recognition needs to happen in a centralized server farm overloading it as the users will grow. Such a system offers poor performances in term of scalability, responsiveness and network bandwidth requirements. By decentralizing the speech recognition intelligence, close to the microphone sensor where voice is captured, previous issues will be resolved. On the other side this type of field device must ensure accurate recognition and low complexity. It is well known that accurate recognition usually requires huge amount of data for the learning phase of the AI recognition and high complexity to deploy local learning. However when the field device objective is the accurate detection of inherent changes of time varying sensed signals then amount of data and computation complexity can be drastically reduced to be affordable for the scarce resources of a micro controller unit still achieving high detection accuracy of changes.

STMicroelectronics AI Sensing platform adopts the core principles of the edge/pervasive computing paradigm as opposed to cloud-centric approach, where the intelligent processing of sensed data is moved and distributed to the leaves of the system, at field level sensing devices, embedding algorithms based on the highly nonlinear approximation capabilities of artificial neural networks, statistical analysis, distributed computation for increased system scalability, safety and robustness. In particular, the core functionalities of the system are moved at lower levels of the platform and two key aspects are therefore implemented:

⁶ https://www.researchgate.net/publication/6223260_Are_Women_Really_More_Talkative_Than_Men

⁷ <http://sixminutes.dlugan.com/speaking-rate>

- 1) **Local predictive analytics for environmental and inertial data streams:** In the AI Sensing use case, localized edge analytics will be applied which will result in unsupervised IoT behavior where only results and events triggered by these analytics will be propagated to the upper level on the infrastructure to the IoT Gateway locally and to enable a seamless deployment on a Vertical Application at network layer.
- 2) **Local AI Sensing behavior and monitoring:** The sensing of this environmental and inertial data streams acting locally at sensor level allows to process more data in real-time and to precisely identify relevant events by modeling the characteristics of the acquired data thanks to neural network self-learning algorithms. Security is increased as well because complete raw data are almost rarely propagated to the IoT Gateway, but just those identifying anomalies are transmitted for complex processing. This makes the system highly scalable, robust, and largely autonomous on its local behavior.

A significant and meaningful incremental demonstration of the technology will be implemented and validated during the project as described in the following section.

2.3.2. NARRATIVE OF USE CASE

2.3.2.1. SHORT DESCRIPTION

On April 6th, 2009, Student's House in L'Aquila (Italy) was devastated by a strong earthquake and 8 young students died. On January 18th, 2017 a huge avalanche hit Hotel Rigopiano in Farindola, close to Pescara (Italy), causing the death of 29 people present in the structure for winter vacation. Those tragedies taught us that about the importance of an automatic earthquakes events detection system. If the streak of earthquakes events that occurred before these two disasters were notified to the National Protection Department, maybe the early evacuation procedures would have been started. This could have saved many precious lives.

UC3 aiming to provide an innovative technology for enabling AI in distributed low power IoT field devices. The innovation consists on moving the analytics from the cloud server directly as close as possible to the sensors, "close to the edge", empowering sensors analytics by powerful yet optimized AI algorithms specifically designed for low power embedded micro-controllers.

UC3 is also a horizontal UC because it aims to provide an enabling technology that could be thus used to address a wider range of vertical specific scenarios in which event detection is needed.

2.3.2.2. COMPLETE DESCRIPTION

A preventive restructuring toward earthquakes of the Student's House in L'Aquila, Italy hosting university students from all over the world was done in 2000. Despite that on Monday March 30th, 2009 at 15:38 PM CET after a 4th Richter magnitude earthquake, the house was evacuated for about 3 hours to check its structural robustness. It was the fourteenth earthquake since January that year. Floors and walls were violently shacked, and the students were shocked at a point required the house management to intervene. Unfortunately, 5 days after Student's House was devastated by a strong earthquake and 8 young students died. After many investigations and subsequent legal trial, the event was deemed predictable. The area where the building was located presents a highly earthquake risk, unfortunately it was not properly and continuously monitored.

On January 18th, 2017 a huge avalanche hit Hotel Rigopiano in Farindola, close to Pescara, Italy, causing the death of 29 people present in the structure for winter vacation. Following some technical analysis, it resulted that the incident was triggered by a series of local earthquakes in central Italy in

conjunction with the sudden raise of the atmospheric temperature that suddenly melted the snow so increasing extremely fast the risk of avalanches in that particular location.

Those two tragedies taught us that if earlier and automatic such events would have been detected and automatically notified to National Protection Department, early evacuation procedures would have been initiated or detailed monitoring would have been deployed probably resulting on saving of many precious lives.

More in general automated event detection based on systems which acquire knowledge on the field where they are installed is becoming of paramount importance not only because of lives saving and also because they don't need any pre knowledge offline characterization. This will enable their full adaptation to time varying environment in which deployment of preexisting knowledge would be inadequate or too limited as not being able to adapt to an expected mix of conditions which ultimately need to be notified to a service as soon as they are sensed and acknowledged in their time varying intimate nature.

According to the defined context, UC3 main objective is to show how the technology that will be developed within SEMIoTICS could possibly trigger alerts in advance, log relevant events, in an autonomous way, so to alerts through e.g. alarms to any service or even to the population and prevent the occurrence of tragedies like the ones above mentioned. SEMIoTICS eco system will provide the perfect infrastructure in which the Edge Computing and the Local Embedded Analytics could be naturally deployed. SEMIoTICS envisages the deployment on the target field (e.g. environment, homes, public access buildings and areas) of sensors boosted with artificial intelligence capability. Sensors learn in real-time the normal condition to autonomously trigger with value and timestamp any significant deviation of conditions to the SEMIoTICS IoT Gateway. The gateway receives and compare the information from multiple sensors and is able to discern consequently between local (i.e., true just in "that" area and eligible to be better verified of not being an occasional one) or systematic (i.e., a known recurring event not representing a real anomaly). The Gateway locally stores and maintains the filtered events in a local database eliminating false positives or irrelevant logs. The weighted combination of multiple conditions (e.g. abnormal recurring structural vibrations, inclination changes, etc.) in building and structures together with the environmental local condition (temperature, rain, snow, etc.) can justify the activation and the implementation of preventive local safety procedures (e.g. elevators at ground zero, denial of access in certain places). Moreover, a "next level alert" triggering toward a local area server can be activated for best evaluation of the situation (e.g., messages to the relevant owners/operators of the structure, local authorities...) to avoid under-evaluation of any potential critical situation before any significant danger would materialize. At the local area server level, a further monitoring and evaluation of the "ongoing situation" can benefit of queries, matching the already available information with forecast available for the region/city in web-hosted databases (e.g. seismographic events just occurred that could provide justification of the registered abnormal activity, whether expected evolution in the area, people presence in the structure, traffic situation around...). That combination of information could alert the relevant authorities to activate increased level of supervision, inspections, up to implementation of the required public security procedures (such as denial of access in areas, evacuation orders in the more potentially critically judged situation). It's all about utilizing the technology to permit acting "before" potential critical situation degenerate into highly dangerous situations.

Two different scenarios have been identified of a relevance for demonstrating UC3 using SEMIoTICS framework in a lab environment:

- **Scenario 1 - Local vs Global anomalies**

A series of inertial intelligent IoT Nodes are deployed in the public building under control analyzing the vibrations at different places and floors. Equivalent systems are deployed in

nearby buildings. When abnormal vibrations are reported the system analyzes whether similar anomaly is reported by the neighborhood and if observed whether similar events has been reported by area services (e.g. by the high precision seismographs belonging to the INGV⁸ - Istituto Nazionale di Geofisica e Vulcanologia webservice). In case of positive feedbacks, the system reports the alert to owners and operators of the building as well as the local authorities while automatically send the command to the elevators control system of the building to reach level zero and stop working preventing people from use them until the alert is released. Authorities may decide after check to issue an evacuation alarm.

- **Scenario 2 - Causal discovery and inference**

Additional outdoor sensors are utilized to track local temperatures and lighting / sun heating condition. This time the validation is done vs the remote DB of the local ARPA⁹ (Agenzia Regionale Per l'Ambiente) reporting area expected trend and other useful information such as avalanche or flood risks for the zone. In this case there is no “event” triggered as such but rather a more detailed evaluation of the evolution in the local area given the local condition enforced by the area expected trend. Again, owners and operators of the structure would be alerted when in front of potential risky trends better evaluated by authorities leveraging historical statistics and associated safety procedures would be activated as precautionary measures.

These two scenarios will be demonstrated incrementally in two macro steps: a first stage focused on testing the system in isolation on a controlled environment (i.e. in a simulated laboratory environment) in order to effectively demonstrate the effectiveness of the local embedded analytics and edge computing core capabilities at work. After this laboratory validation by ST-I domain experts, a second phase mainly developed during Task 5.6 activities, will be focused on the integration of this platform in the SEMIoTICS framework, enlarging the test environment to dedicated field trials, including proper end-to-end validation from Field Devices (IHES Sensing Units), to IoT gateway up to SEMIoTICS network and backend/cloud infrastructure where the UC3 App will be deployed.

2.3.3. TECHNICAL DETAILS

2.3.3.1. DIAGRAMS OF USE CASE

In this section some drawings related to the horizontal use case depicted in section above are shown. In particular, according to it three main actors are defined to implement the desired functionalities and requirements:

- A set of AI enabled intelligent sensing units for unsupervised online learning from environmental and inertial sensors

⁸ INGV <http://cnt.rm.ingv.it/>

⁹ ARPA https://www.arpalombardia.it/Pages/ARPA_Home_Page.aspx

- One or more supervisor units for event management and system coordination
- A monitoring visualization GUI for the system management and events logging facilities

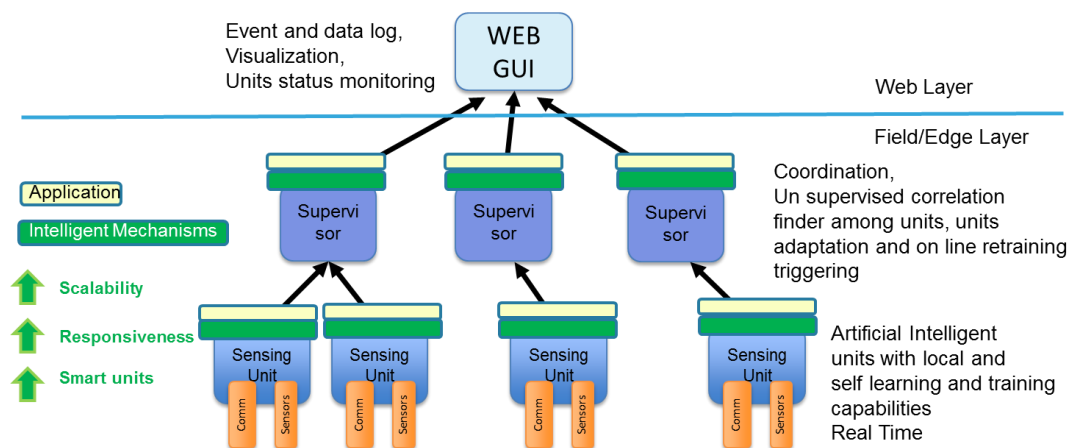


FIGURE 9: ARTIFICIAL INTELLIGENCE SENSING IIOT SYSTEM OVERVIEW

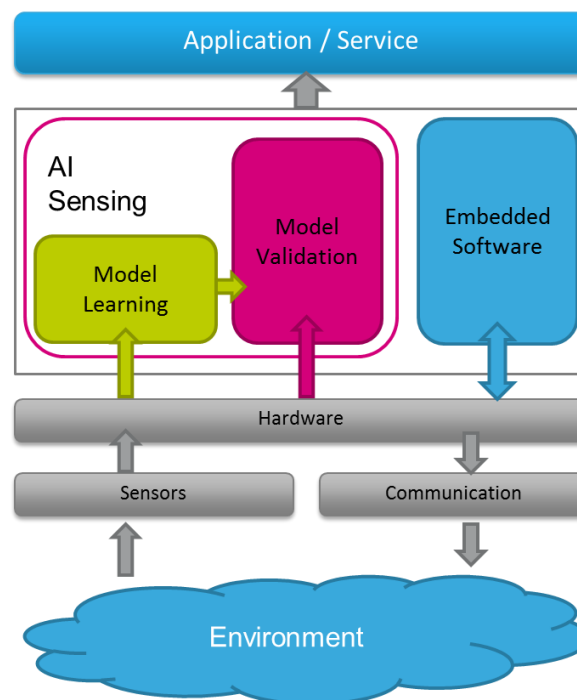


FIGURE 10: ARTIFICIAL INTELLIGENCE SENSING PLATFORM FLOW DIAGRAM

2.3.3.2. ACTORS

- *IHES Sensing Unit*

This component shall be capable to interface itself with the physical world, in order to quantitatively measure physical quantities using environmental (temperature, humidity, pressure, light) and inertial (3-axis accelerometer) sensors. Moreover, it shall be capable to learn the model behind the temporal variations from sample to sample, from time to time of those quantities. That model shall be used as basis to predict the measured quantities in order to compute a real time prediction error. In turn it shall be used to detect changes underlying the measured physical quantities. Since those changes could be associated to the un-capacity of the learnt model to predict the measured quantities a new model shall be learnt. That situation shall be detected and validated as not being a false detection. If it would be a false detection than the already learnt model shall be still applied as valid for quantities prediction. Otherwise if there is an underlying variation of the physical quantity dynamic that shall be learnt as a new model adapting to the new conditions.

The unit shall be capable to support all above operations and signal any event that is underlying a model variation by minimizing false detections. It shall be capable to notify such changes to the IHES Sensing Supervisor and well as portion stream of data for further processing.

The unit shall be capable of connecting to a IHES Sensing Supervisor and coordinates with it for reporting relevant changes from the learnt model and reconfiguring in case.

- *IHES Sensing Supervisor*

This component shall be capable to have an aggregated view of all IHES Sensing Units. It shall be able to compute the dependency graph: its goal is to find any correlation that might exist among connected sensors. For example, a temperature sensor of a unit might be correlated to a pressure sensor of another unit. Those correlations might change over the time and in this case, it shall be able to refresh those correlations and the associated view.

The component shall have also the capability to receive and aggregate data, information and events received by each IHES Sensing Units. It shall be able to integrate change detected by each unit and the dependency graph in order to understand if a change is a localized to that unit (that interface with) or affecting a behavior at global level detected by the other units (that interface with). Consequently, it shall trigger the need for a localized learning process (at unit level) or at global level (units' level that are correlated).

The component shall have the capability to exchange above information between itself, the cloud and the units with some connectivity capabilities.

- *IHES GUI Application*

That component shall be capable to display graphically all the information about the system, the correlations existing between sensors (*dependency graph*) and update any variation using information received by the gateway. Moreover, it shall be capable to capture and display detected changes in term of timing as well as type of sensor unit and associated identifier.

2.3.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The trigger for all of UC3 sub-scenario is the need to detect events any types of events based on data acquired using different (homogeneous and heterogenous) sensors.

To achieve this, different steps are needed: firstly, all the Sensing Nodes must be operative and have successfully perform a training step. Then, when a change is notified by a Sensing Unit the IHES Supervisor, it checks data from the other connected Sensing Nodes and external information (i.e. INGV webserver data, local ARPA online data). If there are similar feedback from other sources the e IHES Supervisor will timely alert those who need to have this information (i.e. local authorities) and eventually perform the self-adapting strategy.

The main outcome of this UC is a SEMIoTICS integrated system of IHES Sensing Units that are coordinated and supervised by an IHES Supervisor in order to implement the described intelligent and self-adapting behavior. Then, all the information about sensing units, the detected events and the self-adapting smart behaviors should be made available through a secure channel in a web monitoring dashboard (IHES GUI Application).

2.3.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

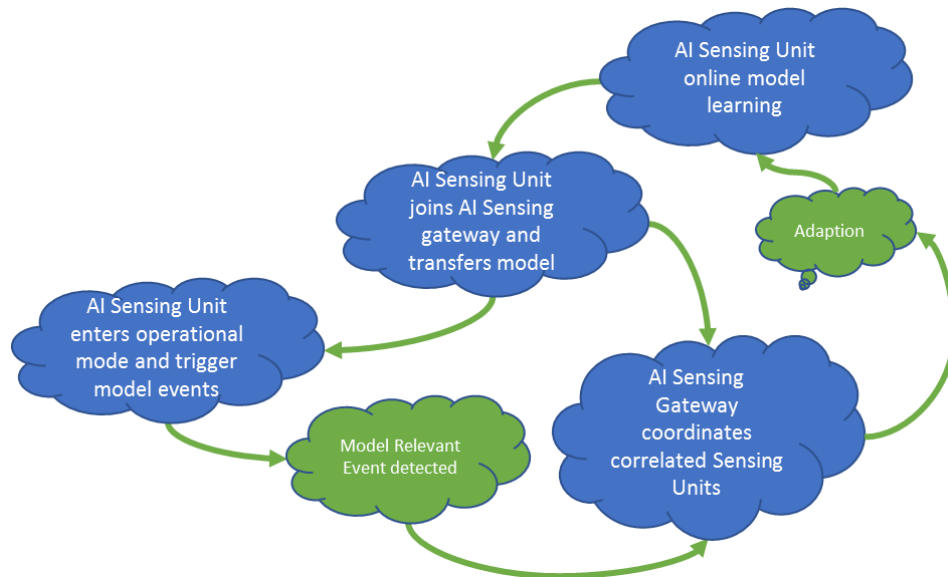


FIGURE 11: IIOT SMART SENSING ACTORS AND INTERACTIONS

2.3.3.5. REQUIREMENTS DESCRIPTION

UC3 requirements described in the table below are defined via domain expert interviews and ST-I expertise in the field.

Req.-ID	Description
R3.1	IoT Sensing unit shall be able to embed environmental (e.g. temperature, pressure, humidity, light) and inertial sensors (accelerometer, gyroscope).
R3.2	IIoT Sensing unit shall be able to interface to the IIoT Sensing gateway in order to coordinate with it. A standard IP based (i.e. TCP transport) 1 to many M2M communication protocol must be adopted to properly handle node communication with components in the gateway.
R3.3	IIoT Sensing unit shall be able to learn a model from observed data in an unsupervised manner. In particular IoT Sensing unit shall be equipped with a low power (tens/hundreds of mW range) 32 bits MCU to support unsupervised learning and unsupervised statistical processing.
R3.4	IIoT Sensing unit shall be able to detect relevant changes from the learned model and report them to IIoT Sensing gateway.
R3.5	IIoT Sensing unit shall be able to adapt to a new model if IIoT sensing gateway requires this.
R3.6	IIoT Sensing gateway shall be able to coordinate a set of IIoT sensing units by finding any correlation between them according to observed data models.

R3.7	IIoT Sensing gateway shall be able aggregate relevant events (i.e. changes) coming from whichever of connected IIoT sensing units deciding if they are global or local changes.
R3.8	IIoT Sensing gateway shall have the capability to exchange relevant information (i.e. events) between itself, the cloud and the sensing units with some connectivity capabilities.
R3.9	IIoT Sensing web GUI shall be able to display correlations between connected IIoT Sensing units and the status related to each IIoT sensing unit.
R3.10	IIoT Sensing web GUI shall be able to display logging about relevant events detected by connected IIoT Sensing units reporting info about unit ID, type of data and type of event detected.

3. Non-SEMIoTICS Use Cases

This chapter contains 7 in-depth use cases not belonging to SEMIoTICS. Each use case has the following uniform structure and the text marked in *blue* contains the respective use case specific information.

Sub-section	Description
1	Scope and Objectives of Use Case: <i>The scope defines the limits of the use case and the objectives mention the motive behind the use case</i>
2	Narrative of Use Case (Short & Complete description): <i>Short description is intended to summarize the main idea of use case and Complete description is a full narrative of the use case from an expert user's point of view, describing what occurs when, why, with what expectation, and under what conditions.</i>
3	Technical Details
	3.1 Diagrams of Use Case: <i>The diagrams show step-by-step interactions. e.g. Use case diagram, sequence diagram, activity diagram, etc.</i>
	3.2 Actors: <i>In this section, actors which are involved in the use case are listed and described. These can for instance include people, systems, applications, databases, devices, etc.</i>
	3.3 Triggering Event, Preconditions, Assumptions, Success criteria/expected outcome: <i>Triggering event describes what event(s) trigger(s) this use case (Actor/System/Information/Contract). Pre-condition(s) describe(s) what condition(s) should have been met before this use case happens. Assumption may be used to define further general assumptions for this use case. Finally, the success criteria or expected outcome of the use case, if foreseen is mentioned.</i>
	3.4 Information exchanged between actors: <i>It summarizes the information exchanged between two actors with high level concepts for SDN/NFV.</i>
	3.5 Requirements description: <i>Requirements coming from high level concepts and technical details written in the use case.</i>

3.1. Use Case 4: SPDI pattern-based management of smart building infrastructure

3.1.1. SCOPE AND OBJECTIVES OF USE CASE

The SEMIoTICS will develop a pattern-driven framework, built upon existing IoT platforms, to enable and guarantee secure and dependable actuation and semi-autonomous behaviour in IoT/IIoT applications.

The objectives of the use case include:

- The development of patterns for orchestration of smart objects and IoT platform enablers with guaranteed Security, Privacy, Dependability, and Interoperability (SPDI) properties.
- And the development of dynamically and self-adaptable monitoring mechanisms supporting integrated and predictive monitoring of smart objects in a scalable manner.

3.1.2. NARRATIVE OF USE CASE

3.1.2.1. SHORT DESCRIPTION

The application setting of this use case is a smart building. The SEMIoTICS will utilize patterns to manage the smart building ecosystem, offering assisted living services to the users during normal operation, while preserving privacy goals. In case of cyber-attacks, the system will be re-configured at runtime, increasing security and countering the threat (e.g. Denial of Service (DoS) attacks, malicious access attempt, etc.) (Hatzivasilis et al., 2017a).

The goal is to enhance the key IoT challenges of:

- Dynamicity,
- Heterogeneity, and
- End-to-end (E2E) security and privacy

3.1.2.2. COMPLETE DESCRIPTION

A popular application of ambient intelligence systems constitutes assisting living services on smart buildings. Intelligence is imported in embedded equipment and the system becomes able to control lights and air-conditioning or provide energy management services. IoT is the main enabler of such environments. However, the interconnection of these cyber-physical systems and the processing of personal data raise serious security and privacy issues. SEMIoTICS will take advantage of the underlying IoT setting, not only for implementing the requested smart building functionality but also for enhancing the modelling and administration of the offered SPDI properties (Hatzivasilis et al., 2016).

Moreover, the system could be configured at runtime in order to retain the desirable SPDI goals. This operation is important, especially in cases of cyber-attacks. Intrusion detection systems will discover ongoing attempts to infiltrate the smart building infrastructure and alert the involved SEMIoTICS components (e.g. the Intrusion Detection System (IDS), the SDN controller or other intelligent management agent). Then, the SPDI controller (placed in the SDN controller) will be able to alter the system architecture automatically, based on pre-defined strategies, and enhance protection. The strategies are designed by the security experts of the organization and tackle a specific set of cyber threats. Also, a remote management service will enable the system administrator to change the system manually and update the reaction plans based on the latest security guidelines (i.e. made by Computer Emergency Report Teams (CERTs)). A CERT is responsible to enable information sharing capabilities and tool support between other security teams to manage incoming events and incidents.

The trust between the CERT/CSIRT community¹⁰ is guaranteed by the accreditation and certification of CERT Team by authorities such as TERENA¹¹/FIRST¹².

Thus, the users would consume safely the deployed services. In case of attack, like DoS, their security will be retained, and the personal data will be safeguarded as the SEMIoTICS's SPDI patterns feature imposes the principles of 'E2E protection'.

The use case targets IoT developers and administrators. The SPDI patterns assist the good practices towards the 'security-by-design'. Furthermore, the system manager is equipped with a systematic methodology that verifies the acquired defense level and preserves protection at runtime.

Currently, there is no concrete method to tackle the four SPDI perspectives in parallel. Moreover, the heterogeneity nature of the IoT ecosystem and the high volume of communicating devices harden the establishment of E2E properties, their monitoring as well as their runtime management. The SPDI patterns technology of SEMIoTICS will accommodate the deployment and operation of modern IoT appliances.

3.1.3. TECHNICAL DETAILS

3.1.3.1. DIAGRAMS OF USE CASE

The following figure depicts the application of SPDI patterns in the smart building scenario. In the knowledge base (KB) the SDN controller maintains locally the core SPDI properties for every individual component of the underlying subsystem (i.e. the IoT devices and their services). Then, the controller estimates the properties of the currently integrated setting. It can permit/block composition activities (e.g. in cases where a node cannot be considered trusted or it does not deploy the required protection modules) or change configurations at runtime in order to comply with the designed E2E properties (e.g. enforce encrypted communication with larger cryptographic keys in order to increase security in case where the system is under attack). As aforementioned, the administrator can also perform the same functionality through a remote management service and update the controller's reaction strategies.

¹⁰ <https://www.enisa.europa.eu/activities/cert/support/guide2/introduction/what-is-csirt>

¹¹ www.terena.org

¹² www.first.org

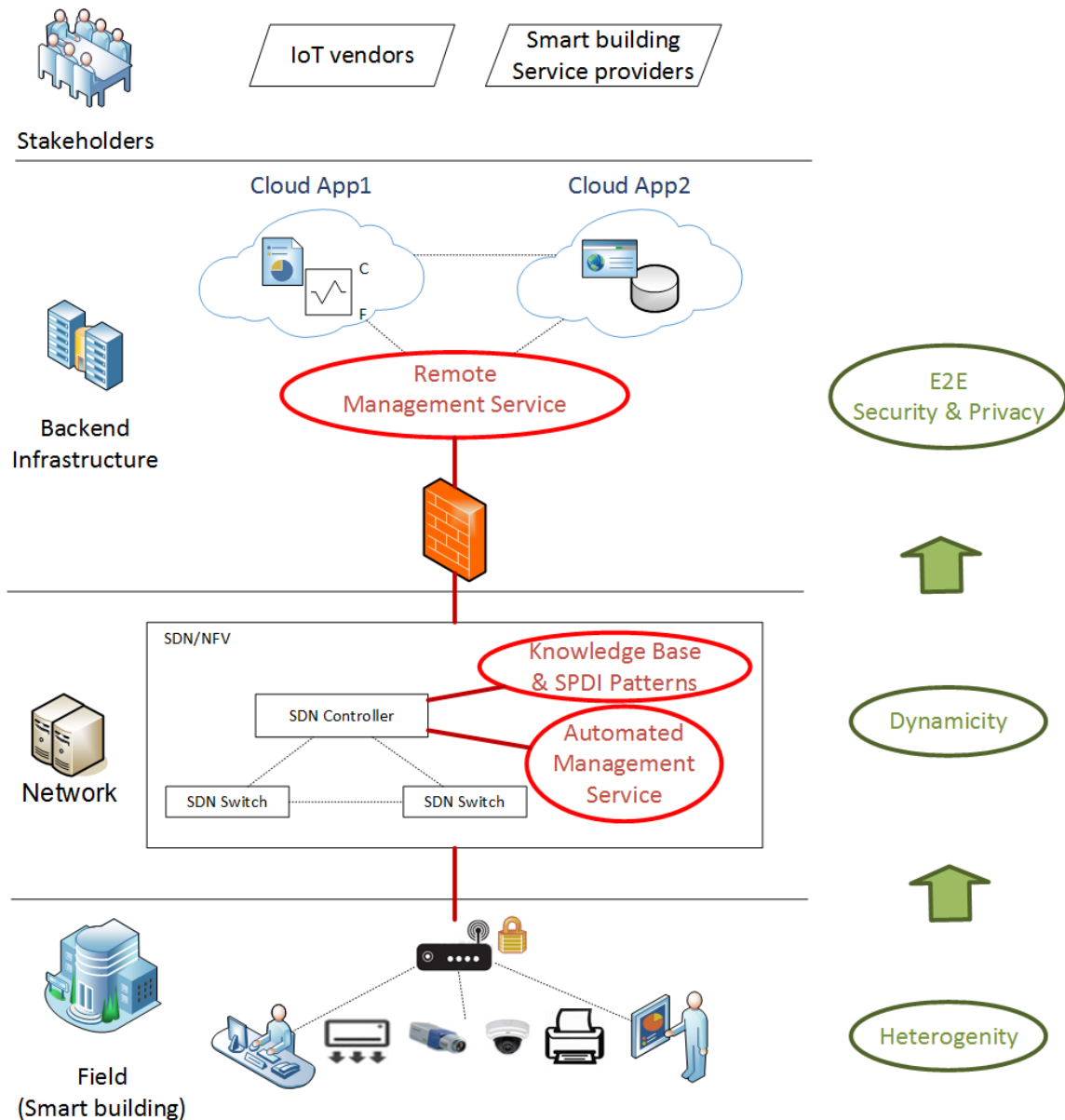


FIGURE 12: SMART BUILDING – SPDI PATTERNS

The relevant UML Use Case diagram is illustrated in the figure below. The main actuators are the system designer and administrator who model the main SPDI features and the reaction plan. The SDN controller evaluates and monitors the composed system at runtime while performing the automated management strategies.

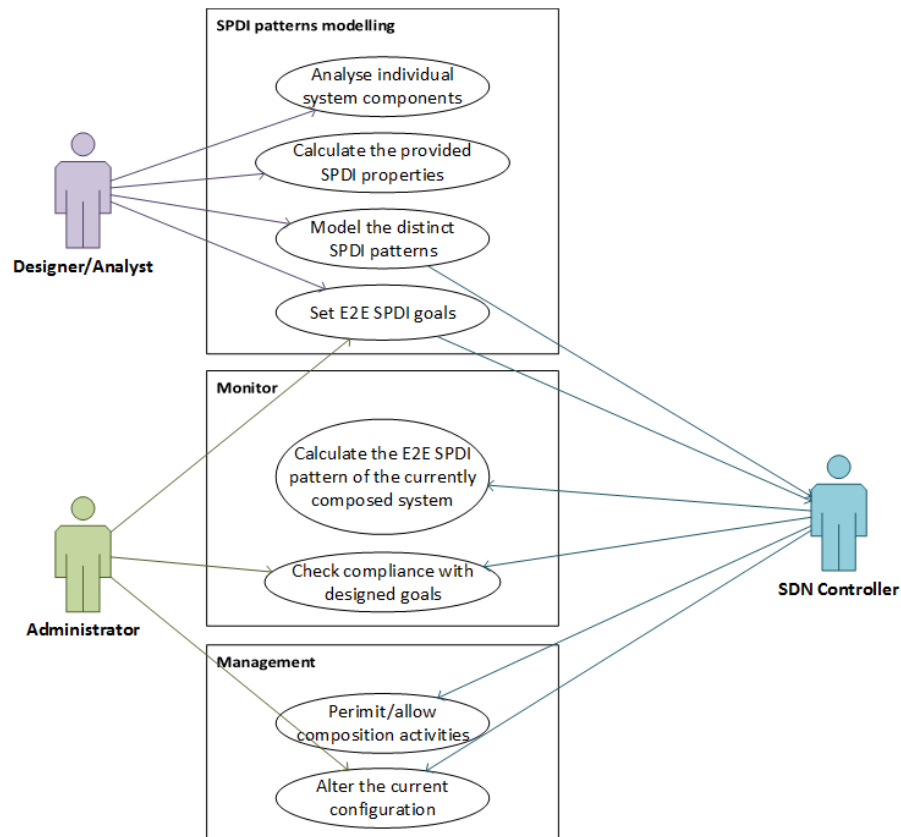


FIGURE 13: USE CASE DIAGRAM OF MODELLING, MONITOR, AND MANAGEMENT OF SPDI PATTERNS

The next figure describes the sequence of events for incident response when a cyber-attack is performed (e.g. unauthorized access attempt). The system detects and counters the attack automatically. The administrator is notified for the incident and, after analyzing the data, reports the event to the CERT. The CERT performs thorough analysis and disseminates the new guidelines for mitigating such attacks to the relevant action team (i.e. a group of people that is responsible retaining the system's security, including administrators and security specialists). The team updates the response policy and the administrator deploys the new SPDI patterns to the SDN controller, which adapts the system to the new state.

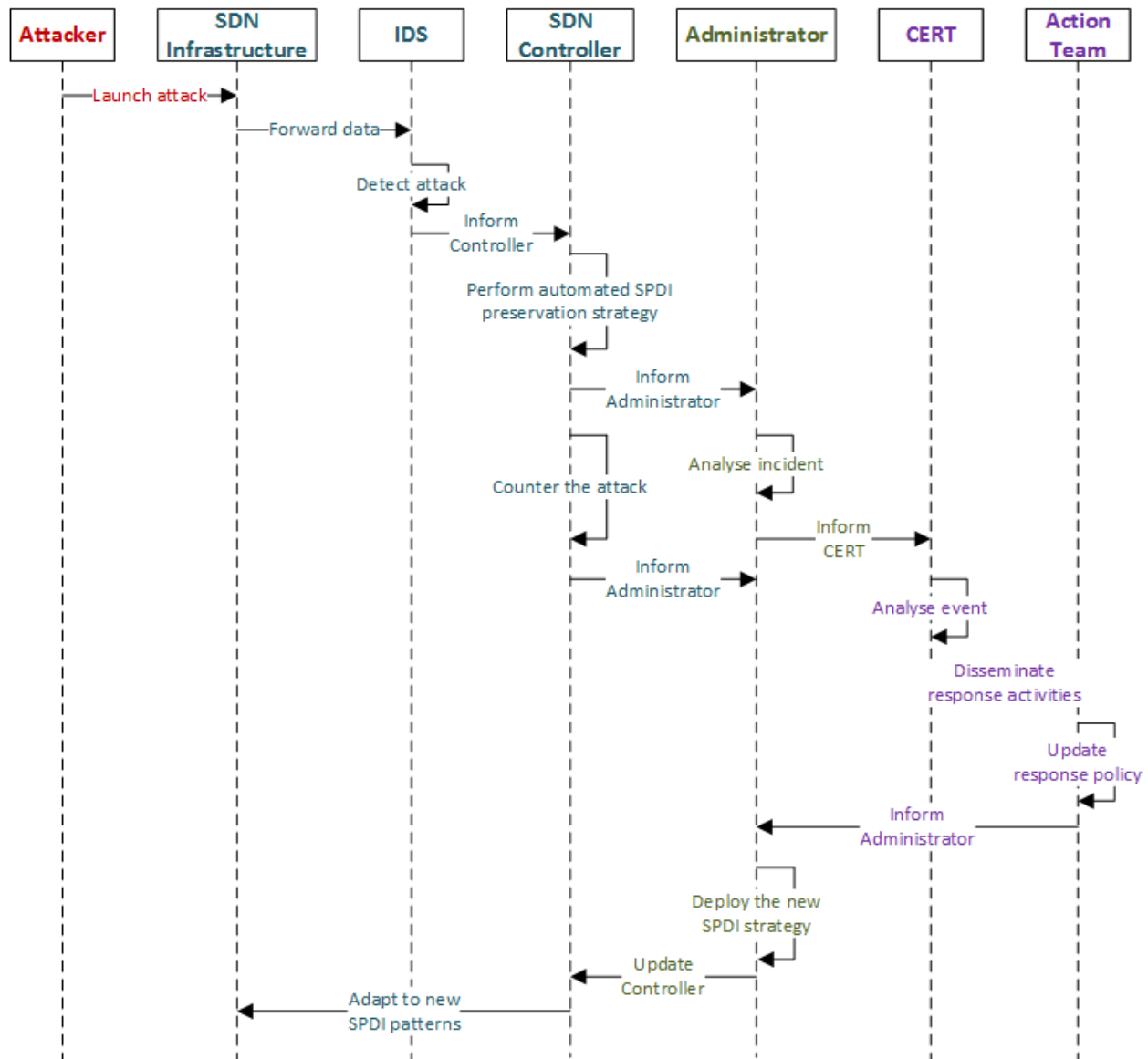


FIGURE 14: SEQUENCE DIAGRAM FOR INCIDENT RESPONSE

3.1.3.2. ACTORS

1. **Network operators:** provide Internet access and deploy networking equipment (routers, switches, SDN) to configure their subsystem.
2. **System designers/analysts:** design the deployed IoT system and model the core SPDI properties of the individual system components and services. A pattern language will be defined for the SEMIoTICS project that will support the specification of all facets of patterns and will enable the automated application of patterns to realize key capabilities offered by the SEMIoTICS framework, like orchestration, verification, and adaptation of smart object compositions at runtime.

3. **SDN controller:** monitors the network traffic and manages the underlying components based on SPDI goals. The controller enforces strategies for ensuring specific SPDI patterns and achieves E2E properties. In case of attack, the controller automatically configures the system at runtime to preserve protection and an adequate SPDI level.
4. **Service providers:** offer IoT services for the smart building. The service types include: i) energy management, ii) assist living, iii) physical safety and disaster mitigation, iv) and cyber-security.
5. **End users:** use the network and consume the provided services.
6. **Administrators:** monitor and manage the system, detect anomalies, and control incidents. Nevertheless, when a security incident occurs, severe intrusion detection admins can forward it to the Computer Emergency Report Team (CERT) (a public or private organization that handles computer security incidents in a global or national scale) which is capable in following specific strategies and multi-level security response.
7. **CERTs:** cooperate with Internet service providers, ICT vendors and government agencies; provide several services to their member and the public, including: i) collection and dissemination of information related to computer security, ii) raising of information security awareness, iii) in depth analysis of security incidents, iv) event handling and response, v) collaboration with other CERTs.
8. **Action teams:** handle security incidents that are reported by a CERT. Among others, they also: i) install mechanisms for filtering the network traffic, ii) detect intrusion actions in the system, iii) perform actions for protecting the system threatened/affected by malicious activity, iv) patch the system, v) restore affected systems, vi) offer solutions from related advisories and alerts vii) deploy other response policies.

3.1.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The main event that is examined in this case study is the automatic response of the system to cyber threats and attacks.

Preconditions:

- The real-time monitoring procedure collects: i) cyber data including system, file integrity, and security logs and ii) reported data from the sensory equipment for physical events like system health messages and critical alerts.
- Management strategies based on pre-defined simple and E2E SPDI patterns, technical support, and cooperation with external organizations (i.e. CERTS) that are established during the design and deployment phases.

Triggering Events:

- Security or privacy events that are resulted from automated processes or human observations.
- Statistical traffic analysis and/or machine learning techniques that detect possible malicious network interactions (e.g. high volume of packets towards the same IP/port could trigger the identification of DoS attacks in the IDS) (Hatzivasilis et al. 2017b).
- External security-related reports made by trust and reputable third parties, such as CERTs), summarizing the latest cyber-security status and mitigation techniques.

Assumptions:

- The SPDI properties of every individual system component should have been evaluated during the design phase.

- The automated reactive plans should marshal the component composition capabilities and the E2E patterns.
- An incident strategy should be deployed, describing how information is passed and the actions to be taken.
- The maintenance policy of the various SPDI perspectives is also part of the response procedure.

Success criteria/expected outcome:

- Once a threat is identified, the IDS raises an internal alarm to inform the involved system components, which is then taken into account during the response phase.
- The reactive plan should constrain the bad effects of the attack or even counter it, while preserving the desired SPDI goals.
- CERT examines the security incident and provide the action team with the adequate procedures that have to be applied in order to mitigate the problem, like changing the user privileges or expel suspicious users.
- The produced data of the incident response phase (e.g. triggering events, taken actions) should be maintained in the knowledge base for feedback and reference point of future similar events.

3.1.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

1. Between the IDS and the SDN Controller:
 - a. The IDS produces an automated event monitoring capture report. The data are transmitted to the SDN Controller through a secure channel that utilizes the Secure Sockets Layer (SSL) (Freier et al., 2011) for confidentiality and integrity.
 - b. The information transfer methods are determined by the capabilities of the device that run the IDS. They may include: Virtual Private Network (VPN) (Wesinger and Coley, 2007), Hypertext Transfer Protocol Secure (HTTPS) (Durumeric et al., 2013), File Transfer Protocol with SSL Security (FTPS) (Xia et al., 2010; Gupta, 2015), File Transfer Protocol (FTP) over Secure Shell (SSH) (Xia et al., 2010), Internet Protocol Security (IPSec) (Doraswamy and Harkins, 2003), and Secure File Transfer Protocol (SFTP) (Gupta, 2015).
2. Between the SDN Controller and the system administrator:
 - a. The SDN controller will automatically process the data and classify the incident based on pre-established mechanisms and the organization's Knowledge Base that securely maintains the previously detected events.
 - b. The administrator is notified respectively and logs in to the system to access the stored information.
3. Between the system administrator and the CERT:
 - a. The administrator processes manually the event report and compares it to known events in the Knowledge Base or other third-party sources, in an attempt to classify the incident and eliminate possible false positives.
 - b. Then, the administrator logs in to the CERT's incident handling system. The event is recorded and forwarded for further process by the CERT experts.
 - c. The administrator receives back a set of actions and suggestions, describing the next steps towards the mitigation of the malicious activity. The administrator integrates this information to the organization's Knowledge Base. Then, he/she models the new SPDI patterns capturing the CERT's recommendations and updates the system respectively resolving the security problem.

3.1.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R4.1	Connectivity of the SDN controller with the various underlying components (e.g. the SDN switches and the IoT gateways and devices) in order to support the automated configuration process
R4.2	Connectivity of the SDN controller with the remote management service in order to support the remote configuration process
R4.3	The SDN controllers must be always available to serve incoming requests
R4.4	Intrusion Detection System (IDS) that captures and processes suspicious traffic
R4.5	Accredited and certified Computer Emergency Report Team (CERT) MAY get informed about an occurring cyber incident (e.g. DoS).

REFERENCES

- Doraswamy, N. and Harkins, D., 2003. IPSec: the new security standard for the Internet, intranets, and virtual private networks. *Prentice Hall Professional*, 2nd edition, pp. 1-262.
- Durumeric, Z., Kasten, J., Bailey, M. and Halderman, J. A., 2013. Analysis of the HTTPS certificate ecosystem. *Internet Measurement Conference (IMC)*, ACM, Barcelona, Spain, October 23-25, pp. 291-304.
- Freier, A., Karlton, P. and Kocher, P., 2011. The Secure Sockets Layer (SSL) protocol version 3.0. *Internet Engineering Task Force (IETF)*, RFC6101, August 2011. Available on-line: <https://tools.ietf.org/html/rfc6101?ref=driverlayer.com>
- Fysarakis, K., Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C. 2016. RtVMF – A secure Real-time Vehicle Management Framework with critical incident response. *IEEE Pervasive Computing Magazine (PVC) – Special Issue on Smart Vehicle Spaces*, IEEE, vol. 15, issue 1, pp. 22-30.
- Gupta, U., 2015. Survey on security issues in file management in cloud computing environment. *Cryptography and Security*, arXiv:1505.00729, pp. 1-5.
- Hatzivasilis, G., Papaefstathiou, I., Plexousakis, D., Manifavas, C. and Papadakis, N., 2017a. AmbiSPDM: managing embedded systems in ambient environments and disaster mitigation planning. *Applied Intelligence*, Springer, pp. 1-21.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2017b. SCOTRES: secure routing for IoT and CPS. *IEEE Internet of Things (IoT) Journal*, IEEE, vol. 4, issue 6, pp. 2129-2141.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2017c. Real-time management of railway CPS, 5th EUROMICRO/IEEE Workshop on Embedded and Cyber-Physical Systems (ECYPS 2017), IEEE, Bar, Montenegro, 11-15 June.

Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2016. Software security, privacy and dependability: metrics and measurement. *IEEE Software*, IEEE, vol. 33, issue 4, pp. 46-54.

Wesinger, R. and Coley C., 2007. *US Application*, US20070101421A1, May 2007. Available on-line: <https://patents.google.com/patent/US20070101421A1/en>

Xia, L., Chao-sheng, F., Ding, Y. and Can, W., 2010. Design of secure FTP system. *International Conference on Communications, Circuits and Systems (ICCCAS)*, IEEE, Chengdu, China, July 28-30, pp. 270-273.

3.2. Use Case 5: Leveraging SDN/NFV for minimizing energy consumption in multiple smart buildings

3.2.1. SCOPE AND OBJECTIVES OF USE CASE

Scope

This use case deals with the overall energy consumption minimization in a multiple smart building scenario. It is assumed that the buildings belong to a single company or a public entity, and thereby the overall energy consumption is of interest. Moreover, the price of the energy is dynamic and set by the energy grid operator, which communicates with the consumers in real-time. Another assumption is that a flexible and scalable approach is considered to minimize the overall energy consumption. Namely, energy controllers can be deployed at the IoT Gateways to perform a distributed optimization or at the IoT backend. A Management and Orchestrator (MANO) framework takes this decision and assigns the corresponding virtual resources at the IoT Gateways or the IoT backend, based on the next criteria. The computational requirements of the energy consumption optimization; the available computational resources at the IoT Gateways and the IoT backend; the cost charged by the IoT backend company, which is related to the computational and storage resources used by the energy consumption optimization; the end-to-end Quality of Service (QoS) of the network between the buildings and the IoT backend. Moreover, an SDN controller has a global view of the communication network, i.e. the buildings' intranets, and the fronthaul networks between the IoT Gateways and the SDN controller. Thereby, the SDN allows a low-latency and reliable communication between sensors/actuators and the energy controllers deployed either at the IoT Gateways or the IoT Backend, see below for further clarifications.

Objectives

The main objective is stated as follows:

- **Main objective:** Leverage SDN and the MANO framework for a scalable, efficient, reliable and low-latency optimization of the overall energy consumption in multiple smart buildings¹³, within a dynamic energy pricing context.

To achieve the main objective the next objectives are envisaged:

- Use NFV MANO to implement an energy management solution leveraging Service Function Chains (SFC), and virtualized applications. VNFs and virtualized applications can run either at a set of locally-deployed distributed IoT Gateways or at the SEMIoTICS backend cloud. The decision depends on several criteria, see the scope section above.

¹³ That is, in the smart buildings there are loads consuming energy. The aim, is to minimize the energy consumed by the loads in all these buildings.

- Observe that the flexibility to instantiate the energy controllers (as virtualized applications) either at the IoT Gateways or the IoT backend guarantees a scalable, reliable and efficient solution. Namely, the MANO framework decides the most convenient placement of the energy controllers depending on: the state of the network; the dimensionality of the IoT devices' measurements; the computational cost of the problem at hand; the available computational resources at IoT Gateways and the IoT backend; and the latency and reliability requirements of the energy optimization problem.
- Low-latency and reliable communication enabled by the SDN controller.
 - When the energy controller is deployed at the IoT backend, the SDN controller must guarantee a low latency and reliable communication between the sensors/actuators and the IoT backend. Namely, in smart grids, both the number of smart meters that collect data from end users of the energy grid, and the sampling frequency of the smart meters are increasing. Thereby, a huge traffic load in the communication net is expected (Leon-Garcia, 2018), which could reduce the reliability, increase the latency and thereby affect severely the QoS of the application at hand. Note that low latency requirements are needed in some cases, e.g. when the objective is to minimize the energy consumption of workstations or data centers. In this case, the consumption depends on the computational resources assigned to perform a set of jobs, which are stored in queues whose state varies dynamically with new jobs to be processed (Giannakis, 2017). The SDN controller alleviates this problem. Namely, via Southbound Interfaces, such as OpenFlow, NETCONF, and OF-CONFIG, the SDN Controller is able to dictate forwarding decisions to the network data paths, as well as configure alternative communication interfaces that guarantee a reliable and low latency communication between the sensors/actuators at the buildings and the energy controllers at the IoT backend.

When the energy controllers are deployed at the IoT Gateways of geo-distributed smart buildings, a reliable and low-latency communication among IoT Gateways is required. Namely, for optimization purposes the energy controllers perform local computations and then they communicate between them to update the local optimization. This procedure is repeated until convergence. The communication between energy controllers must be fast and reliable to guarantee the convergence of the optimization. This low latency and reliable communication is guaranteed by the SDN controller via QoS measures such as isolation through VLANs, traffic prioritization, and bandwidth allocation.

3.2.2. NARRATIVE OF USE CASE

3.2.2.1. SHORT DESCRIPTION

Leveraging SDN and NFV technologies, this use case considers the minimization of the overall energy consumption in a set of smart buildings, within a dynamic pricing context. This optimization is implemented either by a set of distributed energy controllers, deployed at the IoT Gateway (GW) of each building, or at the IoT backend cloud. Namely, a MANO framework is responsible for that

decision, by placing the Virtual Functions that implement the monitoring and energy management functionalities to the appropriate computational resources (either at the set of distributed IoT Gateways or at the backend cloud). To this end, the MANO framework evaluates the computational cost of the optimization problem when it is implemented either in the IoT Gateways or at the IoT backend; the computational resources available locally at the IoT Gateways, and remotely at the IoT backend cloud (and the economic cost that third-party cloud platform companies charge); the end-to-end QoS of the network between the smart buildings and the IoT backend, e.g. packet error rate or estimated delay associated to the overall traffic; the QoS constraints of the energy optimization problem. Thereby, the MANO framework permits to solve the energy optimization problem in a scalable and efficient manner by taking into account the aforementioned QoS requirements. Moreover, the MANO framework is complemented with an SDN controller. Namely, when the energy controllers are implemented at the IoT Gateways, a distributed optimization arises. These types of problems rely on an iterative procedure between IoT Gateways, which need a reliable and low latency communication (McMahan, 2018). This is possible thanks to the SDN controller, which is able to configure forwarding rules that ensure the required reliability and low latency of the fronthaul network that connects the IoT Gateways. On the other hand, the network between the smart buildings and the IoT backend must offer the required QoS, in terms of latency and reliability, when the energy controllers are deployed at the IoT backend. In this regard, note that the energy management of some loads requires low latency, e.g. workstations or datacenters (Giannakis, 2017). Moreover, the trend in smart grids is that the net traffic is increasing dramatically (Leon-Garcia, 2018), thereby the routing paths and forwarding rules must be managed efficiently to avoid a loss in QoS, this is accomplished by means of the SDN controller.

3.2.2.2. COMPLETE DESCRIPTION

The **efficient management of the energy consumption in smart buildings** is of paramount importance in the future smart energy grid. Through dynamic energy pricing policies the operators can **avoid peaks of energy consumption**, which would exceed the capacity of the energy grid and thereby provoke blackouts. An efficient energy management has **societal benefits** as well, as it **reduces the CO₂ emissions**. From the consumer side, besides the societal benefits, the main advantage of an efficient energy management is the **reduction of the energy consumption cost** by taking into account the dynamic energy pricing. Thereby, an efficient, **dynamic and reliable energy control** in smart buildings must be carried out.

In this use case, the optimization of the energy consumption in a scenario where multiple smart buildings are involved is considered. That is, a company owns several geo-distributed buildings, and it aims to minimize the overall energy consumption by managing the loads within the building, subject to a set of QoS constraints and for a dynamic energy price, provided by the energy grid operator. Examples of loads to be managed are Heating Ventilation and Air Conditioning (HVACs) systems and the associated constraints are the users' comfort within the building, which is measured in terms of temperature and humidity. Another example of a load, whose energy consumption is to be minimized, is a data center, which is composed of computational and storage resources distributed among the smart buildings. In this case there are a set of jobs that arrive continuously at the data center and need to be processed (Giannakis, 2018). They are stored in a set of queues in the data center. In this case, the energy

consumption optimization is a type of resource allocation problem, i.e. one must decide which computational resources are assigned to process the jobs in the queues. The more computational resources are assigned, the faster are the jobs processed, though more energy is consumed. Observe that in this case, the constraints consist of latency that the jobs accept and the available computational and storage resources.

The smart buildings are geo-distributed and the number of loads to be managed can be large. Moreover, to minimize the energy consumption, within a real-time energy pricing framework, low latency decisions are required on the management of the loads. The energy consumption optimization problem can be solved either at the IoT backend or in a distributed way at the IoT Gateways.

On the one hand, the IoT backend has a global view of the energy management problem, in terms of the energy consumption of all the loads of the different buildings and the sensor measurements related to the loads' management constraints, e.g. users' comfort. However, all this information must be sent from the IoT Gateways to the IoT backend to carry out the energy consumption optimization. Thereby, the state of the network can affect the performance of the energy optimization problem, in terms of latency or reliability. In fact, (Leon-Garcia, 2018) reported that the net traffic in smart grids is increasing dramatically. Also note that some loads require low latency in the energy management, e.g. the data center example mentioned above. Moreover, the computational resources available in the IoT backend are in general higher than the ones in the IoT Gateways. However, for high dimensional optimization problems the cost to solve the global problem at the IoT backend may be high, both in terms of computational complexity and economic cost, when the IoT backend belongs to a third-party company.

An alternative to the optimization at the IoT backend, is to split the global problem into subproblems and to solve them in a distributed way at the IoT Gateways. That is, the IoT Gateways perform local computations and then update their results with the outputs of the other IoT Gateways, through fast iterations. The distributed optimization approach, implemented at the IoT Gateways, is suboptimal compared to the centralized optimization at the IoT backend, but it faces the low latency requirements and circumvents the high dimensional optimization problem. Namely, in this distributed approach, within the IoT Gateway of each building, an energy controller is deployed in the form of a virtualized application. This energy controller receives, with low latency and reliably, measurements of the energy consumption of all the buildings' loads, through an Intranet network. These measurements are taken by a set of energy consumption sensors. The energy controller also can receive information related to the air quality in a large number of spots within the building, which is important to assess the users' comfort and also to avoid failures of the loads, e.g. workstations or data centers. These measurements are provided by temperature, humidity and CO₂ sensor nodes. Moreover, the energy controller also receives information of the optimization procedure of the other energy controllers with low latency through an access network. Given all this information, the energy controller performs a local computation and communicates its output to the energy controllers, of other buildings, with low latency through an access network. In few iterations the optimization converges, and each energy controller decides how to manage their building's loads to minimize the overall energy consumption.

Thereby, to take into account the benefits of both the centralized energy management at the IoT backend, and the distributed one at the IoT Gateways, the next flexible solution, based on Virtual Function (VF) chaining, is proposed. A MANO controller is responsible for optimally placing the VFs, that comprise the smart buildings' energy consumption optimization, at the virtualized resources of the IoT backend or at the IoT Gateways. To take this decision the MANO framework has to evaluate what would be the computational cost of the energy optimization if it was implemented globally at the centralized IoT backend or in a distributed way at the IoT Gateways. Note that those costs are different and depend on the dimensionality of the problem. Moreover, in this regard, the MANO framework must take into account the amount of computational resources available either locally at the IoT Gateways or remotely at the IoT backend cloud. In the latter case, also the economic cost must be taken into account, as the IoT backend cloud services can be provided by third-party companies and the user can set an economical cost constraint. Moreover, the MANO framework must take into account the end-to-end QoS of the network between the smart buildings and the IoT backend, e.g. in terms of latency and reliability, when the energy controller is deployed at the IoT backend. Or the QoS of the fronthaul network between IoT Gateways, when the energy controllers are instantiated at the IoT Gateways, as the energy controllers at the IoT Gateways perform local computations and exchange updates until convergence. Finally, the MANO controller must bear in mind the QoS constraints of the optimization problem, e.g. the latency and reliability requirements when the load to optimize is a data center. In summary, the MANO framework must evaluate the joint computational cost and the communication net cost to decide if the deployment at the IoT Gateways or at the IoT backend fits more properly the QoS requirements of the energy optimization problem, in terms of computational efficiency, latency and reliability.

Thereby, low latency and reliable communication between the energy controllers, through an access network, is required when the energy controllers are deployed in a distributed way at the IoT Gateways. When the VF is deployed at the IoT backend a low latency and reliable communication of the core network can be also required, e.g. when the smart buildings' loads correspond to workstations or data centers. In this regard, also a low latency and reliable communication is required between the energy controller and the actuators that manage the loads. All these low latency and reliable communication are guaranteed by an SDN approach. Namely, an SDN controller is deployed at the Wide Area Network (WAN) which communicates the buildings. It has a global view of the network state and networking resources of the buildings' intranets and the WAN that communicates the buildings. This is accomplished by a set of southbound software interfaces with the networking equipment such as switches and routers. Also in this regard, the SDN approach will tackle the interoperability between the IoT Gateway and the sensor and actuator nodes due to heterogeneous intranets between them. Namely, BACnet protocol over a wired network is used between the IoT Gateway and the actuators, whereas a multi-hop wireless net such as IEEE 802.15.4 or a Power Line Communication can be used between the sensors and the IoT Gateway. The interoperability between sensors, actuators and IoT Gateway is solved thanks to SDN. The rationale is that SDN splits the control plane from the data plane and it provides software interfaces with the networking equipment. Thereby, the management of heterogeneous networks is much easier than in conventional networks.

3.2.3. TECHNICAL DETAILS

3.2.3.1. DIAGRAMS OF USE CASES

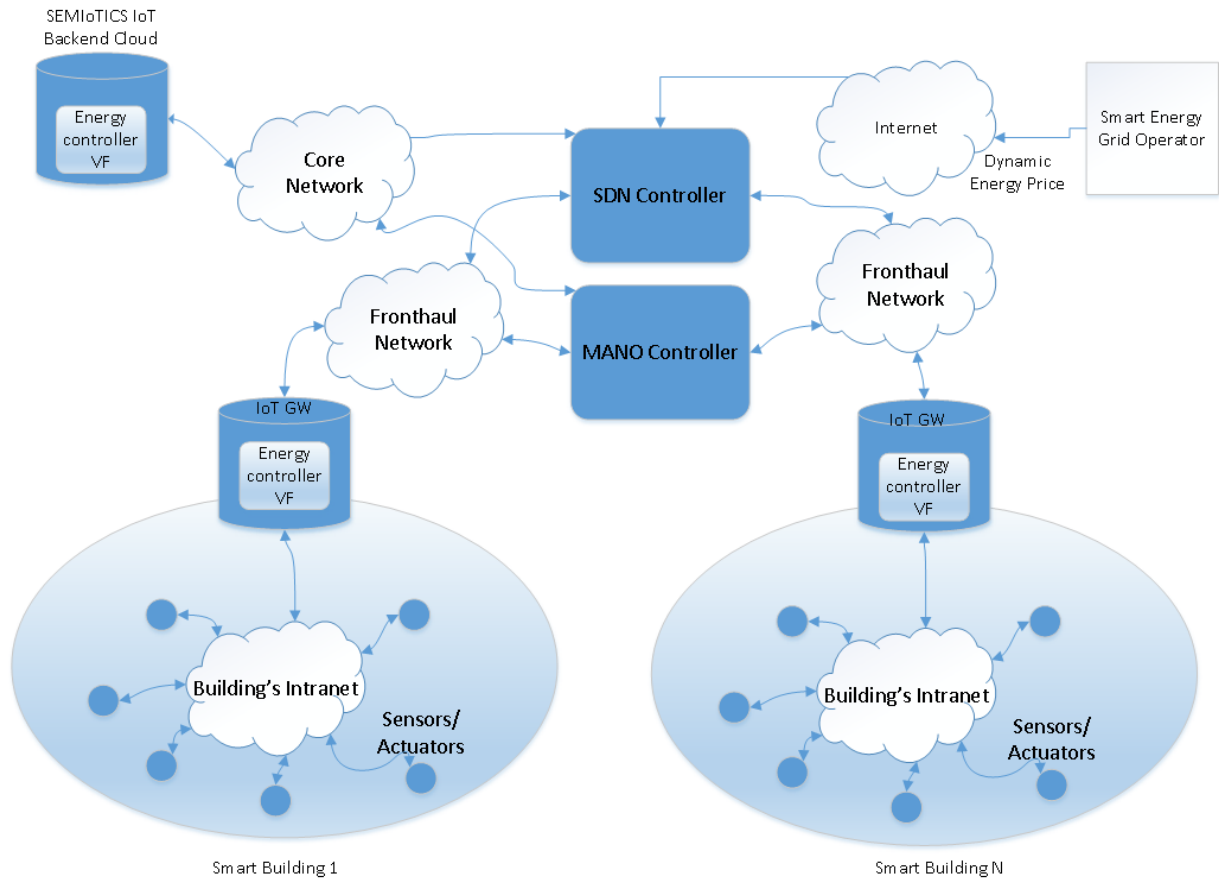


FIGURE 15: BLOCK DIAGRAM OF THE ENERGY CONSUMPTION MINIMIZATION IN MULTIPLE SMART BUILDINGS, LEVERAGING SDN AND SERVICE FUNCTION CHANNELING VIRTUALIZATION

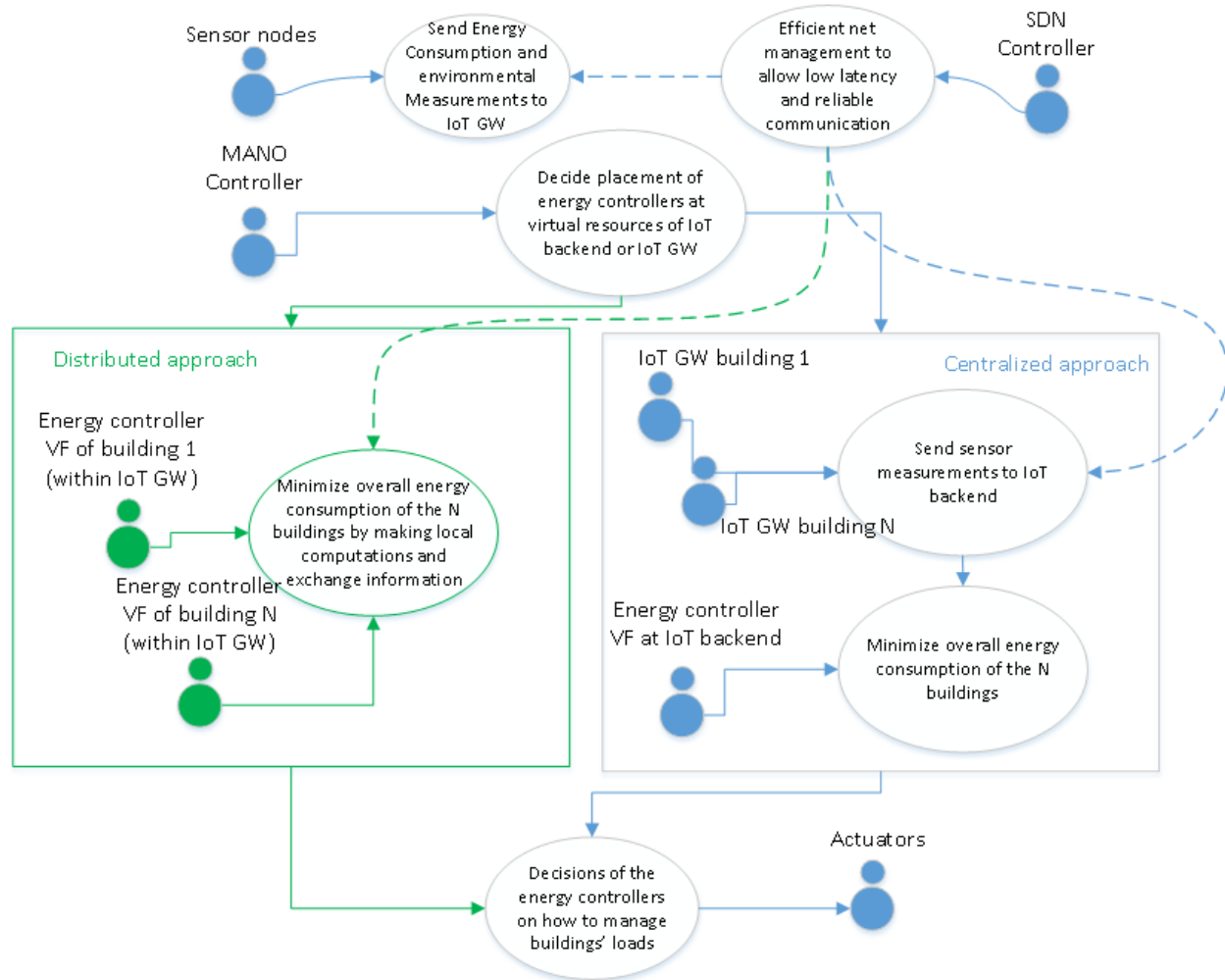


FIGURE 16: USE CASE DIAGRAM OF THE ENERGY CONSUMPTION MINIMIZATION IN MULTIPLE SMART BUILDINGS, LEVERAGING SDN AND SERVICE FUNCTION CHANNING VIRTUALIZATION

3.2.3.2. ACTORS

- **Loads:** Devices consuming energy, e.g. HVACs systems, workstations or data centers.
- **Sensors and actuators:** The sensors measure the energy consumption of the loads (commonly known as smart meters) and other parameters related to its state (e.g. temperature and pressure, to avoid failures or malfunctions). The actuators control the loads according to the decisions of the energy management controller. Moreover, there are environmental sensors of temperature and gas concentration levels, e.g. CO₂, which give information of the users' well-being within the building. Note that the management of HVACs has a direct influence on those parameters and thereby on users' well-being.

- **Building's intranet:** It enables the communication of the IoT devices (i.e. sensors and actuators) with the IoT Gateway. This is typically a heterogeneous network, as the sensors and actuators use different type of networks. As an example, the sensors can use a multi-hop wireless network such as IEEE 802.15.4, whereas the actuators can use a BACnet protocol for controlling the loads with an underlying wired network. SDN permits an efficient management of the networking resources in this heterogeneous intranet. Thereby, it improves the reliability and the latency of the communication.
- **IoT Gateway:** It is responsible for the interoperability between the local network at the building and the access network that connects the smart building with the SDN controller. To this end, southbound software interfaces managed by the SDN controller will be available. Moreover, depending on the computational and latency and reliability requirements, the MANO controller will decide to locate the energy controller's VF within the IoT Gateway.
- **Energy controller at smart building:** As it is commented in the previous point, the MANO controller decides whether an energy controller VF is deployed within the IoT Gateway or at the IoT backend. In the former case, thanks to the SDN management, it will receive reliably and with low latency the next information:
 - The energy consumption of all the loads within the building and its state, which is measured by the sensors.
 - The dynamic energy pricing (provided by the energy grid operator).
 - The environmental information of the building at a large number of spots, i.e. temperature, humidity, CO₂ concentration levels, which has a direct influence on users' well-being and loads' state.
 - The outputs of the optimization procedure carried out by the energy controllers of the other buildings in the past iteration. The aim is to minimize the overall energy consumption of all the buildings, through a distributed optimization approach. This implies that the energy controllers do local computations and then exchange information with the other energy controllers. Very fast iterations are needed, i.e. low latency, to guarantee the convergence of the optimization procedure in a fast way and to permit that the optimization procedure faces the dynamic pricing framework. This is guaranteed thanks to the efficient management of SDN and its global view of the network.

Given all this information, and after some iterations with the other energy controllers, the energy controller decides how to control the loads for the next time intervals to minimize the energy consumption subject to the buildings' constraints. These constraints are related to e.g. the users' well-being.

- **Energy controller at the IoT backend:** The MANO controller can decide to implement an energy controller VF at the IoT backend cloud. In this case, it will receive the environmental

sensor measurements and the energy consumption measurements of all the loads within all the buildings. Given this information and the dynamic energy price, it will decide the optimal management for all the loads in all the buildings.

- **MANO controller:** It is the responsible of deciding where to place the energy controller VFs, either at the IoT backend or at the IoT Gateways. To take this decision it considers the latency requirements of the energy management, the dimensionality of the optimization problem or the latency and the reliability of the communication network.
- **Access network:** It permits to communicate the energy controllers of the geo-distributed buildings. This communication is managed by the SDN controller.
- **SDN controller:** It is deployed within the WAN. It has a global view of the networking resources and network state of the WAN, the buildings' intranets and the access networks. This global view leverages on splitting the control plane and the data plane of the networking devices and by establishing a set of software interfaces with them. Thereby, global view of the network permits to establish a low latency and reliable communication. When the energy controllers are deployed at the IoT Gateways, a low latency and reliable communication among the IoT Gateways is mandatory for the convergence of the distributed optimization. When the energy controller is deployed at the IoT backend, a reliable communication is required through the core net, and depending on the load to be optimized low latency requirements can be needed as well, e.g. when optimizing the consumption of data centers.
- **Energy grid operator:** Provides information of the dynamic energy pricing.

3.2.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The energy consumption sensors and the environmental sensors trigger the energy controller periodically with new measurements. The energy controller also receives updates on the dynamic energy price and weather conditions. In the case that the MANO controller decides to implement the VF at the IoT GWs, the IoT GW performs local computations and exchange information with the rest of the energy controllers. This local computation and iteration procedure is repeated until convergence. The SDN controller manages the communication to provide a low latency and a reliable communication. The outcome of the optimization is a set of decisions on how to manage the buildings' loads, to be implemented by the actuators. The expected outcome is to minimize the overall energy consumption of a set of buildings within a dynamic pricing context and respecting the users' constraints.

3.2.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- **Information between sensor devices and the IoT Gateway:**

Energy consumption of the buildings' loads. Environmental information such as temperature, humidity, CO₂ concentration levels, at different spots of the building.

- **Information between the energy controller within the IoT Gateway and the actuators:**

This applies when the MANO controller decides the distributed energy management approach at the IoT Gateways. The exchange of information is the decision on how to manage the loads controlled by the actuators. These decisions optimize the energy consumption and respect the users' constraints.

- **Information between the energy controller within the IoT backend cloud and the actuators:**

This applies when the MANO controller decides the centralized energy management approach at the IoT backend cloud. The exchange of information is the decision on how to manage the loads controlled by the actuators. These decisions optimize the energy consumption and respect the users' constraints.

- **Information between the energy grid operator and the energy controllers:**

Provides the dynamic energy pricing.

- **Information between energy controllers of different buildings.**

Recall that, when the MANO controller decides to deploy the energy controllers' VFs at the IoT Gateways, the overall energy consumption minimization relies on a distributed optimization. This is implemented by means of a set of distributed energy controllers. The distributed optimization approach relies on local computations at each energy controller and exchange of information related to those local computations between the energy controllers.

- **Information between the IoT Gateways and the IoT backend cloud:**

Buildings' sensor measurements must be sent to IoT backend cloud when the VF energy controller is deployed within it.

3.2.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R5.1	Low latency and reliable communication between the energy controllers, when they are deployed at the IoT Gateways. This is mandatory for the iterations among the energy controllers that required in the distributed optimization.
R5.2	Interoperability between the IoT Gateways and the IoT devices (sensors/actuators), due to the heterogeneous intranets communicating them.

R5.3	Low latency and reliable communication between the sensor measurements and the IoT Gateway
R5.4	Low latency and reliable communication between the energy grid operator and the energy controllers
R5.5	Low latency and reliable communication between the energy controller VFs, within the IoT Gateway, and the actuators, which manage the loads, when the VFs are deployed at the IoT Gateways. Or between the IoT backend and the buildings' actuators, when the VFs are deployed at the IoT backend.
R5.6	Reliable communication between the IoT Gateways and the IoT backend, when the energy controller VF is deployed at the IoT backend. Moreover, low latency can be required depending on the loads to be managed, e.g. energy consumption optimization of data centers or workstations.

REFERENCES

H. Brendan McMahan et al. (2018), Federated optimization: Distributed machine learning for on-device intelligence, arXiv preprint arXiv:1610.02527. Available online: <https://arxiv.org/abs/1610.02527> (accessed on April 2018).

G. Giannakis et al. (2017), Stochastic Averaging for Constrained optimization with application to online resource allocation, *IEEE Transactions on Signal Processing*, vol 65, no. 12, pp. 3078-3093.

A. Leon-Garcia et al., (2018), OpenAMI: Software-Defined AMI Load Balancing. *IEEE Internet of Things Journal*, IEEE, vol. 5, no. 1, pp. 206-218.

3.3. Use Case 6: IoT platform interoperability in case of a power plants

3.3.1. SCOPE AND OBJECTIVES OF USE CASE

The scope of this use case has been built based on the coal power plant as an example of a system responsible for monitoring, control and optimization. The coal burning process is a subject of control in this particular exemplary system. The main aim of the use case is to extend current system architecture by creating direct connections between sensors and steering devices through IoT Gateways as well as allowing connectivity and integration with various IoT platforms and their components. SEMIoTICS framework will allow such an architecture refinement in order to provide direct IIoT sensor actuation process as well as flexibility and interoperability within the area of the IoT platforms depending on the specific business and technical needs.

This use case is designed in order to build an universal solution within SEMIoTICS framework. It will fully support SPDI patterns and it will be ready to be leveraged in any power plant ecosystem. The developed solution will support different communication protocols including OPC which is a well-established communication standard on many power plants.

The main objectives of this use case are:

1. Development of IoT platforms interoperability to enable monitoring and actuation process of IoT devices as an autonomous or semi-autonomous activity in the power plants. This objective assumes that IoT platform solution could be easily interchanged with other service supplier.
2. Development of patterns orchestration for IIoT smart objects specific for power plants
3. OPEX reduction in power plants due to the expected labor reduction

3.3.2. NARRATIVE OF USE CASE

3.3.2.1. SHORT DESCRIPTION

In the most general scenario few steps are taken:

- 1) At first user buyer buys needed sensors and/or actuators.
- 2) Then user buyer wants to provision smart things to the IoT platform, creating some system based on his needs.
- 3) Depending on the demand user buyer can connect the smart thing in direct or indirect, but always in a secure way.
- 4) After that, he wants to monitor gathering data, make actions based on them and finally optimize some parameters or process.

In this presented particular scenario sensors and actuators (1) of coal burning process are used. From ETA use case perspective (2), leveraging IoT platforms and direct connection (3) with devices (bypassing OPC client or by leveraging parallel communication) will provide added value for the power plant operators, since the data will be processed in real time allowing actuation and constant efficiency optimization (4).

The solution can elaborate a base for any system responsible for monitoring & controlling process especially generation of energy in power plant systems. Thus this use case can serve as a reference for other similar use cases. Such a system will be designed leveraging SPDI patterns in order to support interoperability at all levels especially focusing on the level of IoT platforms.

3.3.2.2. COMPLETE DESCRIPTION

As of today, ETA system is deployed in one of Polish coal power plant (ETA (η) stands for efficiency symbol in physics). It works as a typical monitoring system with dashboards and support for big data processing. The main functionality of this system is monitoring and optimizing critical combustion parameters, based on advanced algorithms. Combustion parameters mentioned above can optimize the burning process and increase the block's efficiency by several percent. Currently, the main limitation of this system is that the end user (operator) has to set changes manually based on dashboards. The second very important disadvantage is that events from devices are stored in an OPC Server and this approach requires exporting data to flat file format (CSV) via OPC client. Data processing is not smooth-flowing since data is not available online for processing and this prevents to have autonomous actuation in the system. The diagram below describes the current ETA system architecture – which is a passive mode system used for computing and visualizing optimal burning process parameters.

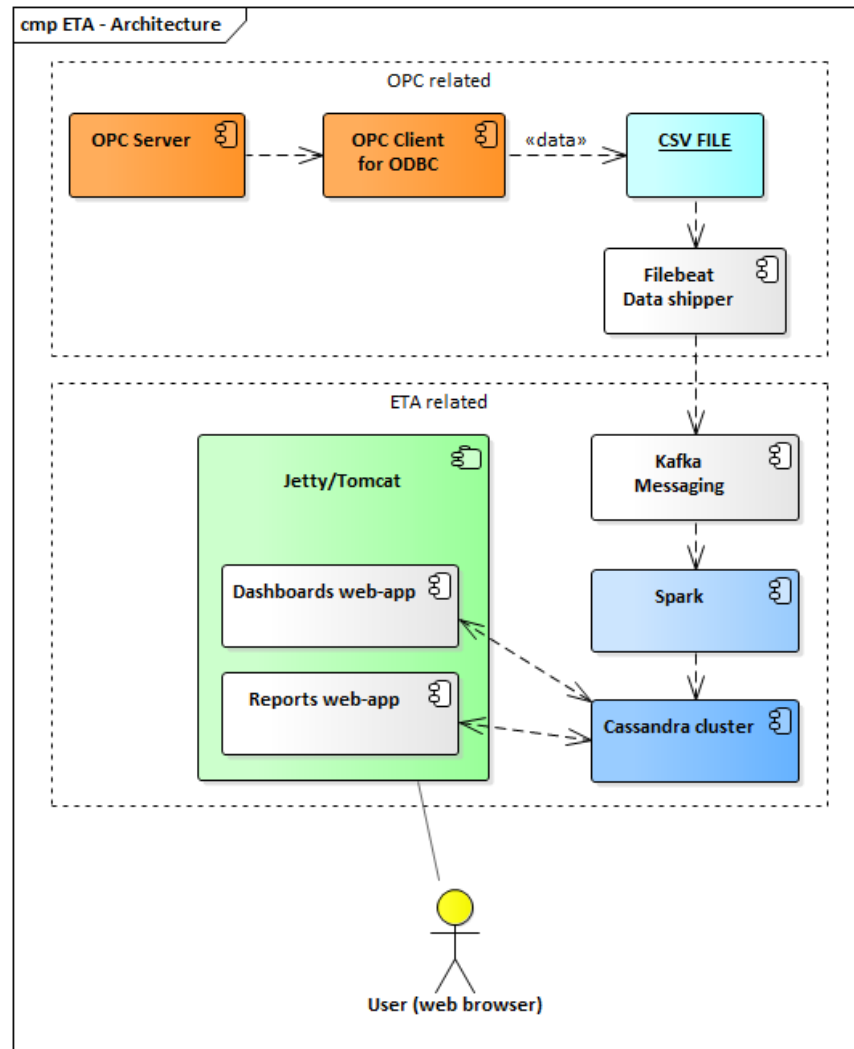


FIGURE 17: ETA USE CASE

The aim of architecture refinement is to build three communication layers between IIoT smart objects, through IoT gateway up to IoT platforms by implementing SEMIoTICS approach. Such a path will lead to development of UC in AI mode. In this approach the development of dynamically and self-adaptable monitoring of power plan efficiency will be possible and that will also allow autonomous efficiency optimization. Moreover, platform interoperability will trigger flexibility in choosing the best IoT platform components for specific business needs.

IIoT sensors installed within the power plant infrastructure will send events through an IoT gateway while the event messages will be routed to specific components of IoT platforms for further processing. Appropriate routing will be based on business rules implemented in the middleware. Every event will contain specific information e.g. temperature, generated power and other parameters required for algorithms calculation.

Such an approach will allow us to eliminate manual operator interaction and enables the actuation process as an autonomous or semi-autonomous activity while the operator will be only responsible for process supervision with no manual interaction needed.

IoT platform interoperability provided by SEMIoTICS will allow us to install key processing system modules (e.g. an advanced calculation module) in either IoT backend and/or in IoT platforms depending on the business need and expectations. Such a component will be in charge of generating calculation events to be processed in the IoT platform. It would be responsible for model learning and evolution by estimating trends and predictions. In such approach the IoT platform might be responsible for triggering appropriate actuation processes while “Advanced calculation module” would give the required input to start such dedicated processing.

Thanks to development of specific SPDI patterns, the target group of such IoT Platform use case can be any power plant. Data can be gathered from any kind of sensor, thanks to semantic interoperability. The optimization algorithms used for calculation can be easily drawn from external sources. IoT interoperability would bring critical scalability needed by specific derived deployments such as wind or solar power farms.

Building a universal systems responsible for the control and optimization of the energy production process in power plants requires the following architectural changes.:

- a) Communication between sensors, actuators and the IoT gateway. Several communication interfaces exist in the State-of-Art (SoA) for this purpose. Thereby, one of the most important IoT gateway features is to provide a flexible and reliable interface with IoT platforms, while facing the heterogeneity in the communication interfaces and patterns.
- b) Providing an interface for communication between sensors/actuators and IoT gateway. Due to the underlying application, this communication has to be reliable and resilient against security threats.
- c) Designing and deploying a solution responsible for the actuation process. This point can be fulfilled in two ways: direct communication with devices to change its mechanism or communication with the control system using a dedicated API.

Architecture:

The diagram “ETA System Overview” depicts the target system architecture based on the SEMIoTICS framework. ETA system components like actuators/sensors will be directly connected via IoT Gateway with the IoT Backend. An advanced computing module will be one of backend system applications.

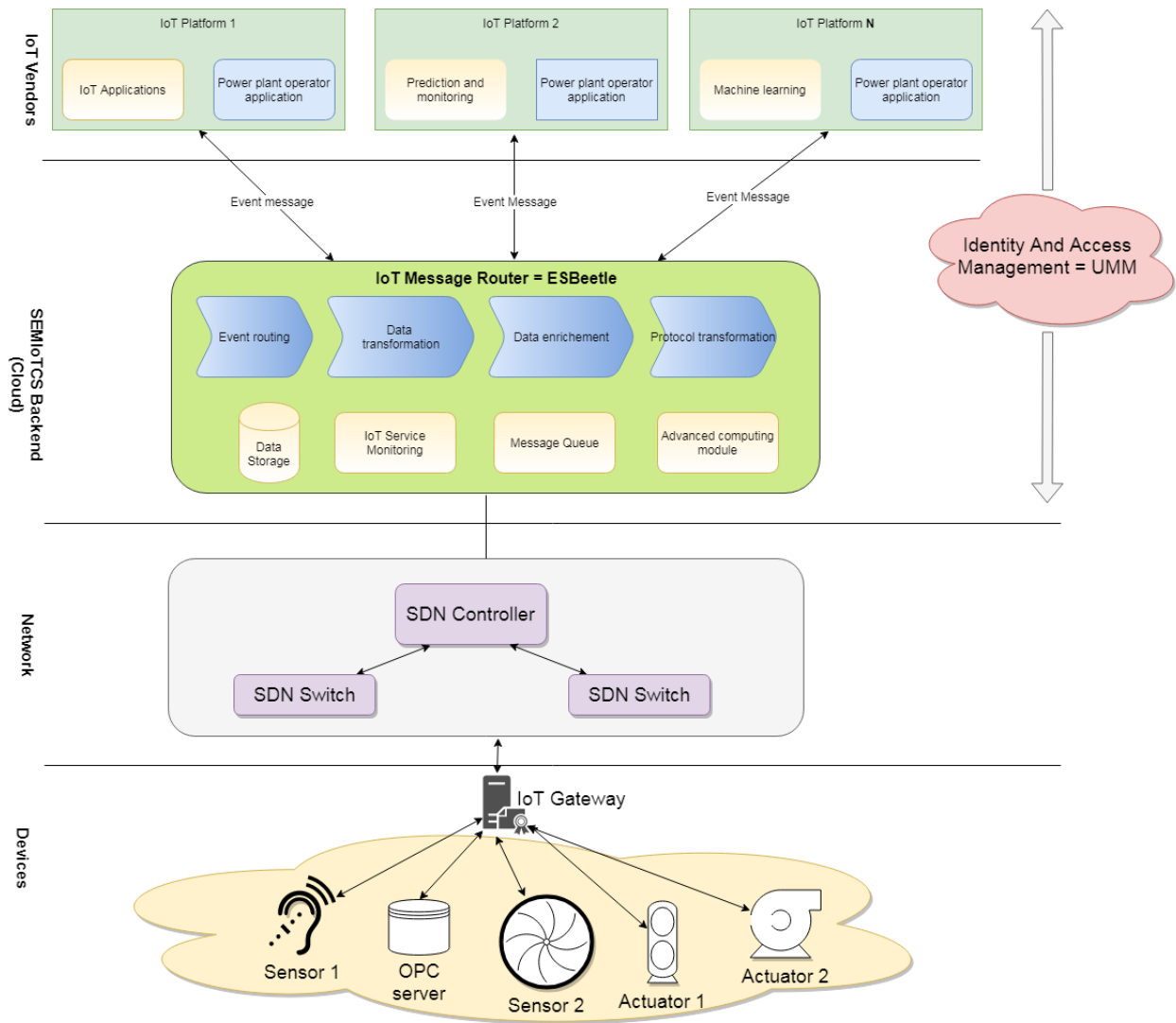


FIGURE 18: ETA SYSTEM OVERVIEW

Communication between IoT platform components

There exist several technologies for communication between sensors/actuators and IoT Gateways in the SoA of IoT systems. Nowadays the IoT scenario is characterized by a heterogeneous set of communication technologies. As an illustration, in industrial automation applications, popular protocols providing a wired communication between sensors/actuators and networking equipment include:

- SCADA (Supervisory Control And Data Acquisition) [Pyavchenko, 2017]
- Modbus [Modbustools.com. 2018]
- OPC (Open Platform Communications) [OPC Foundation, 2018]

- Optomux (which relies on RS485 serial interfaces) [Opto22.com, 2018].

For communications from the IoT gateway to the core network and IoT platform, several technologies are available such as:

- Wireless Cellular technologies: 2G, 3G, 4G [Phone Area, 2018], LoRa [leeexplore.ieee.org, 2018], Sigfox [Sigfox.com, 2018].
- Wired: DSL [Wired, 2010], Optical-based fiber (FTTX) [Lightwaveonline.com 2018].

That is, the IoT gateway manages traffic between networks that use different protocols and it is responsible for protocol translation and other interoperability tasks. Also, in this regard, the IoT gateway may carry out translation between low layer protocols used by the IoT devices and higher layer protocols provided by the Cloud. A simple example is e.g. the translation from OPC protocol implemented at sensor level to MQTT protocol available at the cloud platform.

Moreover, for the type of application at hand, it is of paramount importance to provide a reliable, efficient and flexible end-to-end communication between the sensors/actuators, IoT-Gateway and IoT Platform. To this end, recent advances on networking technology such as Software Defined Networking (SDN) technologies can be used. The idea is that a central SDN controller located at the core network receives the overall network state information. Then, through a set of software API interfaces it can configure and manage the network both to interact upwards with the IoT platforms and downwards with the access to the network layer where the sensors/actuators, intranet nodes and IoT Gateway are located. Thereby, compared to legacy systems it has a global vision of the network, allowing easier re-configurability and network resources allocation, so as to improve the reliability of the network against communication impairments and to deal with different communication requirements, e.g. latency, data rate, etc.

For instance, in the automation of power plants, SDN controller could dynamically allocate resources to find the most reliable and secure path in the end-to-end communication to meet the requirements of different types of industrial protocols such as OPC or Modbus Moreover, SDN allows for better programmability of the IoT gateway and increased interoperability between networks.

3.3.3. TECHNICAL DETAILS

3.3.3.1. DIAGRAMS OF USE CASE

In order to represent actors and system interactions more deeply, the diagram “System Interactions” depicts some more detailed system functionalities based on the IoT platforms than usual practices for use case modelling would require.

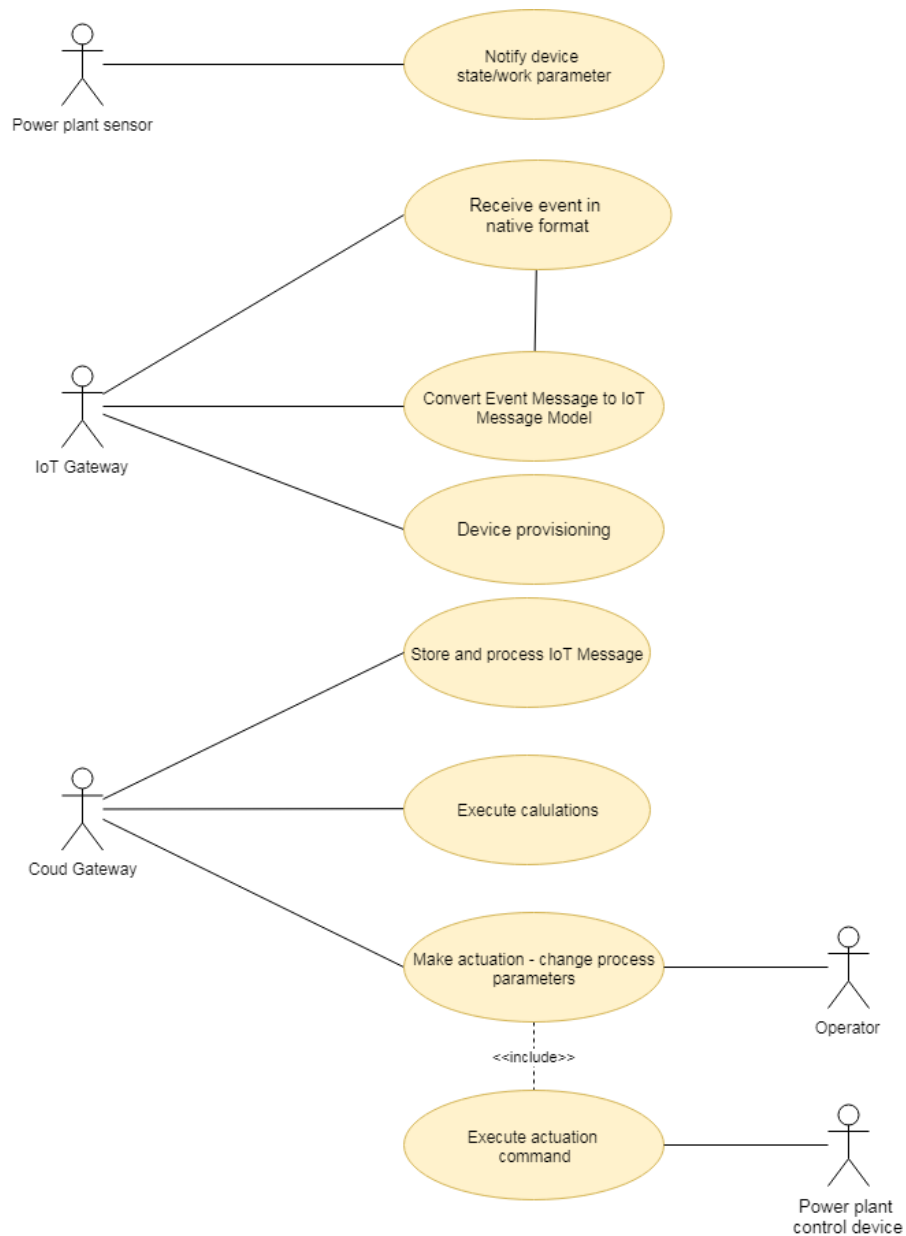


FIGURE 19: SYSTEM INTERACTIONS

Sequence diagram - data flow

This diagram depicts two stories. The first path presents common data processing with actuation as a provisioning process. The second path presents the case, when the data are not needed for the actuation processes hence are published only to the monitoring component.

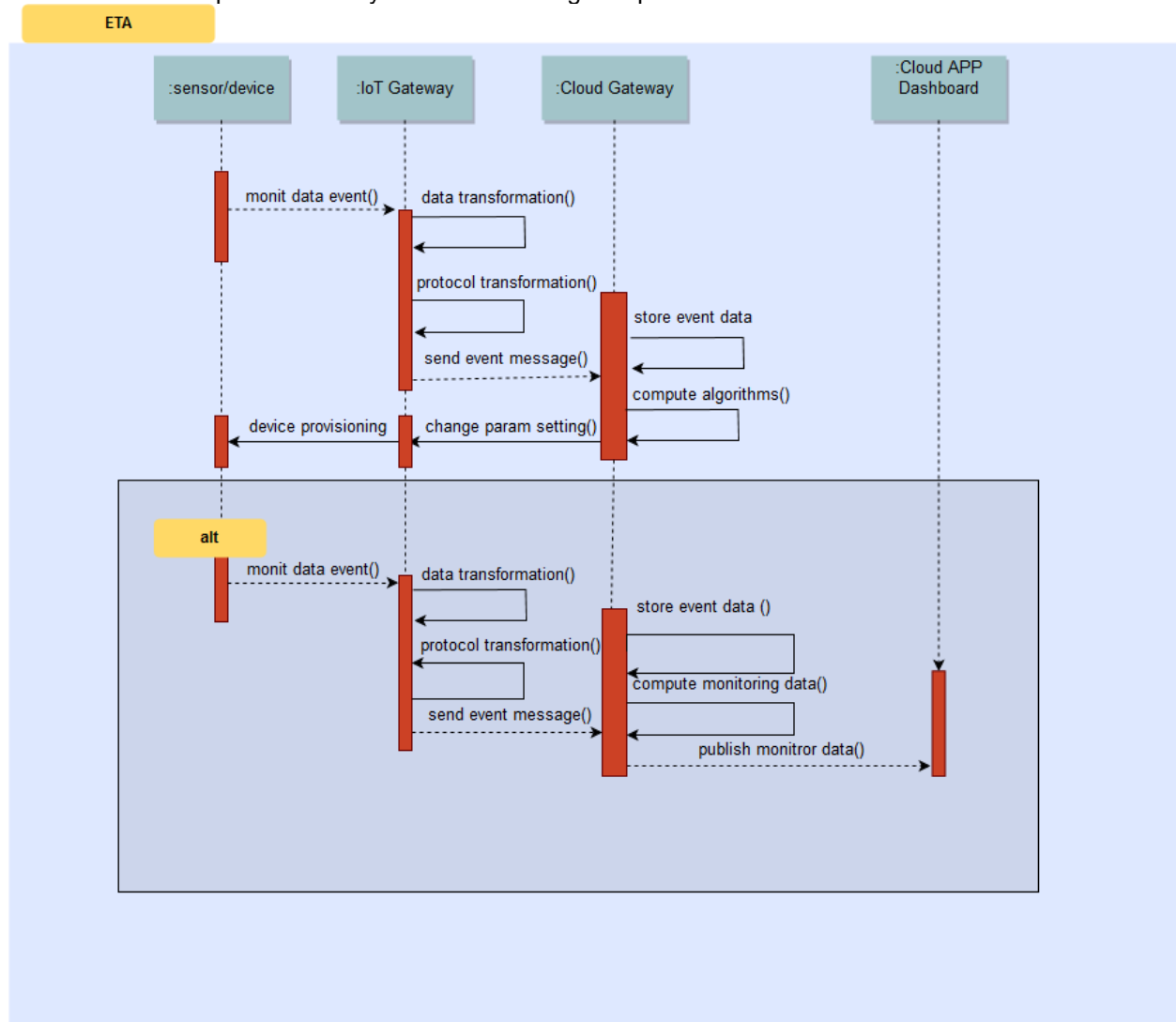


FIGURE 20: SEQUENCE DIAGRAM - EVENTS PROCESSING

3.3.3.2. ACTORS

- Power plant sensor – sensor responsible for collecting information about the combustion process and measuring the efficiency of the block
- Power plant control device – devices responsible for changing settings to control work of other devices.

- IoT Gateway – network component of SEMIoTICS infrastructure, is responsible for direct communication with IoT platform.
- Operator (human) – Power plant operator, who is responsible for the correct burning process of the plant block.
- Advanced computing module – dedicated module for data processing based on advanced algorithms and BigData.
- OPC system – component directly connected with sensors, collects and saves data for further processing.

3.3.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

Triggering Event

The triggering event for ETA platform are messages which contain the data from power plants sensors. Use case is triggered each time new data is sent to the platform.

Preconditions

- All devices like sensors and actuators are installed in system which should be supported by IoT platform
- All data gathered by sensors is sent to the IoT platform directly or indirectly
- Calculation algorithms are present in the platform backend

Triggering Event

- Event containing data from, sensors is sent to the platform
- Operator request to start actuation procedure manually

Assumption

- Platform is processing the data in real time, to satisfy the given energy needs
- During calculation data is checked for possible plant failures and abnormalities

- Calculation results are returned as new parameter values, which can be automatically send to actuators or semi-automatically send to operator who will put them into the system

Success criteria/expected outcome

The success criteria of this use case is leveraging IoT platforms and direct connection with devices (in ETA achieved by bypassing OPC client or by leveraging parallel communication) providing specific added business value for performed operators typical for the system. The main success criteria is creating universal framework for systems responsible for control and optimization of business processes like energy production in power plant.

Expected outcome is processing the data in real time and providing interfaces between all layers allowing autonomous or semi-autonomous actuation. The other important outcome is constant efficiency optimization of business specific operations and processes especially coal burning process in systems like ETA. Platform interoperability is expected to trigger flexibility in choosing the best IoT platform components for specific business needs.

3.3.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- Power plant sensor \leftrightarrow IoT Gateway – Sensor sends a message to IoT Gateway in native (OPC) format. Message payload contains information about current device state.
- IoT Gateway \leftrightarrow IoT platform (Cloud) – message forwarded from device sensor is converted to appropriate format (additionally the message content is encrypted)
- IoT platform (Cloud) \leftrightarrow IoT Gateway – the result of IoT platform data processing. Typical message will be a list of commands to execute on device.

3.3.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R6.1	IoT Gateway must be able to establish connection with devices via OPC protocol
R6.2	Connectivity between the control devices and IoT Cloud components for automated configuration purposes
R6.3	System scales even when new power plant block is monitored and managed

R6.4	Cloud applications to manage power plant can handle control process for many independent power blocks
R6.5	IoT Cloud component with advanced computing module have to generate business events for next subsequent processing. The IoT platform will decide, what changes should be done.
R6.6	Advanced computing module needs to be developed based on SPDI patterns which would support any IoT platform.
R6.7	IoT platform should provide the possibility to report list of executed actuation commands.

REFERENCES

- Pyavchenko, T. (2017). CONTROL OF TECHNOLOGICAL PROCESSES BASED ON SCADA. Polythematic Online Scientific Journal of Kuban State Agrarian University.
- Modbustools.com. (2018). Modbus Protocol. [online] Available at: <http://www.modbustools.com/modbus.html> [Accessed 18 Apr. 2018].
- OPC Foundation. (2018). What is OPC? - OPC Foundation. [online] Available at: <https://opcfoundation.org/about/what-is-opc/> [Accessed 18 Apr. 2018].
- Opto22.com. (2018). Optomux Protocol Guide. [online] Available at: http://www.opto22.com/site/documents/doc_drilldown.aspx?aid=1941 [Accessed 18 Apr. 2018].
- ieeexplore.ieee.org. (2018). Evaluation of LoRa and LoRaWAN for wireless sensor networks - IEEE Conference Publication. [online] Available at: <http://ieeexplore.ieee.org/document/7808712/> [Accessed 18 Apr. 2018].
- Sigfox.com. (2018). Sigfox - The Global Communications Service Provider for the Internet of Things (IoT). [online] Available at: <https://www.sigfox.com/en> [Accessed 18 Apr. 2018].
- Phone Arena. (2018). 1G, 2G, 3G, 4G: The evolution of wireless generations. [online] Available at: https://www.phonearena.com/news/1G-2G-3G-4G-The-evolution-of-wireless-generations_id46952 [Accessed 18 Apr. 2018].
- WIRED. (2010). DSL. [online] Available at: <https://www.wired.com/2010/02/DSL/> [Accessed 18 Apr. 2018].
- Lightwaveonline.com. (2018). FTTX. [online] Available at: <http://www.lightwaveonline.com/fttx.html> [Accessed 18 Apr. 2018].

3.4. Use Case 7: Adaptive monitoring of the smart micro-grid in a cluster of buildings

3.4.1. SCOPE AND OBJECTIVES OF USE CASE

Scope

This SEMIoTICS use case is related to the smart energy domain and applies to a cluster of buildings that form a micro-grid of energy producers (via solar panels) and energy consumers (i.e., the building loads). SEMIoTICS adaptive monitoring mechanisms will be deployed to identify patterns in the energy flow and subsequently trigger actions when certain conditions are satisfied with an ultimate goal to increase the energy efficiency of the micro-grid.

Objectives

In the proposed micro-grid use case, the energy management system of a cluster of smart buildings will allow the bi-directional flow of energy using demand-response algorithms for optimal energy utilization. To that end, the objectives of the adaptive monitoring use case are the following:

- Design an adaptive monitoring procedure that will be executed using advanced machine learning algorithms that:
 - Identify patterns on the energy consumption
 - Take action based on the identified patterns preemptively
 - Recognize changes in the patterns to re-adapt the actions accordingly
- Ensure a fast reaction and resilient operation of the energy management system based on adaptive monitoring:
 - Rapid and reliable control of energy flows within the micro-grid.
 - Guarantee a smooth transition from grid connection mode, to islanding mode, where the micro-grid consumes its own generated energy.
 - Re-adapt the actions taken by the energy management system based on pattern changes in the energy consumption of the micro-grid.
- Design demand-response algorithms and methods for the optimal utilization of energy generated by solar panels to reduce peak energy demand by 40%.
- Satisfy the timing requirements for rapid control of the micro-grids with execution of the decisions based on the adaptive monitoring at the local cloud.

3.4.2. NARRATIVE OF USE CASE

3.4.2.1. SHORT DESCRIPTION

This use case proposes an energy management system with the capability of bi-directional flow of energy among clusters of urban buildings that act both as energy producers (via solar cells) and energy consumers (via building loads). SEMIoTICS adaptive monitoring mechanisms are employed

to keep track of real-time energy generation and consumption, to ensure efficient energy allocation based on the needs of the buildings while reducing peak energy consumption, and thus, decreasing the cost to the end user and promoting energy self-sufficiency. The adaptive behaviour of SEMIoTICS will enable a rapid and proactive reaction to energy generation and consumption patterns, optimizing energy utilization and reducing peak demand by as much as 40%.

3.4.2.2. COMPLETE DESCRIPTION

As environmental issues are constantly in the limelight during the last decade, the use of IoT and massive machine-type communication (mMTC) for efficient energy management has attracted a lot of attention and many ideas have been proposed that could transform the cities of the near future. One of these ideas is the concept of smart micro-grids, in which dense metropolitan areas are divided in multiple clusters of buildings, to address their energy needs collectively. The bi-directional flow of energy among energy producers (through solar cells) and energy consumers (building loads) can be implemented with SEMIoTICS IoT based inverters, as well as IoT monitoring devices (i.e., smart meters) at the Field layer. SEMIoTICS IoT nodes are responsible for actuation and adaptive real-time monitoring of energy generated and consumed. The smart meters are monitoring several energy-related parameters, such as the consumed energy per building load, energy generated per solar panel, as well as environmental conditions. All IoT nodes transmit the monitored parameters to a building-wide SEMIoTICS IoT Gateway. Then, the IoT gateway delivers the data to the energy management system which is implemented as a collection of services that run at the SEMIoTICS backend (i.e., private and public clouds). The energy management system is responsible for processing building data and implementing the adaptive behavior. More specifically, the energy management system is where the machine learning services are implemented in order to identify certain patterns that control the bi-directional flow of energy via the actuating IoT-based inverters. Also, using adaptive demand response algorithms, it is able to reduce the peak load. Hence, using all the collected data from the smart meters, the energy management system is able to calculate various important metrics, such as:

- Times of day that certain loads are expected to be active
- The amount of excessive harvested energy at all buildings that can be used at another building with higher energy requirements,
- Energy-related patterns that could be useful for preventing black-outs at the micro-grid
- Cross-compare the expected and measured energy generation per solar panel, identifying problems (e.g., ageing panels, shading, or inappropriate angles)

Moreover, in order to decrease even further the time required to trigger certain actions, a local cloud (i.e., mobile edge computing (MEC)) is employed at each micro-grid. To be more specific, when the energy management system identifies a pattern by analyzing the data from the smart meters, it populates a list of actions at the local cloud. Then, if a pattern in this list is recognized by the local cloud, it immediately executes an action, decreasing the execution time of the action significantly. Finally, the energy management system continuously analyzes data from smart meters to identify patterns that trigger energy-saving actions or unexpected behavior and adapts the trigger list accordingly. In this way, the system can also identify deficiencies or failures to apply certain actions, thus proactively correcting future problems.

3.4.3. TECHNICAL DETAILS

3.4.3.1. DIAGRAMS OF USE CASE

In the following figure, we demonstrate a general plan of the micro-grid architecture. As it can be seen, each building is equipped with a certain number of smart meters that act as the IoT nodes. The smart meters communicate with the IoT gateway to deliver their monitoring data, which are then forwarded to the energy management system (i.e., public cloud) via the internet.

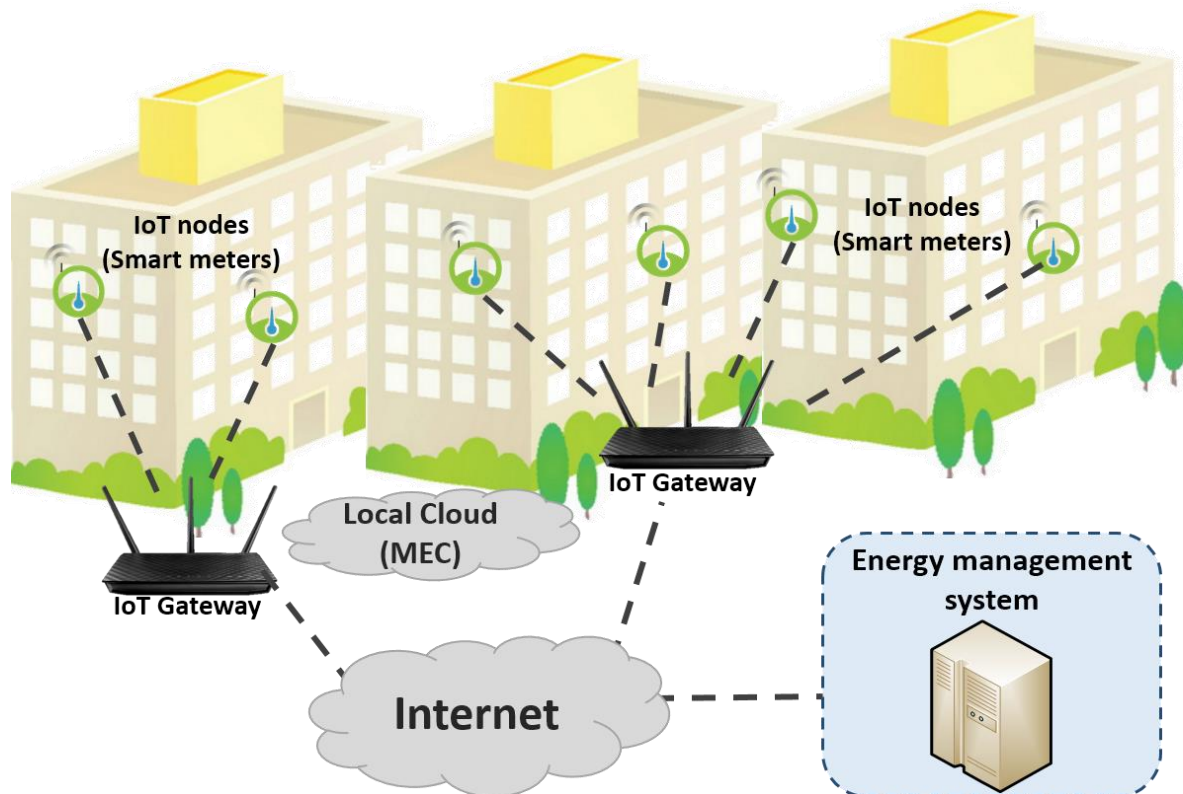


FIGURE 21: DIAGRAM OF THE MICRO-GRID

While the monitoring data are delivered at the energy management system, certain patterns are being identified at the public cloud that trigger actions to benefit the residents and handle the power consumption and generation. In Figure 22, we present the flow of data and actions from the buildings to the public cloud and vice versa. Moreover, we can notice that the local cloud communicates both with the IoT gateways and with the public cloud, as it receives the updated actions list from the public cloud to apply them to the buildings and, thus, ensure a fast and reliable adaptive monitoring operation.

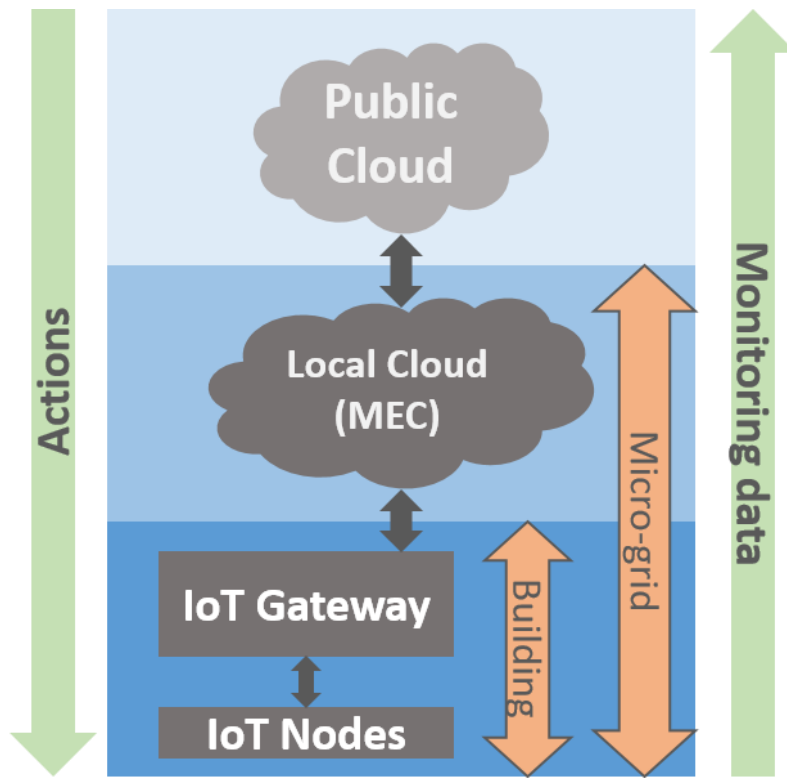


FIGURE 22: DIAGRAM OF THE ADAPTIVE MONITORING USE CASE

Next, in Figure 23, we present a sequence diagram for the adaptive monitoring use case that describes the sequence of events once the smart meters begin forwarding their monitoring data to the IoT gateways. Upon reception of the data at the public cloud, patterns are identified, and actions are forwarded at the micro-grid that provide incentives at the residents or allocated the energy appropriately to increase the micro-grid efficiency.

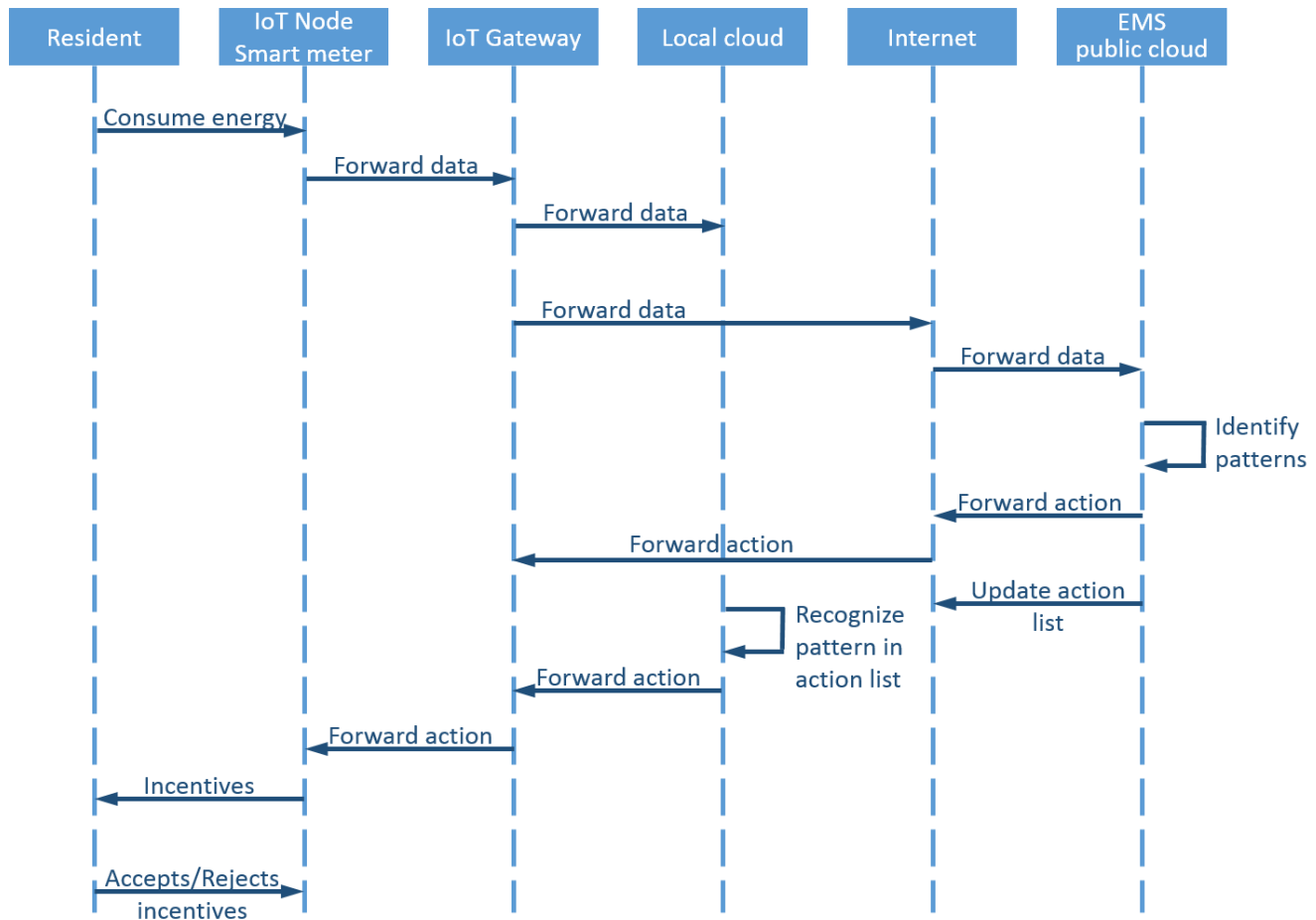


FIGURE 23: SEQUENCE DIAGRAM FOR ADAPTIVE MONITORING

3.4.3.2. ACTORS

The main actors in this use case are the following:

- **Micro-grid:** The micro-grid involves all the local actors of the use case and consists of a cluster of buildings. Each micro-grid should be energy self-sufficient and it controls the energy flow among the different buildings using adaptive demand response algorithms.
- **Building:** Each building of the micro-grid consists of several apartments with a smart meter installed at each one. For each building, there is a SEMIoTICS IoT gateway that collects the energy consumption data from all apartments and reports them to the energy management system.
- **End user (resident):** Each building is inhabited by several end users. They are responsible for reducing their peak demand from the system, as they receive incentives from the adaptive monitoring system.
- **Smart meter (SEMIoTICS Field devices):** A smart meter is able to identify the current loads in an apartment and report it to the gateway.

- **Energy management system (Public cloud):** This actor is the most important for the adaptive monitoring. It receives all the energy-related metrics and information from the smart meters and, using machine learning, identifies patterns that trigger certain actions. In this way, it is possible to allocate the energy among the buildings in an optimal way and reduce peak demand.
- **Local cloud (MEC):** When the actions are determined by the energy management system, a local cloud located near buildings is responsible for monitoring the energy-related metrics and deciding whether a specific pattern triggers an action. In this way, it is possible to achieve a rapid and reliable operation.
- **IoT Gateways:** The gateways are responsible for the collection of the data from multiple smart meters and their delivery to the energy management system (public cloud) and the local cloud.
- **Network Operator:** The network operator is responsible for the successful delivery of the monitoring data from the field devices to the public cloud through the Internet.

3.4.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The main event that is examined in this use case is the automatic action of the system using adaptive monitoring to increase the efficiency in the energy consumption and to trigger actions when certain conditions are met.

Preconditions

As smart meters start to log energy consumption and other energy-related data, the energy management system actively processes this data to identify patterns.

Triggering event

When certain patterns have been established that require appropriate actions to take place, such as redirecting energy flow from a building with excess energy, the energy management system updates a list of actions at the local cloud. This list describes the necessary triggering requirements that have to occur in order for an action to take place. For instance, one action could be to suggest incentives at the end user, or to allocate the excessive harvested energy from one building to another in order to reduce the energy consumption and increase the efficiency of the micro-grid.

Assumptions

The energy management system continuously identifies changes in these patterns, in order to update the trigger list based on differences in the recurring data.

The system can also identify unexpected behavior or failures to apply certain actions, to proactively avoid future problems.

Success Criteria/expected outcome

The expected outcome for the adaptive monitoring use case is to successfully identify patterns that trigger energy-saving and cost-effective actions that reduce peak demand and promote energy self-sufficiency of building clusters.

3.4.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- **Between the Smart meter and the Resident** – The smart meter notifies the resident regarding incentives that have been defined through the demand response algorithms that target the reduction of the peak demand. The resident chooses to follow or disregard the incentives.
- **Between the Smart meter and the Gateway** – The smart meter collects power consumption data and delivers them to the gateway.
- **Between the Local cloud and the Gateway** – The data from smart meters are delivered to the local cloud through the gateway for rapid decision/identification of patterns that have been already established by the energy management system.
- **Between the Energy management system and the Smart meters** – The energy management system logs every information that receives from all smart meters through the communication network in order to identify patterns in the energy consumption and other energy-related metrics, and defines the actions that have to be proactively taken to achieve an optimal operation.
- **Between the Local cloud and the Energy management system** – The local cloud is close to the micro-grid to offer rapid decision making in the monitoring process. To that end, whenever the energy management system identifies an introduction or changes on certain patterns in the energy data, it updates the list of triggers at the local cloud.
- **Among different buildings of the same micro-grid** – The amount of energy generated from solar panels in the micro-grid is constantly monitored by the energy management system to ensure efficient energy allocation and peak demand reduction.

3.4.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R7.1	Ensure high connectivity probability (>99%) among all the communication network devices
R7.2	The local cloud should be able to recognize the established patterns and act in less than 10ms
R7.3	The energy management system has to be able to handle data from a massive collection of smart meters, i.e. several micro-grids employ the same energy management system, and be able to scale when new micro-grids are included in the network
R7.4	The demand response algorithms have to decrease the energy consumption of the micro-grid by 40%
R7.5	The adaptive monitoring should be able to identify changes in the patterns and readapt the established actions in a timely manner
R7.6	The energy management system should monitor and guarantee a smooth transition from grid connection mode, to islanding mode, where the micro-grid consumes its own generated energy
R7.7	When there are discrepancies in the load of different buildings, the energy management system should guarantee the flow of energy from the low-load buildings to the high-load buildings to achieve load balancing

3.5. Use Case 8: Machine learning and edge analytics for smart cities

3.5.1. SCOPE AND OBJECTIVES OF USE CASE

The proposed vertical use case describes a framework for a monitoring service within a smart city infrastructure towards enhancing safety, efficiency and maintenance of a vital part of the smart city. Specifically, we focus on a smart pipeline monitoring IoT-based system exploiting an AI embedded sensing platform.

3.5.2. NARRATIVE OF USE CASE

3.5.2.1. SHORT DESCRIPTION

The use case focuses on a monitoring/prediction solution that can be implemented in critical smart city infrastructures, such as pipeline networks. For this solution, a set of sensors that belong to an AI embedded sensing platform is employed, to collect various qualitative characteristics of the pipeline or any other type of infrastructure (Hatzivasilis et al., 2017a). The sensors are able to communicate with each other as well as with the IoT gateway, which in turn communicates with the cloud/data center. Moreover, the collected information is leveraged to detect possible threats leading to failure, leakage, and corrosion of various infrastructure parts.

3.5.2.2. COMPLETE DESCRIPTION

IoT technology is highly disseminated and used in a variety of everyday life tasks during the last years. Smart cities constitute a popular application field, where IoT infrastructures can be efficiently applied to promote quality of life, security, optimized resource management, etc. (Fysarakis et al., 2016; Hatzivasilis et al. 2017c). A vital smart city building block corresponds to management, monitoring or maintenance of critical infrastructures such as traffic light management systems, camera surveillance systems, air quality monitoring, integrity of engineering structures, pipeline monitoring etc. In the current use case, we assume that the focus is given on a smart pipeline monitoring/prediction system. However, it is important to notice that the use of the described architecture is not only limited to smart pipeline monitoring/prediction but could be naturally extended and adapted to other smart city IoT applications.

Smart pipeline monitoring/prediction is heavily based on the AI embedded sensing platform used in the field (layer) to sense the physical world, obtain the raw signals and thus, collecting the data in a parallel fashion. Specifically, the AI embedded sensing IoT platform is composed by a set of sensors, micro controllers and a coordinating gateway. The gateway acts as a supervisor controlling sensing units and triggering their self-learning, self-adaptive, self-configuration and self-management capabilities. The platform proposes a sophisticated design methodology imposing a hierarchical, naturally distributed, thus scalable, processing chain.

Towards this twofold direction (of monitoring and prediction), the set of AI sensing units can be placed in the specific locations of the pipeline infrastructure measuring a wide range of qualitative characteristics such as temperature, pressure, velocity of the flow along the pipeline, pipeline orientation, humidity etc. The structural integrity of the pipeline network can be reflected in the aforementioned parameters. Structural integrity could be compromised by aging, sudden environmental conditions (e.g., storms, floods, fires), construction failure etc. Those threats can lead to corrosion, leakage and failure of pipelines resulting in serious environmental and economic consequences affecting wide areas of the city and its inhabitants.

During monitoring, raw/physical data are collected and processed (by each sensing unit) in a real-time fashion. It should be highlighted that monitoring is strongly related with event detection and thus, online machine learning algorithms can be applied in order to identify specific abnormal patterns within the streaming data. Moreover, feature extraction and feature selection can be applied in each sensing unit (in a distributed manner) to reveal the hidden inter-sensor data information structure. The estimated features could then be transmitted to the gateway for further processing, i.e., performing event detection. Several off-the-shelf approaches could be adopted for event or anomaly detection such as Hidden Markov Models (HMMs), autoencoders, Long Short-Term Memory (LSTM) autoencoders, k-NN (nearest neighbor), Local Outlier Factor (LOF), etc. Besides, data fusion-based event detection algorithms could be devised to achieve higher detection rates by exploiting multisensory data fusion.

The analyzed data sent to the gateway can be further transmitted into the cloud or the data center for prediction purposes. More specific, we need great amount of storage for data aggregation and thus, the cloud or data center can be efficiently used for (offline) streaming data storage. The aggregated data can be exploited for long-term prediction towards identifying specific time of failures, estimating the parts of the infrastructure that are about to malfunction and in general ensuring that potential problems are discovered well before they actually occur. Prediction of critical events can be performed using decision trees, support vector machines (SVMs), deep learning and other machine learning time-series forecasting/prediction techniques.

It is obvious from the description above that monitoring is taking place at the edge of the IoT system related with online, real-time detection, while prediction is related to short or long-term estimation of critical events. The extracted special data transferred from the lower edge level to the upper cloud/data

center level can be used for prediction. However, continuous data aggregation can be exploited to retrain the predictor and an updated version of the monitoring algorithm can be obtained by sending down to the gateway data computed during the prediction procedure. As a result, this information from the upper level can be used to obtain an updated robust monitoring version at the edge level, since we can determine what type of sensed data is needed to be acted on quickly and what can be analyzed at a later time. In some cases, it may be determined that certain data points are not worth capturing and analyzing at all, particularly in instances where enormous amounts of data are being generated.

3.5.3. TECHNICAL DETAILS

3.5.3.1. DIAGRAMS OF USE CASE

The following figure depicts the application of machine learning in the smart pipeline monitoring/prediction task. The critical smart city infrastructure is monitored through an AI embedded sensing platform. The collected data are processed locally and then transmitted to the upper IoT architecture level through the IoT gateway. Next, the data flow can be transmitted via the network to the next upper level which is the data center capable of further analyzing the acquired information. Finally, the stakeholders can update the policy accordingly if needed and disseminate it in a top-bottom fashion through the IoT system. More specific, stakeholders such as IoT vendors or smart city service providers can exploit the processed data in order e.g. to replace failed or malfunctioned IoT equipment which is critical for the smart city monitoring/prediction capabilities.

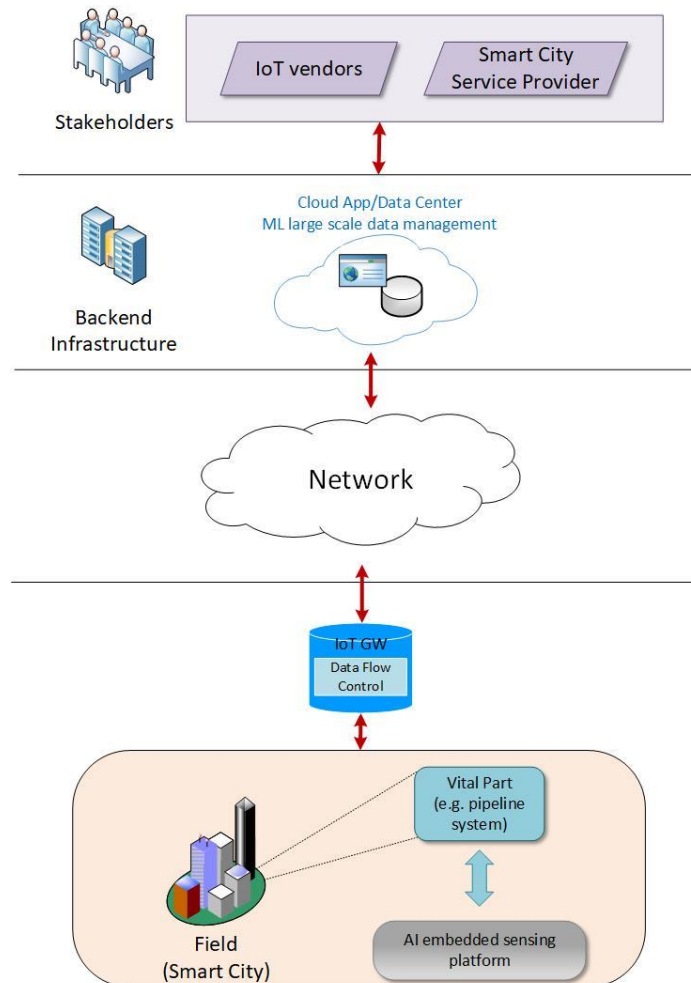


FIGURE 24: SMART CITY - SMART PIPELINE MONITORING

The next figure describes the sequence of events for monitoring and event detection response (e.g. failure, leakage etc.). The system detects and counters the incident automatically.

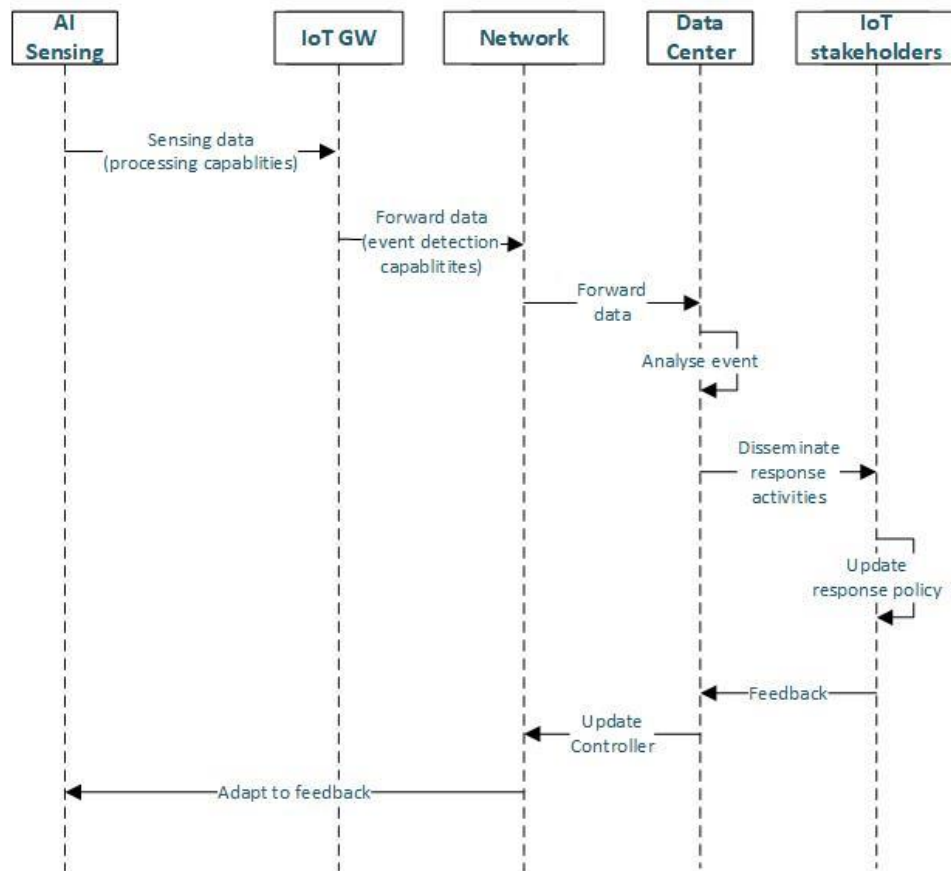


FIGURE 25 SEQUENCE DIAGRAM FOR MONITORING/PREDICTION AND EVENT DETECTION RESPONSE

3.5.3.2. ACTORS

1. **AI IoT sensing units:** sense the physical world and obtain raw data. Based on the streaming raw data, feature extraction and feature selection is performed.
2. **AI IoT gateway:** receives features from sensing units for event detection. According to the detected event send the appropriate flag to the lower level. It also transmits the obtained data to the cloud level for prediction.
3. **Cloud/data center:** aggregate data from lower level. Design prediction process for pipeline maintenance or estimating failure time etc. Send information to the stakeholders for policy adaptation. Send also information to the gateway for monitoring process adaptation.
4. **IoT stakeholders:** offer IoT services for the smart city pipeline monitoring/prediction. The service types include: i) resource management, ii) physical safety and disaster mitigation, iii) policy dissemination.

3.5.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

The AI sensing units are triggered and collect (periodically or not depending on the sensing unit type) new measurements corresponding to the qualitative as well as quantitative characteristics of the smart city infrastructure. The collected measurements can be either locally processed or

transmitted to the IoT gateway performing further computations and exchanging information with the rest of the AI sensing units. The outcome of the local computations is a decision on how to act in case of a detected event. In some cases, the decision making is going to be affected by the feedback through the network from the cloud/data center as a result of a long-term monitoring and prediction procedure of the whole system. The expected long-term outcome is to minimize the risk of a smart city infrastructure failure and achieving the unobstructed operation of the infrastructure.

3.5.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- AI IoT sensing unit <> AI IoT gateway – Sensor sends message (extracted/selected features) to IoT gateway in appropriate format.
- AI IoT gateway <> Network <> Cloud/Data Center – Message forwarded (through the network infrastructure) from device sensor is converted to appropriate format (additional in message content is encrypted).
- Cloud/Data Center <> IoT Stakeholders – Disseminate information corresponding to the processed cloud data to the IoT Stakeholders for policy update.
- IoT Stakeholders <> Cloud/Data Center – Updated policy (related with smart city monitoring/prediction) provided as feedback to the data center in order to update the algorithmic parameters accordingly.
- Cloud/Data Center <> Network <> AI IoT gateway – The result of cloud data processing. Typical message will be a list of commands to execute on device.

3.5.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R8.1	IoT sensing unit shall be able to interface to the physical world
R8.2	IoT sensing unit shall be able to interface to the IoT gateway in order to coordinate with it
R8.3	IoT sensing unit shall be able to perform distributed feature extraction/selection from the acquired data
R8.4	IoT gateway shall be able to estimate abnormal detection based on (un)-supervised model
R8.5	IoT gateway shall be able aggregate relevant events (i.e. changes) coming from whichever of connected IoT sensing units deciding if they are global or local changes
R8.6	IoT gateway shall have the capability to exchange relevant information between itself, the cloud and the sensing units with some connectivity capabilities
R8.7	Cloud shall be able to aggregate features/data coming from the edge level through IoT gateway
R8.8	Cloud shall be able to perform prediction based on the aggregated data based on a supervised model

R8.9	Cloud shall be able to send updated model data/parameters to the IoT gateway
R8.10	IoT gateway shall be able to update the monitoring model based on the information coming from the cloud

REFERENCES

- Doraswamy, N. and Harkins, D., 2003. IPsec: the new security standard for the Internet, intranets, and virtual private networks. *Prentice Hall Professional*, 2nd edition, pp. 1-262.
- Durumeric, Z., Kasten, J., Bailey, M. and Halderman, J. A., 2013. Analysis of the HTTPS certificate ecosystem. *Internet Measurement Conference (IMC)*, ACM, Barcelona, Spain, October 23-25, pp. 291-304.
- Freier, A., Karlton, P. and Kocher, P., 2011. The Secure Sockets Layer (SSL) protocol version 3.0. *Internet Engineering Task Force (IETF)*, RFC6101, August 2011. Available on-line: <https://tools.ietf.org/html/rfc6101?ref=driverlayer.com>
- Fysarakis, K., Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C. 2016. RtVMF – A secure Real-time Vehicle Management Framework with critical incident response. *IEEE Pervasive Computing Magazine (PVC) – Special Issue on Smart Vehicle Spaces*, IEEE, vol. 15, issue 1, pp. 22-30.
- Gupta, U., 2015. Survey on security issues in file management in cloud computing environment. *Cryptography and Security*, arXiv:1505.00729, pp. 1-5.
- Hatzivasilis, G., Papaefstathiou, I., Plexousakis, D., Manifavas, C. and Papadakis, N., 2017a. AmbiSPDM: managing embedded systems in ambient environments and disaster mitigation planning. *Applied Intelligence*, Springer, pp. 1-21.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2017b. SCOTRES: secure routing for IoT and CPS. *IEEE Internet of Things (IoT) Journal*, IEEE, vol. 4, issue 6, pp. 2129-2141.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2017c. Real-time management of railway CPS, 5th *EUROMICRO/IEEE Workshop on Embedded and Cyber-Physical Systems (ECYPS 2017)*, IEEE, Bar, Montenegro, 11-15 June.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2016. Software security, privacy and dependability: metrics and measurement. *IEEE Software*, IEEE, vol. 33, issue 4, pp. 46-54.
- Wesinger, R. and Coley C., 2007. *US Application*, US20070101421A1, May 2007. Available on-line: <https://patents.google.com/patent/US20070101421A1/en>
- Xia, L., Chao-sheng, F., Ding, Y. and Can, W., 2010. Design of secure FTP system. *International Conference on Communications, Circuits and Systems (ICCCAS)*, IEEE, Chengdu, China, July 28-30, pp. 270-273.

3.6. Use Case 9: Security and privacy enhanced smart wearables

3.6.1. SCOPE AND OBJECTIVES OF USE CASE

This use case connects smart wearable devices with apps running in the cloud. As wearable devices collect very sensitive data, privacy and security are particularly important. Thus, this use case operates on two layers: field devices and the cloud layer; it has the following objectives:

- Allow sensor¹⁴ owners to define fine-grained privacy policy specifying which apps may use data in which ways
- Provide dynamic analysis on cloud platforms to enforce policies
- Prevent malicious apps from misusing data

3.6.2. NARRATIVE OF USE CASE

3.6.2.1. SHORT DESCRIPTION:

Wearable sensors can generate data which are both very useful and also very privacy-sensitive, for example heartbeat rate for an athlete or acceleration data to detect falls and accidents. Therefore, it is essential to have strong mechanisms to protect users' privacy, also to meet legal requirements, but it is nevertheless desirable to allow various apps to make use of the sensor data. Therefore, this use case combines fine-grained privacy policies with dynamic analysis to monitor apps and to enforce policies in the cloud.

3.6.2.2. COMPLETE DESCRIPTION

This use case aims to provide better security (as compared to legacy systems) for two groups of users: First, users, who wear wearable sensors, achieve more control over their personal data by supporting fine-grained policies. The second group are the users of apps creating value out of this data. In legacy systems, here either privacy, i.e. control over sensor data, would be lost, or the number of apps which can be used would be limited to a set of predefined apps. As we will explain below, with dynamic monitoring of apps, we can achieve both good data privacy and allow arbitrary apps.

When a user deploys a new wearable sensor, he first has to define a privacy policy specifying how the data generated by the sensor may be used. This can include: limits on data retention period, limits on users who may access data, anonymization of data, etc. The sensor will then attach this policy to all data it sends to the cloud platform. In this step the sensor also needs to be connected to the gateway; this includes identification of the sensor. A sensor can either identify itself with an ID entered by an authorized user, or using securely stored credentials from a security module. If the sensor successfully authenticated itself towards the gateway, it can then exchange keys for encryption with the cloud platform.

¹⁴ When we speak of sensors, this equally applies to actuators. However, for the sake of brevity we only write "sensor" instead of "sensor or actuator".

The cloud platform has three main tasks: it receives data from sensors, and it runs apps installed by users. In addition, it is responsible for enforcing security and privacy policies. The cloud platform receives the privacy policies associated with sensor data together with this sensor data; furthermore, the owner of the cloud platform can also define additional security policies for this cloud platform. This means that cloud platform generally receives all data, or can request it from the sensor; only at level of the cloud platform the decision is then taken who may or may not access the data. This also implies that the cloud platform must be fully trusted by all actors in this use case, as there is no control against a cloud platform acting maliciously. Should a policy be violated, the platform owner will be notified, and the app be stopped. The dynamic analysis of the cloud platform will also support more complex mechanisms such as a finite state machine to model benign or malicious behavior of apps. On a simpler level the cloud platform can also deploy whitelisting of known benign apps to grant them more privileges, whereas unknown apps are monitored more closely.

The next type of actor in this use case is an app developer. We assume that we can only achieve the full benefit of wearable sensors and actuators with open platforms supporting third-party apps. An example of such an app distribution platform is the Google Play Store. Of course, such an app store should already run security checks which are beyond the scope of this use case. However, as demonstrated against the Google Bouncer, these checks can always be detected and therefore also circumvented. Thus, in this use case, this also leads to the threat of malicious apps misusing sensitive data. Therefore we propose to add monitoring using dynamic analysis to create two more benefits: First, at the app store level, only a general security policy can be enforced. On the cloud platform, we can enforce the policy specific to an item of sensitive data. Second, we can dynamically analyze the code actually being executed, i.e. the analysis cannot be circumvented anymore.

The last type of actor are the end users of apps. They can install apps using the cloud platform to make use of the data generated by the wearable devices. The users may or may not be the same as the sensor owners; for example, in the case of a sensor monitoring the heartbeat rate of an athlete, it may be the athlete himself, but it may also be a coach analyzing the performance or a researcher interpreting anonymized data of many athletes.

3.6.3. TECHNICAL DETAILS

3.6.3.1. DIAGRAMS OF USE CASE

Figure 1 shows the main stakeholders (blue) and how they interact with the different systems (green), mainly cloud platform and sensor.

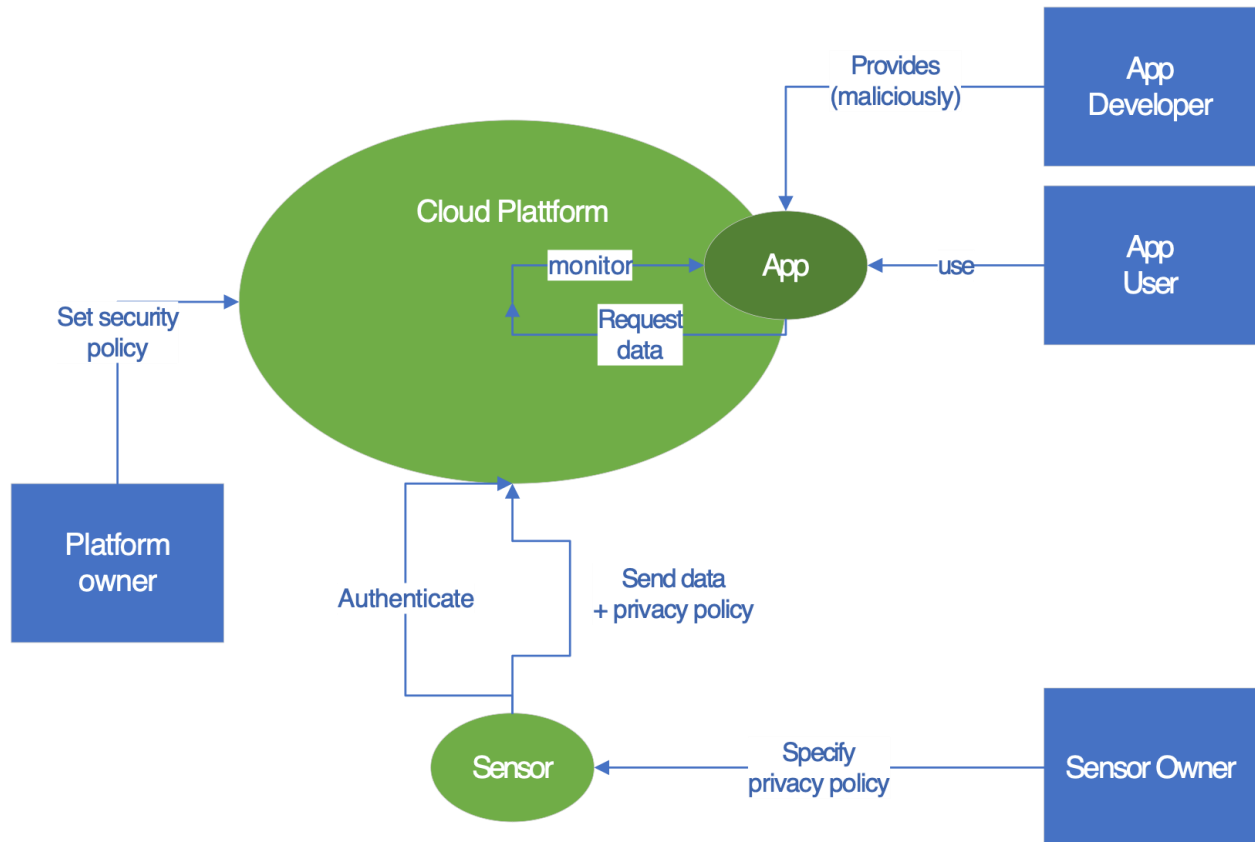


FIGURE 26: STAKEHOLDERS (BLUE) AND THEIR RELATIONSHIPS TO ENTITIES (GREEN)

3.6.3.2. ACTORS

- **Sensor owner:** The sensor owner is responsible for adjusting the privacy policy of a sensor during setup. The sensor owner can also be the user of the sensor, but this is not always the case. For example, in a smart health scenario monitoring a person suffering from dementia, the sensor user may not be able to configure the sensor so that a separate sensor owner is needed for this task.
- **Cloud platform owner:** The cloud platform owner runs and manages cloud platform. He can also define security policies governing which behavior of apps is admissible. In case of policy violations, the platform owner will be notified of the violation. In this context, the cloud platform refers to the software environment. The hardware provide may be a different actor; however, the hardware provider does not play an active role in this use case, and therefore is not discussed further.
- **App user:** The app user can install and run apps on the cloud platform to generate smart behavior from combining various sensors' data. He can install apps from different sources, e.g. app stores; however, he may not be able to distinguish a benign from a malicious app. Thus, this use case considers the scenario of malicious apps being installed on the cloud platform.
- **App developer:** The app developer provides apps to app stores for the cloud platform. For their own self-centered benefit, some app developers may also provide malicious apps.

3.6.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/EXPECTED OUTCOME

In this use case, we assume that cloud platforms are already configured and running, i.e. users can execute apps there. We also assume network connectivity between sensors and cloud platform to be available. This may include networks using NFV or SDN technology, but that is transparent to this use case.

Triggering Event: On the field device level, this use case is triggered when a new sensor is added to the system. On the cloud platform level, this use case is triggered when apps are being executed and monitored.

Assumptions:

- Cloud platform can be trusted.
- Ownership of sensors is permanent, i.e. the owners of sensors do not change.
- Sensor can connect to cloud platform identified by a URL.
- There are malicious apps which legitimate users may want to install

Success criteria/expected outcome: This use case is successful if malicious apps on the cloud platform, i.e. apps which violate platform security policies and/or sensor privacy policies, are detected before they can cause any damage with their harmful behavior. In other words, this use case is successful if users can install apps from arbitrary sources without lowering the overall level of security and privacy.

3.6.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

- **User – sensor:** The user defines and enters into the sensor the privacy policy how the sensor data may be used and by whom.
- **Sensor – cloud platform:** The sensor provides the data it senses to the cloud platform, together with the privacy policy. All communication between sensor and cloud platform is encrypted.
- **App developer – cloud platform:** The cloud platform notifies the app developer if one of his apps was stopped on the platform due to a policy violation.
- **Cloud platform – platform owner:** The cloud platform notifies its platform if a violation of the security policy occurred.

- App user – cloud platform: The cloud platform notifies the user if one of his apps was stopped on the platform due to a policy violation, together with an explanation which sensitive data may possibly have been affected.

3.6.3.5. REQUIREMENTS DESCRIPTION

Req.-ID	Description
R9.1	Platforms, e.g. cloud platform and sensor, SHALL be trusted.
R9.2	Sensors SHALL to be identifiable, i.e. they either need a TPM module/smartcard, or they need a user interface.
R9.3	Sensors need to be able to encrypt the data they generate, i.e. their CPU and memory need to be sufficient to perform these cryptographic operations.
R9.4	The cloud platform needs to be able to monitor the execution of an app, in particular its interactions with other apps, the network interface, and APIs.

3.7. Use Case 10: Semantics in IIoT engineering and orchestration in industry automation systems

3.7.1. SCOPE AND OBJECTIVES OF USE CASE

The objective of this use case is to derive requirements related to the provision of *semantic interoperability* in IIoT applications, which are in the scope of SEMIoTICS project.

Semantic interoperability ensures that data can be comprehended unambiguously by both human users and software programs across different platforms and domains. It offers interaction between heterogeneous things, machines, and smart objects on a higher level of abstraction. This is a prerequisite for the creation of value added open and interoperable IIoT services and applications. Semantic Interoperability also targets the problem of maintaining many different APIs for communication with devices or systems, coming from different manufacturers.

In order to motivate the need for semantic interoperability in IIoT, we consider typical tasks such as discovery, engineering, re-engineering, bootstrapping, and orchestration in industry automation systems. As an indicate example, we use the FESTO Process Automation workstation¹⁵ as a typical example of an industry multipurpose workstation. It consists of two tanks, equipped with several sensors and actuators. Thus, we need to ensure that updated equipment and services can be integrated to the workstation, while retaining the proper functionality as it was described during the system design phase.

The main objectives are:

¹⁵ <http://www.festo-didactic.com/int-en/learning-systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and-temperature-controlled-systems.htm>

- The semantic modelling of field devices and their capabilities
- The semantic description for requirements of added value services
- The capability of plugging a new device into a running system and functioning with respect to an already engineered system.

The main benefit of semantically interoperable IIoT systems is the realization of more cost-effective automation systems. The first aspect related to the cost saving is about reducing the costs of engineering, bootstrapping, operation and maintenance of IIoT systems. The second aspect is related to the enablement of added value services on IIoT systems. In practice automation systems often contain underused equipment (e.g., sensors or actuators that are not used in an implemented process workflow). If we could enable easier re-engineering of these systems with less effort (e.g., by using semantic technologies), then it would be cheaper to implement added-value services on automation systems. In particular, it would be beneficial to deploy added-value services on underused equipment.

3.7.2. NARRATIVE OF USE CASE

3.7.2.1. SHORT DESCRIPTION

Current automation systems are fully integrated vertical systems. They are efficient, but inflexible. Once they are engineered and operational, they cannot be changed easily. For example, it is not straightforward to plug a new device into a running system and expect it to be functional with respect to an already engineered system. Similarly, it is not effortless to develop a new added value service for an existing system. In both cases, the reason is a know-how contained by experts, but not explicitly represented in machine-interpretable form. We need semantic description of capabilities of field devices, as well as requirements of added value services. The following use case describes problems found in the current vertically integrated automation systems and sketches the role of semantics in IIoT in order to amend these problems.

3.7.2.2. COMPLETE DESCRIPTION

In this section, we describe our industrial use cases for discovery, engineering, re-engineering, bootstrapping, and orchestration of field devices in an automation station (AS) on a FESTO Process Automation workstation, shown in Figure 1. The workstation consists of two liquid tanks on which various sensors and actuators can be attached to. In the case of our workstation there exist: an ultrasonic sensor, float sensors, proximity sensors, temperature sensor, heater, overflow sensor, pump, a pneumatic valve, and so on. The workstation captures the process of steering liquid, measuring level of liquid, protecting overflow of liquid, protecting a pump from dry run, measuring flow of liquid, and keeping liquid within certain temperature range and within certain level range.

The workstation is a typical prototype as found in a real production environment. It means that it is a set of networked field devices, highly integrated in a closed operating system. Engineering and configuration of these devices require special know-how about their capabilities. Information about interfaces and other characteristics of field devices exist, but it is decoupled from them. Typically, it resides in catalogs, engineering and development environments, or just contained as expert know-how. Therefore, once an automation system has been engineered, it is hard to change it, e.g., to plug a new field device and automatically make it to be a part of a running system, or to re-engineer an existing system whenever a malfunction occurs. It is already a challenge to discover an existing functionality in complex workstations. Further difficulties occur when a new application or added-value service is to be added over a running system. In order to illustrate this, let us consider a

concrete example. To ensure the overflow protection in Tank 1, we must be able to detect the rise of liquid over a certain threshold, and in that moment to pump out the liquid from Tank 1 to Tank 2.

Discovery - For the first task, we have an ultrasonic sensor and a float sensor available in Tank 1. The float sensor is a binary sensor, which detects the overflow of liquid in Tank 1. The ultrasonic sensor measures the level of liquid in Tank 1. For the second task, a pneumatic valve, which controls liquid flow from Tank 1 to Tank 2 can be used. This information (Schemas), together with the information about parameterization and localization of devices, is valuable for discovery purposes. It needs to be formally represented in a standardized semantic format (by a Knowledge Engineer) so that machines or engineers can easily discover relevant devices (e.g., a level sensor or pneumatic valve).

Engineering - An engineer is supported by semantic technologies during the engineering process in order to lower the engineering tasks. A new functionality can be deployed based on semantic engineering templates, called Recipes. A Recipe is a formal specification of a typical engineering task, such as the overflow protection in a tank. Since the Recipe is semantically described, it can be discovered via semantic search. Once the engineer finds a relevant Recipe for her engineering task, the Recipe can be easily implemented. That is, relevant ingredients can be automatically found (i.e., IIoT filed devices with the functionality required by the Recipe), and the engineer can easily put the ingredients into interaction. In this way, the new functionality can be engineered with less effort, and the documentation about it can be automatically produced.

Re-engineering - An engineer can update the functionality of an AS with minimum effort simply by installing a new functionality on it. If any of the devices on the FESTO workstation used in the above engineering process is malfunctioning, then the devices should be re-engineered to ensure overflow protection on Tank 1.

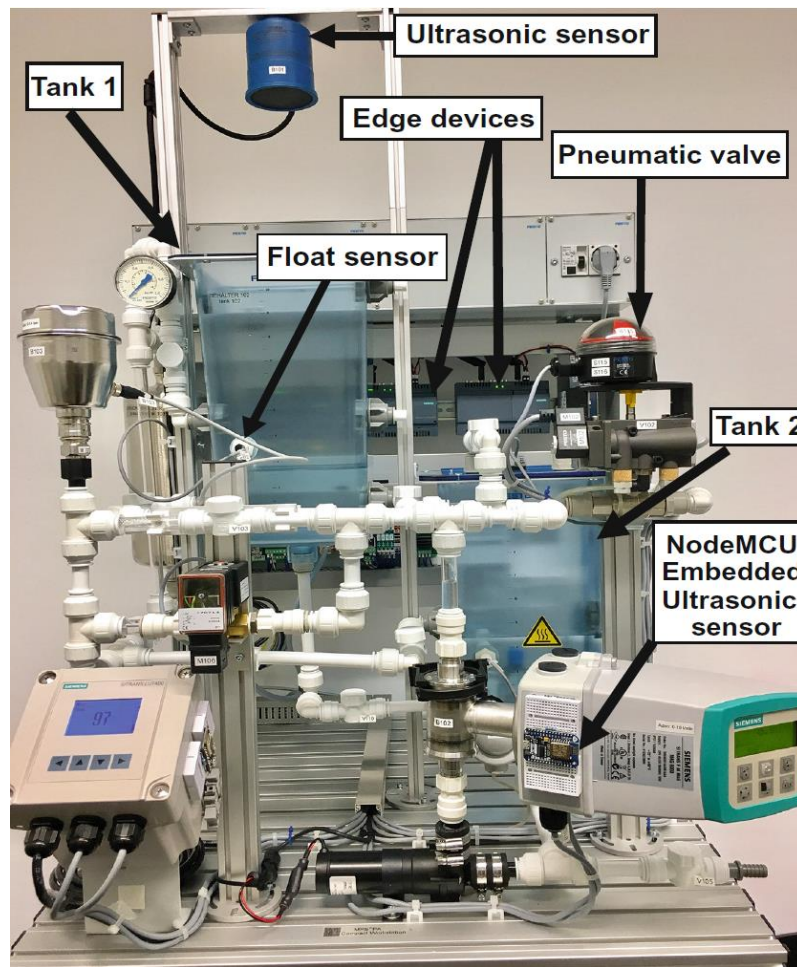


FIGURE 27: FESTO PROCESS AUTOMATION WORKSTATION

Bootstrapping – An engineer can add new IIoT field devices or replace existing ones with less effort. In existing Ass, the integration of a new field device is a complex task. It involves manual checking whether the device can be used for a certain automation task and whether it is compatible with the existing devices involved in that task. Further on, its configuration is not trivial since the device is just a part of an ecosystem of heterogeneous devices. Therefore, a field device needs to come with a semantic description of it, which will help in the process of bootstrapping the device into the system. The semantic device description is an enabler for simplifying the process of configuration and deployment of IIoT field devices.

A semantic model, which enables device descriptions for the purpose of bootstrapping, needs to be applicable for SEMIoTICS use case and needs to be standardized as much as possible. Further on, it needs to be applicable for IIoT applications. Therefore, for this requirement, the SEMIoTICS solution will rely on the industry standard OPC UA¹⁶, and W3C standardization work Thing Description (Web of Things¹⁷), as well as on the IoT application semantics created by iotschema.org¹⁸.

¹⁶ <https://opcfoundation.org/about/opc-technologies/opc-ua/>

¹⁷ <https://www.w3.org/WoT/WG/>

¹⁸ Currently available from: <http://iotschema.org/>

Orchestration – In this phase, we envision an application developer that targets to develop an application. For instance, it can be a Cloud App, which demands certain quality of service (QoS), network related criteria to be fulfilled, as typically required by IIoT applications. QoS network-related criteria can be semantically described and interpreted by SDN controllers prior to the deployment of the application in order to check whether the communication infrastructure can meet the requirements of the Cloud App. When developing an application, the application developer needs to be in position to orchestrate network resources in the same vein as resources exposed by field devices. It is a goal of our work in SEMIoTICS to provide a semantic model for describing QoS network-related parameters and SDN/NFV infrastructure, so that an automated evaluation of both is enabled.

3.7.3. TECHNICAL DETAILS

3.7.3.1. DIAGRAMS OF USE CASE

The following figure shows the focus areas and tasks related to the role of semantics in IIoT and SEMIoTICS project. This is shown in ovals on the left-hand side of the figure. The focus areas and tasks are roughly sorted out so that they show at which level of the SEMIoTICS architecture levels they fit. We see that semantic descriptions need to be provided at each of the three levels, thereby describing field devices, edge devices (gateways), network infrastructure, as well as assets in the backend/Cloud. Semantic artifacts will be stored in a semantic (knowledge) repository.

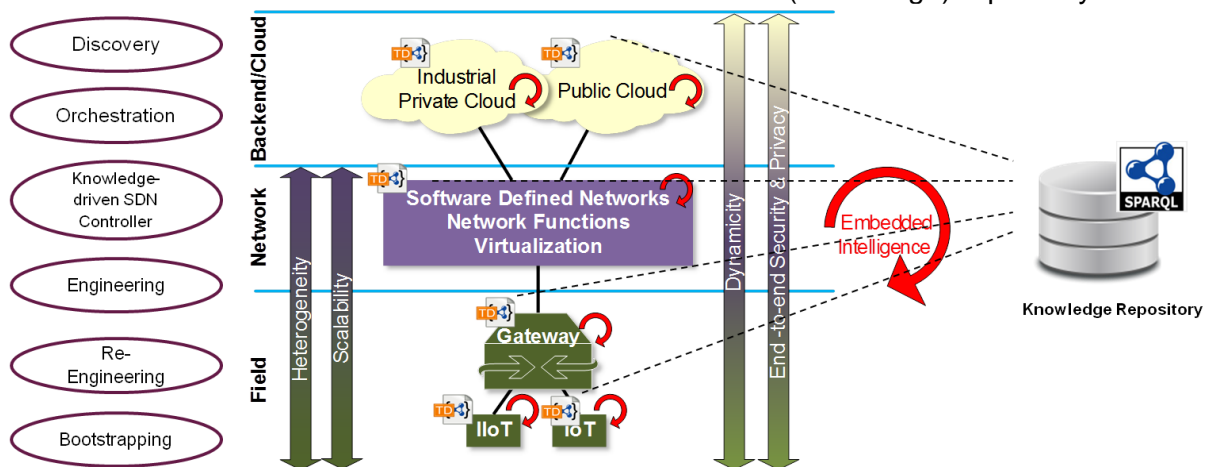


FIGURE 28: SEMANTICS IN SEMIoTICS - DISCOVERY, ENGINEERING, RE-ENGINEERING, BOOTSTRAPPING, AND ORCHESTRATION IN INDUSTRY AUTOMATION SYSTEMS

The following figure depicts different actors related to the use case and typical roles they undertake. In the next section, we will describe details about these actors and their roles.

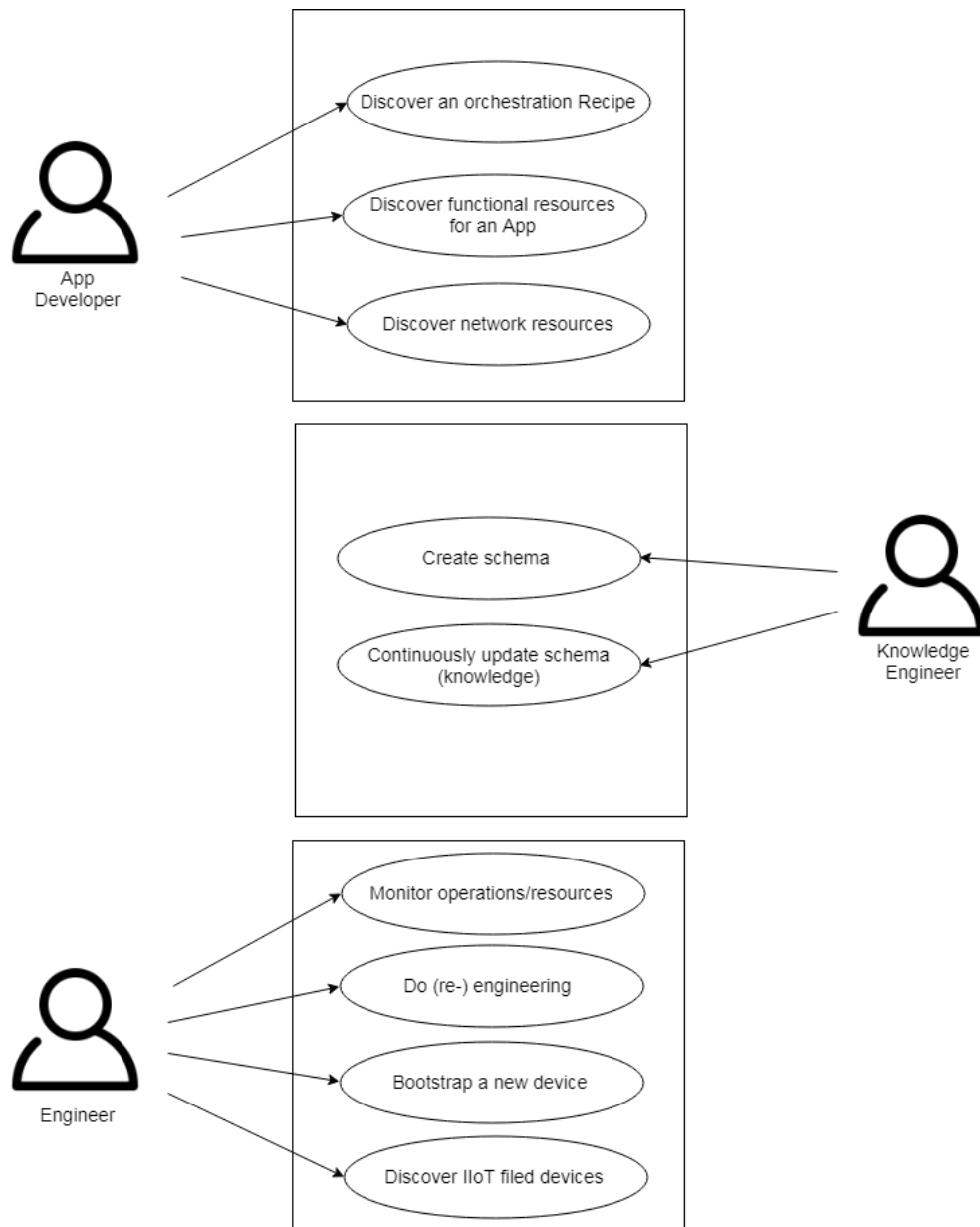


FIGURE 29: USE CASE DIAGRAM

3.7.3.2. ACTORS

1. An **Engineer** – is responsible for an Automation System. The actor performs tasks of engineering and re-engineering, and later the tasks of deployment, operation, and monitoring. This role is normally divided in several ones. However, in the scope of this use case it is important to note that all these roles are related to bottom part of the architecture (field level and control). For simplicity, we keep these roles gathered around one actor.
2. A **Knowledge Engineer** – provides semantic (knowledge) artifacts for all three levels of the SEMIoTICS architecture. These include semantic schema, ontologies or knowledge graphs required for describing assets in a concrete implementation of the architecture. Further on, the

artifacts include applications and tools, which help other actors to use semantics. The Knowledge Engineer continuously updates the semantic artifacts depending on new assets and use cases.

3. An App Developer – develops new applications (Apps). Apps bring added value to existing services and infrastructure. Therefore, it is important to support the App Developer to efficiently provide her task. This is not an easy task as Apps are built on complex Automation Systems. While an engineer possesses a lot of domain know-how related to Automation Systems, an App Developer expects this know-how to be available to her, so that she can focus on her own task. Semantic (knowledge) artifacts, provided by Knowledge Engineer, serve to close this gap.

3.7.3.3. TRIGGERING EVENT, PRECONDITIONS, ASSUMPTIONS, SUCCESS CRITERIA/ EXPECTED OUTCOME

This information is provided based on different possible scenarios related to this use case.

1. Scenario – Bootstrapping a new field device

Event:

- A new field device is added in a SEMIoTICS automation system.

Precondition:

- The field device is semantically described in accordance to a semantic model used by a SEMIoTICS automation system.

Assumption:

- The field device can be used (bring certain value) to an operating SEMIoTICS automation system.

Expected outcome:

- The new field device is successfully and with less effort (when compared to traditional, i.e., non-semantic based systems) integrated in the SEMIoTICS automation system.

2. Scenario – Development of a new application with Recipes

Event:

- App Developer starts the development process of a new App.

Precondition:

- The new App can be developed with a Recipe (based on SEMIoTICS Recipe Model).

Assumption:

- The new App requires semantic artifacts that already exist in the semantic (knowledge) store.

Expected outcome:

- The new App is successfully developed based on a Recipe. Since the tasks of semantic discovery and orchestration are automated, the overall effort to develop the new App is significantly decreased (when compared to traditional, i.e., non-semantic based approach).

3. Scenario – Specification and updates of Semantic artifacts to support bootstrapping and development of new applications

Event:

- A new field device or network device is plugged to the SEMIoTICS automation system but its Thing (device) Description cannot be annotated with existing SEMIoTICS semantic schema. The same event is triggered if Backend/Cloud assets cannot be semantically annotated based on existing semantic schema.

Precondition:

- No precondition.

Assumption:

- No Assumption.

Expected outcome:

- The new filed device, network device or Backend/Cloud asset is successfully described with the SEMIoTICS semantic schema. Thus, it is semantically integrated into SEMIoTICS infrastructure and available to be used in SEMIoTICS use cases.

3.7.3.4. INFORMATION EXCHANGED BETWEEN ACTORS

Information exchange is automated between actors. When an Engineer plugs a new field device, the device registers itself with the semantic (knowledge) repository. This is done by uploading its Thing Description. In this way, the Knowledge Engineer can discover the device and annotate its description w.r.t semantic schema used in the system. The App Developer accesses the same repository whenever she needs to discover assets/resources required for the development of a new App.

3.7.3.5. REQUIREMENTS DESCRIPTION

No additional requirements are needed regarding the infrastructure. Below, we list the requirements that need to be taken into account in order to provide semantic interoperability in IIoT, in SEMIoTICS, w.r.t tasks of discovery, engineering, re-engineering, bootstrapping, and orchestration in industry automation systems.

Req.-ID	Description
R10.1	Semantic schema to enable semantic description of field and edge devices.
R10.2	Semantic schema/metadata to enable semantic description of network (SDN and NFV) infrastructure.
R10.3	Semantic schema to enable application semantics in Backend/Cloud.
R10.4	Semantic format to describe assets (e.g., W3C WoT Thing Description)
R10.5	Semantic mapping between various semantic models, e.g., OPC UA IM, MindSphere IoT Model, W3C WoT TD etc.
R10.6	Infrastructure/tools to support semantic discovery.
R10.7	Infrastructure/tools to support orchestration of application, network and other resources based on Recipes.
R10.8	Infrastructure/tools to support semi-automated device Plug & Play and bootstrapping.



4. Requirements classification for SEMIoTICS architecture

4.1. Classification of requirements

The requirements identified in this document are classified in the categories as described in the following table:

Classification	Description
SPDI-by-Design	SPDI (Security, Privacy, Dependability, and Interoperability) is a pattern-driven framework, built upon existing IoT platforms, to enable and guarantee secure and dependable actuation and semi-autonomous behaviour in IoT/IIoT applications.
SDN & NFV	Software Defined Networking and Network Functions Virtualization are the key core 5G technologies which are enablers for programmable network.
IoT Platforms	IoT platforms are designed to store and process Internet of Things (IoT/Industrial IoT) data. The platforms are built to take in the massive volumes of data generated by devices, sensors, actuators and applications and initiate actions for their timely responses.
Monitoring and Adaptation	The SEMIoTICS framework will support evolving runtime management and adaptation of IoT applications and smart objects. Adaptation will be triggered by monitoring the guard conditions of the patterns used by the IoT application of interest, and applying the actions defined in the patterns when such conditions are satisfied. SEMIoTICS framework will incorporate learning and evolution mechanisms supporting the analysis of any adaptation and configuration actions undertaken for an IoT application.
Machine Learning	In SEMIoTICS, it is foreseen to enrich the generated and collected data from the Sensors/Actuators with semantic information at the source and intermediate stations, process them locally with machine learning algorithms to extract the most important features of the data and only then transfer the learned local features to the cloud for further, global, processing and feature analysis.
Security and Privacy	SEMIoTICS will combine novel and standardized technologies to provide lightweight and usable mechanisms for the end-to-end authentication of its entities (devices, applications, users etc.)

Semantic Interoperability	As IoT/IIoT should integrate an extremely large amount of heterogeneous entities, these entities need to be consistently and formally represented and managed. Such entities are sensor and actuation devices as well as applications that utilize them. SEMIoTICS will extend the W3C approach so that it takes into account the needs of the IIoT domain, in particular so that it enables the semi-autonomic bootstrapping & interfacing of things to achieve end-to-end semantic interoperability in an IoT system.
Field network	Field network typically connects sensors, actuators, and other I/O devices to IoT/IIoT Gateway/s.
Connectivity network	In order to provide programmability in the network “Connectivity network” offers SDN/NFV enabled 5G network between Field network and its backend network.
Cloud / Backend	Cloud or Backend is an efficient, scalable, affordable way to both manage IoT/IIoT devices and handle all that information they generate and make it work for the business applications it is hosting.
Requirement type (F/N)	Functional (F) or Non-functional (N) type of requirement coming from specific use case.

The requirements may also be classified as per following scenarios.

Scenario	Description
Demonstration Scenario 1	IIoT integration in Wind Park Control Network providing value added services (IWPC) (e.g. 3rd party vendor access, Predictive Maintenance and Monitoring, Local smart behaviour, etc.)
Demonstration Scenario 2	Socially Assistive Robotic Solution for Ambient assisted living (SARA)
Demonstration Scenario 3	Intelligent Heterogeneous Embedded Sensors for future IoT systems (IHES)

The above classification criteria are then applied to different use cases identified in this document as shown in the following table.

Nr.	Req.-ID	Description	SPDI-by-Design	SDN & NFV	IoT Platforms	Monitoring and Adaptation	Machine Learning	Security and Privacy	Semantic Interoperability	UC1	UC2	UC3	Field	Connectivity	Cloud	Requirement type (Functional [F] / Non-functional [N])
1	R1.1	Automatic establishment of networking setup to establish end-to-end connectivity between different stakeholders		X	X					X			X	X	X	F
2	R1.2	Automatic establishment of computing environment in IIoT Gateway necessary for the minimum operation of the IIoT devices through 5G network controller based on SDN/NFV		X			X		X	X			X	X		F
3	R1.3	Enabling the definition of network QoS on application-level and automated translation into SDN controller configurations.		X	X					X				X		F
4	R1.4	Network resource isolation for guaranteed Service properties – i.e. reliability, delay and bandwidth constraints.		X						X			X	X		F
5	R1.5	Fail-over and highly available network management in the face of either controller or data-plane failures.		X		X				X				X		F
6	R1.6	Identification and exclusion of decisions made by unreliable, i.e. faulty or malicious SDN controllers.		X		X				X				X		N
7	R1.7	Scalable operation of the SDN control to cater for a massive IoT device integration and large-scale request handling in the SDN controller(s) using a (near-) optimal		X									X	X	X	N

		IoT client – SDN controller assignment procedure.													
8	R1.8	Semantic and robust bootstrapping/registration of IIoT sensors and actuators with IIoT gateway										X	X		F
9	R1.9	Semantic interaction between IIoT Gateway and legacy turbine control system										X			F
10	R1.10	Local analytical capability of IIoT Gateway to run machine learning algorithms (e.g. specific to 2 specific sub-use cases)				X				X		X			F
11	R1.11	Support of device composition and application creation through template approach.		X	X			X					X	X	N
12	R1.12	Standardized semantic models for semantic-based engineering and IIoT applications.						X		X		X			F
13	R2.1	The SARA system MUST provide a User Interface (UI) for the GP to define system configurations for each Patient.			X	X					X			X	F
14	R2.2	The SARA system MUST provide a User Interface (UI) for the Technician to access system configuration data for Patients.						X			X			X	F
15	R2.3	The SARA system MAY provide a User Interface (e.g. on a remote client) to allow the Technician to configure the system for a Patient.	X		X			X			X			X	F

16	R2.4	The SARA system MUST provide a UI for the GP to define 'daily activity rules' for each Patient.						X			X				X	F
17	R2.5	The SARA system MUST (via the SARA Hubs) continuously monitor the Patient's daily activities.		X	X			X			X		X		X	F
18	R2.6	The SARA system MUST validate monitored Patient activities against the prescribed 'daily activity rules'.						X			X				X	F
19	R2.7	The SARA system MUST provide (via Dialog devices) timely activity suggestions to the Patient – e.g. for scheduled events (taking medicine, programmed exercises) and in case of 'violations' of the 'daily activity rules'.			X						X				X	F
20	R2.8	The RA MUST assist the Patient in performing a variety of cognitive rehabilitation exercises.			X		X				X		X		X	F
21	R2.9	The SARA system MUST recognise (via Dialog devices) and respond appropriately to Patient requests to perform cognitive exercises.			x		X				X		X		X	F
22	R2.10	The SARA Hubs MUST be able to detect, recognise, identify and determine the positions of significant objects (including people) in the environment.	X		X		X				X		X		X	F
23	R2.11	The SARA system MUST keep track of the changing positions of significant objects (including people) in the environment.	X		X		X				X		X		X	F

24	R2.12	The SARA system MUST recognise (via Dialog devices) and respond appropriately to Patient requests to: <input type="checkbox"/> be informed of locations of objects/people. <input type="checkbox"/> receive physical support from the RR. <input type="checkbox"/> be escorted to a location by the RA.	X		X		X				X		X		X	F
25	R2.13	The SARA system MUST (via SARA Hub devices) continuously monitor the Patient to detect, and raise an alert in case of, incidents (i.e. critical medical events such as falls, heart attacks, ...).	X	X	X	X		X	X		X		X	X	X	F
26	R2.14	The SARA system MUST reliably and rapidly notify assigned Caregivers of Patient incidents.	X	X				X			X		X			F
27	R2.15	The SARA system MUST (in conjunction with the RA) provide on-demand telepresence (bi-directional, real-time, high-quality, audio-video streaming) between the RA's tablet device (Patient side) and either the Caregiver's mobile phone (UC 6) or GP's desktop computer (UC 7).	X	X			X				X		X	X	X	F
28	R2.16	The SARA system MUST (for reasons of privacy) notify the Patient whenever a telepresence session has started/ended, and also inform the Patient of the identity of the remote operator.	X		X			X			X		X		X	F

29	R2.17	The SARA system MUST provide a UI for the GP to access historical logs of Patient bio-medical data from the ACS.						X				X			X	F
30	R2.18	The SARA system MUST periodically log (Hub monitored) Patient bio-medical data to the ACS.	X		X			X				X			X	F
31	R2.19	The SARA system MUST recognise (via Dialog devices) and respond appropriately to Patient requests for physical exercises.			X		X					X			X	F
32	R2.20	The SARA system MUST assist the Patient in performing a variety of physical rehabilitation exercises.	X		X	X						X			X	F
33	R2.21	The SARA system MUST provide (via the SARA Hubs) the GP remote access to real-time (Hub monitored) Patient biomedical data.	X	X	X			X	X			X	X		X	F
34	R2.22	The SARA system SHOULD support and track the dynamic addition/removal of system components (e.g. health monitors temporary added to the BAN).	X		X	X						X			X	F
35	R2.23	The SARA system SHOULD facilitate the integration of legacy and new IoT Devices (Hubs, Sensors & Actuators) - e.g. through standardised low-level API conformance.			X				X			X			X	F

114

39	R2.27	The SARA system MUST perform with consistent quality (as assured through version control of source code and strict, systematic and high-coverage regression testing) in all target operating environments.			X			X				X			X	N
40	R2.28	The SARA system MUST fully comply with all relevant local government laws governing the privacy, security and storage of sensitive Patient health-related data.	X					X				X		X	X	N
41	R2.29	The SARA system MUST ensure that all health-critical communications (in particular: incident alerts) between components are prioritised in terms of receiving network resources and ensuring reliable transmission to destination.	X	X		X						X		X		N
42	R2.30	The SARA system MUST reliably and rapidly respond appropriately to health-critical events (in particular incident alerts).	X	X	X	X		X	X			X		X	X	N
43	R2.31	The SARA Mobotic devices SHOULD NOT (as far as this is preventable) cause any harm or damage to individuals and objects while navigating around the home.	X									X		X	X	N
44	R2.32	The SARA Mobotic devices MUST include emergency failsafe 'stop' mechanisms allowing them to be shut down in case of impending or	X									X		X		N

		actual risk to persons or objects.														
45	R3.1	IIoT Sensing unit shall be able to interface to the physical world.			X											F
46	R3.2	IIoT Sensing unit shall be able to interface to the IIoT Sensing gateway in order to coordinate with it.			X											F
47	R3.3	IIoT Sensing unit shall be able to learn a model from observed data in a unsupervised manner.					X									F
48	R3.4	IIoT Sensing unit shall be able to detect relevant changes from the learned model and report them to IIoT Sensing gateway.				X										F
49	R3.5	IIoT Sensing unit shall be able to adapt to a new model if IIoT sensing gateway requires this.				X	X									F
50	R3.6	IIoT Sensing gateway shall be able to coordinate a set of IIoT sensing units by finding any correlation between them according to observed data models.			X	X										F
51	R3.7	IIoT Sensing gateway shall be able aggregate relevant events (i.e. changes) coming from whichever of connected IIoT sensing units deciding if they are global or local changes.			X											F
52	R3.8	IIoT Sensing gateway shall have the capability to exchange relevant information (i.e. events) between itself, the cloud and the sensing units with some connectivity capabilities.		X	X											F

53	R3.9	IloT Sensing web GUI shall be able to display correlations between connected IloT Sensing units and the status related to each IloT sensing unit.			X									X			F
54	R3.10	IloT Sensing web GUI shall be able to display logging about relevant events detected by connected IloT Sensing units reporting info about unit ID, type of data and type of event detected.			X									X			F
55	R4.1	Connectivity of the SDN controller with the various underlying components (e.g. the SDN switches and the IoT gateways and devices) in order to support the automated configuration process	X	X		X					X	X	X		X		F
56	R4.2	Connectivity of the SDN controller with the remote management service in order to support the remote configuration process	X	X		X					X	X	X		X		F
57	R4.3	The SDN controllers must be always available to serve incoming requests	X	X		X					X	X	X		X		N
58	R4.4	Intrusion Detection System (IDS) that captures and processes suspicious traffic	X	X		X		X			X				X		N
59	R4.5	Accredited and certified Computer Emergency Report Team (CERT)						X			X						N
60	R5.1	Low latency and reliable communication between the energy controllers, when they are deployed at the IoT Gateways. This is mandatory for the iterations among the energy controllers that		X	X			X					X		X		N

		required in the distributed optimization.														
61	R5.2	Interoperability between the IoT Gateways and the IoT devices (sensors/actuators), due to the heterogeneous intranets communicating them.		X					X				X	X	X	N
62	R5.3	Low latency and reliable communication between the sensor measurements and the IoT Gateway		X				X				X	X	X		N
63	R5.4	Low latency and reliable communication between the energy grid operator and the energy controllers		X				X				X		X	X	N
64	R5.5	Low latency and reliable communication between the energy controller VFs, within the IoT Gateway, and the actuators, which manage the loads, when the VFs are deployed at the IoT Gateways. Or between the IoT backend and the buildings' actuators, when the VFs are deployed at the IoT backend.		X	X			X				X	X	X	X	N
65	R5.6	Reliable communication between the IoT Gateways and the IoT backend, when the energy controller VF is deployed at the IoT backend. Moreover, low latency can be required depending on the loads to be managed, e.g. energy consumption optimization		X	X			X	X			X		X	X	N

		of data centers or workstations.														
66	R6.1	IoT Gateway must be able to establish connection with devices via OPC protocol								X		X	X	X		F
67	R6.2	Connectivity between the control devices and IoT Cloud components for automated configuration purposes		X	X			X		X	X	X	X	X	X	F
68	R6.3	System scales even when new power plant block is monitored and managed		X	X	X		X		X	X	X	X	X	X	F
69	R6.4	Cloud applications to manage power plant can handle control process for many independent power blocks			X	X		X		X	X	X	X	X	X	F
70	R6.5	IoT Cloud component with advanced computing module have to generate business events for next subsequent processing. The IoT platform will decide, what changes should be done.			X	X	X			X		X			X	F
71	R6.6	Advanced computing module needs to be developed based on SPDI patterns which would support any IoT platform.	X		X		X		X		X			X	X	F
72	R6.7	IoT platform should provide the possibility to report list of executed actuation commands.			X	X				X	X	X		X	X	F
73	R7.1	Ensure high connectivity probability (>99%) among all the communication network devices		X	X							X	X	X		N

74	R7.2	The local cloud should be able to recognize the established patterns and act in less than 10ms		X		X	X					X			X	N
75	R7.3	The energy management system have to be able to handle data from a massive collection of smart meters, i.e. several micro-grids employ the same energy management system, and be able to scale when new micro-grids are included in the network		X	X							X	X			N
76	R7.4	The demand response algorithms have to decrease the energy consumption of the micro-grid by 40%				X	X					X	X			F
77	R7.5	The adaptive monitoring should be able to identify changes in the patterns and readapt the established actions in a timely manner		X		X	X					X		X	X	F
78	R7.6	The energy management system should monitor and guarantee a smooth transition from grid connection mode, to islanding mode, where the micro-grid consumes its own generated energy		X		X	X					X	X		X	F
79	R7.7	When there are discrepancies in the load of different buildings, the energy management system should guarantee the flow of energy from the low-load buildings to the high-load buildings to achieve load balancing				X	X					X	X		X	F
80	R8.1	IoT sensing unit shall be able to interface to the physical world							X			X	X			N

81	R8.2	IoT sensing unit shall be able to interface to the IoT gateway in order to coordinate with it						X			X	X			N
82	R8.3	IoT sensing unit shall be able to perform distributed feature extraction/selection from the acquired data					X				X	X			F
83	R8.4	IoT gateway shall be able to estimate abnormal detection based on (un)-supervised model					X	X			X	X			F
84	R8.5	IoT gateway shall be able to aggregate relevant events (i.e. changes) coming from whichever of connected IoT sensing units deciding if they are global or local changes						X			X	X			F
85	R8.6	IoT gateway shall have the capability to exchange relevant information between itself, the cloud and the sensing units with some connectivity capabilities						X			X	X	X	X	F
86	R8.7	Cloud shall be able to aggregate features/data coming from the edge level through IoT gateway						X			X	X	X	X	F
87	R8.8	Cloud shall be able to perform prediction based on the aggregated data based on a supervised model					X				X			X	F
88	R8.9	Cloud shall be able to send updated model data/parameters to the IoT gateway				X		X			X	X	X	X	F
89	R8.10	IoT gateway shall be able to update the monitoring model based on the information coming from the cloud				X	X				X	X	X	X	F
90	R9.1	Platforms, e.g. cloud platform and sensor, can be trusted.			X			X			X	X	X	X	N

122



100	R10.7	Semantic infrastructure/tools to support orchestration of application, network and other resources based on Recipes.	X	X	X				X	X					X	N
101	R10.8	Semantic infrastructure/tools to support semi-automated device Plug & Play and bootstrapping.	X		X	X			X	X			X			N

Although each requirement from separate use cases (SEMIoTICS and non-SEMIoTICS) fall in different classification areas, SEMIoTICS consortium did extensive exercise to map their relation to Demo scenario 1, 2 and 3 of SEMIoTICS with functional and non-functional nature of the individual requirement.

Based on “Classification” and “Scenario” criteria, the mapping of the selected requirements is shown in the following tables. *Note: The total number of requirements and sum of requirements applicable to each demo scenario will not match due to overlapping applicability of different requirements.*

Classification areas with **Blue** numbers of requirements represent the influence of those areas for each Demo Scenario.

# Requirements			
Classification	Demo Scenario 1	Demo Scenario 2	Demo Scenario 3
SPDI-by-Design (29)	8	23	3
SDN & NFV (36)	14	13	17
IoT Platforms (48)	13	28	19
Monitoring and Adaptation (29)	12	13	17
Machine Learning (22)	3	7	13
Security and Privacy (27)	4	19	8
Semantic Interoperability (29)	12	11	11

In the above table, top 3 classification criteria are marked in **blue** colour pertaining to each Demo Scenario.

There may be no specific classification area related requirements e.g. Security and Privacy, specified in SEMIoTICS use cases (UC1, UC2 and UC3). This is due to the fact that each use case focusses on specific technical aspects which are important for realization. Task 2.3 will come up with abstracted requirements considering different areas to be applied to SEMIoTICS use cases.

# Requirements			
Classification	Demo Scenario 1	Demo Scenario 2	Demo Scenario 3
Cloud / Backend (57)	12	35	20
Connectivity network (50)	16	19	27
Field network (62)	13	26	27
Functional Requirements (63)	15	30	31
Non-functional Requirements (38)	13	14	14

In the above table, top 2 classification criteria are marked in **blue** colour pertaining to each Demo Scenario.

# Requirements			
Classification	Demo Scenario 1	Demo Scenario 2	Demo Scenario 3
Total (101)	28	44	45

In the above table, total requirements and their applicability to each Demo Scenario is shown. Given the complexity of the diverse requirements from different use cases (both SEMIoTICS and Non-SEMIoTICS), it is not possible to find out common denominator of requirements. In task 2.3, above requirements will be considered as basis; and UC1-3 and UC agnostic requirements (from UC4-10) will be followed in order to showcase 3 SEMIoTICS use cases in lab trials.

5. Summary

This deliverable has highlighted different use cases for smart sensing and actuation from 5 different domains namely Energy, Healthcare, Smart City, Smart Building, and Industry Automation. The detailed requirements coming from 10 use cases in this document reflect the implementation of 3 Use cases of SEMIoTICS and exploitation directions in 7 Non-SEMIOTICS use cases in their respective domains.

The detailed SEMIoTICS use case definitions, specifically Use cases 1-3 and the associated requirement analysis (done in section 4.1) will form the basis for selection in “Specification of Infrastructure requirements” (Task 2.3) and “SEMIOTICS architecture design” (Task 2.4) of Work Package 2 (WP2). The final requirements coming from WP2 will be implemented in WP3 and WP4; and eventually be demonstrated in 3 different demo/usage scenarios in WP5. The relation of the requirements considered for Task 2.3 with respect to the KPIs of the project (and project objectives) will be covered in Task 2.4 as well as technical deliverables of WP3 and WP4.