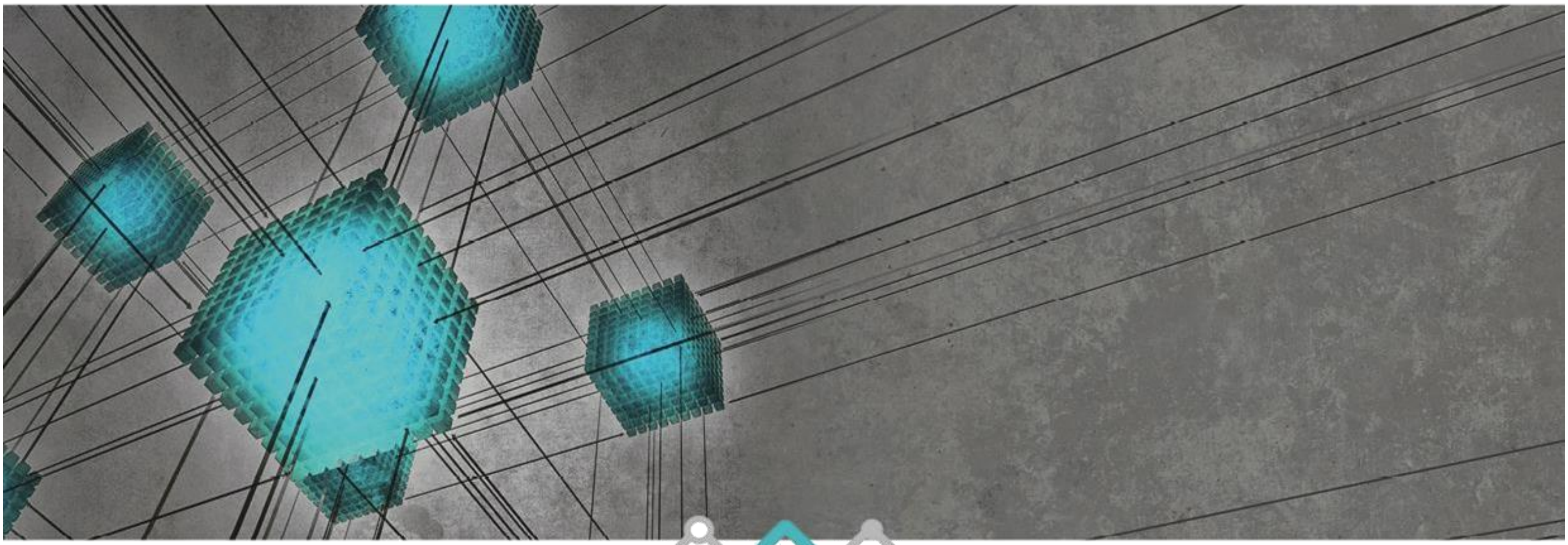


interiot

INTEROPERABILITY
OF HETEREOGENEUS
IOT PLATFORMS.

# D6.1

## System Integration Plan

August 2017

Version 1.0

# INTER-IoT

INTER-IoT aim is to design, implement and test a framework that will allow interoperability among different Internet of Things (IoT) platforms.

Most current existing IoT developments are based on “closed-loop” concepts, focusing on a specific purpose and being isolated from the rest of the world. Integration between heterogeneous elements is usually done at device or network level, and is just limited to data gathering. Our belief is that a multi-layered approach integrating different IoT devices, networks, platforms, services and applications will allow a global continuum of data, infrastructures and services that will enhance different IoT scenarios. Moreover, reuse and integration of existing and future IoT systems will be facilitated, creating a de facto global ecosystem of interoperable IoT platforms.

In the absence of global IoT standards, the INTER-IoT results will allow any company to design and develop new IoT devices or services, leveraging on the existing ecosystem, and bring them to market as fast as possible.

INTER-IoT has been financed by the Horizon 2020 initiative of the European Commission, contract 687283.

# INTER-IoT

# System Integration Plan

*Version:* Final

*Security:* Confidential

31st August 2017

The INTER-IoT project has been financed by the Horizon 2020 initiative of the European Commission, contract 687283

## Disclaimer

This document contains material, which is the copyright of certain INTER-IoT consortium parties, and may not be reproduced or copied without permission.

The information contained in this document is the proprietary confidential information of the INTER-IoT consortium (including the Commission Services) and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the project consortium as a whole nor a certain party of the consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is subject to change without notice.

# Executive Summary

This document describes the integration plan of the INTER-IoT system. The integration is divided into 2 sections, namely, the factory acceptance test (FAT) and site acceptance test (SAT). Both will be carried out during the execution of WP6.

During integration phase the developed modules will be tested separately on the developers' computers. When the separate modules are working properly they are combined during the FAT and deployed onto the test platform to test interfaces and communication between modules. The project considers three large scale pilots: INTER-HEALTH, INTER-LogP and INTER-DOMAIN with the integration of the different third party contributions.

Once the system is fully functioning on the test platform it is ready for field integration which will be done on actual systems and will encounter real-life situations. Both tests and the setup of the FAT are described in this document. SAT will be addressed in WP7 (Evaluation) and will be fully analysed in the evaluation process proposed in D7.1.

Additionally, regarding the requirement of the Ethical Review a section related with the problems associated with ethics and privacy during the integration are addressed in the documents.

Finally, the structure of this document is divided into the following sections:

- Section 1: Introduction
- Section 2: System description
- Section 3: Test strategy and approach
- Section 4: Defect Reporting
- Section 5: Test environment
- Section 6: Integration environment
- Section 7: Documents associated with the integration
- Section 8: Ethics and Security

## List of Authors

| Organisation | Authors | Main organisations’ contributions |
|---|---|---|
| Neways | Dennis Engbers,<br>Johan Schabbink | Initial setup/structure of the document.<br>Finalization of the document |
| UPV-SABIEN | Álvaro Fides<br>Gema Ibañez | Contribution on INTER-HEALTH<br>Contribution to Ethics |
| VPF | Pablo Giménez Salazar | Contribution on INTER-LogP |
| UNICAL | Giancarlo Fortino | Contribution on INTER-HEALTH |
| PRO | Miguel Montesinos<br>Miguel A. Llorente | Contribution to INTER-LogP<br>Contribution to INTER-DOMAIN<br>Contribution to Ethics and Security |
| TU/e | Georgios Exarchakos | Contribution to different sections of the document<br>Document review |
| XLAB | Flavio Fuart | Contribution to the integration plan<br>Contribution to INTER-DOMAIN |
| SRIPAS | Paweł Szmeja<br>Wiesław Pawłowski | Contribution to different sections of the document |
| RINICOM | Eric Carlson, Garik Markarian. | Contribution to INTER-HEALTH<br>Document review |
| NPV | Francisco Blanquer | Contribution to INTER-LogP<br>Contribution to INTER-DOMAIN |
| ASLTO5 | Anna Costa<br>Marina Mortara | Contribution to INTER-HEALTH<br>Contribution to Ethics |
| AFT | Moncef Seminchi | Contribution to different sections of the document |
| ABC | Alessandro Bassi | Contribution to different sections of the document |
| UPV | Eneko Olivares<br>Carlos E. Palau | Contribution to the integration plan<br>Contribution to INTER-DOMAIN<br>Final review of the deliverable |
| Irideon | Bastian Faulhaber | Contribution on Senshook |
| SOFOS | Haris Koumaras | Contribution on A software-defined end-to-end IoT gateway with virtualization capabilities |
| University of Twente | João Moreira | Contribution on Interoperable Situation-Aware IoT-Based Early Warning |
| Universitat Pompeu Fabra | Toni Adame Vázquez | Contribution on INTER-HARE platform |
| CNR | Gianfranco Modoni | Contribution on A Semantic Middleware for the information synchronization |

| | | |
|---|---|---|
| E3TCity | Javier Escalera | Contribution on E3Tcity Smart City Platform and Devices Integration |
| CEA - Commissariat à l'énergie atomique et aux énergies alternative | Levent Gurgen | Contribution on Integrating sensiNact platform with INTER-IoT Framework |
| Vrije Universiteit Brussel (VUB) | Kris Steenhaut | Contribution on INTER-OM2M |
| Nemergent Solutions S.R.L. | Joseoscar Fajardo | Contribution on Mission Critical operations based on IoT analytics (MiCrOBIoTA) |
| Athens University of Economics and Business | George Polyzos | Contribution on ACHILLES: Access Control and autHenticatIon deLegation for interoperabLE IoT applicationS |
| TU Wien | Hong-Linh Truong | Contribution on INTER-HINC: Interoperability through Harmonizing IoT, Network Functions and Clouds |
| AvailabilityPlus GmbH | Günther Hoffmann | Contribution on SecurIoTy |

## Change control datasheet

| Version | Changes | Chapters | Pages |
|---|---|---|---|
| 0.1 | Creation and completion | All | |
| 1.0 | Version ready for formal review | All | |

# Contents

## List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AIOTI | Alliance for Internet of Things Innovation |
| API | Application Programming Interface |
| App | Application |
| AS2AS | Application & Services to Application & Services |
| BMI | Body Mass Index |
| CCB | Change Control Board |
| CoAP | Constrained Application Protocol |
| D2D | Device to Device |
| DDOS | Distributed Denial-Of-Service |
| DIY | Do It Yourself |
| DS2DS | Data & Semantics to Data & Semantics |
| EC | European Commission |
| ECG | Electrocardiography |
| ECH | Empty Container Handler |
| EWS | Early Warning System |
| FAT | Factory Acceptance Test |
| ICT | Information and Communication Technology |
| FIPA | Foundation for Intelligent Physical Agents |
| GW | Gateway |
| IEEE | Institute of Electrical and Electronics Engineers |
| INTER-FW | INTER-IoT Interoperable IoT Framework |
| INTER-HEALTH | INTER-IoT Platform for Health monitoring |
| INTER-LAYER | INTER-IoT Layer integration tools |
| INTER-LogP | INTER-IoT Platform for Transport and Logistics |
| INTER-METH | INTER-IoT Engineering Methodology |
| IPR | Intellectual property rights |
| IPSM | Inter-Platform Semantic Mediator |
| IoT | Internet of Things |
| IoT-EPI | IoT European Platform Initiative |
| JADE | Java Agent Development Framework |
| JSON | JavaScript Object Notation |
| LPLAN | Low-Power Local Area Network |
| LPWAN | Low-Power Wide Area Network |
| MQTT | Message Queuing Telemetry Transport |
| MW2MW | Middleware platform to Middleware platform |
| NFV | Network Functions Virtualization |
| N2N | Network to Network |
| OS | Operating System |
| OSGi | Open Services Gateway initiative |

| | |
|---|---|
| PLC | Programmable Logic Controller |
| PWT | Professional Web Tool |
| QoS | Quality of Service |
| REST | Representational State Transfer |
| RTG | Rubber-Tyred Gantry cranes |
| RTT | Round Trip Time |
| SAT | Site Acceptance Test |
| SDN | Software Defined Networking |
| SQL | Structured Query Language |
| STA | Station |
| STS | Sea-To-Shore cranes |
| TT | Terminal Tractor |
| UPF | Universitat Pompeu Fabra |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| Wi-Fi | Wireless Fidelity |

# 1 introduction

INTER-IoT project is aiming at the design, implementation and experimentation of an open cross-layer framework, an associated methodology and tools to enable voluntary interoperability among heterogeneous Internet of Things (IoT) platforms. The proposal will allow effective and efficient development of adaptive, smart IoT applications and services, atop different heterogeneous IoT platforms, spanning single and/or multiple application domains.

Most current existing sensor networks and IoT device deployments work as independent entities of homogenous elements that serve a specific purpose, and are isolated from "the rest of the world". In a few cases where heterogeneous elements are integrated, this is done either at device or network level, and focused mostly on unidirectional gathering of information. A multi-layered approach to integrating heterogeneous IoT devices, networks, platforms, services and applications will allow heterogeneous elements to cooperate seamlessly to share data, infrastructures and services as in a homogenous scenario.

This document describes the integration plan which is part of the experimentation step.

There will be 2 major use-cases and several smaller ones to prove the functionality of the platforms.

The first major use case is INTER-LogP which involves:

Two-level verification involving: (i) Platform and (ii) Application Service. INTER-LogP will be validated against functional and non-functional indicators. The effectiveness of the transport and logistics service will be evaluated with respect to current operational metrics according to qualitative and quantitative performance indicators.

The second major use case is the INTER-HEALTH which involves:

Two-level verification involving: (i) Platform, and (ii) Application Service. INTER-HEALTH will be validated against functional and non-functional indicators as well as by performing usability analysis. The effectiveness of the lifestyle monitor service will be evaluated with respect to well-established traditional methods according to qualitative and quantitative medical indicators.

One of the main goals of INTER-IoT is to overcome fragmentation caused by typical IoT platforms being oriented to a specific solution, stakeholder and application domain. The cross-domain use case will show how verticality is avoided in INTER-IoT. The rationale behind this use case is that future IoT applications will not aim at a single application domain but multiple domains in which devices, networks, platforms, services or generated data will interact.

The scenarios defined in the cross application domain use case will integrate platforms from the two application domains in consideration, and also from different application domains (e.g. smart grid or smart cities). This use case will prove the extendibility of the project outcomes, achieving interoperability between IoT platforms from different application domains. Several scenarios have been foreseen in which IoT platforms from different application domains may be required to interoperate, e.g. logistics and health monitoring of transport workers for labour risk prevention, however new cross domain scenarios will be defined during the execution of the project and after the resolution of the Open Call, including e.g. road IoT ecosystems; supply chains or emergency response services IoT ecosystems used in fire brigades, ambulances or security forces.

According to the Grant Agreement the field trials will be successful once the following conditions are met:

Trials of INTER-IoT concept, with involvement of 400 smart objects in the logistics use case and 200 subjects (with wearable devices) in the m-Health use case, and ~500 IoT units in the cross domain use case. Extensive testing of results of application of the INTER-IoT framework to instantiate multi-IoT-platform systems in real-world scenarios, validated by the corresponding stakeholders.

# 2 System description

To overcome the fragmentation of vertically-oriented closed systems, architectures and application areas and move towards open systems and platforms that support multiple applications. The project will provide open solutions including an API, a methodology and tools, to allow integration of IoT platforms and therefore ease the development of new applications and services by third parties. The development of an interoperability framework will overcome fragmentation caused by typical IoT ecosystems orientation to a specific solution, stakeholder or environment. A cross-domain use case will prove how verticality is avoided in INTER-IoT

## 2.1 Component

The Inter-IoT system is build up in 2 sections, on is the gateway (GW) section and the other is the middleware (MW) section. The GW section is taking care of the lowest 2 layers, namely the D2D layer and the N2N layer. The gateway is forming the link between the virtual and physical world. The MW is taking care of the upper layers, namely the MW2MW layer, the AS2AS layer and the DS2DS layer.

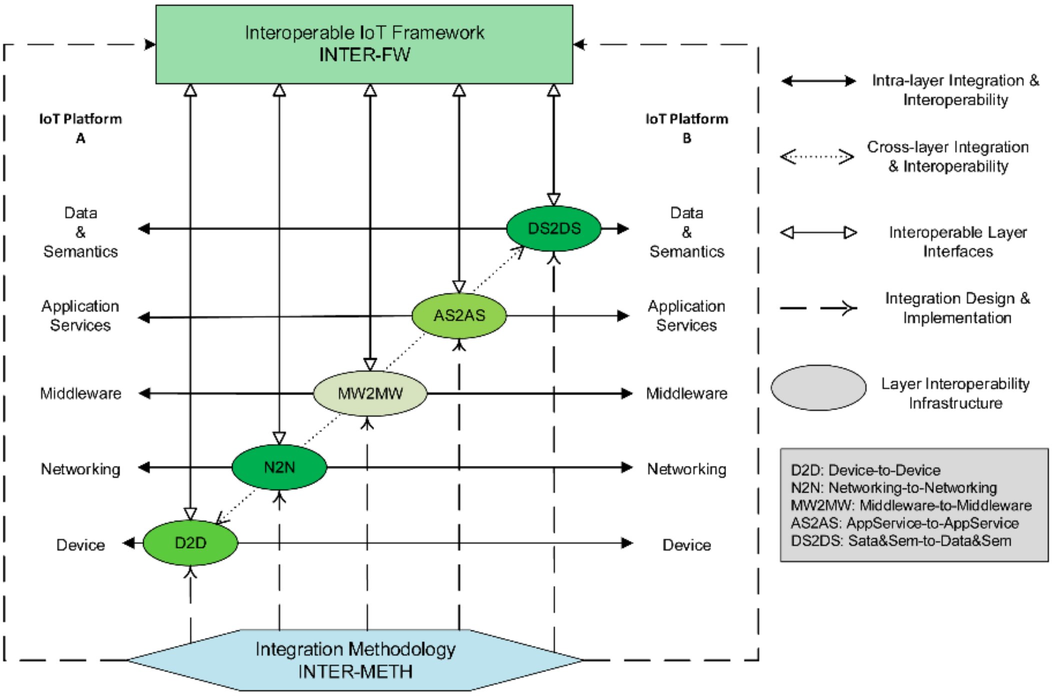

Interoperable IoT Framework
INTER-FW
IoT Platform A
IoT Platform B
Data & Semantics
Application Services
Middleware
Networking
Device
DS2DS
AS2AS
MW2MW
N2N
D2D
Integration Methodology
INTER-METH
Intra-layer Integration & Interoperability
Cross-layer Integration & Interoperability
Interoperable Layer Interfaces
Integration Design & Implementation
Layer Interoperability Infrastructure
D2D: Device-to-Device
N2N: Networking-to-Networking
MW2MW: Middleware-to-Middleware
AS2AS: AppService-to-AppService
DS2DS: Sata&Sem-to-Data&Sem

**Figure 1: INTER-LAYER**

Both the gateway as well as the middleware section are described in detail in D3.1

### 2.1.1 Layers

The gateway is built in layers to allow different classes of hardware to be connected. Wireless sensors can connect at the access network module (A.N. module), simple IoT devices can connect to the dispatcher connector and advanced IoT systems (which have an internal gateway) can connect to the MW bridge.

The middleware is also built in layers, the first layer is the bridge that allows different platforms to be connected. The translated information of the bridge is than routed through the communication and control layer towards either MW2MW services, if needed trough DS2DS IPSM towards the API which connects to the AS2AS and inter-framework.

## 2.2 Pilot project

In order to validate the Inter-IoT system several pilots have been chosen in order to validate the defined use-cases. Each pilot is described in the following paragraphs, starting with the 2 large pilots, INTER-LogP and INTER-HEALTH. After the these open call pilots will follow in paragraph 2.3.

### 2.2.1 INTER-LogP

The goal of INTER-LogP pilot is to demonstrate the need of a system that allows the exchange of data and messages among the different actors in the port community. In Figure 2 an overview has been given. There are three main actors: the port, the terminal and the haulier company. INTER-IoT has to provide interoperability between the IoT platforms of the port and the terminal, and give access to other devices from other companies, like trucks.

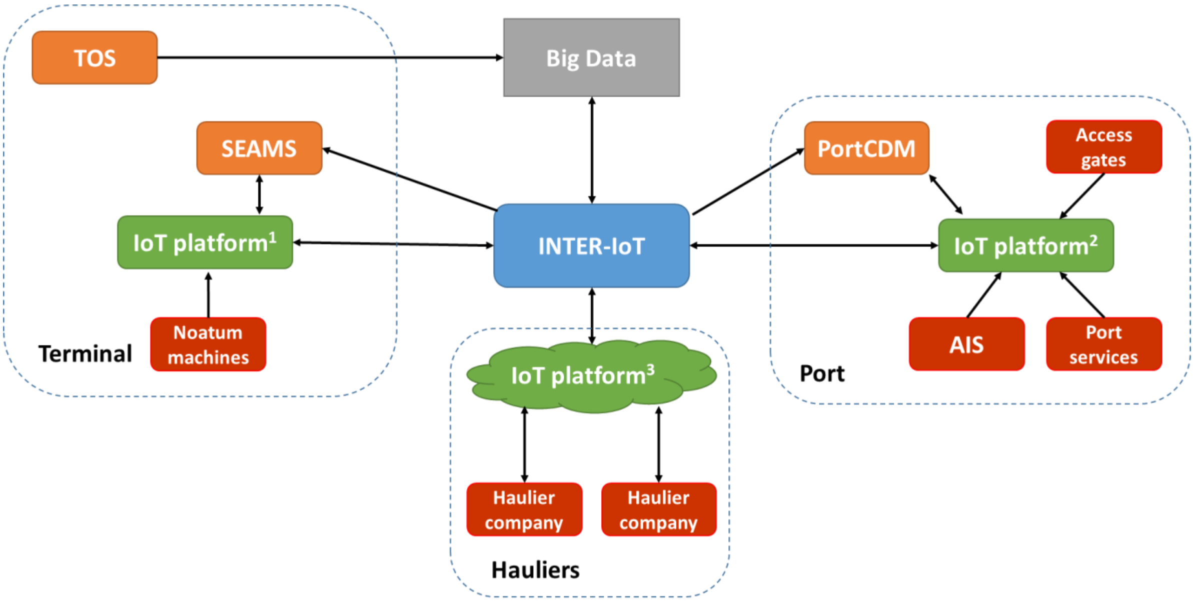


**Figure 2: INTER-LogP pilot design**

Both the port and the terminal have a large number of sensors and devices that produce large amounts of data, which can be interesting for other entities. Furthermore, they need data from other companies to provide a better service to their clients.

The main objective in the defined scenario is a service to control access, monitor traffic and assist the operations at the port. Several IoT platforms will identify trucks and drivers using devices. They share data under predefined rules through interoperability of platforms. This information can be used to monitor the truck by the Port Authority, control security and safety and manage all resources in the terminal more efficiently. This will also avoid queues in the access gates to the port and the terminal and improve overall logistics. The overall scenario and use cases of LogP are given in Figure 3.

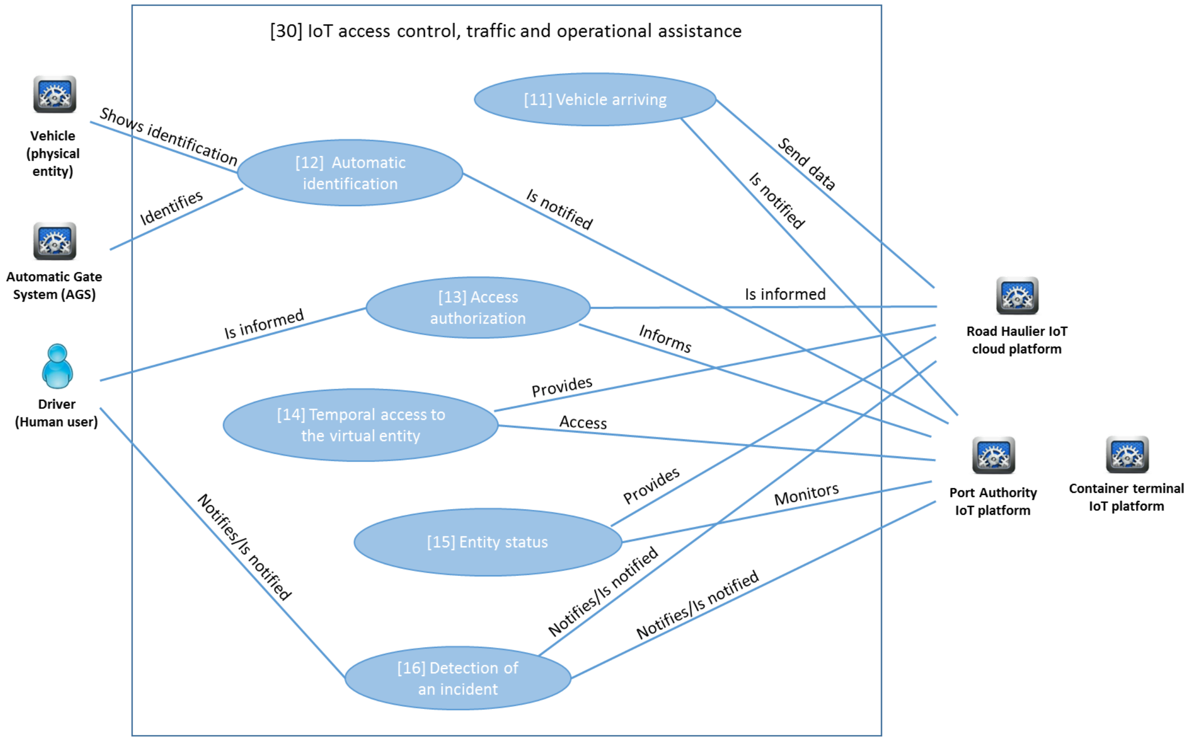

[30] IoT access control, traffic and operational assistance
Vehicle (physical entity)
Automatic Gate System (AGS)
Driver (Human user)
Road Haulier IoT cloud platform
Port Authority IoT platform
Container terminal IoT platform
[11] Vehicle arriving
[12] Automatic identification
[13] Access authorization
[14] Temporal access to the virtual entity
[15] Entity status
[16] Detection of an incident
Shows identification
Identifies
Is informed
Notifies/Is notified
Send data
Is notified
Is notified
Is informed
Informs
Provides
Access
Provides
Monitors
Notifies/Is notified
Notifies/Is notified

Figure 3. INTER-LogP scenario with use cases

### 2.2.2 INTER-HEALTH

The goal of the INTER-HEALTH pilot is demonstrating how to foster a healthy lifestyle and how to prevent chronic diseases by monitoring subjects' physical characteristics, nutritional behavior and activity.

The pilot will consist of 200 test subjects: 100 subjects following traditional monitoring without IoT devices and 100 subjects with devices. The latter are the ones using the INTER-IoT solution.

They will first attend a nutritional counseling session at ASL TO5 where their initial physical characteristics are measured, using IoT Devices on the premises (BMI, waist circumference, weight, blood pressure…).

Each subject will then receive a management program. Then at home, while they follow the program, they measure their characteristics using their phone and IoT devices.

The subjects will visit ASL TO5 each 6 month for check-ups. The healthcare professional in charge of monitoring each user will have access to the history of all the measurements through a dedicated web application.

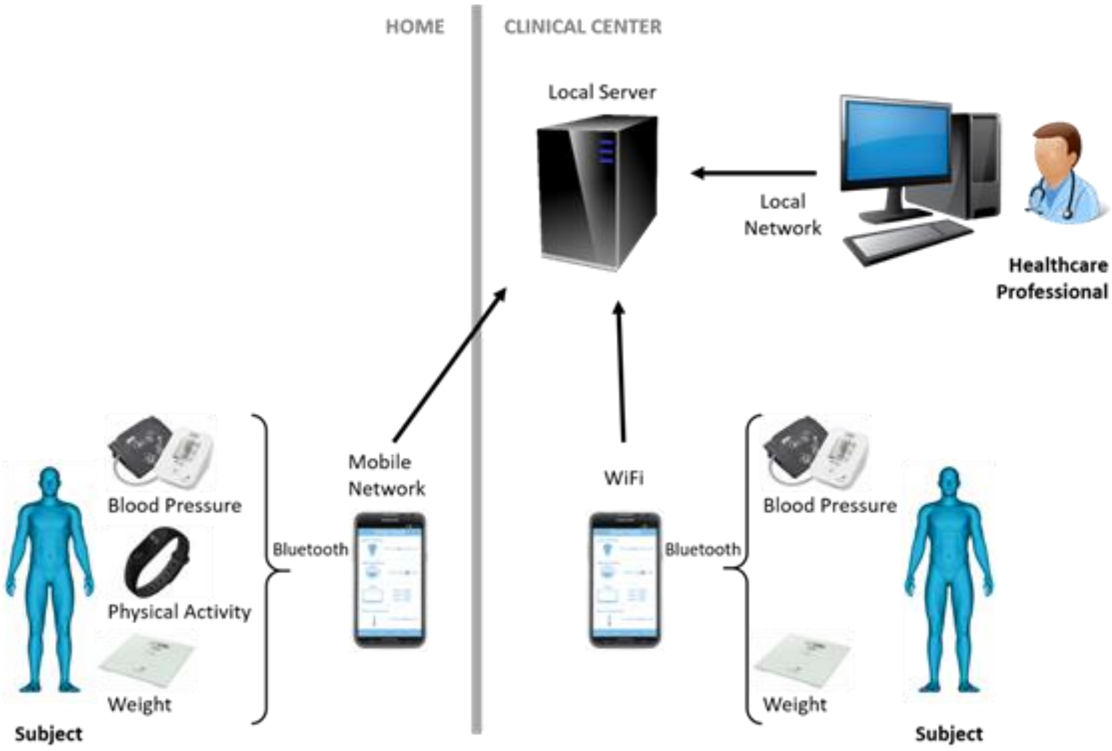


**Figure 4: INTER- Health System overview**

The Local Server is located at the premises of the clinical centre.

It runs the following components: The INTER-IoT Framework, the instance of universAAL, the Professional Web Tool (PWT) in .NET and its Database in a SQL Server.

The INTER-IoT Framework runs in its Virtual Machine and contains all INTER-IoT Modules needed for the pilot, of which the ones of interest to INTER-HEALTH are the INTER-IoT Middleware and INTER-IoT API.

The universAAL instance is an OSGi container with all the universAAL Modules needed for the pilot, of which the ones of interest to INTER-HEALTH are the REST API, and all the modules that compose the basic universAAL Middleware.

The .NET Framework hosts the PWT Web Application, which will finally allow healthcare professionals to manage all the data within the pilot.

The SQL Server hosts the Database used by the PWT to store its data.

The setup of the mobile phone used at the clinical centre differs from those used at each subject's home. The mobile phone used at the clinical centre is an Android Phone running the universAAL Android App and a dedicated app for getting measurements from Bluetooth devices. The mobile phone used by the subjects is an Android Phone running the BodyCloud Android App.

The Bluetooth devices used at the clinical centre differ from those used at each subject's home. The models used at the clinical centre are A&D Medical UA 767PBT (Blood Pressure) and A&D Medical UC 321PBT (Weight) which are regular Bluetooth devices. The models used by each subject are A&D Medical UA 651BLE (Blood Pressure) and A&D Medical UC 352BLE (Weight) which are Bluetooth Low Energy devices, in addition to the Xiaomi Band 2 (Physical Activity).

The PCs used by Healthcare Professionals at the clinical centre to access the PWT are their own regular PCs.

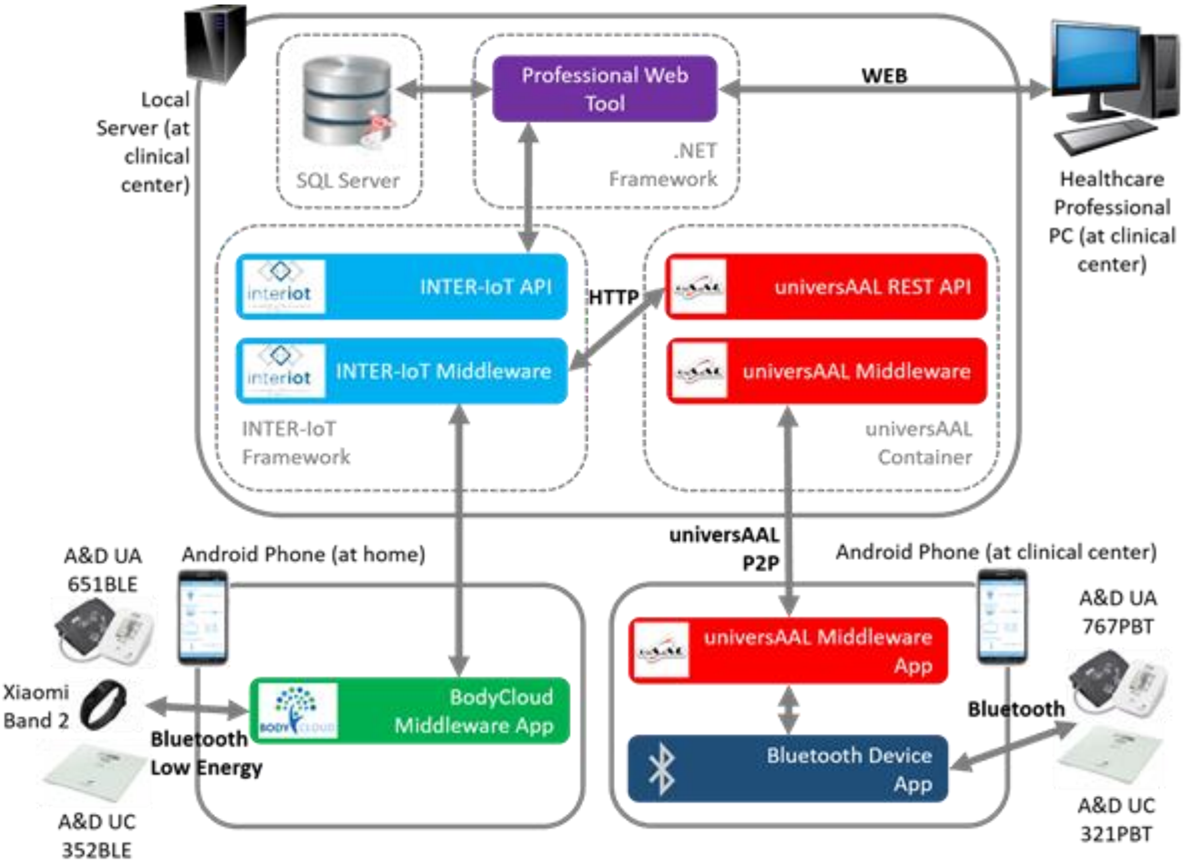


**Figure 5: INTER- Health System architecture**

The universAAL instance communicates with the INTER-IoT Framework by means of the universAAL REST API. The INTER-IoT Middleware contains the universAAL Bridge, which handles the communication towards universAAL through a REST API, which pushes communication back to the INTER-IoT Middleware. Through an HTTP callback endpoint the universAAL Bridge registered.

The PWT connects directly to the SQL Server through the dedicated port.

The PWT obtains all the subject measurement data from the INTER-IoT API, an API is used for accessing it.

The Healthcare Professionals use their PCs to open the PWT Web Application through a regular Web Browser. The PWT is only accessible by systems from within the local network.

Within the Android Phones used at the clinical centre, the dedicated Bluetooth Device App communicates the measurement data to the unviersAAL Middleware App through regular Android Intents.

The measurement devices (Blood Pressure and Weight) used at the clinical centre communicate with the Android Phone through regular Bluetooth, as managed by the Bluetooth Device App, that handles the connection and the data format encoding.

The Android Phones used at the clinical centre communicate with the local server through universAAL: The universAAL Android App connects to the local server universAAL instance either through regular automatic universAAL peer-to-peer communication or through the point-to-point universAAL Gateway. This connection is only allowed for devices in the local network. The Android Phones connect to the local network through Wi-Fi.

The measurement devices (Blood Pressure, Weight and Physical Activity) used at subject's home communicate with the Android Phone through Bluetooth Low Energy, as managed by the BodyCloud App, which handles the connection and the data format encoding.

The Android Phones used by subjects communicate with the local server through the BodyCloud Bridge in the INTER-IoT Middleware. This bridge opens a public endpoint to which the local server allows external connections. The BodyCloud app in the subject's phone connects to this endpoint to send the measurement data through the internet (using mobile network or any Wi-Fi access).

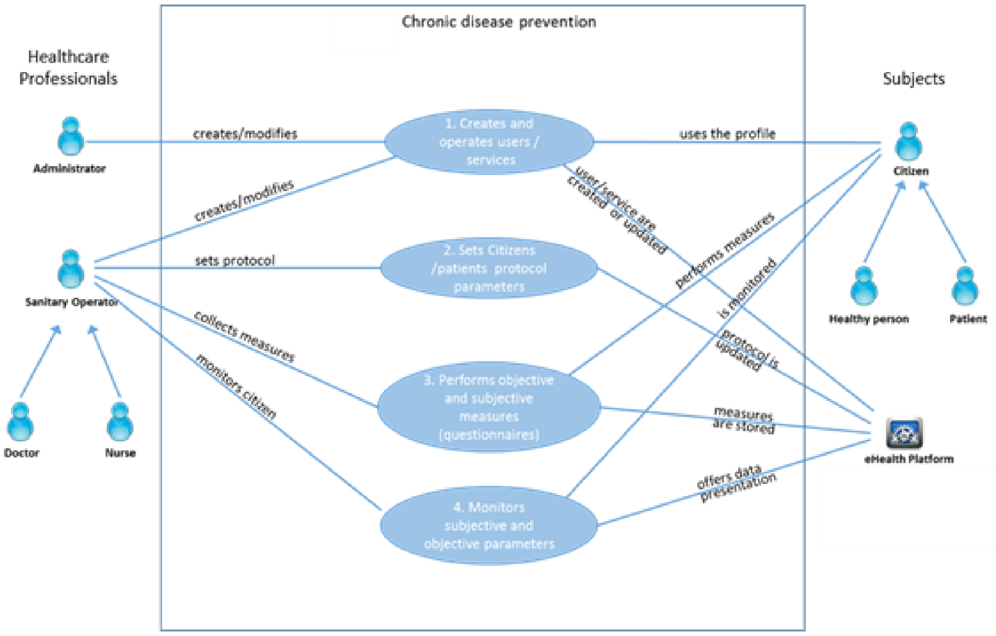


**Figure 6: INTER- Health Main use cases**

The “main overall” use case is Chronic Disease Prevention, which aggregates the most important use cases in the pilot:

- Create and operate users and associated services: Healthcare professionals can use the PWT to create users and manage subject’s and professional’s personal information.
- Set citizens/subjects protocol parameters (kind of measures, thresholds, periodicity): The healthcare professionals in charge of monitoring each subject progress can set up their prevention program.
- Performing objective measures (weight, blood pressure, activity) and subjective measures (lifestyle questionnaires): Subjects can use the IoT devices, managed by their mobile phones, to record their measurements, which will be uploaded to the system at ASL TO5. Subjects can fill in questionnaires about their nutritional habits, which will also be uploaded. During the visits to ASL TO5, healthcare professionals can also use the IoT devices and phones at their premises to upload this data which they can measure during the visit.

Monitoring measures and trends: Healthcare professionals can access the PWT to check the current status and history of the measured values for each subject, along with their questionnaire results.

## 2.3 Third party projects

In the following paragraphs the 12 open call pilots are described.

### 2.3.1 sensiNact integration

sensiNact is a horizontal platform dedicated to IoT and in particularly used in various smart city and smart home applications. sensiNact aims at managing IoT protocols and devices heterogeneity and provides synchronous (on demand) and asynchronous (periodic or event based) access to data/actions of IoT devices, as well as access to historic data with generic and easy-to-use API. To achieve these objectives, sensiNact comes with two complementary frameworks: sensiNact Platform interconnects IoT devices using different southbound IoT protocols such as Zigbee, EnOcean, LoRa, XBee, MQTT, XMPP, as well as platforms such as FIWARE and allows access to them with various northbound protocols such as HTTP REST, MQTT, XMPP, JSON RPC and CDMI. The gateway can also host applications and manage them using an application manager module. □ sensiNact Studio proposes an IDE (Integrated Development Environment) based on Eclipse to manage the existing devices, in addition to develop, deploy and manage IoT applications.

More information about sensiNact can be found at the http://open-platforms.eu portal, which is managed by the IoT-EPI initiative.

### 2.3.2 INTER-OM2M

In order to complement the measurements serving the logistics applications of the port (e.g. INTER-LogP), we will provide some tracking/environmental related measurements, featuring Sigfox/Lora as underlying radio technologies. These measurements will be made available through the oneM2M framework, maintained at Vrije Universiteit Brussel (VUB).

On top of that, some environmental measurements and measurements of the city of Valencia, available under the FIWARE framework will also be made available under the OM2M VUB platform to complement the measurements made by VUB itself. Also, the applications of the city of Valencia will be able to access and exploit the extra measurements produced by the equipment installed by VUB at the port. To this end, an INTER-IoT – OneM2M bridge will be developed by VUB. Figure 1.3.1 presents the efforts that VUB needs to do in relation with the use cases of the main consortium.

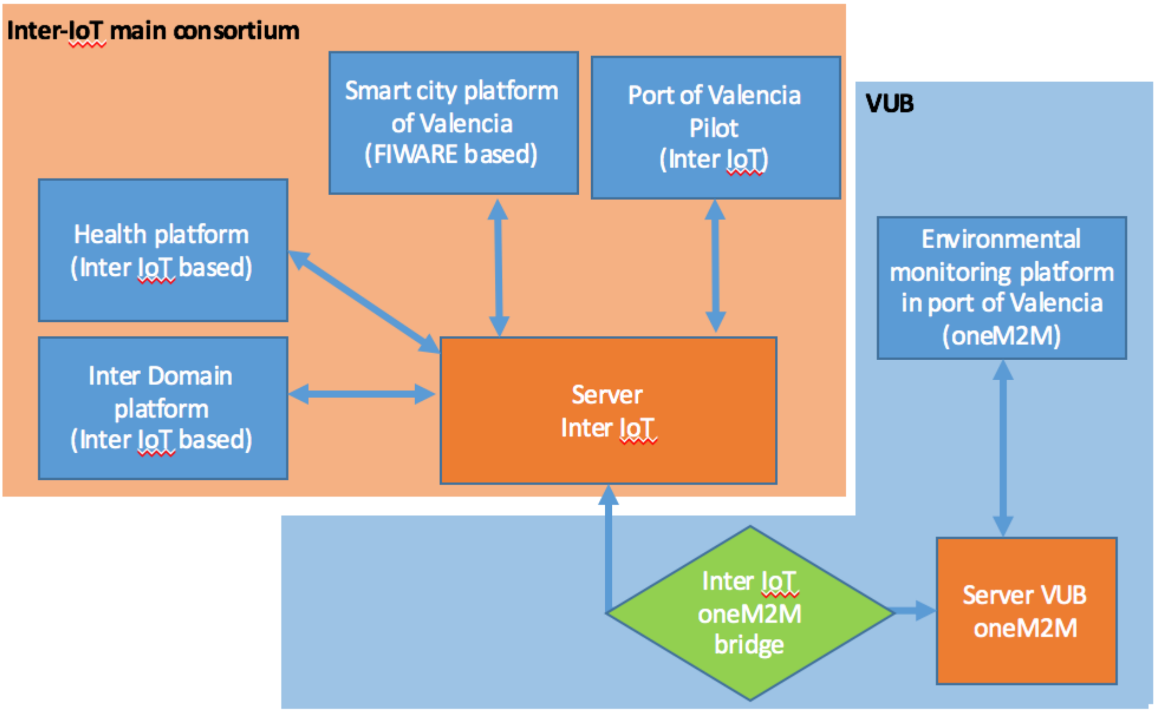

Inter-IoT main consortium
Smart city platform of Valencia (FIWARE based)
Port of Valencia Pilot (Inter IoT)
VUB
Health platform (Inter IoT based)
Environmental monitoring platform in port of Valencia (oneM2M)
Inter Domain platform (Inter IoT based)
Server Inter IoT
Inter IoT oneM2M bridge
Server VUB oneM2M

Figure 7: oneM2M system integration overview

### 2.3.3 INTER-HARE

The **INTER-HARE platform (*Integration of multiband IoT technologies*)** sets out to create an IoT platform based on a cluster-tree network, where a low-power wide area network (LPWAN) acts not only as data collector, but also as backhaul network for several so-called low-power local area networks (LPLANs), as shown in Figure 7. The INTER-IoT gateway becomes then the central element of the LPWAN, with full vision of the whole platform, and entitling cluster-heads to manage their corresponding LPLAN in a hierarchic way.

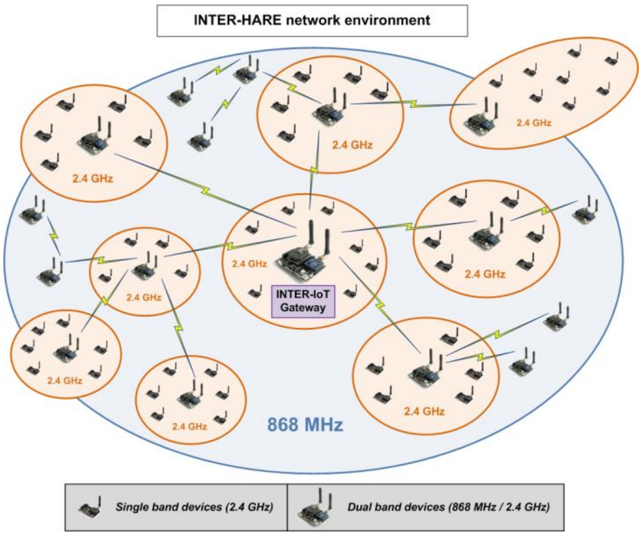

INTER-HARE network environment
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
2.4 GHz
INTER-IoT Gateway
868 MHz
Single band devices (2.4 GHz)
Dual band devices (868 MHz / 2.4 GHz)

Figure 8: INTER-HARE network environment

Communication within both the LPWAN and the LPLANs is based on the HARE protocol stack[1], ensuring transmission reliability, low energy consumption by adopting uplink multi-hop communication, self-organization, and resilience. Under these premises, LPWAN boundaries are extended beyond typical 868 MHz coverage range and easily integrate devices coming from adjacent/overlapping 2.4 GHz LPLANs. Use of separated frequency bands in overlapping networks results in an overall reduction of interferences. Lastly, thanks to the hierarchic system proposed, scalability is enforced by a management based on subnetworking techniques.

In line with the stated motivations, INTER-HARE main objectives are presented:

- To guarantee transparent network interoperability and unified, centralized, self-organized control of heterogeneous devices.
- To design, develop and test a novel end-to-end communication protocol between network devices (both from the LPWAN and from the LPLANs) and the central gateway, capable of periodically sending collected and/or simulated data.
- To build a simulation platform based on open-source technologies for preliminary testing of communication mechanisms.
- To create a preliminary testbed with heterogeneous devices in laboratory.
- To perform a pilot execution of the whole platform in a real IoT environment/use case.
- To collect statistics regarding network's resilience and self-organization as well as communication performance.

### 2.3.3.1 Collaboration approach with INTER-IoT

INTER-HARE directly aims at the INTER-LAYER (Methods and tools for providing interoperability among and across each layer of IoT platforms) building block. Specifically, INTER-HARE allows the INTER-IoT gateway to encompass under the same platform two technologies (LPWAN and LPLAN) working in different frequency bands (868 MHz and 2.4 GHz).

[1] T. Adame, S. Barrachina, B. Bellalta, and A. Bel. "HARE: Supporting efficient uplink multi-hop communication in self-organizing LPWANs". January 2017. https://arxiv.org/pdf/1701.04673v1.pdf Accessed: 2016-01-12

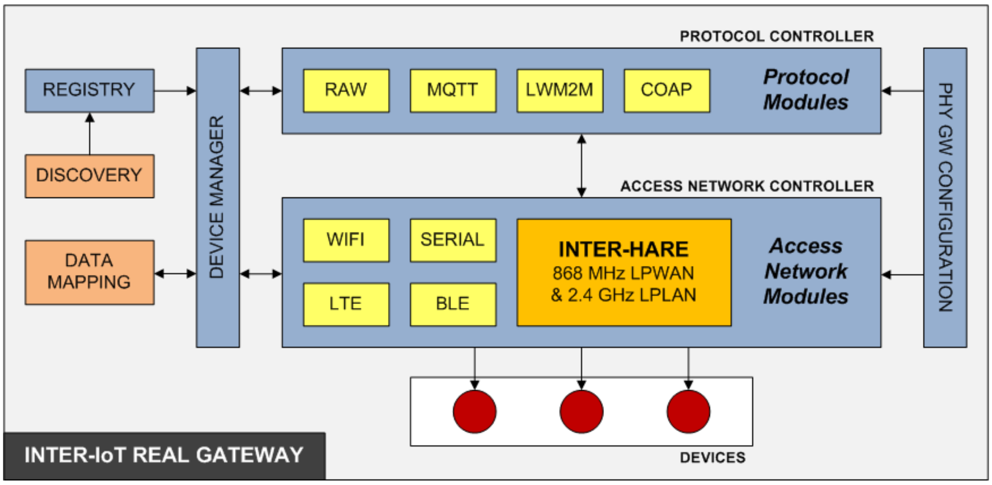


**Figure 9: INTER-HARE location within the INTER-IoT real gateway scheme**

As a novel access network module within the INTER-IoT real gateway (see Figure 9), INTER-HARE provides device-to-device interaction based on access mechanisms, being able to deliver the gathered information to the protocol controller.

Similarly, main configuration parameters of the INTER-HARE platform can be set from upper modules via proper connections. In line with the overall goal of the INTER-IoT project, INTER-HARE access network module allows seamless integration of multiband IoT technologies from different application domains with the requirements from Table 1.

*Table 1: Common requirements of INTER-HARE use cases*

| Requirement | Value | Requirement | Value |
|---|---|---|---|
| Coverage range | Up to several km. | Throughput | <100 bits/s |
| Geographic coverage | Excellent even in remote and rural areas | Latency | Non-delay-sensitive |
| Penetration | Good in-building and in-ground penetration | Mobility | Static devices |
| Device density (per base station) | High (up to thousand) | Cost | Low hardware and operating cost |
| Power profile | Unassisted, battery-powered devices | Maintenance | Unassisted and self-organizing network |
| Battery lifetime | From some months up to several years | Delivery model | Continuous data delivery model |

### 2.3.4 Mission Critical operations based on IoT analytics

The anticipated benefits of interoperable “Mission Critical operations based on IoT analytics” (MiCrOBIoTa) are unquestionable. A typical situation in mission critical operations support systems is to include information coming from specifically deployed devices to gather environmental measurements. Examples of these devices are temperature sensors, meteorological and hydrological probes, traffic monitoring cameras, etc. We propose to add the MC-IoT system, which includes a new monitoring and analytics component and an evolved Control Room interface tailored to the specific needs of the use case. In the case of a simulated crisis, significant information from on-body health-related sensors and port logistics devices will provide life-saving information to the mission critical operations support system. Besides, the available mission critical communications components can be used to demonstrate the crisis handling use case.

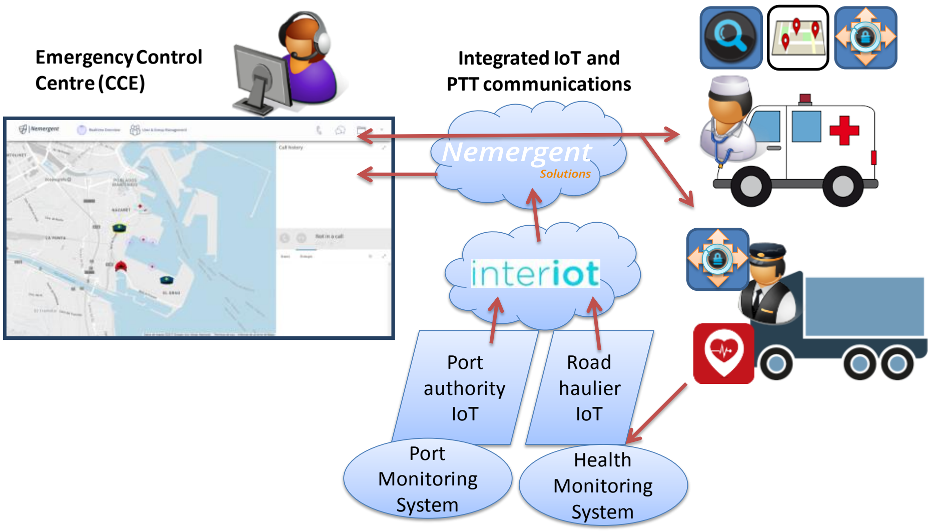


Figure 10: IoT-aided Mission Critical operations scenario

Taking into account the overall picture and the availability of different IoT platforms, a complex use case could be created for an emergency simulation exercise. This scenario would include a typical emergency intervention, enhancing the operations support through the use of new communication technologies over commercial networks.

An example operational procedure is provided hereafter:

1. A road haulier comes into the port area. Upon an incident / health issue, the on-board health monitoring sensor reports the anomalous data to the INTER-IoT system through the road haulier company IoT platform.
2. The relevant data arrives at the Port Authority emergency control centre (CCE), which manages incidents taking place within the port and coordinates with other first responders (police, firefighters, ambulances, etc.).
3. The CCE operator accesses the MC-IoT system through the web-based GUI, which will provide different types of icons for the different sources of information, and different views targeted at different emergency response units. Potential sources of information are
4. The CCE operator can use this platform to communicate with field response units (e.g., ambulance driver), providing them not only with location and navigation support but also with specific context information useful for the intervention.
5. Besides the IoT-related data processing, the extended use case will make use of the Nemergent Mission-Critical Push-To-Talk (MCPTT) communication systems in order to resemble real-time communication between the different entities involved.

### 2.3.5 Interoperable Situation-Aware IoT-Based Early Warning System

The goal of the project is to support the semantic and syntactic interoperability among IoT artifacts and early warning systems (EWS), i.e. to enable data to be understandable for both sender and receiver. In particular, we focus on coordinating emergency services based on IoT devices, alerting the involved parties (e.g. emergency command control, first responders and employees) when an accident occurs.

INTER-IoT-EWS will be deployed at the application layer of the INTER-IoT framework, reusing the available services offered by the AS2AS layer, to detect accidents at the port area, i.e. scenario number 9 of INTER-LogP/Health (cross domain). Therefore, we aim at creating composite services on top of AS2AS layer, following the INTER-IoT reference model. When necessary, the INTER-IoT-EWS will consume messages that have been translated by the IPSM using a dedicated component exposed in the AS2AS layer to enable the integration among IoT platforms and devices using different ontologies. In particular, the INTER-IoT-EWS project collaboration will provide semantic translations between the W3C Semantic Sensor Network (SSN) and the Smart Appliances REFerence (SAREF) ontologies to be configured within IPSM. Therefore, the IoT platforms and/or devices used in this scenario shall be semantically enriched with ontology representations, especially with SSN and SAREF, but can also make use of the IPSM central ontology. Figure 12 illustrates INTER-IoT-EWS collaboration approach.

The EWS aims at addressing the scenario of accidents at the port area, which requires integration between logistics and health domain, by integrating data provided by devices within trucks delivering goods at the port. These data providers include devices carried by the truck’s drivers, such as wearable medical devices and mobile phones, providing the location of the truck, accelerometer and driver’s vital signs. These data will be integrated with data about incidents at the port, which can be extracted from the port authority IoT platform, and data about the goods transported and the terminal to be shipped, which can be extracted from the terminal IoT platform.

To support the medical devices, an IoT platform will be used, i.e. the INTER-HEALTH solution from UNICAL. This solution is based on the BodyCloud architecture, which comprises four modules: body, cloud, analyst and viewer. Body module includes medical wearable devices communicating with Android-based mobile phone (via Bluetooth) and is implemented with the SPINE framework. “SPINE (Signal Processing in Node Environment) is a software Framework for the design and fast prototyping of Wireless Body Sensor Network (BSN) applications. SPINE enables efficient implementations of signal processing algorithms for analysis and classification of sensor data through libraries of processing functionalities. It also embed an application-level communication protocol. SPINE is organized in two interacting macro-components, which are respectively implemented on commercially available sensor devices and on the personal coordinator (such as an Android smart-phone or tablet, or a personal computer). Communication among these devices is wireless, using the Bluetooth 2.1 or IEEE 802.15.4 standards”.

SPINE is compatible with the Shimmer (www.shimmersensing.com) devices’ family (rev. 1.3 and 2R). An application based on the TinyOS (version 2.1) operational system, used by Shimmer devices, enables the real-time data transfer (through Bluetooth) from the device to a mobile running the SPINE’s data collector. This approach follows the examples of applications that can be deployed within the Shimmer device microcontroller. These application examples provided by Shimmer can also be useful for detecting vehicle collisions based on accelerometer data, such as the “apps/SimpleAccel”, which streams accelerometer data over Bluetooth to a mobile phone. One of the main Shimmer devices supported by the SPINE framework is the Shimmer ECG for cardiac monitoring. The Shimmer 3 ECG functions include electrocardiogram, respiration, accelerometer and gyroscope.

The communication between the body module and the cloud module of UNICAL solution includes an Android application (the body side) that streams data to a server (the cloud side) with the UniversAAL open source IoT platform. Initially, it was planned to use the cloud module of BodyCloud, but it is based on the Google App Engine, which brings up health data privacy issues. Therefore, UniversAAL platform was chosen to replace this cloud module. UniversAAL is a not-for-profit project (open source) that enables seamless interoperability of devices, services and applications through rapid development of innovative IoT solutions. The rapid evolution of UniversAAL IoT ensures that new features and applications are regularly added to integrated systems, bringing possibilities to the middleware platform. “With users able to effortlessly share valuable data between devices and systems, there is the flexibility and versatility to build a completely unique lifestyle environment. Interoperability is now a possibility”. These data will be consumed by the EWS by subscribing to the equivalent service exposed by the AS2AS layer.

Data regarding logistics information can be provided by the mobile phone used by the driver, the truck’s tachograph and GPS. This data will be accessible through the IoT haulier platform , with services exposed in the AS2AS layer.

The Valencia port emergency control centre (CCE) is the main stakeholder in this scenario, since it is the responsible to respond emergencies at the port area, thus, it should be notified if an accident is predicted (risk identified) and/or when an accident is detected by the INTER-IoT-EWS. The notification services of the EWS will be included in the AS2AS services catalogue as a publisher informing when accident is either predicted or detected. This service will follow OASIS EDXL standards, mainly EDXL-SitRep, EDXL-CAP and EDXL-TEP.

The major risk to INTER-IoT-EWS project is the lack of definition of the haulier IoT platform, since the requirements to be achieved by the project state the need of integrating this platform with others. To overcome this risk, we suggest to integrate:

- Health (driver) data: UNICAL platform (with Shimmer device)

- Logistics data:

*Table 2: Device overview*

| **Data type** | **Device and/or gateway** | **IoT platform** |
|---|---|---|
| Location | Android mobile app (open source) | Port authority IoT platform<br>Haulier IoT platform |
| Accelerometer | Shimmer ECG (UNICAL)<br>Android mobile app (open source) | UNICAL IoT platform<br>Haulier IoT platform |
| Speed | Android mobile app (open source) | Haulier IoT platform |
| Truck routes (destination) | | Terminal IoT platform |

The Android mobile app can play the role of data provider of the haulier IoT platform and may be implemented with a broker receiving data from the mobile and exposing this data for the node-RED approach (via REST/JSON-LD).

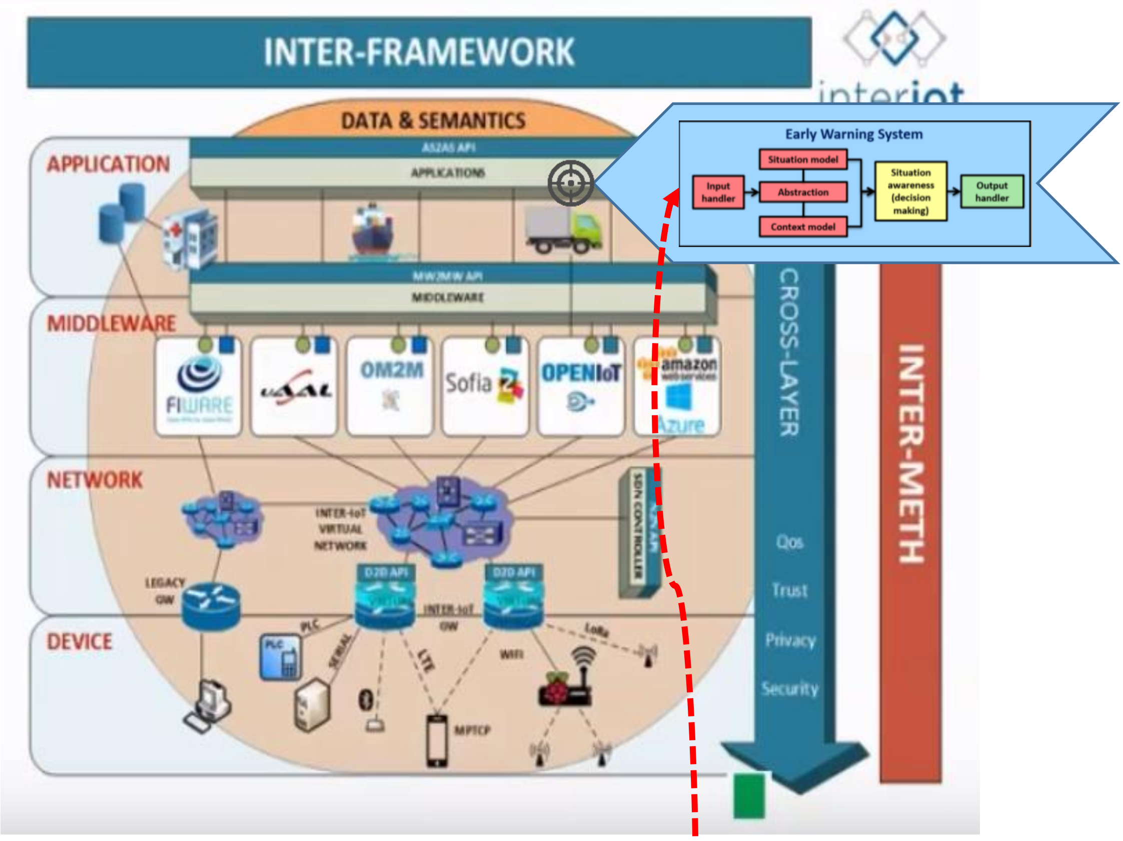


Figure 11: Early warning system in architecture

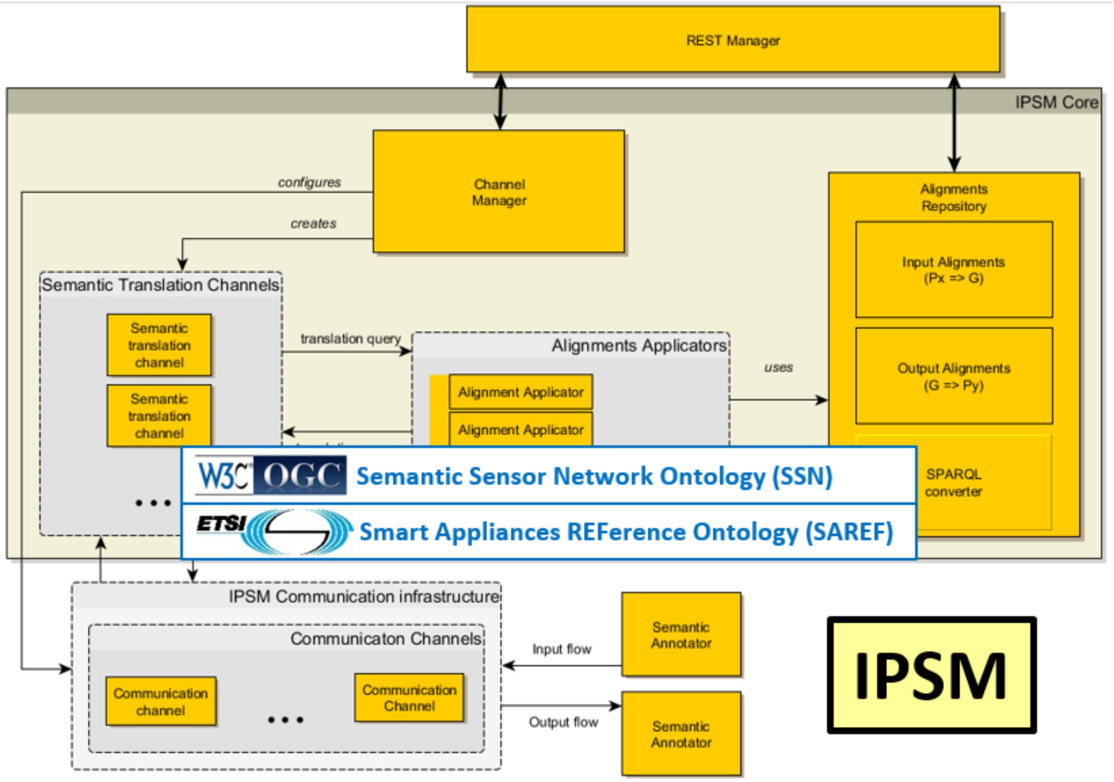


Figure 12: INTER-IoT-EWS collaboration approach

### 2.3.6 SENSHOOK

The port of Valencia is one of the most important hubs in the world and thus a critical point of entry of invasive species that must be monitored, according to the European Centre of Disease Control.

The pilot will consist on deploying a surveillance network of 5 observation static IoT nodes in critical points of the port of Valencia.

Each node is composed of a Smart Mosquito Trap capable of mimicking the human body (scent and respiration) and of automatically counting captured mosquitoes, identify the gender and the species. The information collected by each node is then sent to a server.

The pilot will start in May-June 2018 and will last until October 2018. This corresponds exactly with the period of the year when disease-vector mosquitoes are active and must be monitored.

Following is a diagram that gives an high-level overview of the system. When a mosquito enters the trap it gets detected by the sensor which also captures the necessary data to identify it.

The sensor is connected to a Senscape board which sends the information to the server running Senshook.

A client can retrieve the gathered information.

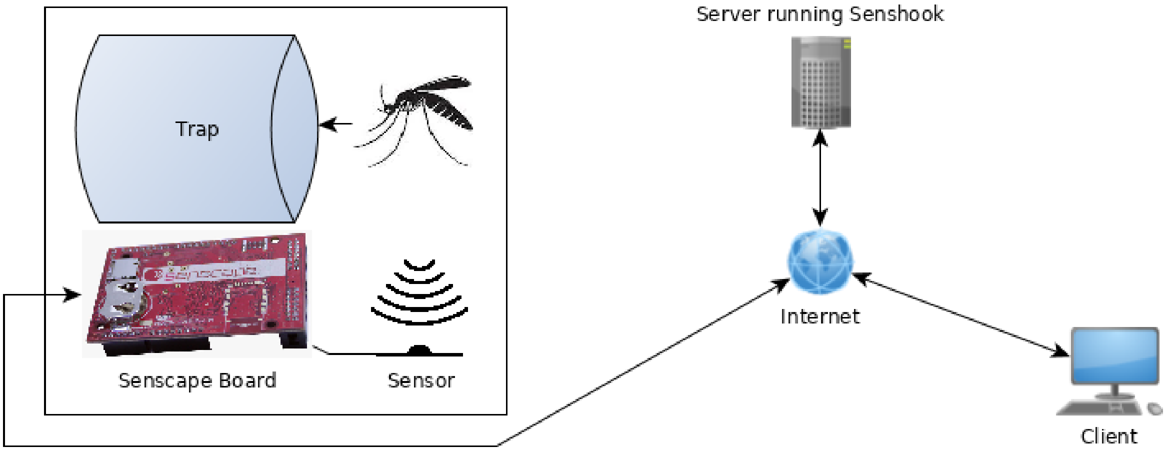


Figure 13: Architectural overview of Senshook

#### 2.3.6.1 Objectives of the project

The specific objectives of the project are to:

- Perform a technical feasibility assessment of the Senshook solutions as part of the INTER-IoT project
- Implement Senshook according to INTER-IoT requirements
- Carry out a series of tests/pilots to evaluate the performance and benefits of the tool.

#### 2.3.6.2 Collaboration approach

Irideon will contribute to the INTER-IoT project by providing a new open tool for the INTER-LAYER building block, which will allow the evolution of products based on INTER-IoT, but at the same time will allow us to evolve our products in order to add new interoperability features.

By contributing to the development of INTER-IoT, Irideon will be able to address new IoT scenarios in which different IoT platforms, apart from those based on Senscape, are involved, and also in those in which more than one application domain is addressed.

#### 2.3.6.3 Senshook Architecture Overview

The system consists of the Senscape hardware and a D2D virtual gateway which provides connection to the middleware platform.
Following is a description of the different components of the gateway.

Dispatcher
The central part of the virtual gateway is the dispatcher, it consists of a communication layer which connects to the hardware via MQTT and a service layer which implements the IEEE 1451 standard.
IEEE 1451 is a set of smart transducer interface standards developed by the Institute of Electrical and Electronics Engineers (IEEE).
Irideon has developed a specific lightweight implementation of this standard for the Senscape hardware devices.
This implementation includes communication protocols, transducer electronic data sheet (TEDS) and common functions.
The dispatcher is implemented in two OSGi bundles:

- Dispatcher API bundle: This bundle defines and exports the interface.
- Dispatcher provider bundle: This bundle implements the service layer, which includes the IEEE 1451 standard and the communication layer with the MQTT connector. Furthermore it registers a service with the dispatcher API.

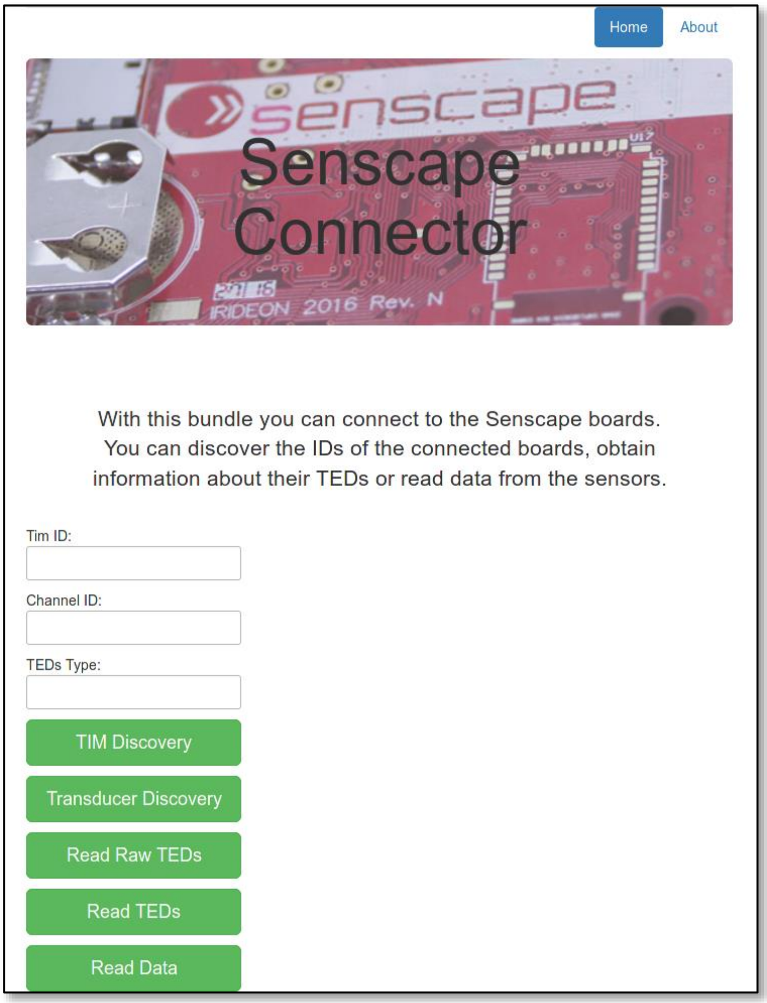
Middleware Controller
The middleware controller consists of the three bundles:

- Middleware controller API Bundle: This bundle defines the OSGi interface for the bundle and exports it as a package.
- Middleware Controller Provider Bundle: This bundle implements the middleware controller and provides it as a OSGi service with the middleware controller API.
- Middleware Controller Application Bundle: This bundle consumes the service published by the provider and implements an application that exposes a REST API.

Measure Storage
The measure storage has two bundles:

- Measure storage API Bundle: This bundle defines the interface for the bundle and exports it as a package.
- Measure storage provider bundle: This bundle provides the connection to a local database to store and retrieve the values sent and requested by the dispatcher. This bundle implements the interface defined by the API bundle and registers it as a service.

API / Web Application
This component provides a REST API to the dispatcher and also includes a web application which serves as a front-end to interact with the Senscape hardware.
It consists of one bundle which is a OSGi application that consumes the service offered by the dispatcher.
Below you can see a screen shot of the web front-end:



With this bundle you can connect to the Senscape boards. You can discover the IDs of the connected boards, obtain information about their TEDs or read data from the sensors.

Tim ID:

Channel ID:

TEDs Type:

And following, a diagram of the virtual gateway with their components and respective bundles:

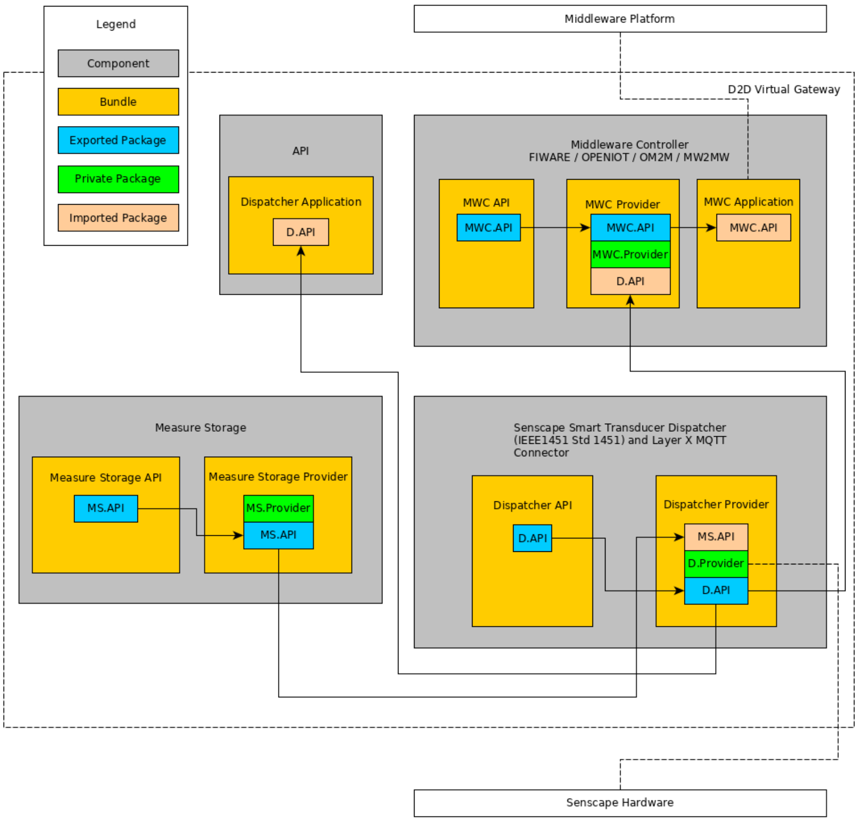


Figure 14: Architectural overview

### 2.3.6.4 System interfaces

The virtual gateway offers interfaces to two of its components, the middleware controller and the dispatcher.

Dispatcher

The dispatcher comes with four interfaces:

- Terminal interface: This interface features an Apache Felix Gogo Shell. It is thought for debugging purposes.
- OSGi interface: The component registers an OSGi service API which can be used to incorporate the dispatcher in a modular OSGi application.
- REST API: The dispatcher also offers a REST API, so it is possible to communicate with it over the Internet making it possible to use it in a web application.
- Web front-end: Serves for testing and demonstration.

Middleware Controller

The middleware controller comes with three interfaces:

- Terminal interface: This interface features a gogo shell. It is thought for debugging purposes.
- OSGi interface: The component registers an OSGi service API which can be used to incorporate the middleware in a modular OSGi application.
- REST API: The middleware controller also offers a REST API, so it is possible to communicate with the component over the Internet making it possible to use it in a web application.

### 2.3.7 SOFOS

The imminent arrival of the Internet of Things (IoT), which consists of a vast variety of devices with heterogeneous characteristics, means that future networks need a new architecture to accommodate end-to-end IoT networking, dealing with: i) the expected increase in data generation, ii) the problems related to the end-to-end IP networking of the resource-constrained IoT devices, iii) the capacity mismatch between devices, and iv) the rapid interaction between services and infrastructure.

Software defined networking (SDN) and network function virtualization (NFV) are two technologies that promise to cost-effectively provide the scale and versatility necessary for IoT services in order to address efficiently the aforementioned challenges. Moreover, given that SDN and NFV are considered a fundamental component in the 5G landscape, since it is widely recognized that 5G networks will be software-driven and most components of future heterogeneous 5G architectures should be capable to support software-network technologies, both SDN and NFV are promising candidate technologies for a Software Defined Approach of end-to-end IoT Networking.

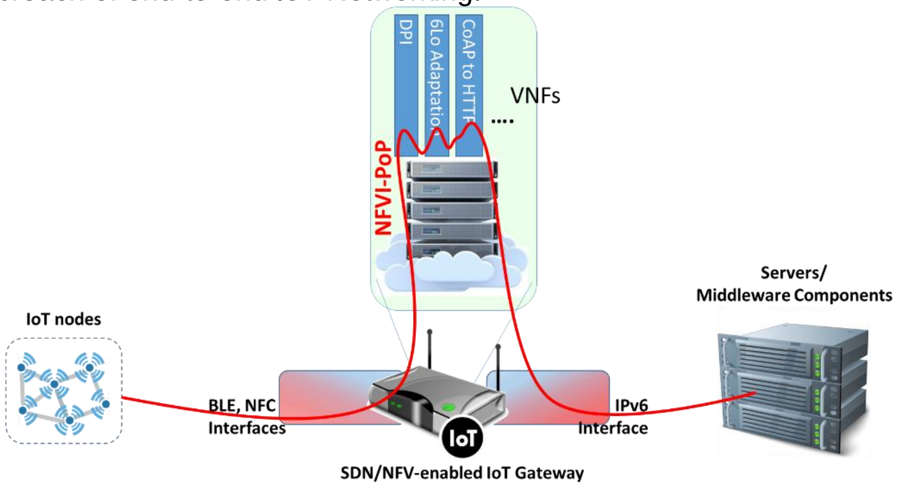


**Figure 15: The proposed SDN/NFV end-to-end IoT Gateway overview**

SOFOS aims at advancing the existing INTER-IoT framework with SDN and NFV functionalities towards a Software-defined end-to-end IoT infrastructure with IoT service chaining support. The main objective of the proposed SDN/NFV-enabled framework is to enhance the interoperability of the INTER-IoT framework in order to facilitate the interoperable management of a large number of diverse smart objects that currently operate utilizing a variety of different IoT protocols.

In this framework, specific objectives of the proposal include:

- To add SDN/NFV Automation and Verification in IoT Infrastructure
- To relocate various IoT functions from HW appliances to Virtual Machines (VMs) (i.e. Virtual Network Functions - VNFs).
- To enhance the interoperability support of the INTER-IoT platform by deploying VNFs that map IoT protocols (such as 6Lo, CoAP, DICE, ACE, etc…) to standard IP networking
- To connect and chain the software-defined IoT functions (i.e. VNFs) together.
- To abstract the IoT's control plane by exploiting the SDN concept and advances.
- Inter-IoT Infrastructure with the proposed advances can be enhanced by means of NFV with integration of SDN, making it more agile and introducing a high degree of automation in service delivery and operation—from dynamic IoT service parameter exposure and negotiation to resource allocation, service fulfillment, and assurance.

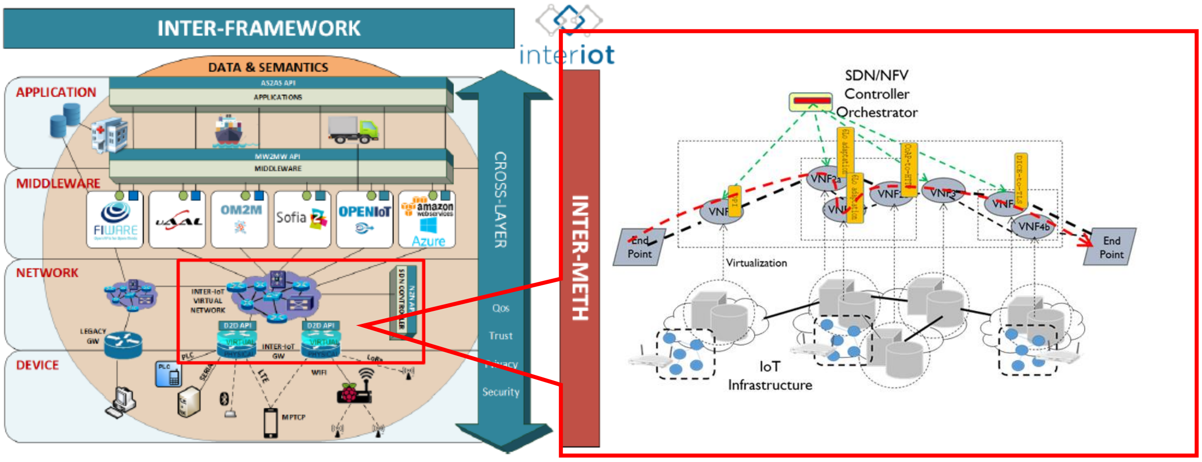


**Figure 16: Collaboration approach of the proposed SDN/NFV end-to-end IoT infrastructure within the INTER-IoT architecture.**

More specifically, the proposed advances are aiming at enhancing the INTER-layer of the INTER-Framework by adding SDN/NFV orchestration capabilities. In specific, the proposal focuses on integrating SDN/NFV Controller/Orchestrator (such as OpenDaylight SDN Controller and OpenBaton NFV Orchestrator) at the INTER-IoT GW, allowing the unified management of the INTER-IoT virtual network by appropriate service chaining between physical smart objects and virtual functions as Figure 16 depicts.

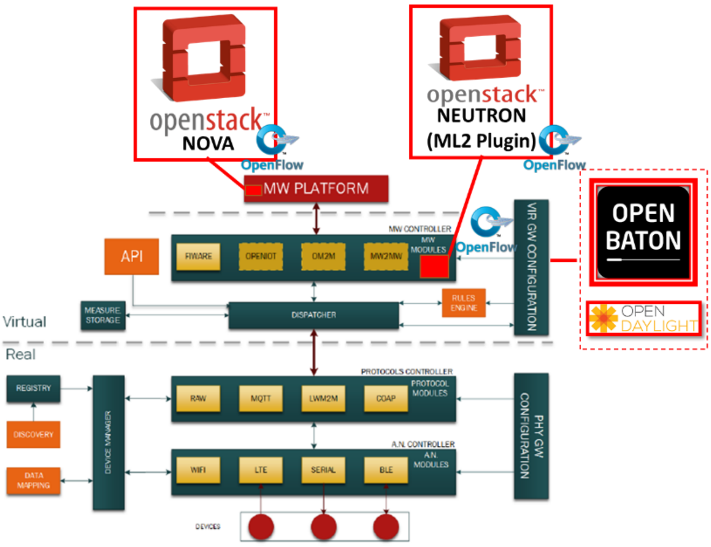


**Figure 17: Proposed SDN/NFV components forming INTER-IoT architecture.**

Figure 17 depicts a more detailed collaboration approach of the proposed SDN/NFV framework with the INTER-IoT platform, highlighting in red color the integration/positioning of the proposed SDN/NFV within INTER-IoT platform. For clarity reasons in Figure 3 the most preferable platforms have been used, namely the OpenBATON NFV Orchestrator, the OpenDaylight SDN Controller, and the OpenStack Cloud Computing Platform, although alternative platforms may be used, such as the INFOLYSiS NFV Orchestrator (depending on the compatibility with Inter-IoT platform). Thus, the proposal envisages the enhancement of the INTER-IoT FW with NFV and SDN orchestration capabilities through an appropriate integration of the NFV Orchestrator and the OpenDaylight platform with the INTER-IoT VR GW Coordinator.

### 2.3.8 E3Tcity Smart City Platform and Devices Integration

The system and used architecture overview (solution, usage of the IoT framework and integration interfaces)

E3Tcity devices connect port lights to the cloud so that light systems will be controlled from InterIoT. Smart behaviour will be given to lights so they can improve visibility conditions in the facility.

Additionally, several sensors, such as presence detectors, light level measurements and environmental condition sensors can be connected to devices to add functionality to the system.

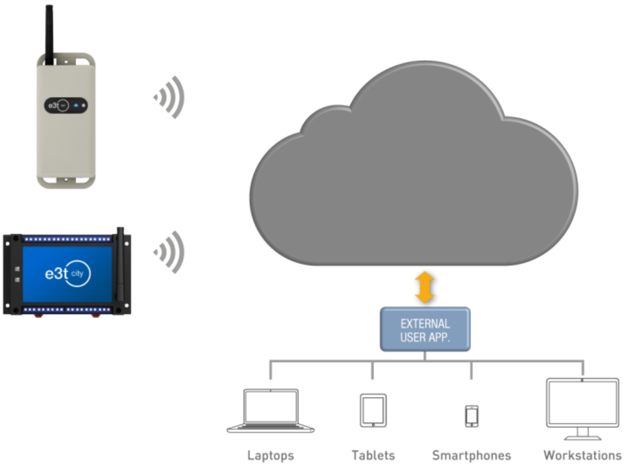


**Figure 18: Integration overview**

This is a schematic on the architecture of the solution:

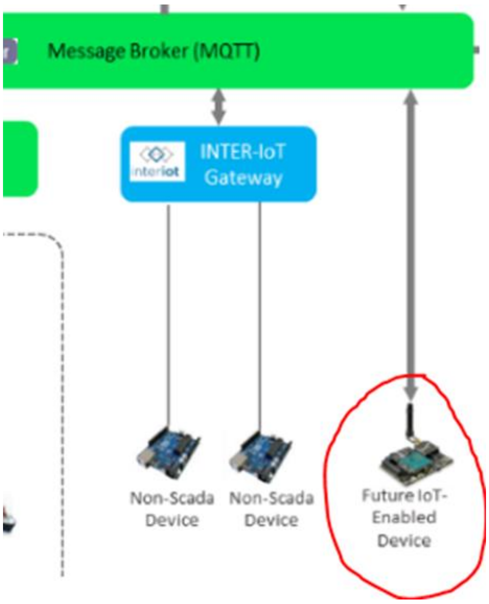


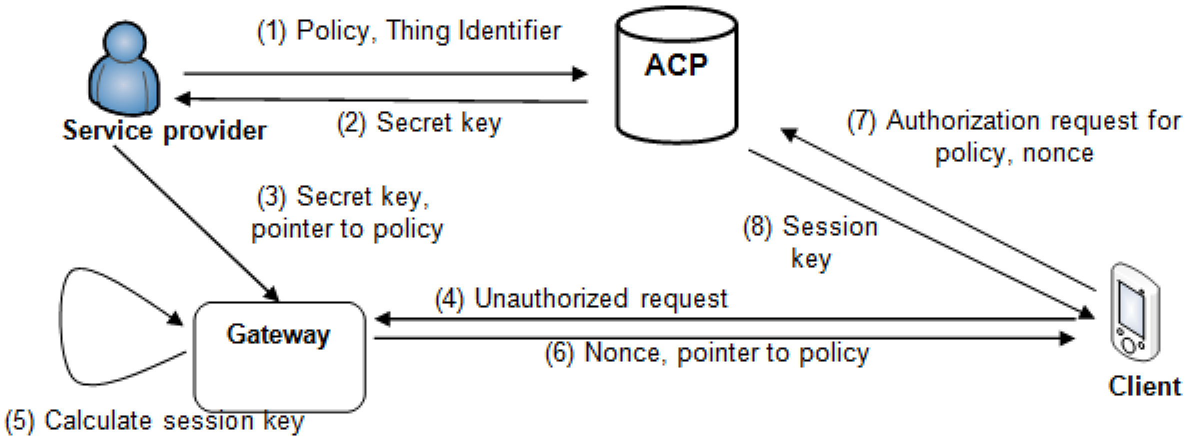
**Figure 19: Sensors connected directly to the MQTT broker**

Due to the current architecture which includes the MQTT protocol, devices fit exactly on this part of the port pilot.

### 2.3.9 ACHILLES

Access control and endpoint authentication in the IoT is a challenging problem. Things are usually small devices with limited storage capacity, power, energy, and processing capabilities, in order to be inexpensive and practical. In many cases Things are “exposed” to tampering, whereas in many application scenarios, after Things are deployed, it is not easy to access them. Things usually are not able to perform “heavy'” tasks, such as complex cryptographic operations. Storing user credentials or any other sensitive information in a Thing creates security risks, adds storage overhead, and makes security management an impossible task. When it comes to interoperable applications, Things (or even gateways) cannot interpret complex business roles and processes. Moreover, companies are not willing to share sensitive information about their users with a Thing (or a gateway), even if this information is required by an access control mechanism, neither do they want to invest in yet another security system.

The ACHILLES project overcomes these limitations by allowing the delegation of security operations to a third party, referred to as the Access Control Provider (ACP), which can be implemented by a trusted separate entity, or even the service provider itself. The ACHILLES concept is depicted in Figure 20.



(1) Policy, Thing Identifier
ACP
Service provider
(2) Secret key
(7) Authorization request for policy, nonce
(3) Secret key, pointer to policy
(8) Session key
(4) Unauthorized request
Gateway
(6) Nonce, pointer to policy
Client
(5) Calculate session key

**Figure 20: ACHILLES overview**

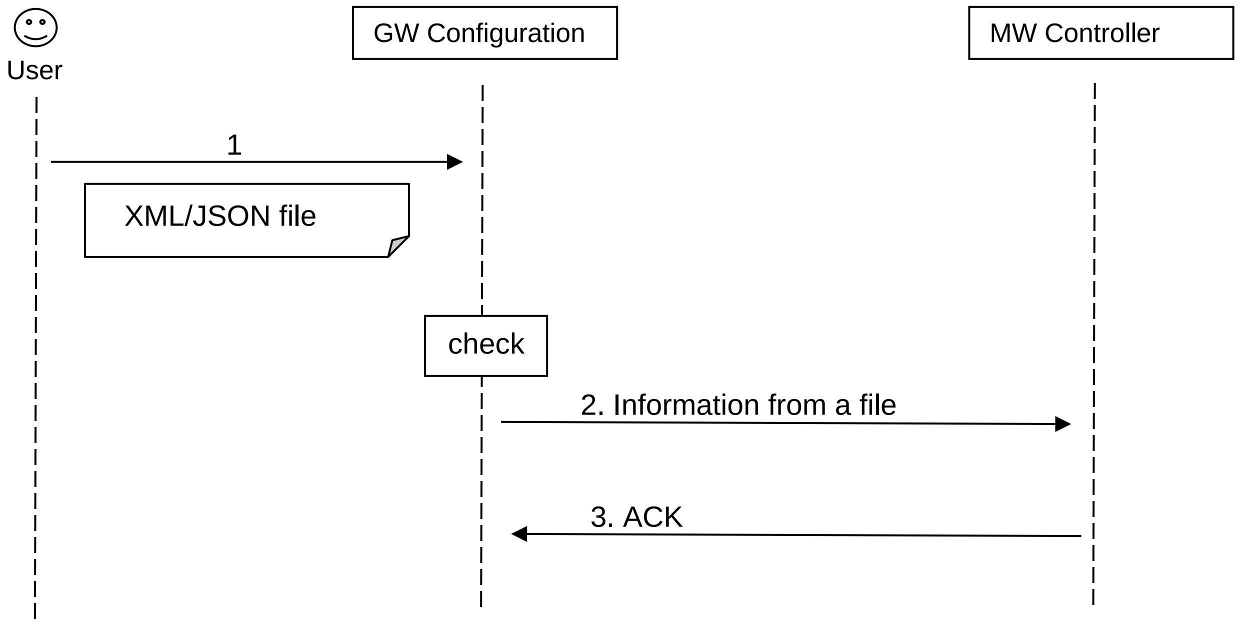
The main idea of the ACHILLES concept is that IoT service providers store access control policies in ACPs and in return ACPs generate secret keys which are stored in Things. These keys are generated, during a setup phase, using a secure hash with input the Thing identifier. Additionally, Things are configured with *pointers* (e.g., a URL that points to an ACP and a particular file) to the access control policies that protect sensitive resources. Every time a client requests access to a protected resource the Thing uses a secure hash function to generate a session key. The secret key used by that function is the key generated by the ACP and the hash inputs are: (a) the pointer to the policy that protects the resource and (b) a random nonce. The Thing transmits the nonce and the pointer to the client, which in return requests authorization from the appropriate ACP (over a secure channel). The ACP has all the necessary information required to calculate the session key: *if the client is authorized*, the ACP calculates the session key and transmits it back to the client. Providing that: (i) the Thing has not lied about its identity and (ii) the messages exchanged between the client and the Thing have not been modified, the Thing and the client end up sharing a secret key. This key can be used for securing subsequent communications (e.g., by using DTLS).

The ACHILLES project will extend the INTER-IoT gateway by providing new protocol modules. These modules will allow (i) gateway-Thing mutual authentication using the ACHILLES approach, (ii) access to CoAP resources protected by ACHILLES, (iii) support for CoAP over DTLS, using the pre-shared key ciphersuite for Transport Layer Security (TLS).

The ACHILLES project will provide software for Things, end-user applications, **Java OSGi modules for the INTER-IoT gateway**, and it will leverage **INTER-FW API** and **Tools** so that end-users will be able to create, modify, and access protected services**.** The ACHILLES OSGi modules will target the **D2D layer of the INTER-LAYER.**

We now define some use cases to illustrate with which INTER-IoT components our project will interact. These cases are based on the cases defined by the INTER-IoT project in deliverable D3.1. Modifications are highlighted with text in **bold.**

| Use case | A platform supporting ACHILLES is configured on the Gateway (extends #61) |
|---|---|
| **Description** | The configuration of a platform that will be connected to the INTER-IoT gateway and will send/receive **protected information** from/to the devices.<br>The configuration can be done using a configuration file that will include, among other things, a list of ACPs |
| **Objectives** | To configure correctly the platform so to be able to receive protected information, as well as to be able to send information to devices and configure access control policies. |
| **Components Involved** | The MW Controller Module.<br>The MW specific platform Module.<br>The GW Configuration Module |



**Step 1:** A user starts the gateway framework

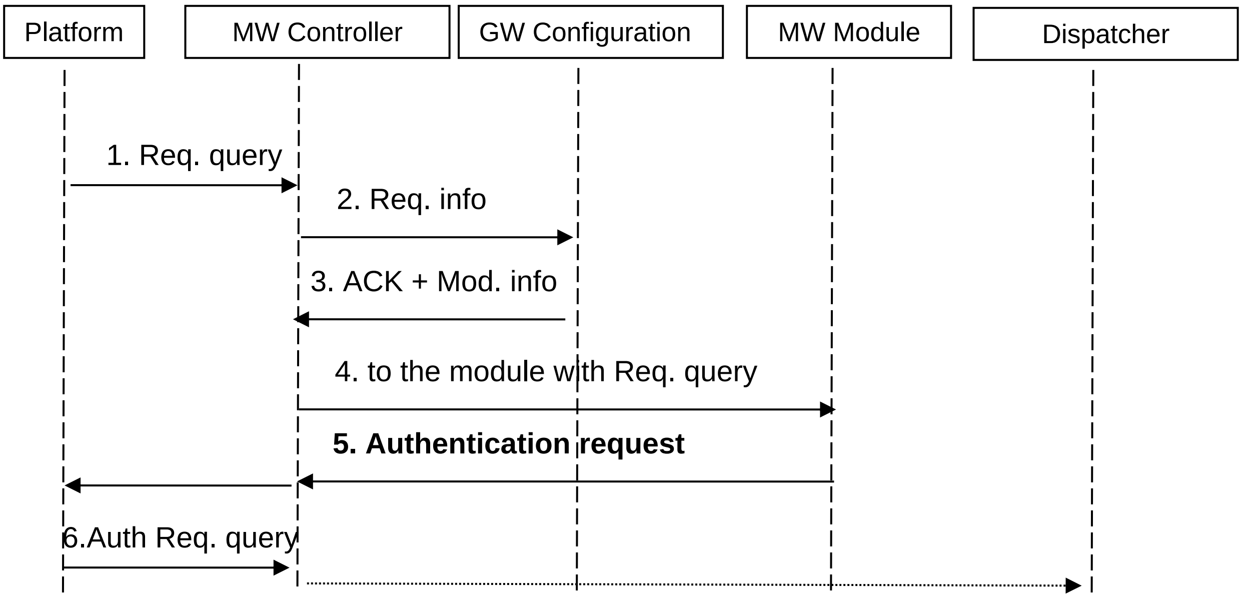
**Step 2:** The GW configuration module is activated and reads form a file all the gateway configuration entries. **These entries include lists of policies and ACPs.**

**Step 3:** The MW Controller is activated and gets from the Configuration Module all the information related with the configuration of the MW platform, **including ACHILLES related information.**

**Step 4:** The MW Controller performs a test of communication with the MW platform **and the ACPs** and throws an Exception if there is a problem.

| Use case | Platform requests protected information from a device (sensor) (extends #63) |
|---|---|
| **Description** | The Gateway receives the request from the Platform. **If the request is for a protected resource and the platform is not authorized, it redirects it to an ACP.** Otherwise it re-directs it to the Device, to obtain specific information.<br>If no change in the value has been performed in a short period, the response will be provided directly from the Measurement Storage. |
| **Objectives** | To obtain an information item requested by the **authorized** Platform form a concrete Device. |
| **Components Involved** | The platform module connected to the IoT platform.<br>The Platform Controller.<br>The Dispatcher. |

| | |
|---|---|
| | The Measurement Storage.<br>The Protocol Module and Controller.<br>The AN Controller and Module. |



**Step 1:** The platform sends a query requesting **a protected resource** from a device.

**Step 2:** The request is handled by the Middleware Controller

**Step 3:** The MW Controller consults in the Gateway Configuration Module about the bundle in charge of manage and parse the request message.

**Step 4:** The Gateway Configuration Module informs the Controller about with bundle is adequate.

**Step 5:** The request message arrives to the MW Module; **the module determines that the platform is not authorized and issues an authentication request, which eventually reaches the platform**.

Step 6: The platform obtains an authorization token form an ACP and sends an authorized request. The rest of the messages are the same.

### 2.3.10 INTER-HINC

The following figure describes the current design of INTER-HINC, which fits well into the design of the Inter-IoT framework. Two main components that will interact with other IoT platforms and services outside the Inter-IoT framework are:

- LocalManagementService: instances of Local Management Service are used to interface to IoT providers. In our case, the providers provide access to IoT devices in terminal (Noatum), in Trucks (Hauliers), and in Ports (using other INTER-IoT middleware)
- Global Management Service: instances of Global Management Server are used to the application and other middleware to use INTER-HINC to control IoT devices, networks and services and acquire IoT data.

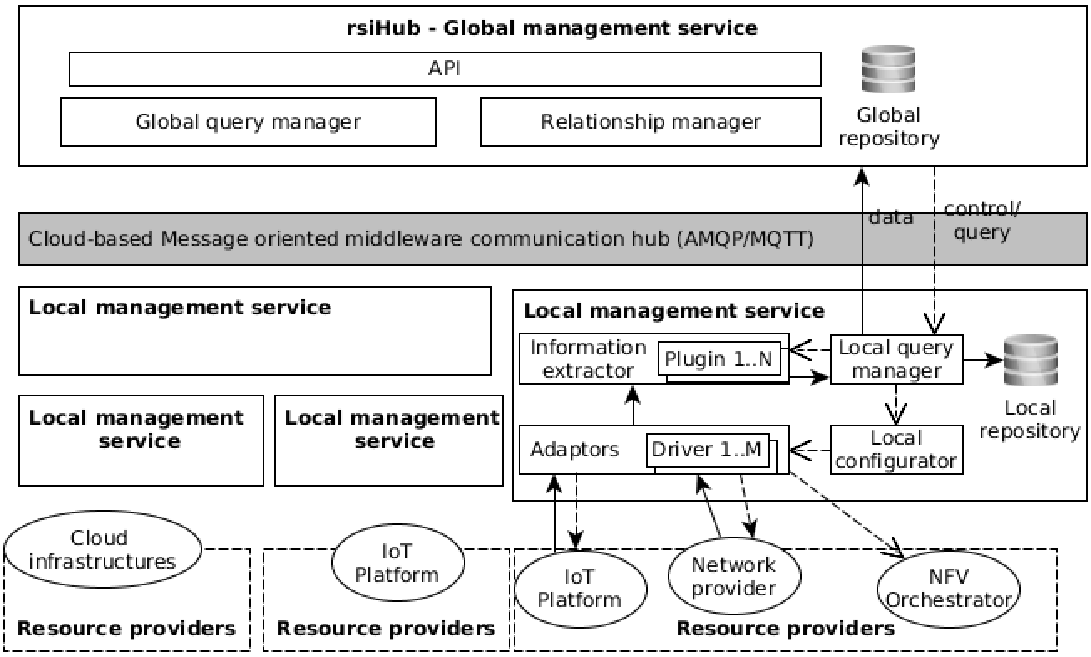


**Figure 21: INTER-HINC architecture**

Both Local Management Service and Global Management Service provide

- REST APIs: standard REST APIs for any clients to use our services.
- Client libraries: for applications to program calls to Service.

### 2.3.11 A Semantic Middleware for the information synchronization of the IoT devices

This project proposes the development of a new component, called Semantic Middleware, to be added within the set of middleware modules supported by the INTER-IoT approach. The new component aims to enable near real-time signaling capabilities to all the devices connected the IoT platform, also thanks to the interaction with various modules provided by of the INTER-IoT approach. The current state of the art points out that many solutions available in literature enable the data exchanges among the devices. However, as the content meaning of the exchanged data expressions is left to the interpretation of the receivers, these solutions do not support the information synchronization, thus limiting the semantic interoperability of the involved devices. In order to contribute to bridge this gap, Semantic Middleware allows to express all the exchanged information (included the synchronization requests) under the form of semantic model. This way, it allows a more flexible and adaptable characterization of the data subscribers' needs and of the data providers' capabilities, while it enhances the interoperability between all the involved devices.

Since the major issue hindering the information synchronization in an IoT platform is the limited interoperability of its involved components, we propose as key–feature of the Semantic *Middleware* its capability to express both registrations and the following alerts of changed information under the form of semantic data. In particular, the requests will be expressed in SPARQL and for this reason the agents are required to know the SPARQL syntax and the T-BOX structure of the semantic model. In addition, the changed information will be notified through RDF. In this way, the connected devices can operate synergistically on the basis of a semantic model through which the involved devices can share information, while the middleware supplies the central point which dispatches information through mechanisms that are transparent to their clients. Behind each device there is an agent that can act both as a publisher to send changed information and as a subscriber to receive all the updates related to subscribe information, while the evaluation of the knowledge-based updates is performed by another agent, which recognizes the emergence of the asked information. The agent-oriented system leverages the FIPA-compliant JADE framework. Fig. 1 reports the information flows concerning this dispatching service.

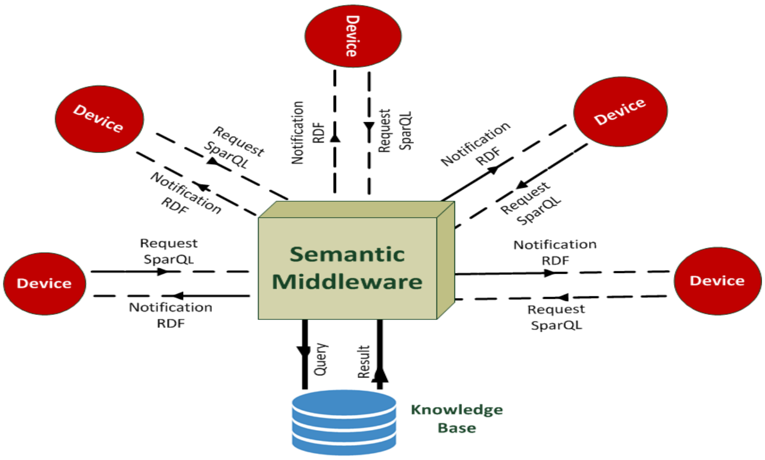


Figure 22: System overview

In the following, a set of its requirements are reported to define key aspects of the Semantic Middleware.

**R1: Prompt notification of a state change among devices and services.** The Semantic Middleware must propagate contextual state messages coming from sensors to keep the other interested components updated about interesting events. Notification are distributed as messages to other devices, so that the latter can perform further evaluations and actions, where needed.

**R2: Subscription to per-consumer relevant information.** Consumers should be able to subscribe to events of interest simply specifying the kind of information they are interested in without bounding to specific knowledge about the producers capabilities.

**R3: Processing efficiently huge IoT stream.** The middleware must be supported by a scalable architecture that reduces memory and computational cost of a massive amount of real-time data produced by devices.

**R4: Supporting private interactions** The middleware must implement a private peer-to-peer messaging politic. Thus, each message can be received by one single consumer, exactly the one subscribed to its content.

The *Semantic Middleware* is agnostic to the meta model (TBOX) of the IoT platform ontology, i.e. the behavior of the *Semantic Middleware* does not depend on a specific semantic structure of the ontology. However, the *Semantic Middleware* uses the GOIoTP (D3.1), which is taken as global common semantic model that all the devices share. If the IoT platform uses also other semantic models, they are aligned with GOIoTP through the Inter Platform Semantic Mediator (IPSM) (D3.1).

The ontology model will be stored into the repository provided by INTER-IoT infrastructure through the SPARQL engine also provided by the same infrastructure.

Figure 2 depicts the overall architecture of the Semantic Middleware, focusing on its semantic information, integration and dispatching capabilities. The diagram outlines the components in charge of supporting it: Update Manager (UM) and Semantic Broker (SB), on its turn made up by the Subscription Manager (SM) and the Messaging System (MS) supported by a multi-agent System. Each sensor, after having gathered the information which oversees, affects the knowledge base hosted on the shared semantic repository by updating or deleting semantic assertions. This is done through a web service exposed by UM. On the other hand, information consumers (smart services and sensors) subscribes to the SB, providing their profile of interest.

SM is the component which is always listening on the queue that manages the new subscriptions, leveraging the Apache ActiveMQ (ActiveMQ) messaging system. Whenever SM receives a subscription request from a network client, it activates server-side an agent (the ClientAgent) which takes care of the client interests. Namely, SM records this interest activating an agent in charge of signaling emerging new information to the consumer. If such agent already exists, SM simply notifies the new consumers’ interest. In each moment, the consumer can unsubscribe by cancelling the request. Each time a sensor authors new knowledge, the UM informs the SB of the occurred event so that, in a continuous query processing fashion, the SB can evaluate emerging information and notifies it to the consumer through the MS.

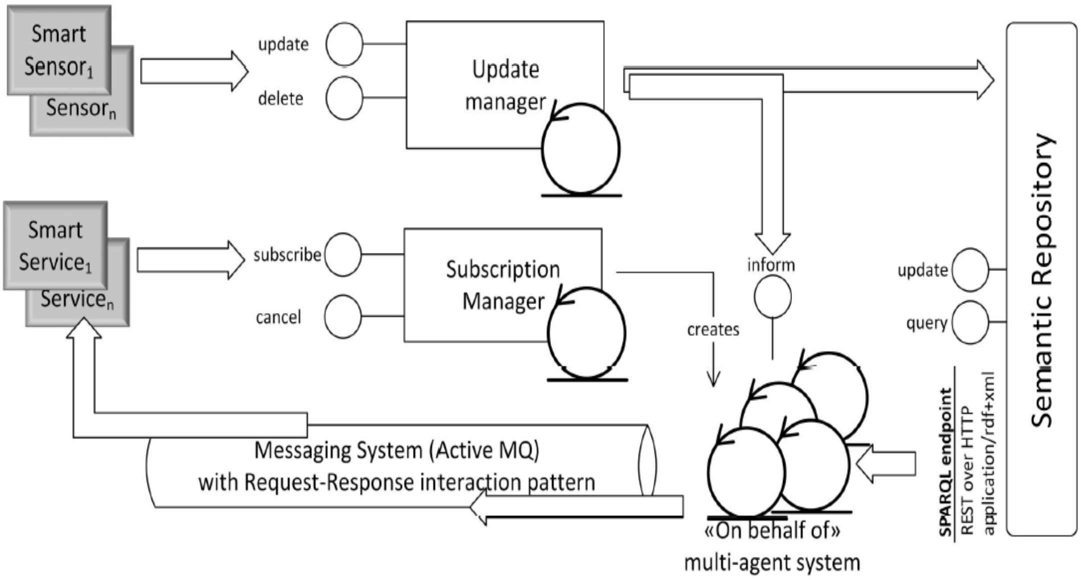


Figure 23: Overall architecture

### Negotiation of machining services through the dispatching events

A valid business scenario for the pilot of the Semantic Middleware is offered by the Cyber Physical System (CPS) Lab located at the ITIA-CNR's headquarters in Milan, where recently it has been developed a cutting-edge CPS system in order to monitor and optimize the position of various pallets along a conveyer belt within an industrial scenario (Figure 3). The idea behind this system is that processes in a production facility can be optimized with the aid of indoor localization and route analysis. In addition, an asset tracking solution makes it possible to retrieve location and nearest available services provided by the servitization of the factory. The pallets contain hardware components that have to be worked through operations performed in different working stations following a specific order. Moreover, the working stations can perform different operations (e.g. drilling, milling, etc.), and the path of the pallet is optimized by a simulation application in order to send it to the closest available working station.

The provided system leverages the paradigm of the IIoT (Industrial IoT), and a network of smart objects. The various devices involved in the CPS system can be synchronized through the *Semantic Middleware*, also exploiting the connection with various modules of the INTER-IoT platform. In particular, all the devices share the GOIoTP (D3.1), which is taken as the global common semantic model.

The sensors monitoring the pallet position will play the role of publisher as they will send through the middleware the information concerning the pallet position (through a SPARQL UPDATE). Also the working stations will publish their availability status. These information will be then consumed by a simulation tool which has previously subscribed to the changes applied to the pallet position and the availability status of the working stations (using a proper SPARQL query) with the goal to identify the optimized pallet route. In addition, the information concerning the route are then published and in its turn consumed by the IoT actuators which allow to change the route of the pallets along the conveyor belt.

It should be underlined that the proposed scenario “Position and Optimization of the pallet” suits well the specifications of the domain INTER-LogP (D2.4), as the latter focuses mainly on logistics and transport. For this reason the scenario ""Position and Optimization of the pallet" can be proposed to extend the business scenarios list for INTER-LogP, reported in Table 3 of the D2.4.

Figure 24: The conveyor belt

### 2.3.12 SecurIoTy

Security is paramount for the safe and reliable operation of IoT connected devices. Security is the foundational enabler of IoT. The security aspect of IoT was demonstrated vigorously in a recent attack in October 2016 carried out in a series of Distributed Denial of Service (DDoS) attacks which caused widespread disruption of legitimate internet activity in Europe and the US. Because the attacks targeted the Domain Name System (DNS) that makes sure information requests on the internet are delivered to the right address, a lot of non-IoT activities such as online shopping, social media interaction, and email communication, were negatively affected for a prolonged period of time.

In settings like healthcare, production, logistics and other verticals the effects may be more severe than what we have seen in past attacks. In the logistics vertical, current threat models (i.e. a process by which potential threats can be identified, enumerated, and prioritized) consider for example how devices but also actuators can be interconnected and operated in a secure way. Avoiding unauthorized access to data, the tampering with data as well as the availability of the device itself. These threat models resample closely the pillars of IT security known as confidentiality, integrity and availability (CIA).

Currently there is consensus that in order for IoT to become a widespread mechanism the security issues have to be resolved. Where there is less consensus is how to best implement security in IoT at the device, network, and system levels. Network firewalls and protocols can manage the high-level traffic coursing through the Internet, but how do we protect deeply embedded endpoint devices that usually have a very specific, defined mission with limited resources available to accomplish it? There is no "silver bullet" in sight that can effectively mitigate every possible cyberthreat.

We believe though, that tried-and-true IT security controls that have evolved over the past 25 years can be an effective starting point for IoT, provided we can adapt them to the unique constraints of the embedded devices that will increasingly comprise networks of the future.

In our proposal, we provide a practical approach to address IoT confidentiality, integrity and availability based on our experience in other arenas such as cloud security.

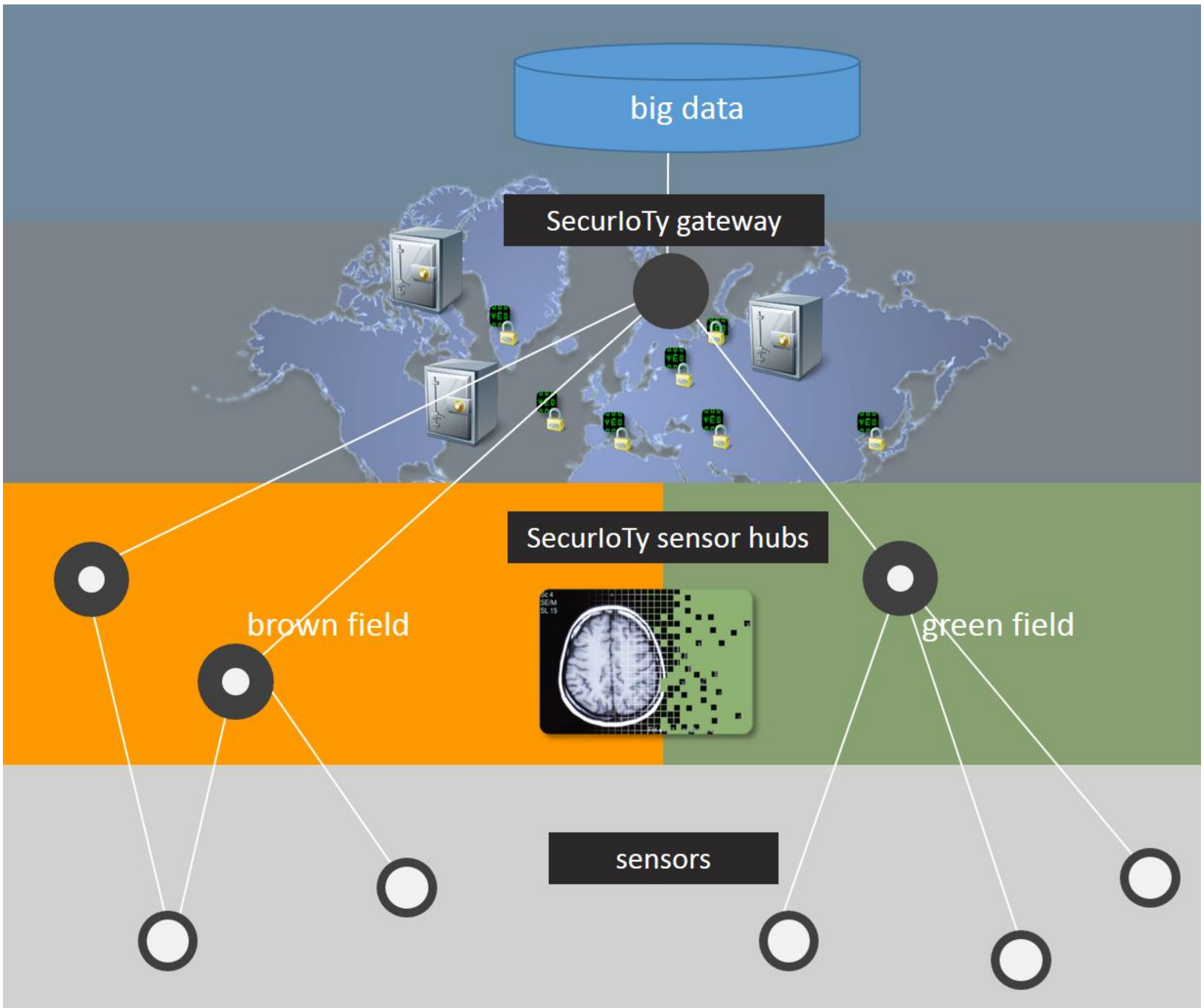


**Figure 25: Basic architecture of an IoT system**

A typical IoT architecture would include some or all of the components depicted in Figure 25. There are sensors which feed data to sensor nodes (or sensor hubs). These sensor nodes would integrate data and deliver the data stream to a data reservoir through some kind of gateway. In a brown field scenario, the architecture would be adapted to the existing framework. In a greenfield approach, the architecture would be set up newly.

SecurIoTy establishes security across all layers of the INTER-IoT Architecture. Knowing no one single control which is going to adequately protect a device, its data traffic, as well as its data storage, how do we apply what we have learned from other sectors? SecurIoTy does so through a multi-layered approach to security. SecurIoTy concretely provides:

(1) Data protection while in transit will be provided through a crypto proxy which will provide encrypted data exchange lines.
(2) Data protection while at rest will be provided by our DocRAID® CloudRAID proxy which is storage agnostic. By providing a CloudRAID proxy, data is being fragmented, encrypted and redundantly distributed across multiple storages, thus yielding unprecedented security, covering a wide range of threat scenarios.
(3) Interfaces are provided towards
   a. the application layer covering HTTP(S) and WebDAV and potentially REST
   b. the middleware layer which covers storages, initially storage will be operated by SecurIoTy, at later project stages other storage have to be explored
   c. the network covering TCP
(4) Encryption key material is maintained and managed using a scalable approach ranging initially from secured storage to hardware based security modules (HSM). In the initial phase of this project we will resort to secured storage to handle key material.

The proposed approach will complement the INTER-IoT Architecture and will provide industry standard interfaces to integrate security as necessitated by the respective application. A high degree of interoperability is achieved by adhering to standard protocols (HTTPS(S), WebDAV, REST, TCP) and by integration of widely used cloud services. In contrast to current approaches to IoT security which mainly focus on single aspects of IoT security, SecurIoTy provides a single framework to cover scalable security from the device level to the application level and which covers all dimensions of security such as confidentiality, integrity and availability (CIA). Put to work in the logistics use case (INTER-logP), SecurIoTy will push the envelope of IoT security well beyond the state of the art.

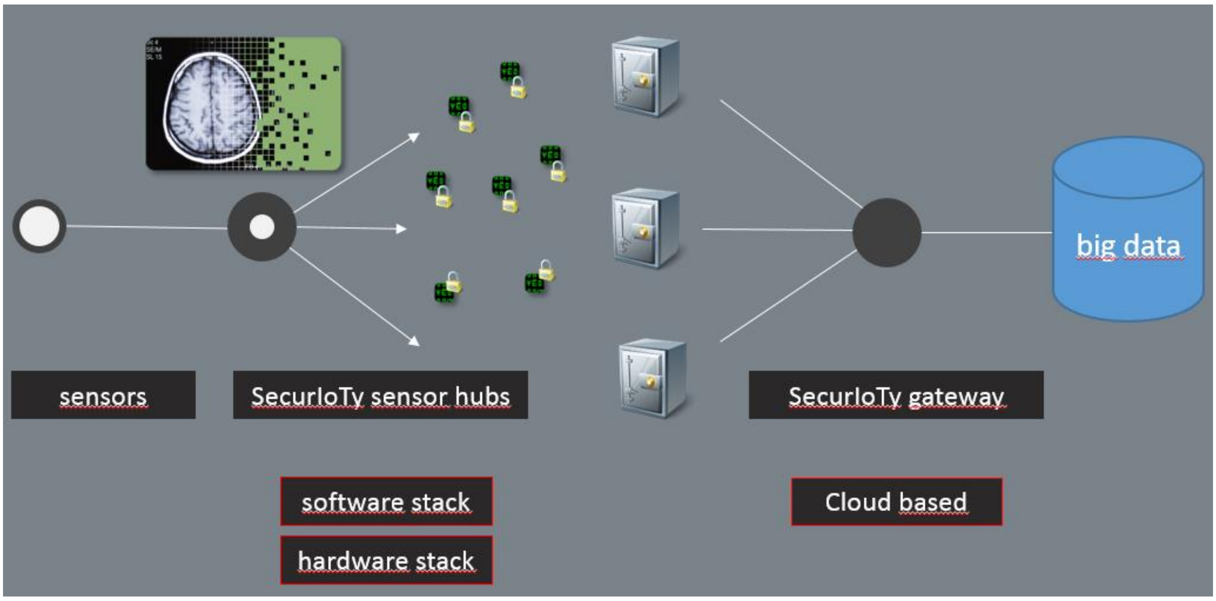


**Figure 26: SecurIoTy overview**

The DocRAID® CloudRAID (see Figure 26) proxy can be deployed as a sensor hub, collecting data from sensors directly or alternatively can be set up as a gateway receiving data from sensor hubs and deliver that data to a data storage. In this project we will assume that DocRAID will be set up as a gateway.

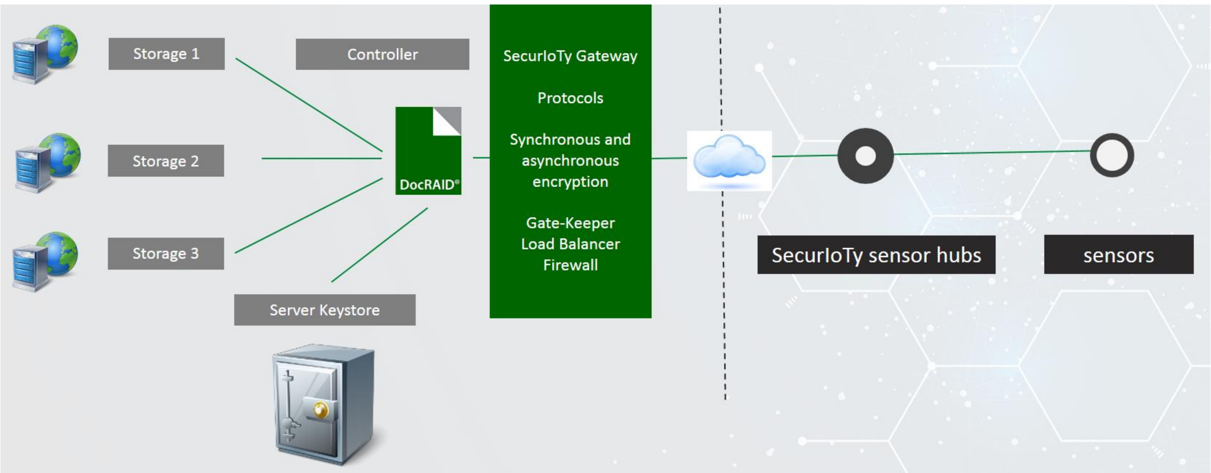


**Figure 27: SecurIoTy architecture with the DocRAID crypto proxy**

The DocRAID crypto proxy works in three phases:

1. Fragmentation

Data is sent through a shredder and fragmented to pieces.

2. Encryption

Each fragment is encrypted using AES256. Key exchange optionally via Diffi-Hellman.

3. RAID distribution

Encrypted fragments are redundantly distributed by the DocRAID® algorithm, no one (1) storage knows all fragments. Distribution across geographies and jurisdictions, keep legacy infrastructure.

### Touchpoints and interface with Inter-IoT

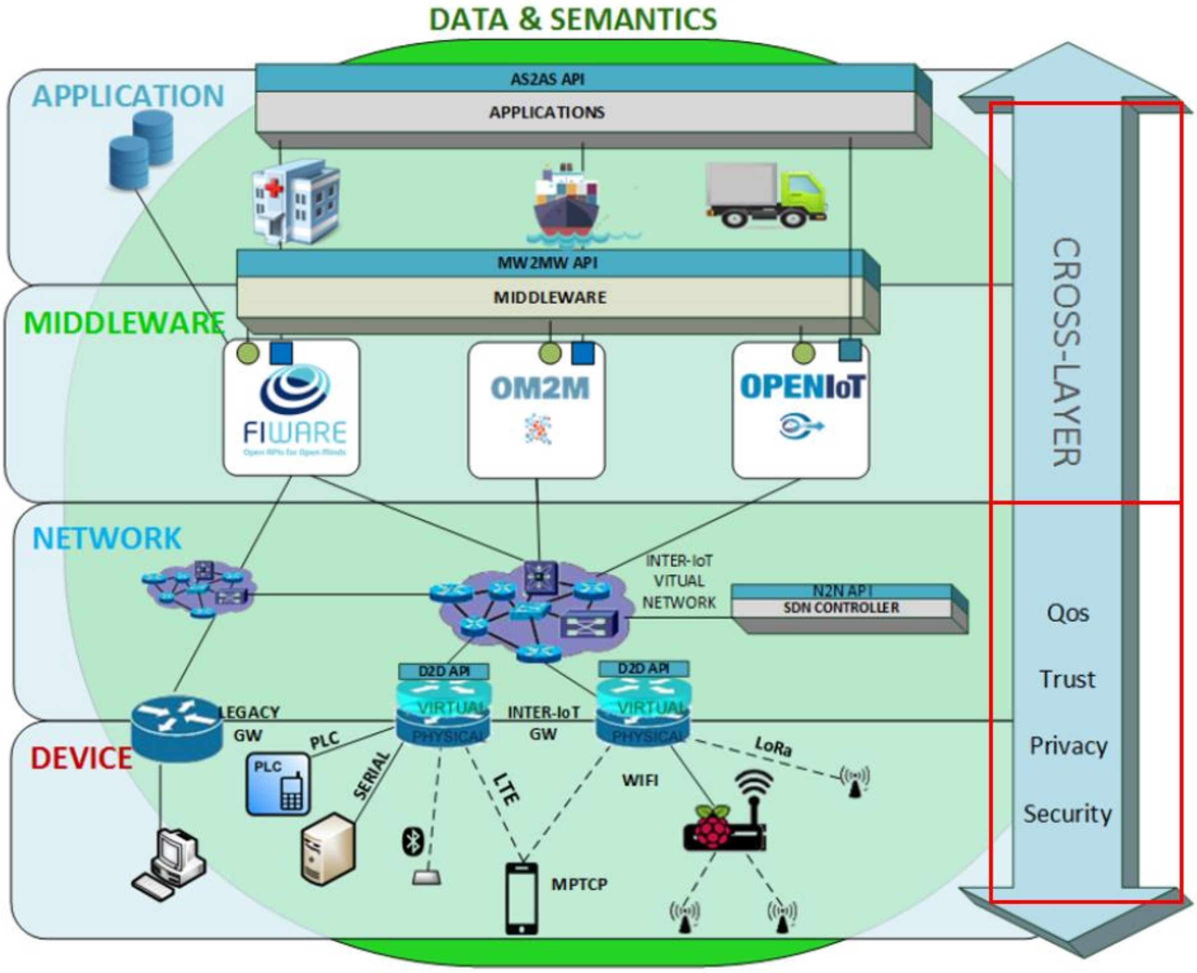


**Figure 28: SecurIoTy touchpoints with Inter-IoT architecture depicted by red frames**

Touchpoints and interfaces with the Inter-IoT architecture will most likely by at the application and middleware level. See the red frames in Figure 28.

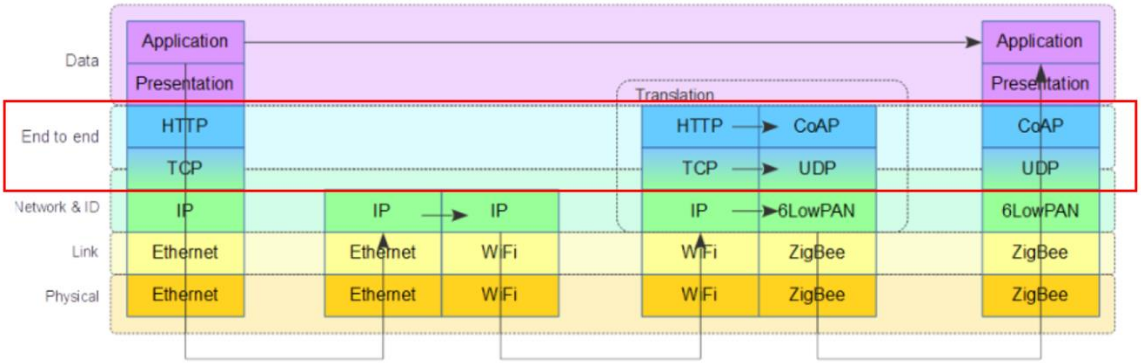


**Figure 29: SecurIoTy touchpoints with Inter-IoT protocols depicted by red frames**

Touchpoints and interfaces with Inter-IoT protocols will most likely by at the end-to-end HTTP/TCP level. See the red frame in Figure 29.

# 3 Test strategy and approach

For testing of the pilot projects Factory Acceptance Test (FAT) and Site Acceptance Test (SAT) testing is done to test and prove the system is operational and functional.

The FAT takes place in a lab environment and tests a solution before it is deployed in the field. The FAT tests if the solution meets the specification, and if it is fully functional. A FAT includes a check of completeness, verifies requirements, and proves functionality. This can be either by simulation or a conventional functional test. When a FAT is not performed correctly or not performed at all finding non-conformities can be delayed until integration in the customer environment. At this stage, there will be less time available to correct the problems without affecting the integration schedule. The FAT Document describes the Factory Acceptance Testing plan and describes or points to previously defined test plans, use cases and test scenarios used during testing. The test outcomes can either be placed in the FAT document itself or a separate test report can be created.

The SAT takes place after integration at the customer site. During Site Acceptance Testing the solution is tested on: Integration, Performance, conformance to specifications and User acceptance testing. The SAT is performed to prove that the solution is integrated correctly into the customers' environment and meets all the requirements. The SAT Document describes the Site Acceptance Testing plan and describes or points to previously defined test plans, use cases and test scenarios used during testing. The test outcomes can either be placed in the SAT document itself or a separate test report can be created.

## 3.1 Testing strategy

For each of the pilots a FAT and SAT document will be created. These documents will describe the test strategy in detail. The FAT will describe the system, test setup, tooling, test strategy, test activities and test results for the lab setup. During these tests the system readiness for field deployment will be tested and proven. For this test, the following stakeholders should be present:

- Project managers (From manufacturer and customer)
- Key engineering personal (System Architect, Lead Engineer, Integrator)
- Operators
- Maintenance personal

During FAT testing the readiness of the system is shown to the customer and the customer can use the system for the first time in an actual system setup to get a better feeling for the new system. The operators and maintenance personal can get a view of how the system is operated and maintained. During the FAT the customer will work with the system for the first time which will most likely lead to some minor changes to the system to fix operation inconveniences before the system is deployed in the field. During testing the result should be written in the FAT document which should be signed off by all the attendees at the end of the test. A FAT will be conducted 2 to 4 weeks before actual deployment in the field.

During the SAT test the actual deployed system is tested and proven. The SAT follows the same principles as the FAT but describes and tests the system integrated in the customer systems. During testing the result should be written in the SAT document which should be signed off by all the attendees at the end of the test. Signing if the SAT is the actual acceptance of the system by the customer.

## 3.2 Entrance criteria

To start a FAT for a project the following deliverables should be present/ready:

- FAT document
- System test setup (as much actual hardware as possible)
- Test applications and tooling (e.g. for performance testing)

To start a SAT for a project the following deliverables should be present/ready:

- SAT document
- Integrated system at the customer
- Test applications and tooling (e.g. for performance testing)

The deliverables will also be listed in the FAT/SAT document to be signed off on start of the test.

## 3.3 Acceptance Criteria

The FAT acceptance criteria is a signed copy of the FAT document as this contains all the needed deliverables and tests to complete the factory acceptance testing.

The SAT acceptance criteria is a signed copy of the SAT document as this contains all the needed deliverables and tests to complete the site acceptance testing.

## 3.4 Testing types

The FAT and SAT document will define the testing types per project in detail. The following list provides an example of testing types one could think of:

- manual data load
- interface using scripted data
- converted data load
- converted data inspection
- backup and recovery
- database auditing
- data archival
- security
- locking
- batch response time
- online response time
- network stress
- security
- live data
- live environment

## 3.5 Suspension and resumption criteria

When one of the entrance criteria is not met at the start of the test the complete or part of the tests will be suspended until it/they are met.

When during testing one of the entrance criteria’s is found to be unsatisfactory the complete or part of the tests will be suspended until it is met.

# 4 Defect Reporting

For defect reporting JIRA will be used. For each integration project, a project will be created in JIRA. The Inter-IoT JIRA sit can be found at http://jira.inter-iot.eu.

## 4.1 Issue tracking process

The used JIRA workflow for these projects is depicted below:

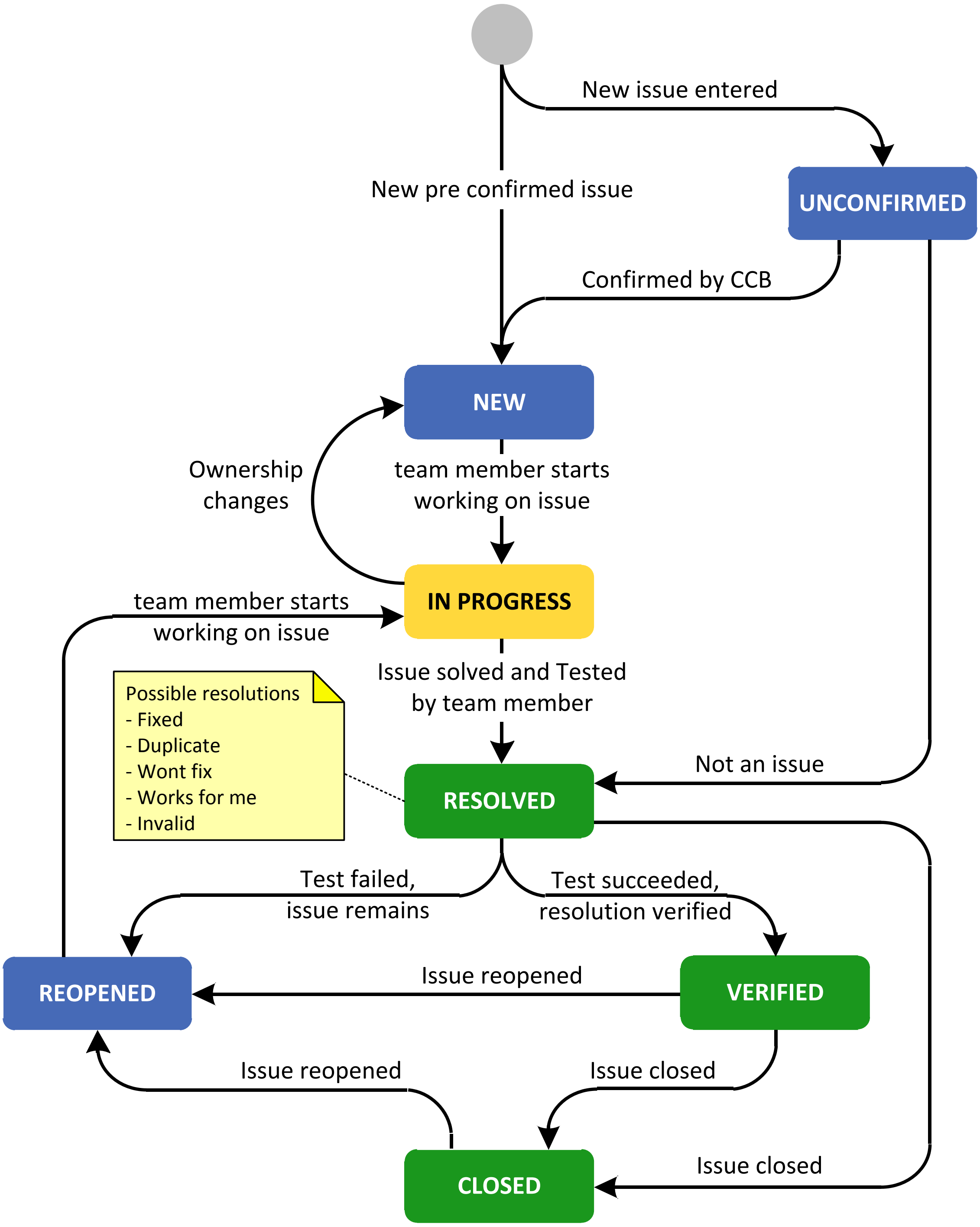


**Figure 30: JIRA workflow**

When a new issues is found during integration testing a new ticket is created in JIRA. The ticket will get the initial status of *unconfirmed*, which means that the issue was seen but it has to be determined if it is actually a defect or not. If the ticket is entered by someone who can determine this (CCB member or the equivalent) the ticket can also be entered with status set to *new*.

A ticket contains a summary, a description and the steps to reproduce.

When an issue is confirmed the status will change to *new.* If not the ticket will be resolved with the appropriate resolution and meaningful comment.

**New**

A ticket with the status *new* is confirmed and needs to be solved by a team member (this could be a developer for a code change or a project manager or architect to define/redefine a requirement or alike). A *new* ticket can be assigned to a team or be self-assigned by a team member when he/she wants to work on it. When work starts on a ticket the team member will change the status to *in progress*.

**In progress**

When a team member starts work on a ticket he will change the status to *In progress* this way it is clear which issues are being worked on.

A ticket can be reassigned to another team member when for instance work on an issue cannot proceed whilst waiting for another team member to help or clarify something related to the issue, or work is handed over to someone else. Reassigning a ticket will change the status to *new*. The ticket will then follow the process again from the status.

When the issue has been resolved, reviewed and tested the ticket can be resolved indicating a resolution and a meaningful comment. This will change the ticket state to *resolved*.

**Resolved**

When a ticket is in *resolved* state it will be incorporated in the next release and tested during a new integration test. When the issue resolution is verified the ticket status will change to *verified*. When the issue remains during testing the ticket will be *reopened*, a description on why the ticket is reopened. From *reopened* the ticket can be worked on by a team member again.

**Reopened**

A *reopened* ticket can be picked up by a team member to work on again, changing its status to *in progress*.

**Verified**

A *verified* ticket has been fixed and verified and will be closed when released in a new production release. When the issue re-merges during further testing the ticket can be reopened.

**Closed**

A closed ticked is an invalid or fixed and released issue. When the issue remerges at a later stage the ticket can be reopened.

## 4.2 Change Control Board

A Change Control Board (CCB) will be defined for each project.

The change control board will consist of the following persons:

| | |
|---|---|
| • Carlos Palau | (UPV) |
| • Eneko Olivares | (UPV) |
| • Flavio Fuart | (XLAB) |
| • Johan Schabbink | (NEWAYS) |
| • Dennis Engbers | (NEWAYS) |
| • Pablo Giménez | (VPF) |
| • Gema Ibáñez | (SABIEN) |

## 4.3 Project configuration

For each pilot project a JIRA project will be created which follows the issue tracking process described in 4.1 4.1Issue tracking process.

The following project will be created:

- INTER-LogP
- INTER-HEALTH
- 25_INTER-OM2M
- 27_INTER-HINC
- 39_INTER-SOFOS
- 42_INTER-HARE
- 43_INTER-MCOB
- 49_INTER-SENSHOOK
- 52_INTER-ACHILLES
- 53_INTER-IOT-EWS
- 66_INTER-SMIS
- 70_INTER-SECURIOTY
- 71_INTER-SENSINACT
- 74_INTER-E3TCITY

# 5 Test environment

## 5.1 Introduction

To test the functionality of the IoT framework a representative test system is needed. The test system needs to approach the “real world” as much as possible. The pilot setups must be recreated and proven. This test environment description is bottom-up and starts with the sensors and works its way up the chain to the cloud.

### 5.1.1 IoT physical layer

There are two different low-level sensor integrations in IoT. The “smart” and “simple” sensors. A smart sensor is a standalone device which runs the physical gateway components (or parts of them) and has it’s sensors physically connected to it and is able to connect to the IoT framework using WiFi, GPRS, etc. The simple sensor needs to connect to a gateway which runs the physical gateway components (or parts of them) and forwards the values generated by the simple sensor. The major difference between them is that the first can migrate over networks itself, the later needs a gateway to do so which in most cases is not easy as the sensor and gateway need to be paired. The difference is depicted below in an example.

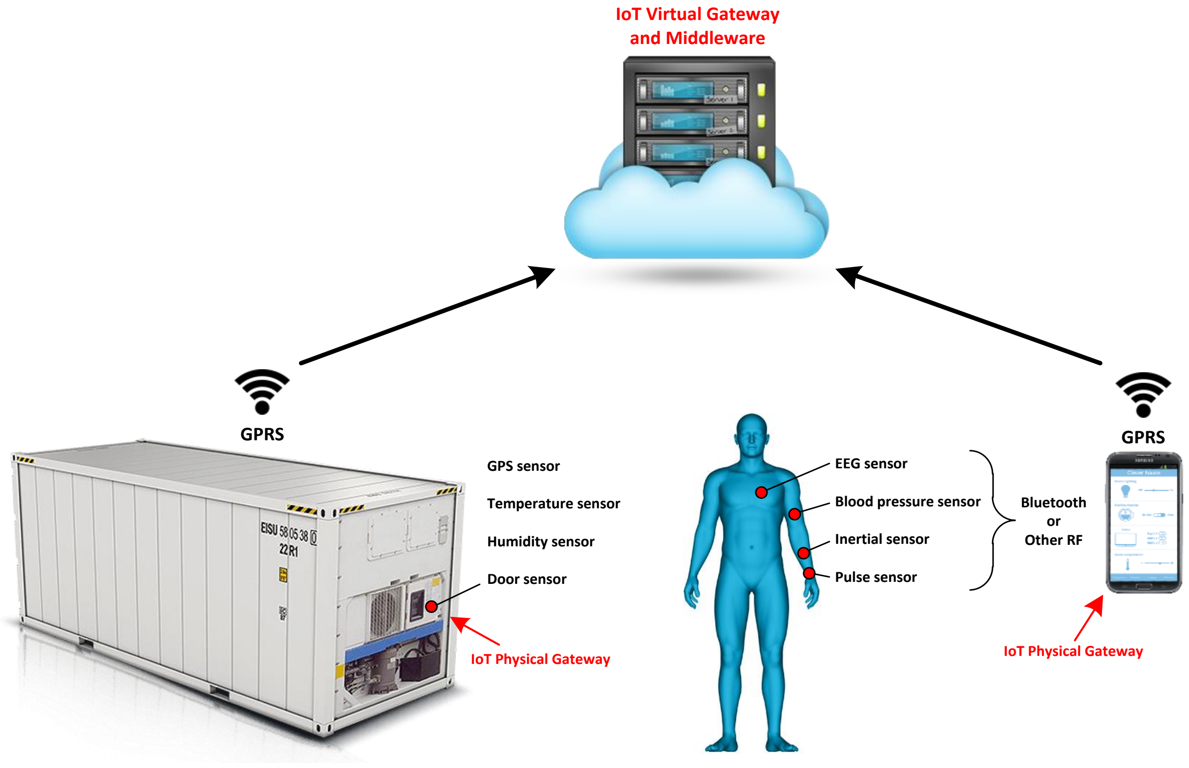


**Figure 31: Sensor type overview**

The simple sensor usually is a low-power device and has low processor and memory capacity. When we look at for instance the panStamp sensor module that runs on a TI-CC430F5137 CPU and compare this to a Raspberry Pi 3 which runs on a BCM2836 and can be compared to a smart sensor we get the following overview:

*Table 3: Sensor platform comparison*

| Processor | Frequency (MHz) | RAM (kilobyte) | ROM (kilobyte) | Architecture |
|---|---|---|---|---|
| PanStamp | 20 | 4 | 32 | 16 bit |
| Raspberry Pi 3 | 4800 (4x1200) | 1048576 (1Gb DDR2) | 67108864 (64 Gb SD) | 64 bit |

When we look at the simple sensor in the example we see that the phone which is the IoT gateway can relocate together with the sensors. This however is not also the case, when the sensors are connected to a container and the gateway for instance a ship and/or a crane will require the sensor to move from gateway to gateway.

For testing of the smart sensor a setup based on a Raspberry Pi 3 with a GPRS and GPS module is used as the gateway with sensors. For testing of the simple sensor a Raspberry Pi 3 is used as the gateway and a panStamp sensor network is used to test the sensors.

### 5.1.2 IoT virtual layer

To test the virtual Gateway and middleware of IoT a set of virtual machine is setup in Microsoft Azure. The machines are setup according to the overview in Figure 32. They are linked together by a private network that can be reached through the stepping-stone server vmmgt01.

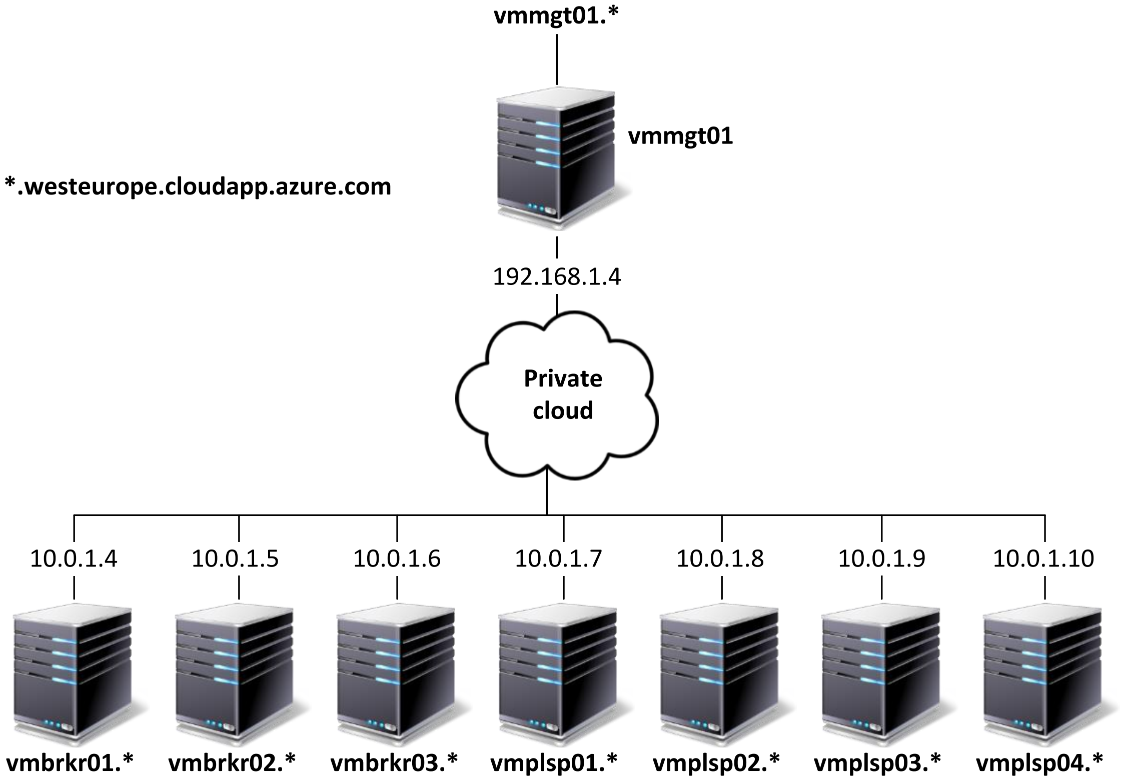


**Figure 32: Azure server overview**

## 5.2 Test setup for Integration

### 5.2.1 Raspberry Pi 3

Source: https://www.raspberrypi.org/help/faqs/#topIntro

The Raspberry Pi is a credit-card-sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video. We want to see it being used by adults and children all over the world to learn programming and digital making.

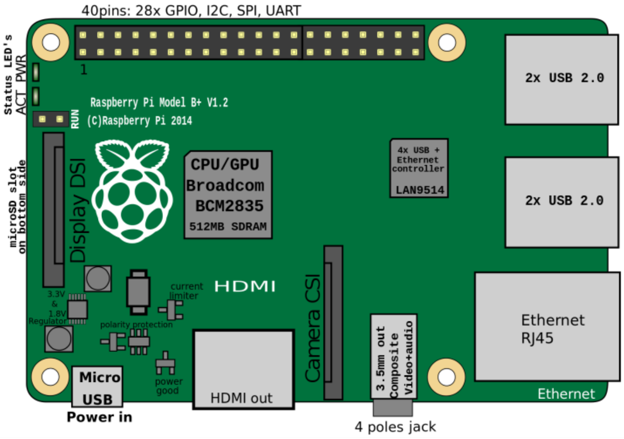


Figure 33: Raspberry Pi

#### 5.2.1.1 GPRS/GSM Quadband module for Arduino and Raspberry Pi

Source: https://www.cooking-hacks.com/gprs-quadband-module-sim900-for-arduino

GPRS/GSM Quadband Module (SIM900) offers GPRS connection to your Raspberry Pi board. It includes the SIM900 communication module. It is necessary to connect an antenna in order to establish communication. It can send SMS, make calls or create TCP and UDP sockets in order to send the information to the Internet.

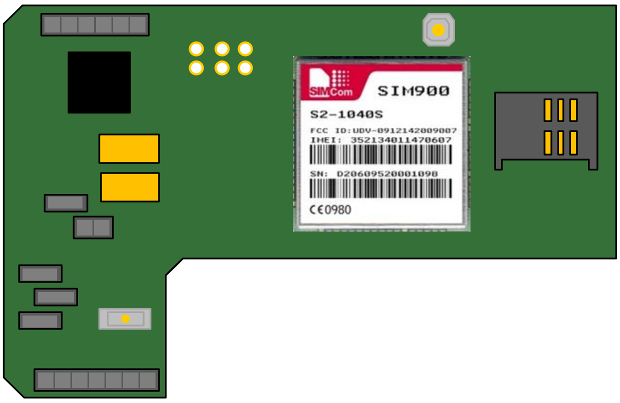


Figure 34: GPRS/GSM Quadband module

### 5.2.2 panStamp sensors low-power wireless modules

Source: https://github.com/panStamp/panstamp/wiki/

panStamps are autonomous low-power wireless modules programmable from the Arduino IDE and made for telemetry and control projects. These modules communicate over the free 868-900-915 MHz bands available worldwide and are designed to last for months and even years when powered from simple alkaline batteries, depending on the duty cycle and transmission interval programmed.

In addition, panStamps form complete wireless ecosystems with direct connectivity to the Internet. The company provides software applications for configuring wireless networks and turning any computer into an automation server. panStamp also provides customized Raspbian images for Raspberry PI giving this popular computer platform a key role in any automation project as a wireless-IP gateway with stand-alone capabilities.

panStamps include an on-board microcontroller so they can be run autonomously without any external processor. There are two different models of panStamp, each using a different microcontroller.

#### 5.2.2.1 panStamp Shield

panStamp shield for Raspberry PI is an elegant way to add low-power wireless connectivity to the most popular compact computer in the world. Featuring a panStamp, this shield releases the Raspberry Pi from having to deal with the low-power wireless communications. Instead, the on-board panStamp acts as a modem connected to the RPI UART (serial port).

This shield also includes a real time IC with battery backup so that we no longer depend on remote NTP servers and Internet connections to get the current time, even after an outage.

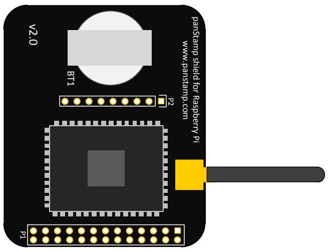


Figure 35: panStamp Shield

#### 5.2.2.2 panStamp Battery board

The panStamp Battery-board is maybe panStamp's most popular carrier board. It features a MAX1724 step-up converter so the board can get a stable 3.3 VDC level from an AA battery providing between 0.8 and 3 VDC. The board supports a variety of sensors, including NTC thermistor, SI7021 for temperature and humidity and BMP180 for barometric pressure and temperature.

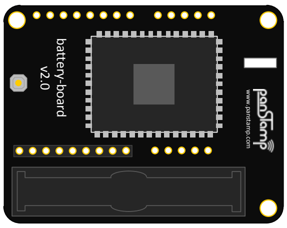


Figure 36: panStamp battery board

## 5.3 Pilot project integration

### 5.3.1 INTER-logP

The scenario set out in INTER-LogP is carried out through the participation of three different actors. Each of them has its own IoT platform and is able to share data with the others.

**Port authority**

The port authority has a large number of sensors distributed throughout the port that provide data for management and operation. Most of that data is confidential, the rest can be shared, adding value to other companies.

The architecture for providing interoperability with the existing infrastructure can be seen in Figure 37. Currently, the port authority has a common database where all the data is stored (in red). It uses WSO2 to provide an IoT architecture in two ways: data in real time through the Message broker and historic data through the Data services server and Enterprise service bus.

Because the port has its own platform, the integration with the INTER-IoT is done through the middleware. It needs a bridge in the middleware layer in order to interoperate with other platforms. Another integration could be done if deployment of new devices or sensor in a place without a wired connection is needed. In that case, the INTER-IoT gateway can be used to connect with the IoT platform sensors.

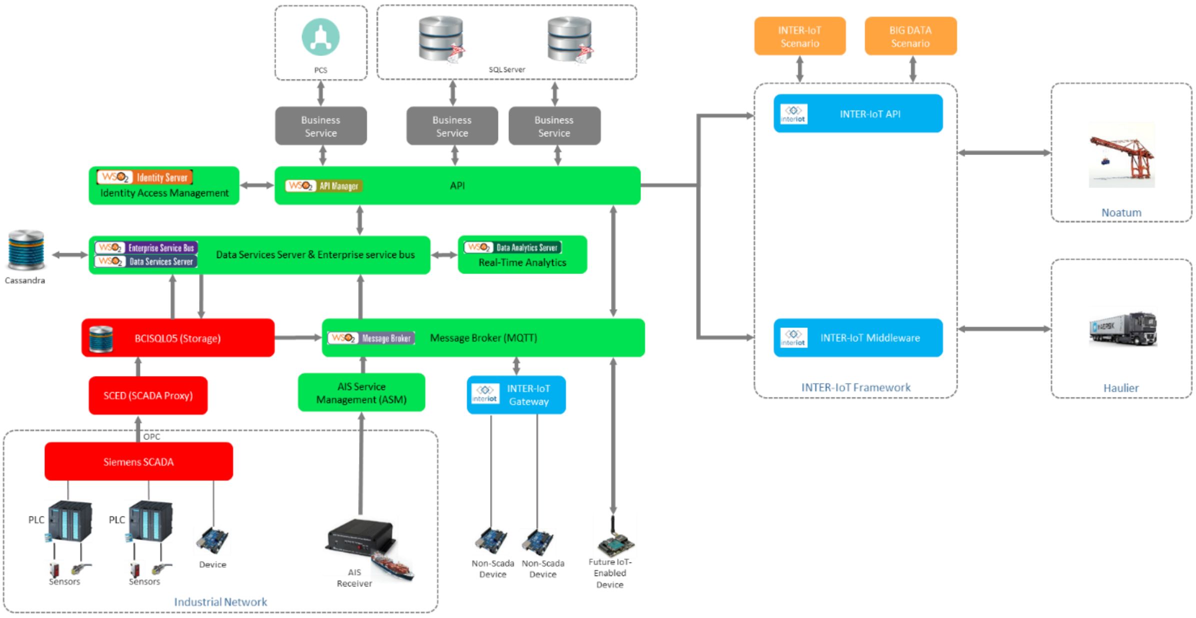


Figure 37: Port IoT platform and integration

The port authority of Valencia has a virtual machine in which its IoT platform is deployed. Through its API, other authorized companies can gain access to certain data. Once the truck arrives at the port facilities it is detected in the automatic gates, and from that moment its position is monitored in real time to guarantee the security in case of an incident.

### Container terminal

A big container terminal should have all the activity that takes place in the yard monitored, including all the machinery, to be able to manage the resources properly. For that reason, in the Noatum terminal each of the machines (vehicles, cranes, etc.) provide massive amounts of data from up to 80 sensors per machine per second.

Over 300 monitored devices among machines and dynamic lighting on lamp posts exist. These are divided into the following types:

*Table 4: Port machine types that carry monitored devices*

| Type | Image |
|---|---|
| STS (sea-to-shore) cranes | |
| RTG (Rubbed-Tyred Gantry) cranes | |
| Reachstackers | |

| | |
|---|---|
| ECH (Empty Container Handler) | |
| TT (Terminal Tractor) | |

In a previous version of the IoT Platform, data of over 200 machines is polled every second and inserted into an SQL Server relational database, which raises scalability issues.

As can be seen in Figure 38, data is sent from the machinery to the IoT Platform in two ways. Legacy sensors are collected once per second and inserted in the IoT Platform. New IoT devices are configured to communicate directly send real-time data through MQTT or REST interfaces. In addition, the data will be stored in a non-relational database, providing faster access to information.

As in the case of the port, the IoT platform of the terminal will be integrated with INTER-IoT through the middleware layer and the API layer.

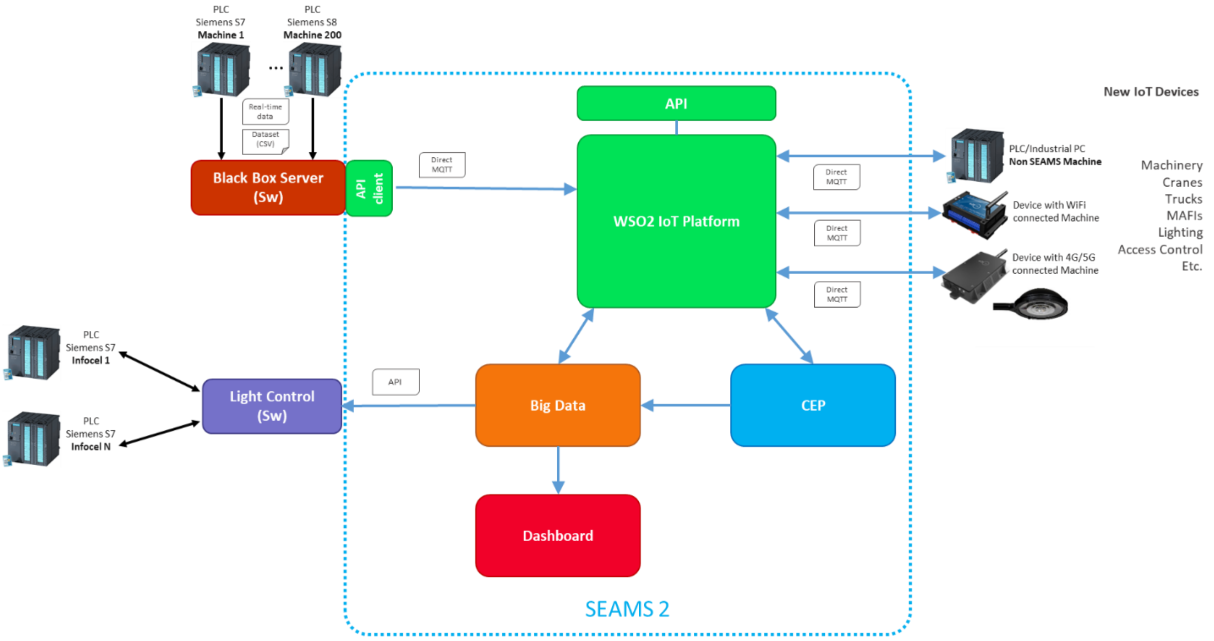


**Figure 38: Terminal IoT platform and integration**

The container terminal has its own server with its IoT platform. They are mainly interested in knowing the estimated time of arrival of the truck to the terminal to be able to manage resources. Furthermore, the terminal gives other companies access to some of their own data, such as the entry and exit of trucks by their access.

### Haulier company

The haulier company has a large fleet of trucks, which access the port daily. Each of them has a mobile app installed on a mobile phone or a tablet that acts as a bridge between the vehicle and the IoT platform of the company. All the devices in the truck and the driver send the data to the IoT platform through an app via Bluetooth.

The haulier company has an Azure IoT platform in the cloud, where trucks send all data about the vehicle. This data will be accessible for other companies as long as they are authorized and certain conditions are met, such as being in the port area.

### Test integration

During the test integration, the three platforms will be deployed in order to exchange data through INTER-IoT. The truck position will be simulated and inserted in the Azure IoT platform of the haulier company.

### 5.3.2 INTER-HEALTH UPV-SABIEN integration setup

The PC used to access the PWT is a developer computer.

The Android Phones used will be available to UPV-SABIEN. The model will be a Motorola Moto G4, but UPV-SABIEN will attempt to try as many other models as possible.

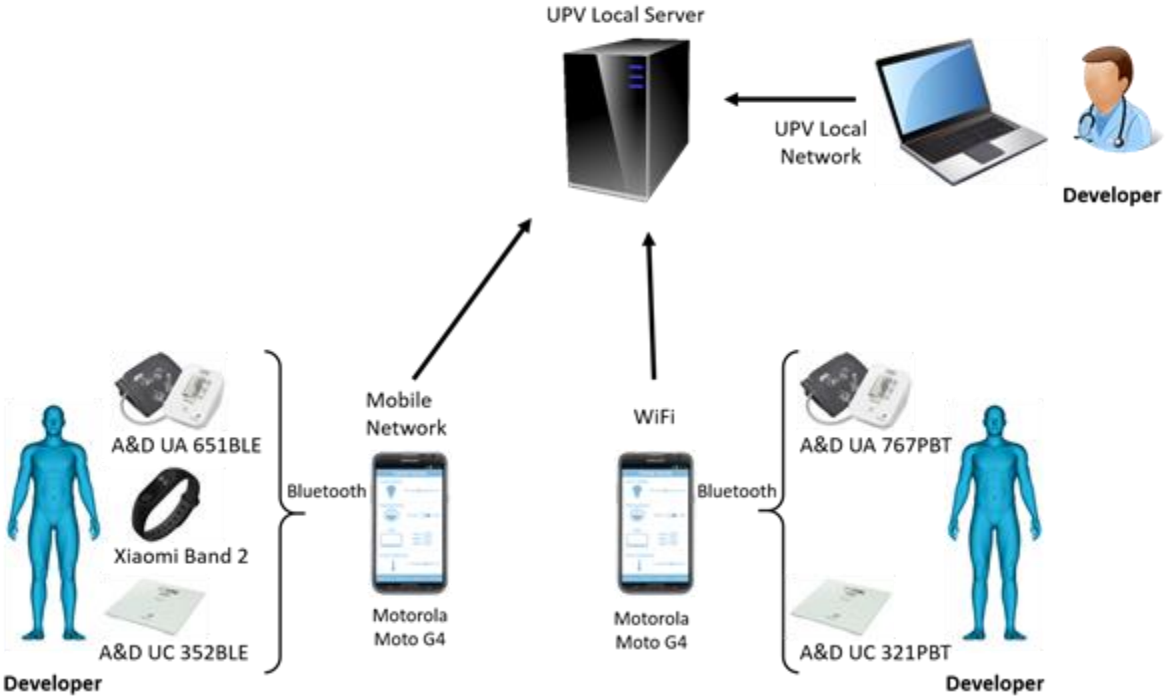


**Figure 39: INTER-HEALTH Integration and Factory test setup overview**

UPV-SABIEN will test all the main use cases described earlier with its local test setup. There are some differences that may make the result differ from what will be obtained from the final integration environment.

The PC used to access the PWT will be a developer PC, which most likely will differ greatly from the PCs used in the final integration environment. Since access to the PWT is through a Web Browser it will most likely not have any impact, but it is still worth taking into account.

The mobile phones used will be initially Motorola Moto G4 phones. UPV-SABIEN will use as many phone models as it can master. Full coverage is impossible because of an abundance of different models, these tests should help identify which limitations may exists when running the apps on different phones. The result is at least a preliminary list of “unsupported phones”.

The network in this test is owned by UPV-SABIEN so it will be possible to configure it to our needs. It will be set up to mimic the network available in the pilot but there is always a possibility of irreproducible, unexpected or unknown characteristics which later pose problems. Another network-related issue is to test the ability to service as many phones as during the pilot. It will be tested by using all phones at the same time with the maximum concurrent users by load-testing tools.

## 5.4 Third party project Integration

### 5.4.1 sensiNact integration

As shown in Figure 40, sensiNact will be inserted to the inter-IoT framework at the middleware layer. sensiNact will implement the APIs provide by the Inter-IoT Framework and build the necessary adapters for data model adaptation and transformation. sensiNact's middleware layer services such as resource discovery and lookup, security, data processing, etc., will be adapted to be compliant with the Inter-Framework.

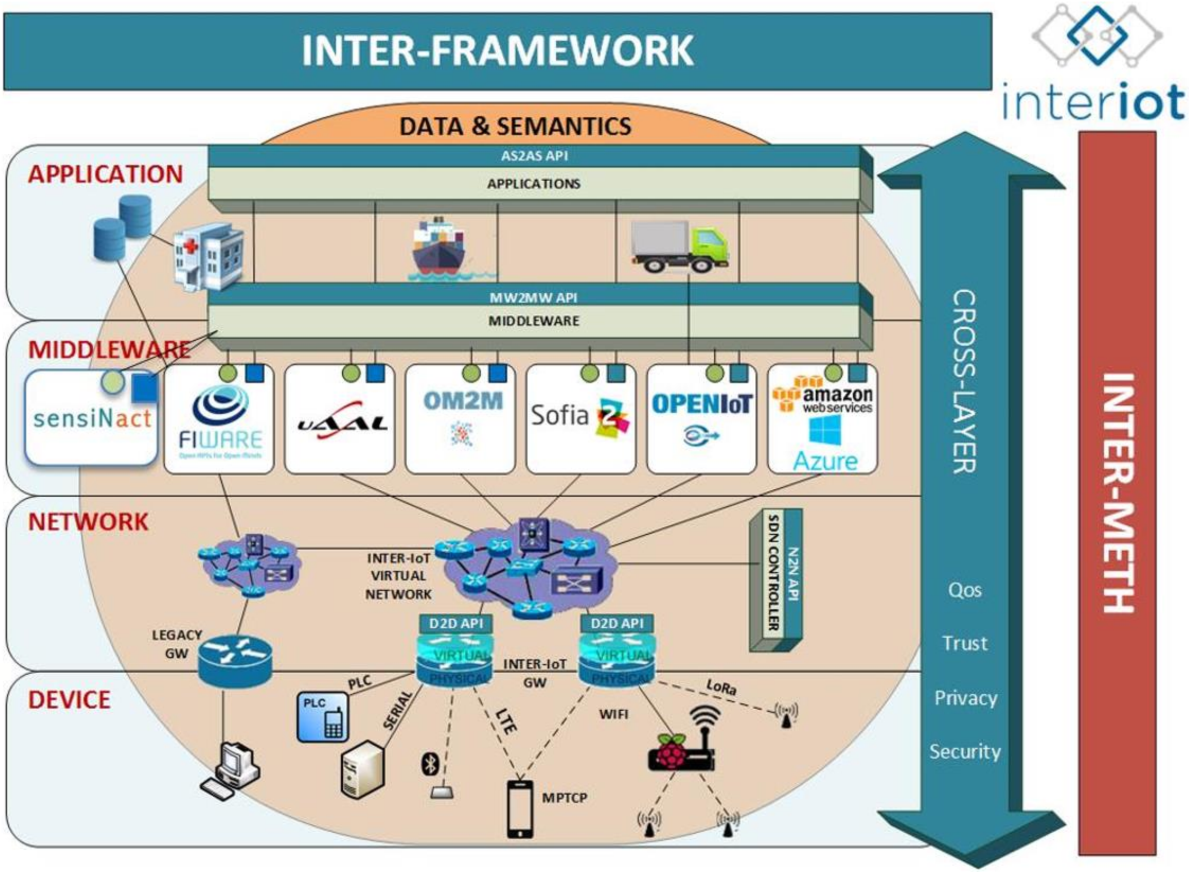


Figure 40: sensiNact layer integration

### 5.4.2 INTER-OM2M

Table **5** summarizes some examples of different scenarios, proposed equipment to be used and the corresponding connectivity protocols. We are open for other scenarios. Also, the exact type of sensors/base stations and their placements needs to be further discussed with the consortium. Inter-OM2M will have to be integrated into the MW-bridge of the Inter_IoT system.

Table 5: Suggested environmental measurement efforts in the port of Valencia.

| Scenario | Device/Equipment + Integration plan | Connectivity |
|---|---|---|
| Tracking of equipment/vehicles in port (position and movements) | Sigfox/LoRa trackers deployment | OM2M<br>CoAP, MQTT<br>Sigfox/LoRa |
| Monitoring of temperature | Sensors for temperature, Sigfox/LoRa motes deployment | OM2M<br>CoAP, MQTT<br>Sigfox/LoRa |
| Monitoring of rainfall level | Sensors, Sigfox/LoRa base station and Sigfox/LoRa motes | OM2M<br>CoAP, MQTT<br>LoRa / Sigfox |

### 5.4.3 INTER-HARE

In the INTER-LogP pilot, we want to demonstrate the need for a system that allows the exchange of data and messages among the different actors of the port community. In this case, as can be seen in Figure 41, there are three main actors: the port, the terminal and the haulier company. INTER-IoT has to provide interoperability between the IoT platforms of the port and the terminal, and give access to other devices from other companies, like trucks.

Both the port and the terminal have a large number of sensors and devices that produce large amounts of data, which can be interesting for other entities. Furthermore, they need data from other companies to provide a better service to their clients.

For instance, it is usual that in the hours of greater activity, there is congestion in the access gates to the port and the terminal. If it could be known the arrival of trucks in advance, you can manage more efficiently resources and avoid queues in the accesses.

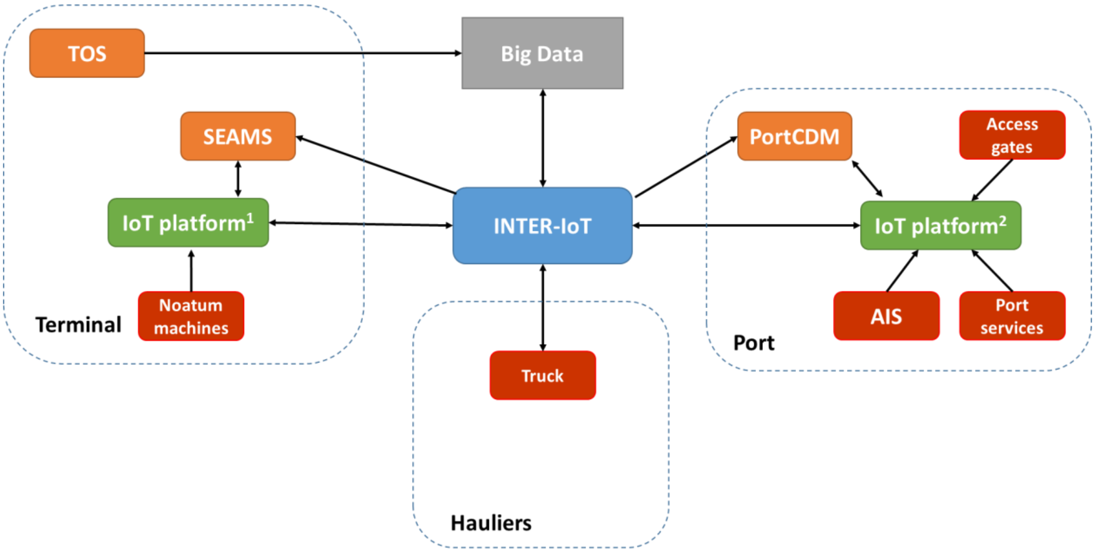


Figure 41: INTER-LogP pilot design

**Port**

The port authority has a large number of sensors distributed throughout the port that provides data for management and operation. Most of that data is confidential, but other can be shared, adding value to other companies.

The architecture for providing interoperability from the existing infrastructure is the one that can be seen in Figure 42. Currently, the port authority has a common database where all the data is stored coming from different systems (in red). It uses WSO2 to provide an IoT architecture in two ways: data in real time through the Message broker and historic data through the Data services server and Enterprise service bus.

Because the port has its own platform, the integration with the INTER-IoT is done through the middleware. It needs a bridge in the middleware layer in order to interoperate with order platforms.

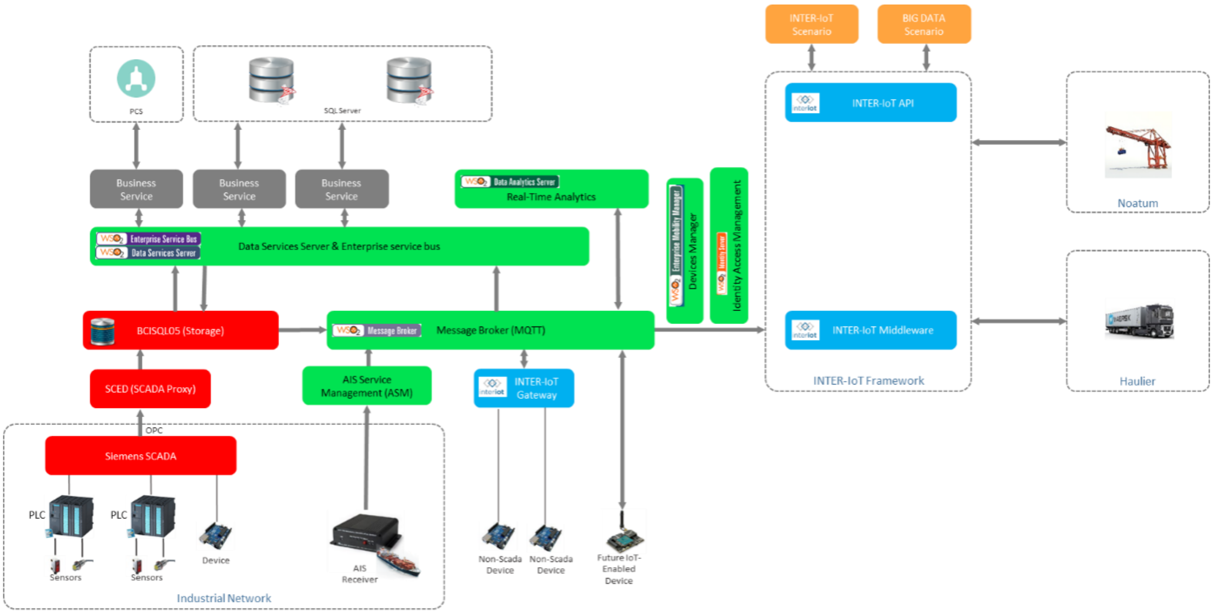


Figure 42: Port IoT platform and integration

## Terminal

A big container terminal should have monitored all the activity that takes place in the yard, including all the machinery, to be able to manage the resources properly. That is why in the Noatum terminal each of the machines (vehicles, cranes, etc.) provide multitude of data per second.

Each machinery has a PLC that is asked every second by updated data that is stored in a relational database. Then this information is used for other systems such as the dynamic lighting system or operational management platform. However, this structure is not scalable. As can be seen in Figure 43, now the data is pushed to the IoT platform instead of asking each second. Furthermore, new devices will only send information when a change happens. In addition, the data will be stored in a non-relational database, providing faster access to information.

As in the case of the port, the IoT platform of the terminal will be integrated with INTER-IoT through the middleware layer.

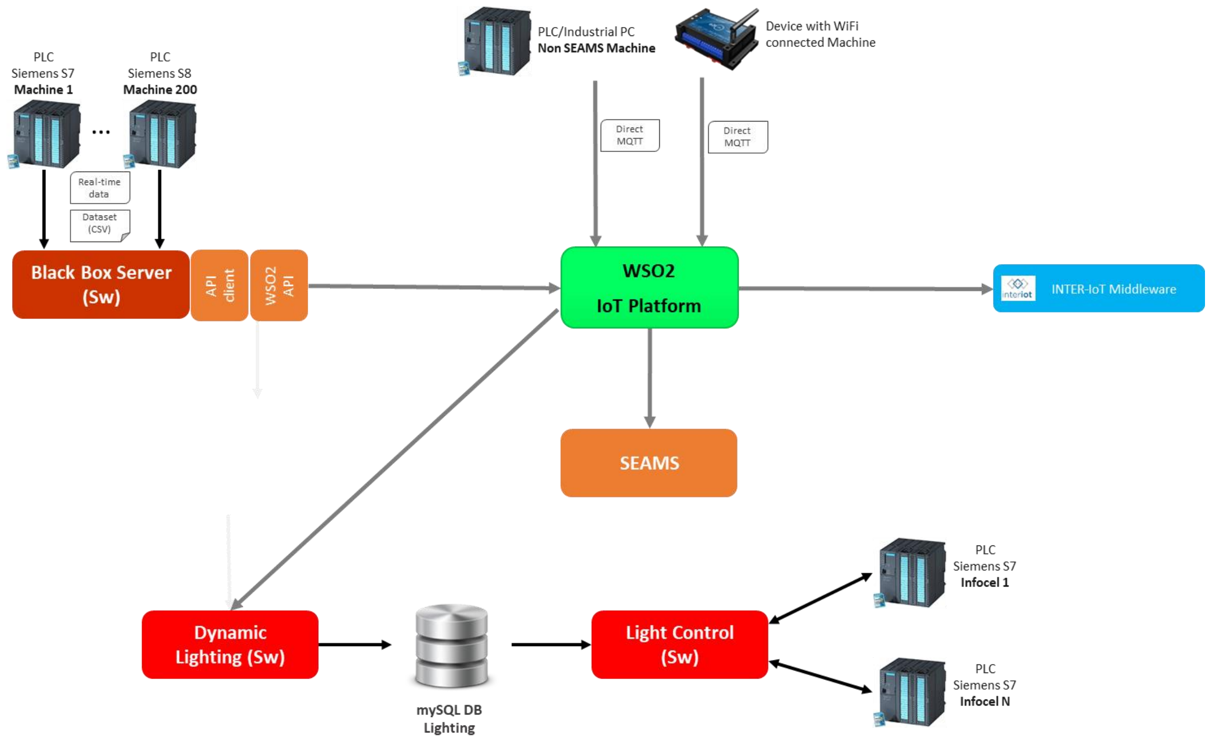


Figure 43: Terminal IoT platform and integration

### 5.4.3.1 Pilots and experimentation of INTER-HARE

In coordination with the INTER-IoT consortium, and after having reviewed the different use cases considered in the INTER-IoT project and compiled in ***Deliverable 2.4: Use cases manual***, UPF proposes to test its INTER-HARE platform in the ***Monitoring reefer container*** use case. The scenario is focused on tracking and monitoring the temperature of the container through different operators along its route, and to obtain faster responses in front of any issue with the temperature of the container.

According to the INTER-HARE network environment, our proposal sets out to deploy several sensors working at 2.4 GHz inside a monitored reefer container. This first data acquisition network will monitor the temperature throughout the container and will keep a constant control of it. All these sensors will be connected to a dual-band device located inside the container or attached to it.

The dual-band device will be equipped with two radio modules: one at 2.4 GHz and another one at 868 MHz. While the first interface at 2.4 GHz will be used to communicate with the data acquisition network, the second interface will transmit the gathered data from that container to the INTER-IoT gateway at 868 MHz, so that longer coverage ranges are achieved. In order to ensure the reliability and the 24-hour availability of this link, some other stations working at 868 MHz could be used as relays. A diagram of the infrastructure proposed to perform the pilot is shown in Figure 44.

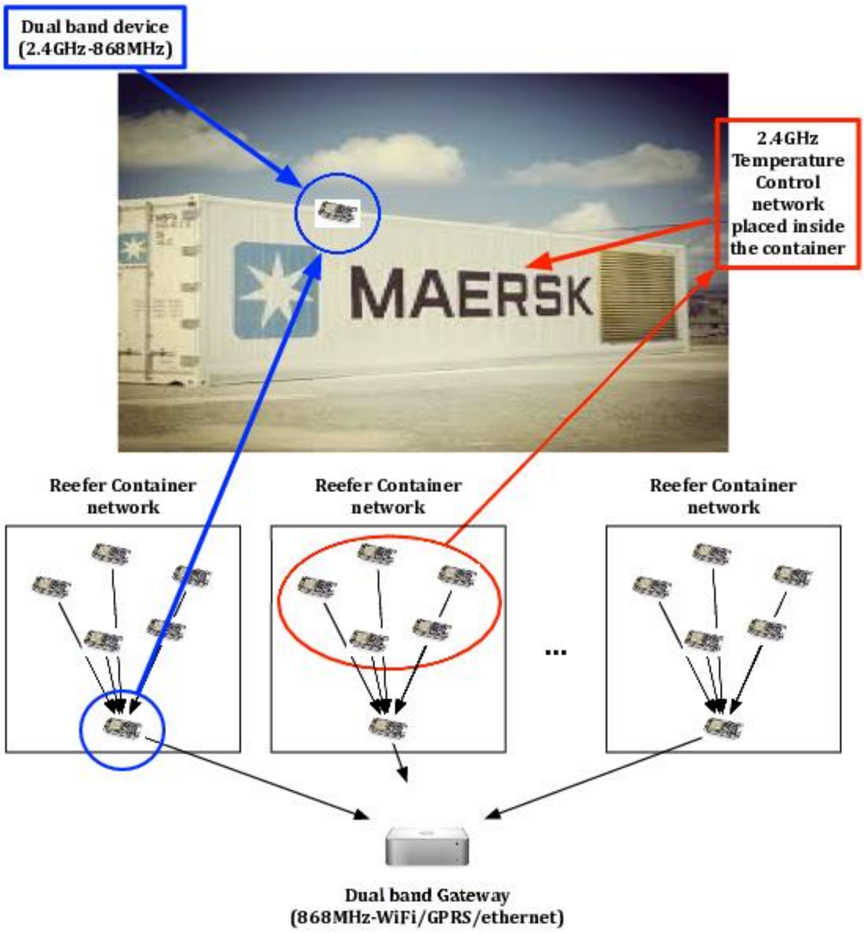


**Figure 44: Proposed diagram of the pilot**

Finally, the INTER-IoT gateway will be responsible for collecting all the information reported by sensors installed inside the containers. After some data processing and encapsulating tasks, the INTER-IoT gateway will transmit the collected data to the INTER-IoT dispatcher by using one of the following technologies: WiFi, GPRS or Ethernet. The dispatcher will in turn process the received data and send it to upper communication layers.

The whole INTER-HARE platform and especially the communication protocols involved in the pilot experimentation will be previously tested in UPF lab facilities in order to validate the viability of the proposal. As for the required equipment to perform the test, it is expected to use the devices depicted in Table 6:

*Table 6: Estimated pilot equipment*

| Device | Quantity |
|---|---|
| INTER-IoT gateways | 1 |
| Dual-band devices (868 MHz & 2.4 GHz) | 1 per reefer container |
| Data acquisition devices (2.4 GHz) | *N* per reefer container[2] |
| Relay devices (868 MHz) | *M*[3] |

[2] The number *N* of data acquisition devices within the reefer container will depend on the dimensions of the reefer container, the criticity of the transported cargo and the accuracy level desired.

[3] The number M of relay devices working at 868 MHz will depend on the distances between reefer containers and the INTER-IoT gateway.

### 5.4.4 Mission Critical operations based on IoT analytics

From a high level architectural standpoint, the foreseen integration of the MC-IoT external application is depicted hereafter.

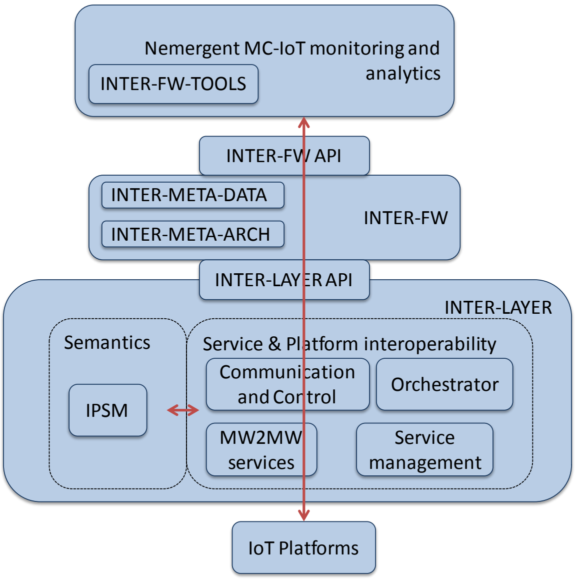


**Figure 45: Depicted high-level integration**

The MC-IoT system will run as an external application to the INTER-IoT platform. In order to access the heterogeneous data from different IoT platforms, the MC-IoT application needs to interact with the INTER-LAYER “Platform interoperability” functional component. Which involves the “Communication and Control”, “Bridges” and “MW2MW services” INTER-LAYER components. This module will also need to interaction with the “Semantics” functional component, which is implemented through the INTER-LAYER IPSM module. Additionally, the MC-IoT external application may need the use of composite IoT-related services. Thus, the invoking of the “Service interoperability” functional component may be needed. This would require the involvement of the INTER-LAYER “Orchestrator” and “Service management” components.

In order to interoperate with the INTER-IoT platform (e.g., registration, authorisation, etc.) and being able to invoke the API functions, the MC-IoT external application will make use of the INTER-FW tools and API. To some extent, the INTER-FW API acts as a wrapper of the INTER-LAYER API, exposing only those methods available to the external applications.

In order to support the test integration, Nemergent will provide access to different equipment endowed with the corresponding SW:

- Nemergent cloud system: access to INTER-IoT system; filtering of data; allowing access to the involved organisations (Port Authority and potentially first responders).
- General-purpose PC with a web browser, in order to access the dispatcher GUI.
- Commercial mobile phones and tablets, resembling the equipment that will be used by the different in-field participants.
    - Smartphone: Nemergent MCPTT Client SW + Simple GUI for accessing relevant INTER-IOT data.
    - Tablet: Nemergent SW with MCPTT communication capabilities + adapted GUI for displaying relevant information for the intervention.

### 5.4.5 Early Warning System

According to the project plan the implementation and integration of the EWS will be executed from September/2017 to October/2018. The complete and final evaluation in the lab is planned to be over along with the implementation (September/2018).

Devices required:

1. Shimmer 3 ECG: wearable medical sensor that will provide real-time data of the driver's vital signs, as heart rate and ECG. It will also provide accelerometer data. This device is compatible with UNICAL solution for BSNs.
   "The Shimmer3 Consensys ECG Development Kit can be utilized to monitor ECG (Electrocardiogram), recording the pathway of electrical impulses through the heart muscle. The ECG Development Kit can also be utilized to monitor (non-invasive) surface EMG, providing a representation of the muscle activity at the measurement site. Combined with Shimmer's integrated 9DoF inertial + altimeter sensor platform, greater context can be given to the wearer's activity and condition in real-time".
2. Android-based mobile phone: UNICAL solution provides an Android application that can receive and process the data from the Shimmer sensors through Bluetooth. This application is responsible for sending the data to the cloud environment. In addition, a mobile application can provide data about accelerometer and geo-localization.

The lab tests of the project include two parts:

1) Semantic translations and decision rules for emergency management:

It will be configured as an ontology translation between SSN and SAREF within IPSM (DS2DS). From a logical data mapping between them, two semantic translations will be implemented, one for each direction. The use cases with the scenario of accident at the port area (1) will guide the creation of decision rules to detect emergency situations based on combinations among parts of other standards and ontologies rather than SAREF and SSN, when necessary. These decision rules will be used to implement the emergency detection services of the EWS.

The lab validation plan of these translations (SAREF→SSN and SSN→SAREF) include unit tests for:

(i) classes/properties with direct mapping (1:1) with and without transformations;
(ii) multiple classes/properties to one class/property and vice-versa (n:1 and 1:n);
(iii) complex mappings requiring the creation of rules; and
(iv) unmapped elements.

The ontology translations will be configured within the IPSM module running on the INTER-IoT test environment. Then, the unit tests will be configured to access this test environment and execute, collecting the measurements resulted, as tests passed and tests failed. The EWS evaluation report will include a description of these results.

The tests include measuring the semantic interoperability offered by the bi-directional translations. A simulation of sending messages between two parties will be executed, where one party relies on SAREF and the other on SSN. The idea is to transform a message originally represented as SSN (m1) to SAREF (resulting in m2) and, then, transform this generated message (m2) to SSN (m3). Semantic similarity will be measured by comparing the initial message (m1) with the final message (m3) after executing the bi-directional translation, i.e. check if m1 is similar to Ta->b(Tb->a(m1)). The opposite way will also be tested, i.e. from SAREF to SSN to SAREF. This test will also be performed in the INTER-IoT testing environment.

The decision rules relating the data represented as SSN and/or SAREF to the EDXL standards will also be evaluated. EDXL messages will be manually generated based on the rules described and the input data. Then, tests will generate automatically the EDXL messages, according to the same input data. The manually and automatic generated EDXL messages will be compared to check inconsistencies and data loss.

2) Situation-aware IoT EWS prototype for accidents at the port area (9):

It will be investigated if it is necessary to include IoT platform components for implementing the EWS, e.g. (i) FIWARE IoT local broker, for local gateway that acts as a publisher of sensor data from IoT devices to the context broker, and (ii) FIWARE context broker, where the IoT EWS will act as an application subscriber, getting data/context from the context broker. During pre-processing of input data, the EWS will use IPSM translation services. Then, harmonization and translation of different data formats can be performed, followed by integration with the EWS core ontology (the internal reference model), which can be an extension of the ontology of IPSM. Decision rules describe the constraints on the contextual elements, triggering the detection of an emergency risk. Data related to the situation are described in a message according to EDXL standards. Then, this message is made available to subscribers through pub-sub REST service with JSON-LD, enabling the EWS to send messages to subscribers.

The description of the use case requirement for translating between SAREF and SSN messages is under progress. Initially, the idea is to represent data sent by the Shimmer and mobile devices as SAREF, once SAREF is recommended to represent energy consumption properties and the scenario of the driver using the devices requires this representation. The internal context model of the EWS, a core ontology, is an extension of SAREF. Therefore, translations from SAREF to SSN is required. The opposite direction can also be useful if the EWS provide notification services linked to the SAREF ontology, thus, from the EWS core ontology, translations from SSN to SAREF are required.

The initial lab tests validation plan include unit tests, organized as illustrated in Figure 46:

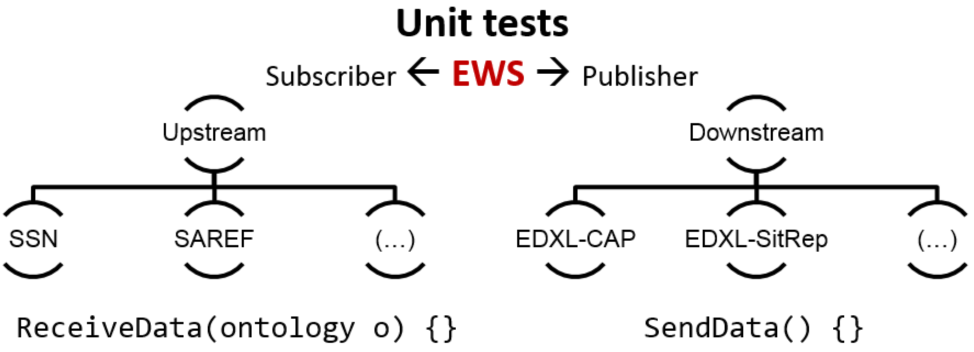


**Figure 46: Organizational structure of IoT EWS**

(i) upstream data: EWS plays the role of subscriber by receiving data. Data tests are used to validate that data represented according to ontologies/standards can be received as input. Initially, the tests consider data sent to the EWS as SAREF only, but the use cases being detailed can also include the requirement of SSN input; and

(ii) downstream data: EWS plays the role of publisher by sending data. Data tests can validate that proper EDXL messages can be generated. It is under discussion whether the EWS will improve EDXL with SSN and/or SAREF through the ValueList, ValueType, ValueListType and ValueKeyType elements of EDXL, which are proper for extending the standard with ontologies.

Sensor data pre-processing will be tested with a combination of unit tests.

Functional tests will validate if input data is framed by the rules of emergency situations, and the EDXL messages generated from the detection of these situations are semantically correct. It includes the simulation of detecting an emergency and sending alerts to different target groups. These functional tests in lab environment will include a person wearing the medical device and the mobile phone transmitting data to the EWS while driving through the university campus. The accident prediction and detection will be tested by changing thresholds on the rules and simulating the detection of emergencies. For example, vehicle collision will be tested by decreasing the threshold compared to the axial energy function in a way that a strong break will be the simulation of a car crash. Changes in vital signs can be tested also with similar approach, for example for tachycardia detection the threshold is decreased and the person simulating the driver is demanded to make exercises.

Test data will include both generated by the functional tests, as described above, and also mockup data test.

With these tests, the components of the EWS is tested as follows:

A. Input handler: data acquisition

The collection of raw data from several sensors will be evaluated, e.g. Shimmer and mobile accelerometers. These sensor interfaces providing through APIs (REST services) will also be tested. The pre-processing of sensor data will also be covered in this evaluation.

B. Abstraction: data enrichment

They data collected by the input handler will follow different formats and represented with SSN and SAREF, according to their requirements, and their integration tested with support of IPSM.

C. Context model: domain ontology

The evaluation covers this component when storing the data according to the EWS internal context model, which can be an extension of IPSM ontology.

D. Situation model: structural and temporal rules

Within this component a CEP technology will be tested for detecting event patterns in real time to detect accidents according to the rules underlying each use case, i.e. the situation models.

E. Situation awareness: decision making

This module will be tested by the inference produced from the application of the situation models over the test data represented according to the context model. Therefore, for each raw test data collected by the EWS, the inference engine will be executed to identify possible accidents in the lab environment.

F. Output handler: emergency situation

This module will also be tested through the notification services exposing data according to the OASIS EDXL standard.

Parts of the system that the tests will not cover include the network level, e.g. packet loss and bandwidth delay. The tests will not cover the data processing in the device level. The focus of this project is the data integration at the application level, giving emphasis to semantic integration in the cloud.

### 5.4.6 SENSHOOK

The Senscape hardware developed by Irideon includes a range of host boards for all kinds of sensors, featuring multiple communication interfaces and power supply configurations. It is a modular, standards-based platform for Internet of Things (IoT) sensor applications. Senscape is a product for professionals which provides tools to develop new marketable Internet of Things (IoT) sensor applications with ease.

Irideon’s Senscape platform is a development environment, which we and our customers use to develop fast IoT solutions for a wide range of applications.

Existing Senscape hardware products (electronic baseboards) include Gecko (for wearable applications), Tuatara (for static and mobile applications) and Flying Dragon (for demanding signal processing applications). The baseboards use our SENSOS embedded operating system, written in C/C++. SENSOS delivers best in class performance vs. power consumption and supports a set of advanced features including:

- Remote data preview
- Multi-device time synchronization
- Remote device smart energy management
- Remote auto-check
- Remote updates
- Secure communications.

Furthermore, it contains a fixed and floating point signal processing library (with a MATLAB counterpart to enable simulations), driver libraries for a range of sensors and actuators and modules for wired and wireless communications.

We currently use third party IoT server technologies to access Senscape devices, which provide common functions such as data storage, data analysis, data visualization, etc., but which do not support some of the more unique features offered Senscape, including many of the IEEE1451 features. Having evaluated and tested a number of 3rd party IoT server technologies, we now plan to develop an open standards-based middleware for integration in any IoT application based on INTER-IoT framework called SENSHOOK, to unlock the unique features embedded into Senscapeas well as allow interoperability among other IoT platforms and services.

#### 5.4.6.1 Senshook / Irideon integration setup

Senshook wants to develop an open standards-based middleware for integration in any IoT application based on INTER-IoT framework and allow interoperability among other IoT platforms and services.

**Unit Tests**
The different bundles and components will have unit tests.

**API Testing**
Furthermore to assure the correctness after deploying, we use API testing.

Dispatcher REST API testing
For integration testing of the dispatcher, we use the following setup. On one hand there is a client which executes the tests, on the other is the Senscape board and in between there is a server running the dispatcher. The connection between client and server uses HTTP and between server and the Senscape board it is establsihed via MQTT.
Postman is a free API development environment. It offers the option to write complete tests for a REST API, so we use it to do status code validation, data type validation, etc.

Below you can see a diagram of the test setup:

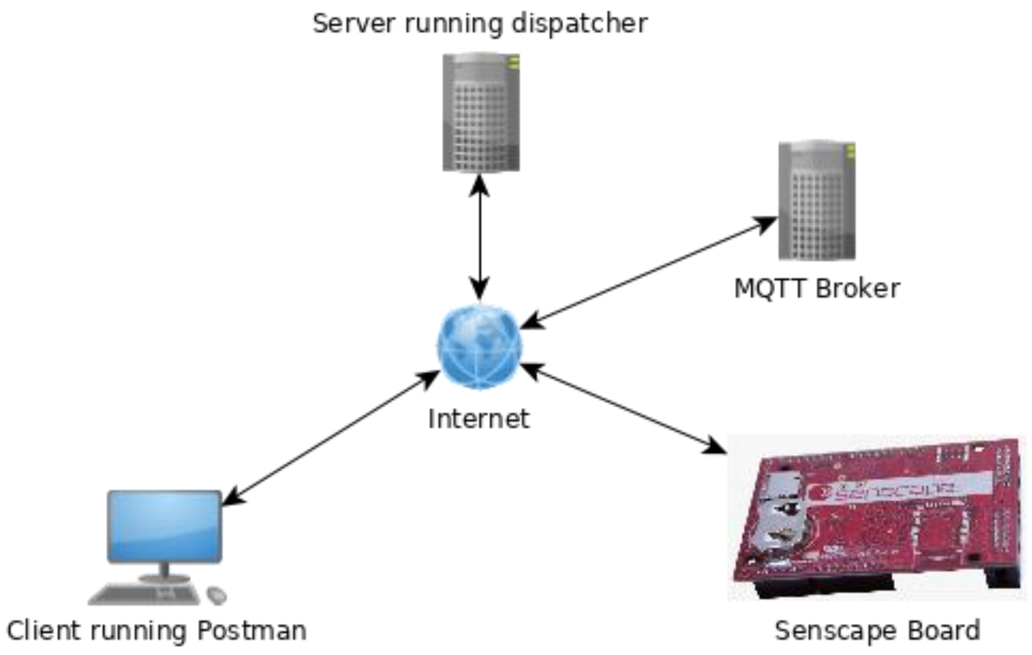


**Figure 47: Senshook test setup**

Middleware controller REST API testing
These tests are not yet implemented.

### 5.4.7 SOFOS/INFOLYSiS integration setup

INFOLYSiS contribution will be integrated within IoT virtual gateway by providing to the INFOLYSiS SDN/NFV Network Manager module administration access to the build-in SDN controller in order to be feasible the appropriate traffic steering to be performed for the provision of the interoperability.

SOFOS aims at piloting the INTER-Domain scenario of INTER-IoT, building on top of the following INTER-IoT use cases:

- Business Scenario #9: Accident at the port area (INTER-LogP)
- Business Scenario #10 Health monitoring system with passengers aboard a ferry (INTER-LogP/Health)
- Business Scenario #30: IoT access control, traffic and operational assistance (INTER-LogP)
- INTER-LAYER Scenario #41: SDN communications: functions virtualization and central management
- Scenario #55 SDN communications: traffic routing

The proposed pilot considers an emergency situation where vessels with casualties are approaching the port where the health units/rescue teams should be prioritized/coordinated depending on the health condition of each casualty. The implementation and use of the SDN paradigm by SOFOS will speed up IoT connections, provide interoperability among different IoT health devices and centralize the management between the vessels domains and the port domain. Moreover, the SDN applicability will allow the prioritization of IoT data flows using traffic engineering, achieving a general overview of the whole network at any time. SOFOS pilot will provide at the first responders commander, who will coordinate the rescue teams, a unified view of IoT data visualisation. More specifically, the proposed SDN/NFV-enabled IoT GW will be used to provide interoperability between eHealth IoT systems on the different vessels with the coordination center at the port with scope to provide a common unified view of the patients/casualties and the location of the available rescue teams. For this purpose, a virtual mapping function that implements an existing interoperability standards commonly used in healthcare information systems will be deployed by the SDN/NFV orchestrator, offering interoperable and continuous data transmission, allowing to the coordinator to allocate at each available rescue unit the appropriate casualty.

**Preliminary KPIs:**

- Provision of interoperability over heterogeneous IoT units based on SDN/NFV techniques (KPI: no # of heterogeneous IoT nodes that are connected in an interoperable way).
- Unified data visualization of heterogeneous IoT units based on SDN/NFV techniques (KPI: no # of heterogeneous IoT nodes that provide data to be visualized in an unified way).
- Agile provision of IoT interoperability over heterogeneous IoT domains as a service based on SDN/NFV techniques (KPI:no # of VNF deployments and SDN rules of heterogeneous IoT nodes that are connected in an interoperable way).
- Performance metrics of SDN prioritization of IoT streams under stressing conditions (KPI: delay, packet loss vs. available bandwidth).

### 5.4.8 E3Tcity Smart City Platform and Devices Integration

E3Tcity platform's main interest is to explore the integration of a complete vertical IoT solution with INTER-IoT environment. Due to the current architecture of E3Tcity platform, the most appropriate layer to do this kind of integrations is the Middleware layer of INTER-IoT Interlayer.

Key points to test in lab:

- Control of e3tcity devices from InterIoT platform
- Correct behaviour of devices (connection to cloud,
- Activation/dimming of a test streetlight

The way to connect e3tcity devices to lights is pretty straightforward:
Test to carry out is realy simple in this stage:

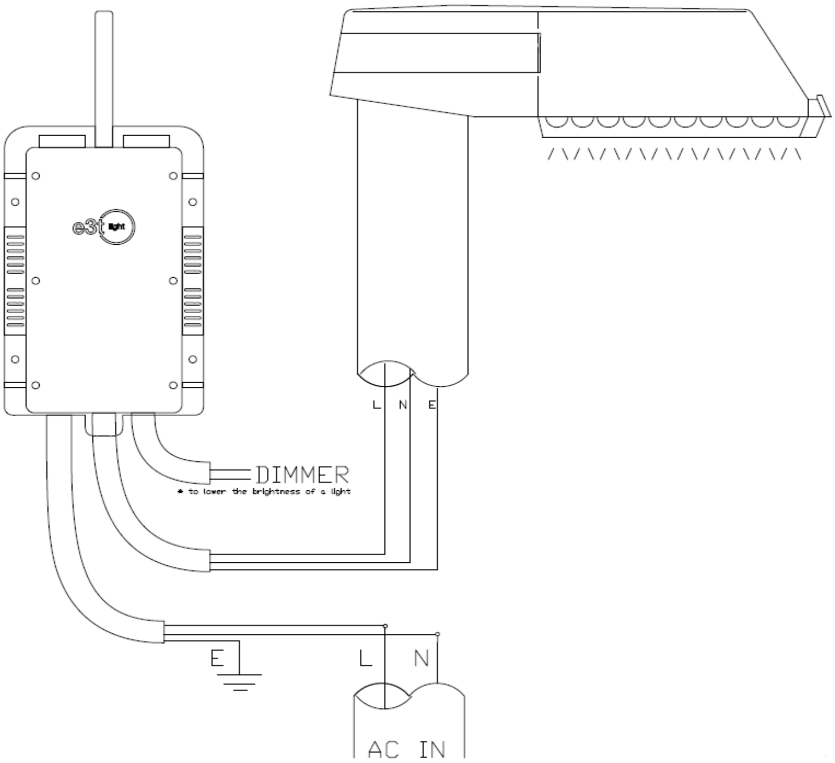


**Figure 48: Wiring overview testsetup**

1. Once connected to the line, the controller will look for a GSM network. Blue led will blink slowly during this period.
2. Once connected to Gsm network, the controller will look for MQTT server, while blue led blinks fast.
3. Once connected to MQTT, a fast switch on/off and dimming test will be done.
4. In case of sensors (not all devices will carry sensors), values from sensors will be read and sensors info update will be verified.

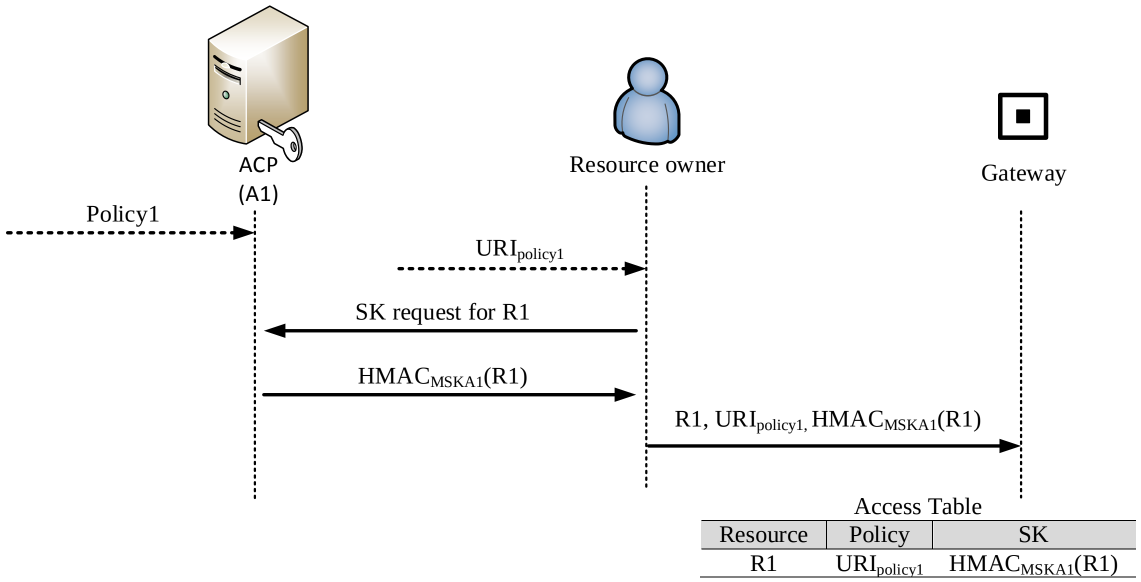
### 5.4.9 ACHILLES

ACHILLES will extend the INTER-IoT platform to provide gateway/Thing mutual authentication, user/gateway (or Thing) mutual authentication, and access control.

The gateway's device manager uses the ACHILLES protocol module implemented in the protocol controller and communicates with the IoT devices.

The pilot will use an external device that will act as the Access Control Provider (ACP). This device can be a PC/Server/VM in the Cloud, supporting Java, Eclipse Jetty and HTTPs. In the FAT setup the ACP will implement a simple user management system, as well as simple access control policies (e.g., based only on usernames). Moreover, external devices capable of running CoAP and TLS will be used as clients.

The test setup will involve the configuration of the Inter-IoT gateway with the necessary information in order to provide access to protected resources. In particular, for each protected resource a gateway should be configured with the resource name (e.g., its URL) a "pointer" to an access control policy, as well as, with a secret key. This secret will be produced by the ACP using the process illustrated in the following figure.



| Resource | Policy | SK |
|---|---|---|
| R1 | $URI_{policy1}$ | $HMAC_{MSKA1}(R1)$ |

**Figure 49: Integration and Factory test setup overview**

In this example, a resource owner wants to offer a resource, identified by R1, protected by the policy Policy1: he requests a secret key (SK) from the ACP and the ACP calculates it using a secure HMAC function (and as a private key an ACP-specific master secret key). Then the resource owner stores all this information to the gateway.

Currently, a functional ACP implementation has been developed and tested.

### 5.4.10 INTER-HINC Situation-aware cross-domain operations based on IoT analytics

INTER-HINC will be developed using Java and NodeJS/Javascript with standard libraries so all integration will be done through RESTful APIs and client libraries implementing MQTT and AMQP protocols. For integration testing, we will stick to these types of APIs so it wont require specific languages or toolkits for integration testing in order to invoke INTER-HINC.

On the other hand, INTER-HINC will also interact with other components through RESTful APIs and MQTT. Therefore, we require other IoT Platforms and INTER-IoT components to provide such APIs through REST and MQTT. From the design of INTER-LogP and INTER-HEALTH using WSO2 with EBS and MQTT-based Message Brokers, it seems that this requirement will be met (but the detail must be examined).

For integration and testing, we will rely on the current scenario to test the following steps:

- activating monitoring containers with sensitive goods in the port,
- analyzing and controlling robotic cranes and trucks to make sure that they do not prevent the emergency responses as well as ready to support the responses,
- sending alarms and controlling vessel arrivals and revising transport planning,
- providing information for operational assistance for the emergency responses, and
- activating systems to support the monitoring of people impacted by the accident using devices and platforms for chronic disease and cognitive decline prevention.

Two achieve the tests, we will perform in-lab testing and field-testing. For in-lab testing, we need to have access to a simplified deployment of other Inter-IoT middleware, emulated IoT platforms and sample data of ports, terminal, trucks, emergency response systems.

- A simplified deployment of other Inter-IoT middleware can be achieved either by downloading such middleware and seting up in our own infrastructure or using an exsiting deployment from the project
- Emulated IoT platforms can be obtained through the providers or be setup by us. If not possible, we will have emulated, simplified actions of such IoT platforms to be implemented for testing (just basic operations for testing). Sample data should be able to obtain by extracting existing data from the current systems
- Our test system will be then deployed in our infrastructure for in-lab testing. The test system will be accessible from the Internet.

### 5.4.11 A Semantic Middleware

1. Turn-on Raspberry and the Ebeacon sensors if they are real, otherwise start the "virtual sensors". The sensors (virtual or real) start to send data containing the position of the corresponding pallet. For this test, we can start using 3 pallets.
2. Open the "virtual application" simulating the working stations. They start to send data containing the availability status and the machining operations that can be performed.
3. Open the simulation application which starts to consume the data sent at the steps 1 and 2 (through a proper subscription performed using a specific SPARL query) and decides the best resources allocation of the pallets. The decided allocation is published in order to be consumed by the pallets.
4. The virtual carriages start to consume the data published at the step 3 subscribing through a proper SPARQL query.
5. The virtual carriages receive the data published at the step 3.

This test allows to verify all the functionalities exposed by the Semantic Middleware, starting from the capability to publish and consume data. No functionality remains untested.

For any IoT middleware, the quality of service (QoS) requirements in terms of latency, efficiency, and scalability are stringent. For this reason, an evaluation of these requirements has been conducted for the Semantic Middleware in a real case study by means of a defined benchmark.

The benchmark consists in processing simultaneously several parameters affecting the systems workload:

- the size of the knowledge base to work on, in terms of triples count (k);
- the number of working sensors (s);
- the number of subscripted consumers (smart services or sensors) under a set of queries (c);
- the number of subscriptions for each consumer (q).

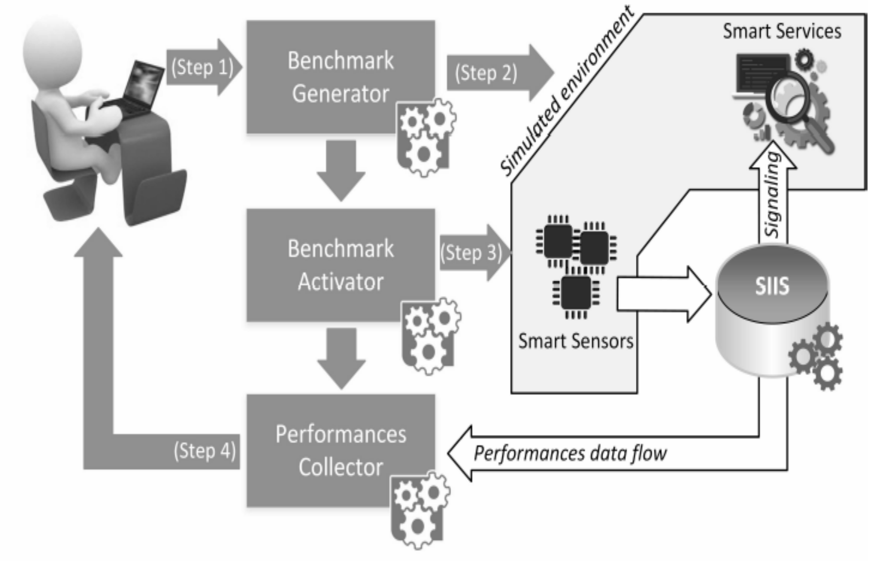


**Figure 50: Test framework**

To perform the evaluation a test framework has been developed as illustrated in Figure 50. Initially, the user specifies the work parameters k, s, c and q, together with its preferred relevance or, alternatively, uses the standard benchmark configuration to create a test setup (Step 1). Then, the Benchmark Generator module generates a simulated environment (Step 2) made up by the desired instances of both producers and consumers, the former notifying the desire information to SM, and the latter subscribing to its signaling services. The generated environment is then activated (Step 3) to start performances measurements. During the execution lifetime, each Stimulus-Reaction Correlation (SRC) is logged and transferred to be persisted to to the Performances Collector. SCRs allow to both track input-output relations and tear down latency correlations. Finally, performances are aggregated and reported to the user (Step 4) as tables and diagrams needed to assess performances. All tools provide an easy to run environment and thus they require very little effort to be executed.

### 5.4.12 SecurIoTy

Initially SecurIoTy will be operated off-premise, i.e. in our data centers. Interface and access will be provided through public internet access. If on-premise operations are mandatory, the use case owner (i.e. Port of Valencia) will have to provide data center access and infrastructure. Details have to be discussed asap for further planning.

As soon as the use case is available to us including all relevant architecture details we will define test and acceptance tests employing the RUP standard.

We employ the Rational Unified Process (RUP), an iterative software development process framework originally developed by IBM. RUP gives us a detailed canvas for software development and integration projects (Figure 51).

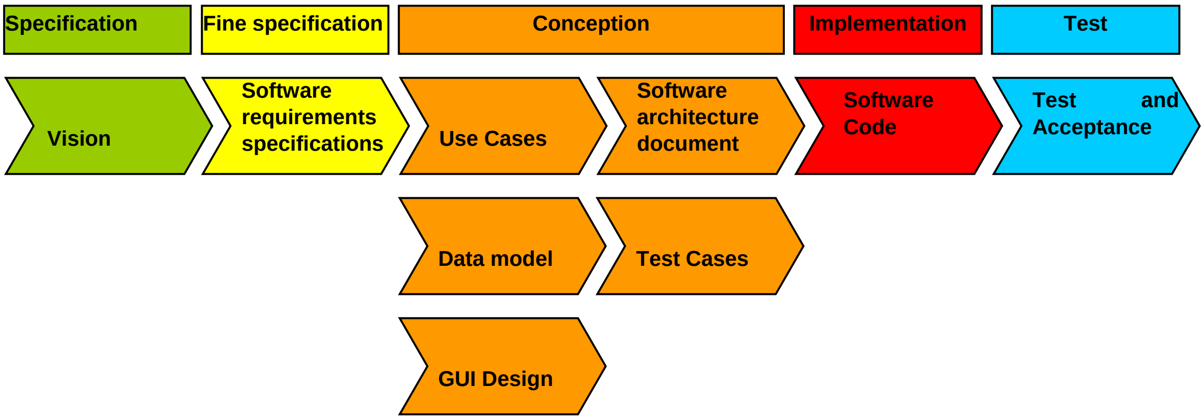


**Figure 51: SecurIoT development and test process based on RUP**

#### Vision and fine specification (SPEC)

The vision and fine specification needs to be synchronized with the INTER-IoT framework.

### Conception (CON)

In the conception phase, we will detail the use cases, test cases and GUI design (which will probably be minimal due to machine to machine communication). The software architecture and data model documents can be drawn from our existing experience and libraries.

### Implementation (IMPL)

The actual coding, respectively adaption and refactoring of our existing code base will be handled in the Implementation phase. Implementation will typically be done in iterations. The INTER-IoT schedule plans for 2 iterations which can be accommodated for in our development cycle.

### Test (TST)

Test plans and acceptance criteria will have to be defined based on the architecture and the INTER-logP use case.

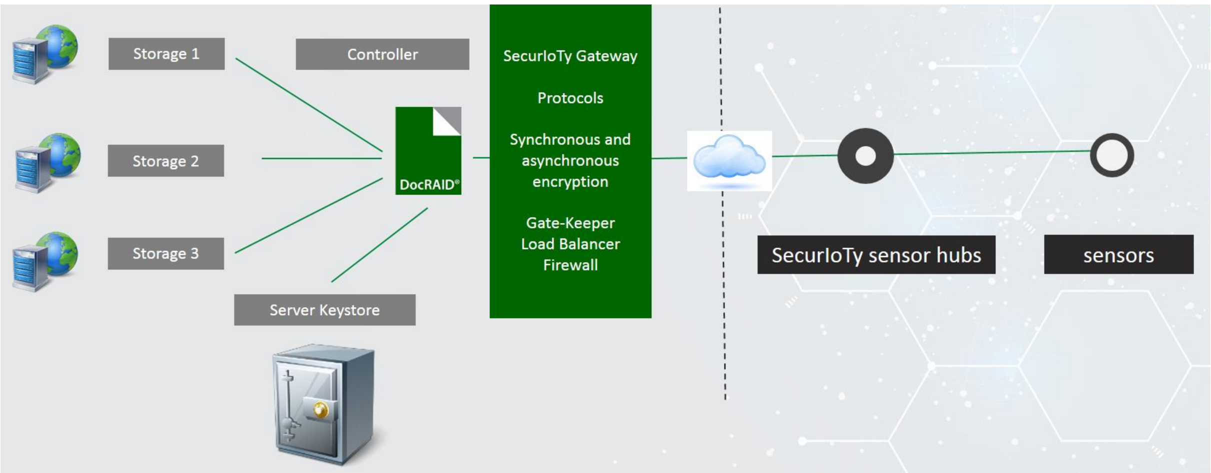


Figure 52: Integration and Factory test setup overview

Assuming that the DocRAID crypro proxy will be set up as a sensor gateway (see Figure 25), we will test functional and non-functional parameters like

- availability and response times
- the gateway (i.e. controller)
- the storages
- optionally the key store, depending on the use case details

# 6 Integration environment

## 6.1 Pilot project integration

For each pilot a high-level description of what the pilot integration test environment will look like and what and how this will be tested is needed.

### 6.1.1 Factory test setup (high-level) – INTER-logP

During integration in the port the INTER-IoT system will be connected to the port systems of Noatum. The integration setup is shown in the following figure.

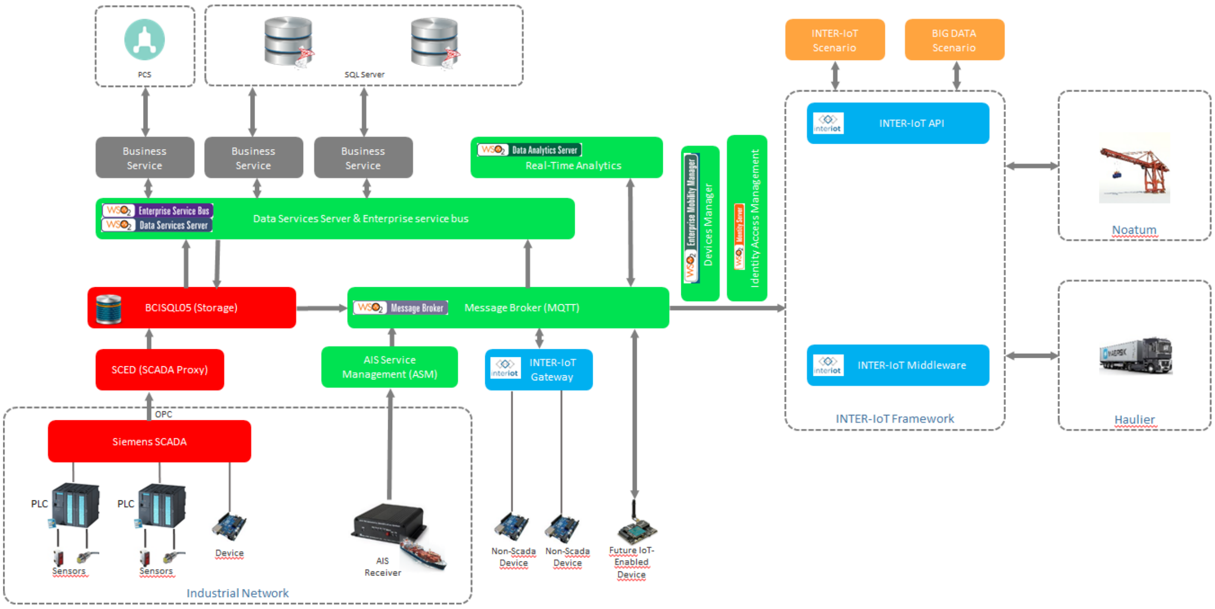


**Figure 53: SAT of the LogP integration**

### 6.1.2 INTER-HEALTH ASL TO5 integration setup

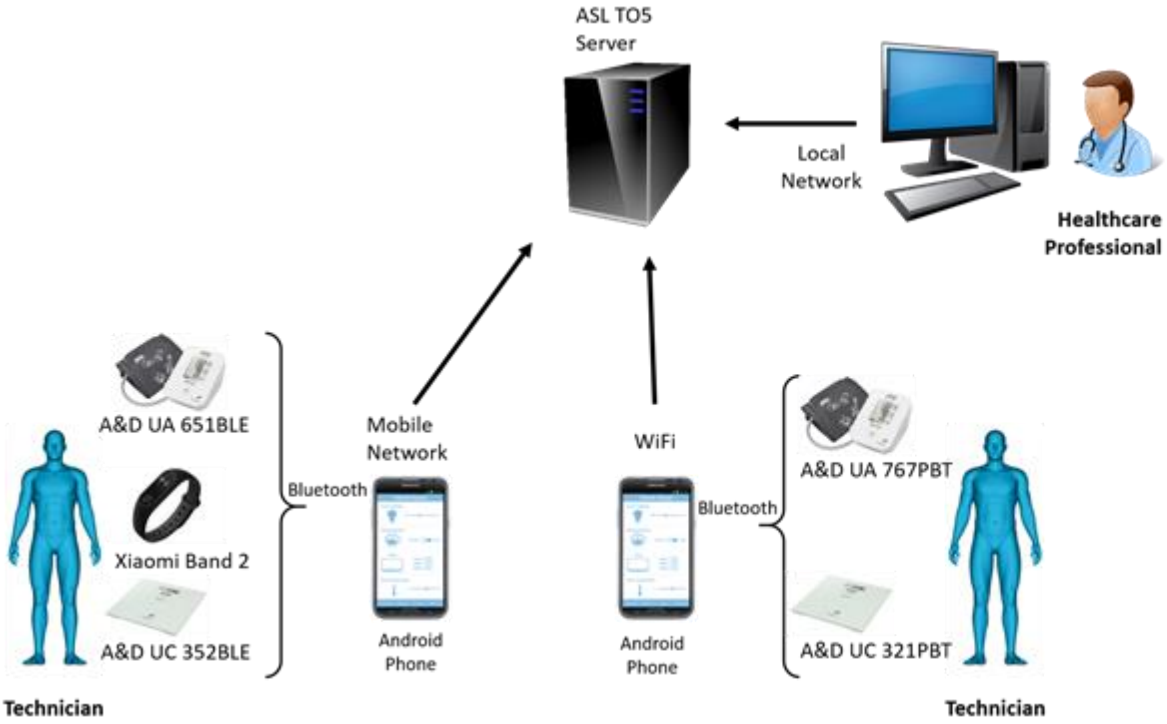


Figure 54: INTER-HEALTH Integration Overview

In this integration environment, the difference with the final pilot deployment is the phones used to simulate the ones of the subjects. In the test setup at UPV-SABIEN this will be tested initially with one model (to be determined), other models will be tested if possible.

All main use cases described earlier will be tested. The main concern during these tests is the setup of the network, and the ability to service as many phones as in the pilot. Similar load tests performed at the UPV-SABIEN test setup will be repeated, only this time the results will be representative for the pilot.

## 6.2 Third party project Integration

For each third party project a high-level description of what the pilot integration test environment will look like and what and how this will be tested.

### 6.2.1 sensiNact integration

The integration of sensiNact to the Inter-IoT Framework will be done at the middleware layer. sensiNact is currently in use as IoT middleware platform in various collaborative projects such as OrganiCity, FESTIVAL, BigClouT, Wise-IoT, ACTIVAGE and IoF2020, in which applications in various domains have been (and will be) developed and deployed in close to real life environments in domains such as smart city, smart home, smart farming, smart living and smart ski resort. Integration. With integration of sensiNact with the Inter-IoT framework, we aim at bringing more devices, platforms, thus more data sources to enrich the "catalogue" of Inter-IoT supported platforms. This will allow, not only to Inter-IoT validating its interoperability methodology and tools but it will also allow sensiNact being compatible with other platforms supported by Inter-IoT.

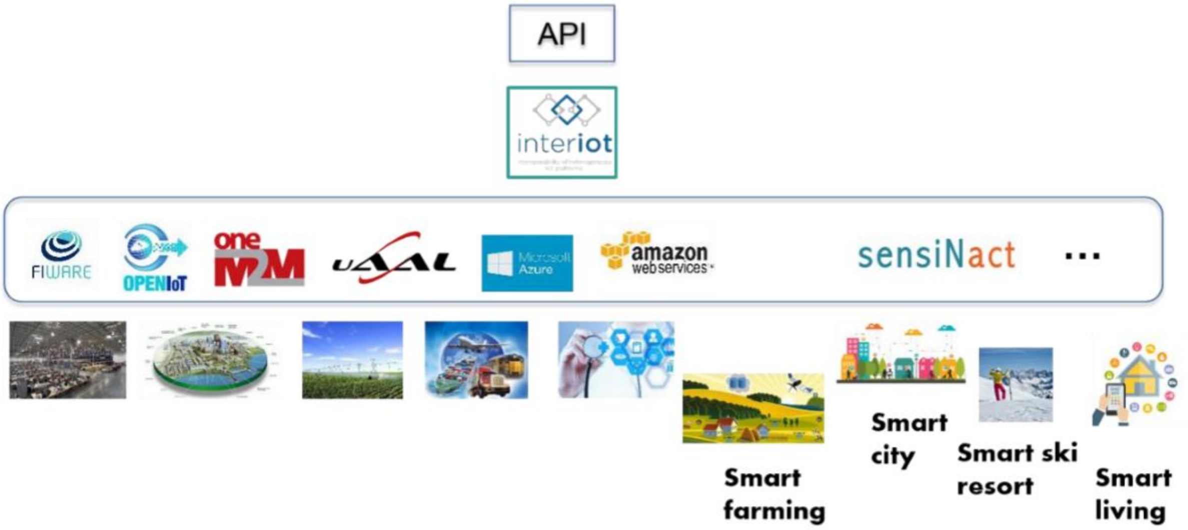


Figure 55: sensiNact platfrom integration

### 6.2.2 INTER-OM2M

This system will be composed by devices using heterogeneous long-range technologies (SIGFOX, LoRa) to track equipment and perform temperature measurements. Some field measurements will be performed to ensure that the radio quality of the links is good enough for the correct reception of the data. This will be done by performing RSSI and LQI measurements for the LoRa motes and the RE-motes, and by verifying the Network Quality information shown in the platform from the manufacturer of the StickNTrack SigFox nodes.

The open source OM2M-based implementation will be deployed as common service layer to allow the use of different application protocols such as MQTT and CoAP in the devices. A raspberry Pi 3B plays a central role in the system acting as a gateway unit for the gathered data. Based on the oneM2M terminology, the raspberry Pi 3B plays the role of a Middle Node Common Service Entity (MN-CSE) which will be registered in a powerful computer in the VUB university running the Infrastructure Node (IN-CSE) that stores the data. In this way, we can directly obtain all the measurements from the computer installed in the university. Since the city of Valencia is already equipped with an alternative monitoring system (FIWARE framework), we planned to develop an INTER-IoT – oneM2M bridge to make available the environmental measurements of the FIWARE framework under the VUB platform. Moreover, the bridge will provide a link to obtain data from the oneM2M VUB server for any applications in the city of Valencia.

Finally, we plan to test the correct functioning of the bridge, by querying services to the OneM2M server from the FIWARE system and from the environmental monitoring system deployed by VUB in the port and the INTER-IoT server.

### 6.2.3 INTER-HARE

The hardware components of the pilot to be performed at Port of Valencia are the same as described in INTER-logP with the addition of the Maersk reefer containers and a container crane.

#### 6.2.3.1 Maersk reefer containers

A refrigerated container or reefer container is an intermodal container (shipping container) used in intermodal freight transport that is refrigerated for the transportation of temperature sensitive cargo.

While a reefer will have an integral refrigeration unit, they rely on external power, from electrical power points (“reefer points”) at a land based site, a container ship or on quay. When being transported over the road on a trailer or over rail wagon, they can be powered from diesel powered generators ("gen sets") which attach to the container whilst on road journeys. Refrigerated containers are capable of controlling temperature ranging from -30 °C, -40 °C, -65 °C up to 30 °C, 40 °C.

A.P. Moller-Maersk Group, also known as ***Maersk***, whose reefer containers will be used in the INTER-HARE pilot[4], is a Danish business conglomerate with activities in the transport & logistics and energy sectors. Maersk has been the largest container ship and supply vessel operator in the world since 1996. The company is based in Copenhagen, Denmark with subsidiaries and offices across 130 countries and around 88,000 employees.

Each Maersk reefer container used in the pilot will contain one cluster-head and several (*N*) data acquisition devices which will monitor the internal temperature and humidity in different container locations.

[4] Maersk smart containers - http://www.maersk.com/en/the-maersk-group/about-us/publications/group-annual-magazine/2015/smart-containers-listen-and-talk

Figure 56: Maersk reefer containers

### 6.2.3.2 Container crane / Port Terminal

A container crane is a type of large dockside gantry crane found at container terminals for loading and unloading intermodal containers from container ships. The cabin of a container crane (or alternatively, any other port facility with enough energy supply and communication capabilities with the rest of network elements) will contain the INTER-IoT physical gateway.

Figure 57: Container cranes at Port of Valencia

### 6.2.3.3 Pilot high-level description

In coordination with the INTER-IoT consortium, and after having reviewed the different use cases considered in the INTER-IoT project and compiled in ***Deliverable 2.4: Use cases manual***, UPF proposes the ***Monitoring reefer container*** use case as Site Acceptance Test (SAT).

The SAT takes place after integration at the customer site. During SAT, the proposed solution is tested on integration, performance, conformance to specifications, and user acceptance testing. The SAT is performed to prove that the solution is integrated correctly into the customer’s environment and meets all the stated requirements.

In the particular case of the INTER-HARE platform, the scenario proposed for the pilot is focused on tracking and monitoring the temperature of the container through different operators along its route, and to obtain faster responses in front of any issue with the temperature of the container.

Therefore, the pilot will finally validate the INTER-HARE platform as a reliable and low-power end-to-end communication system in industrial automation environments, like the one created in port terminals with the monitoring of reefer containers.

#### 6.2.3.4 Deployment

Pilot deployment will be performed in one of the terminals of the Port of Valencia forming a network architecture like the one shown in Figure 58 and including the elements depicted in Table 7.

*Table 7: Estimated pilot equipment*

| Device | Quantity |
|---|---|
| INTER-IoT gateways | 1 |
| Cluster-heads<br>(868 MHz & 2.4 GHz) | 1 per reefer container |
| Data acquisition devices<br>(2.4 GHz) | *N* per reefer container[5] |
| Relay devices (868 MHz) | *M*[6] |

The INTER-IoT gateway would be ideally located in the cabin of a container crane or any other administration/management office in the surroundings with enough energy supply and communication capabilities with the rest of network elements[7].

Each of the available Maersk reefer containers will include one cluster-head device and *N* data acquisition devices. Lastly, relay devices will be placed between the INTER-IoT gateway and the cluster-heads, only if it were necessary to extend the range coverage area of the LPWAN.

[5] The number *N* of data acquisition devices within the reefer container will depend on the dimensions of the reefer container, the criticity of the transported cargo and the accuracy level desired.

[6] The number M of relay devices working at 868 MHz will depend on the distances between reefer containers and the INTER-IoT gateway.

[7] The final location of the different devices used in the pilot will depend on the available port assets.

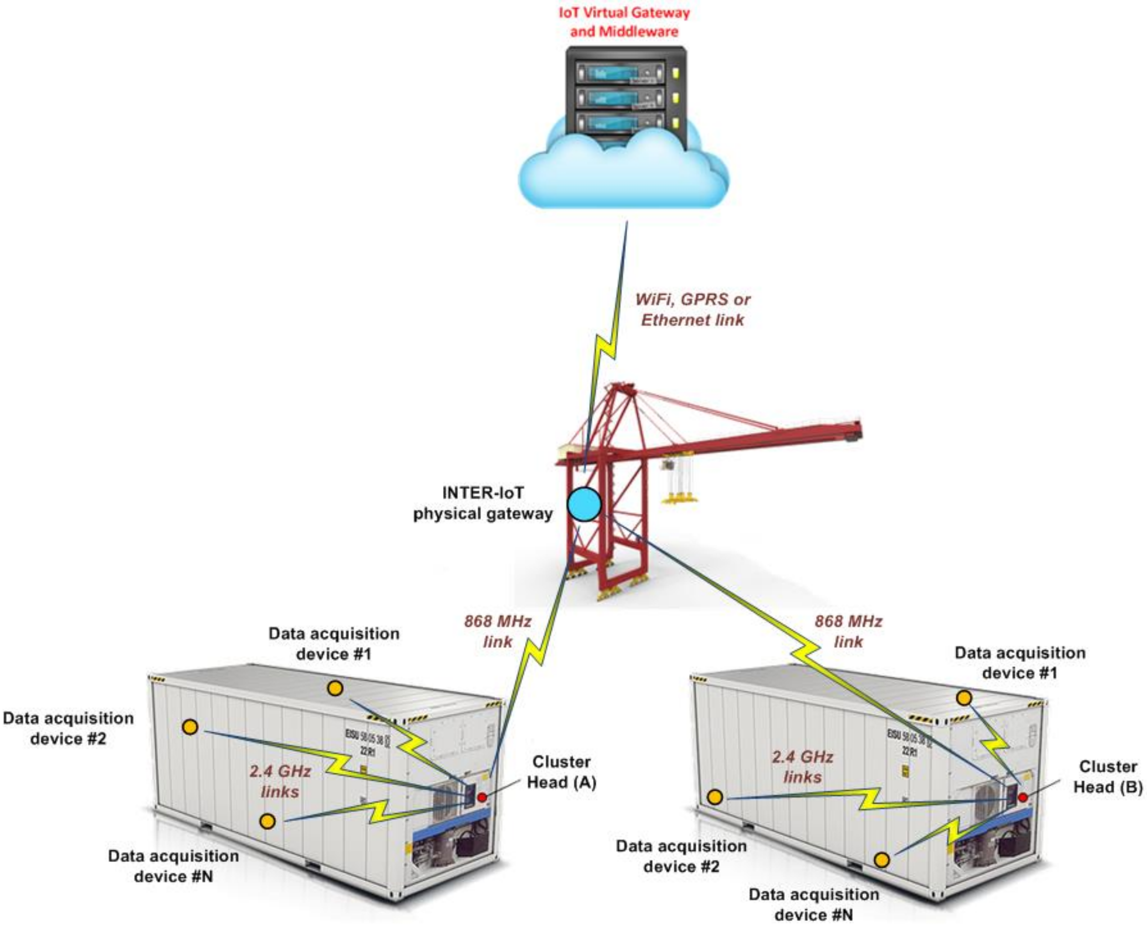

IoT Virtual Gateway and Middleware
WiFi, GPRS or Ethernet link
INTER-IoT physical gateway
868 MHz link
868 MHz link
Data acquisition device #1
Data acquisition device #2
2.4 GHz links
Cluster Head (A)
Data acquisition device #N
Data acquisition device #1
2.4 GHz links
Cluster Head (B)
Data acquisition device #2
Data acquisition device #N

Figure 58 System Integration overview

### 6.2.3.5 Evaluation

The evaluation of the proposed pilot to validate the INTER-HARE platform in the ***Monitoring reefer container*** use case will be circumscribed to the actual end-to-end communication between the data acquisition devices and the appearance of data collected by these devices in the control/management systems of the INTER-IoT project (or alternatively, in the INTER-IoT gateway).

Different metrics inherent to the system’s performance will be assessed, such as transmission reliability, data latency, energy consumption of devices and their battery lifetime, to cite some of them. Any other metric of interest for the final customer (Port of Valencia) will be computed as well.

### 6.2.4 Mission Critical operations based on IoT analytics

Compared to the test environment, the main differences in the integration environment are related to the field trials. For demonstration purposes, we target the use of real vehicles in the area of the port. In order to properly resemble the operational context of the in-field staff, special devices will be used:

- Rugged LTE smartphones for field units (first responders, port staff), running adapted GUI versions for easy-to-use operations (e.g., PTT buttons for its use with gloves).
- Rugged LTE tablet for the road haulier and for the first responder (e.g., ambulance), with additional capabilities for its use as on-board systems (e.g., adapted GUI for displaying only relevant information for the intervention or for the generation of reports).

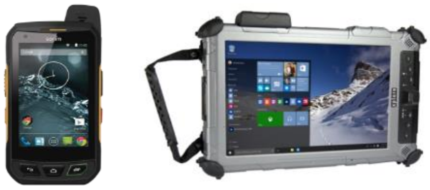


**Figure 59: Target devices for field trials.**

### 6.2.5 Interoperable Situation-Aware IoT-Based Early Warning System

The pilot will include:

- UNICAL IoT platform with Shimmer 3 ECG device
- Android mobile phone with applications for Shimmer and accelerometer data
- Haulier IoT platform (?)
- Terminal IoT platform
- Port authority IoT platform

The site acceptance tests of the project also include two parts:

1) Semantic translations and decision rules for emergency management:

Regarding the SSN x SAREF translations, we will perform user acceptance tests with the INTER-IoT focus group, to evaluate the success criteria of the alignments through the execution of functional tests and comparison with expected results based on variations of data instances (examples) in the same data structure, as quantitative validation. A questionnaire to assess the level of semantic interoperability in these tests will be used for qualitative validation.

Here we are considering that IPSM will be available for these tests, which will use mockup test data generated in the FAT.

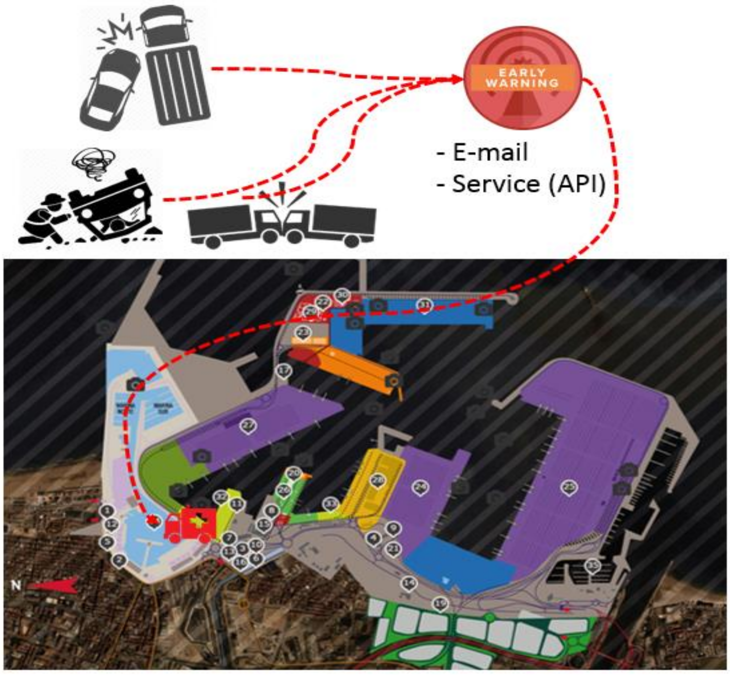
2) Situation-aware IoT EWS prototype for accidents at the port area (9):

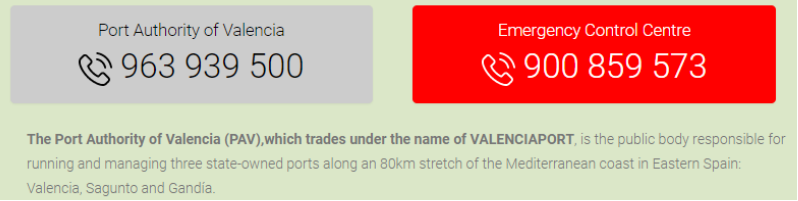
Figure 3 illustrates the EWS detecting accidents and sending messages to the CCE through e-mail and/or through a service, which CCE can subscribe (pub-sub approach). Ideally, the test setup of an EWS should include emergency simulations and emergency response exercises. Simulation of accidents can be performed in a similar way of the FAT, attaching the Shimmer sensor on the chest of a driver and a mobile phone within his/her pocket. The

EWS will be deployed at the cloud, receiving data from the IoT artefacts and sending emergency notifications for the different actors (e.g. CCE), simulating a production environment. The driver can simulate the delivery of goods at the port, for different terminals, and force accident situations. These simulation exercises can be existing or new (modified) emergency simulation exercises that are executed periodically by the CCE at the port area. However, we are still waiting the information about this possibility from the port.

The acceptance criteria of requirement 251 (from D2.3 deliverable) states that "The port IoT platform will be able to coordinate with emergency systems located in the vicinity of the port". And the acceptance criteria of requirement 249 states that "The port IoT platform, the terminal IoT platform and the haulier IoT platform need to exchange data about the trucks and containers entering in the port area. (...) IoT platform has to coordinate with emergency systems". Therefore, the evaluation needs to demonstrate that these requirements were addressed by the EWS. A list of these emergency systems was requested and still needs to be sent by the port authority, probably the ones in the vicinity are hospital systems, but may be also firefighters (civil defense) and the police. For example, in the vicinity of the port there are the Instituto Social de la Marina (Hospital Casa del Mar), Parque Central de Bomberos (firefighter) and Proteccion Civil - Valencia (civil defense). Ideally, the SAT would include a simulation of the EWS sending emergency notifications to these actors through their systems.

Regarding the port emergency control system, the D2.4 (use cases manual) states this requirement: "The haulier IoT cloud platform and the port emergency control system share security and safety information". Therefore, we need to know exactly what "security and safety information" will be exchanged with the port emergency control system (at the application services layer), waiting this information from the port.





Port Authority of Valencia
☎ 963 939 500

Emergency Control Centre
☎ 900 859 573

**The Port Authority of Valencia (PAV),which trades under the name of VALENCIAPORT**, is the public body responsible for running and managing three state-owned ports along an 80km stretch of the Mediterranean coast in Eastern Spain: Valencia, Sagunto and Gandia.

**Figure 60: EWS detecting the simulation of accidents and alerting the CCE**

In general, we are considering that all involved IoT platforms will be ready to use, enabled by the INTER-IoT consortium, as well as the required permissions for the EWS execute as intended. Mockup test data generated within FAT will support the evaluation in SAT along with real data generated by the emergency simulations. The support of the port authority is crucial for the emergency simulations setups and execution; and the support of UNICAL is crucial for the proper work of the health IoT solution.

The parts tested are the same of the FAT. The parts that will not be evaluated are the adequate response to the emergency simulations, once the focus of the EWS is the detection and warning the responsible parties to respond to emergencies (e.g. CCE).

### 6.2.6 SENSHOOK

#### 6.2.6.1 Smart Mosquito Trap

Integrated Pest Management (IPM) relies on the accuracy of the pest population monitoring technique. Without gathering information of population dynamics, and related ecological factors, it is almost impossible to execute the appropriate control at the right location and time. Mosquitoes are usually spread across large areas and boundaries, and it is inefficient to use traditional IPM, strongly dependent on human labor for efficient large scale monitoring.

To solve this, we have developed a trap station with inexpensive optical sensors that automatically detect, count and classify insects entering a trap through bioacoustic analysis, recognizing disease vectors down to the species level. Then, the collected data is transmitted wirelessly to a cloud server, where it can be automatically processed and presented in statistical and GIS (Geographical Information System) format. This data is invaluable as input to mosquito-borne epidemic models that currently receive their input data from manual counting of dispersed traps. This information can be used by public and private organizations to plan optimal intervention strategies with limited resources.

#### 6.2.6.2 Smart Mosquito Trap / Irideon integration setup

Below you can see a graphic of the mosquito trap.
The Insect counter and species identifier modules are installed in a commercial mosquito trap and will start automatically counting and identifying insects, and sending the data wirelessly. The traps can be deployed in the field alone or as a WSN-Wireless Sensor Network.
In C) you can the complete mosquito trap station, including battery, solar panel and CO2 canister.

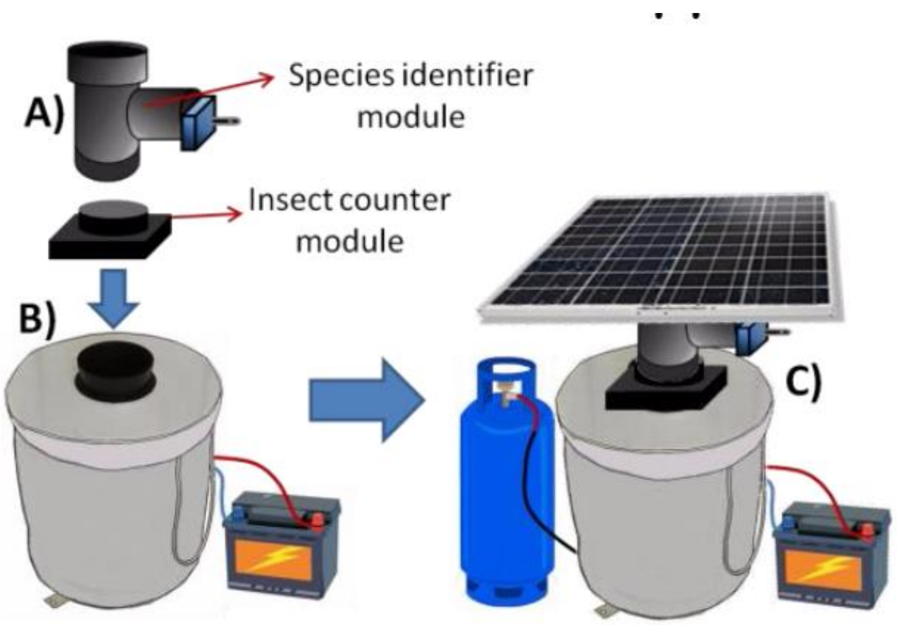


**Figure 61: Graphic of the mosquito trap**

A photo of the mounted trap with the counter and identifier module:

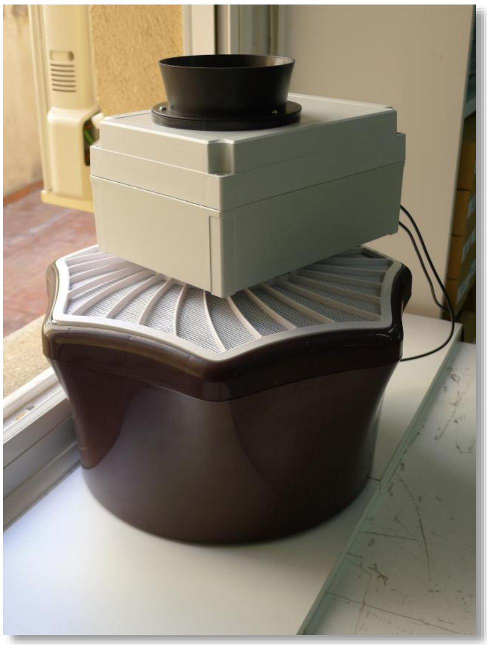


**Figure 62: Mounted trap with the counter and identifier module**

The traps are deployed in the field following the same IPM standard procedures used today. The traps form a low-power mesh network capable of wirelessly communicate between each other to finally send the collected data to the Gateway. This gateway equipped with a meteorological station, transmits the field data to the Management Software (using a link like GPRS/GSM or Wi-Fi). This software is hosted by a Cloud Server, which makes the collected processed data available on Internet. The end-users can use the on-line platform as an easy way to access the IPM tool, using just a laptop, a smart-phone or a tablet PC.

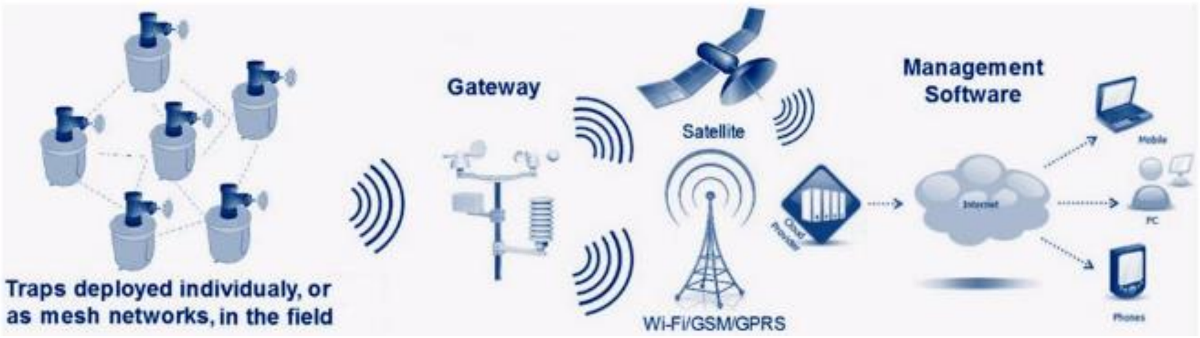


**Figure 63: System overview**

In these days the species identifier and species counter module will be tested in laboratory conditions. Preliminary tests have been done and demonstrated that the systems can detect and identify mosquitoes in free flight.

### 6.2.7 SOFOS

SOFOS experiment considers that INFOLYSiS will deploy on top of INTER-IoT vGW modules that provide SDN/NFV Automation in IoT Infrastructure, such as the INFOLYSiS SDN/NFV Network Manager. By applying appropriate OPENFLOW commands, INFOLYSiS add-on will steer the data traffic from the INTER-IoT vGW to the various VNFs that will have been deployed in order to enhance the interoperability functions of INTER-IoT, allowing to the application layer to represent the received data in a unified way.

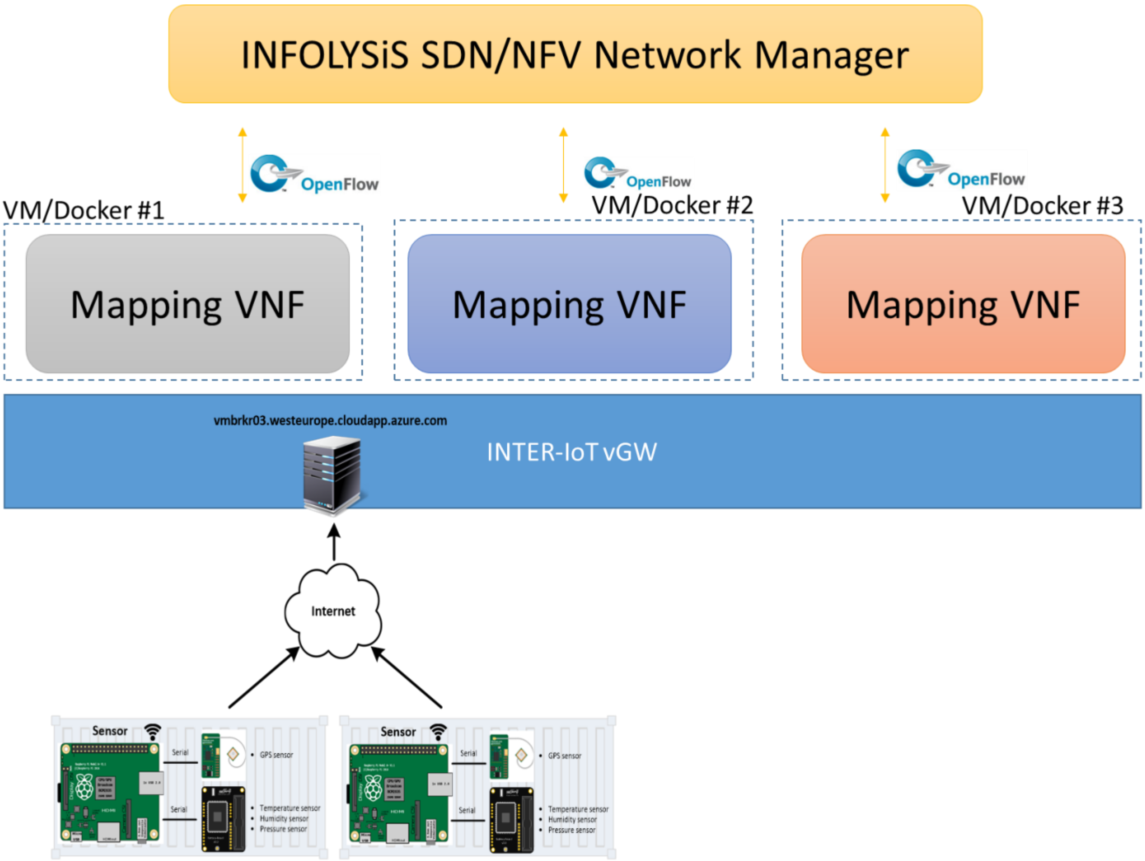


**Figure 64: SOFOS Integration and Factory test setup overview.**

Thus, with the mapping VNFs provided by INFOLYSiS and the support of the SDN/NFV techniques, the data provided by Raspberry and panStamp are mapped to a common protocol (e.g. HTTP). For the instantiation of the virtual functions, an SDN-compatible (i.e. OpenFlow compliant) OpenStack cloud computing platform is considered at the MW layer for enterprise users (in a complimentary way to Docker approach of the rest INTER-IoT framework for simple end-users), which means that at the MW Modules, the OpenStack Neutron with the ML2 plugin is considered, which is the OpenStack SDN networking project focused on delivering networking-as-a-service (NaaS) in virtual compute environments. The Neutron will interface with the OpenStack Nova (at the MW layer), which manages the lifecycle of compute instances in an OpenStack environment and therefore is fully responsible for the instantiation of the softwarised/virtualised IoT functions (i.e. VNFs).

### 6.2.8 E3Tcity Smart City Platform and Devices Integration

Key points to test in the field:

- Integration with actual lights system
- General behavior of the solution

Once the connection an correct integration of the system has been proved, it is time to test the solution in the real field. Test will be the same as in 2) but with devices connected to real lights.
Additional tests will be done to verify specific behaviors of the system once final user has defined it.

### 6.2.9 ACHILLES

Compared to FAT, the SAT modules will differentiate in the followings:

- The ACP will be implemented alongside the user management system used in each demo (interacting using e.g., LDAP).
- Clients will be real devices used in each demo
- Resources will be implemented in real devices/platfroms used in each demo (e.g., Mearks Reefer)

For the SAT the Inter-IoT gateway should be configured with the same information as in the FAT. The difference here is that resource names will refer to real (CoAP) resources implemented in devices used in each demo.

### 6.2.10 INTER-HINC

For field testing, our deployment of INTER-HINC will be configured to connect to the real IoT platforms and INTER-IoT middleware for testing. It should work like in a real scenario, except: many real operations cannot be done (e.g., stopping a crane). In these cases, we will output actions but do not execute them.

The above-mentioned description is a high-level plan for testing. In terms of technical detail, we will have concrete performance tests (e.g., how fast an action will be executed), integration tests (e.g., to see if components can be interoperable), functionality testing (to see if the business logics work as designed), etc. From the software perspective, our approach is to leverage:

- Model-based testing: so that we can generate different test cases based on models.
- Continuous integration and testing: so we can perform the integration and test based on DevOps principles to detect problems as early as possible.

In terms of testing interoperability features from the business perspective, this will be further elaborated based on the scenario and the design of the INTER-HINC, as the work in INTER-HINC is just started.

### 6.2.11 A Semantic Middleware

- Ebeacon sensors tracking the position of the pallet. These sensors can be replaced by “virtual sensors”, i.e. applications that mock and simulate the behavior of real sensors. The sensors will publish, through an UPDATE SPARQL, the position of the pallet. We have to understand if GOIoTP represents the information to model the position of the pallets or a proper domain ontology must be used.
- Working stations performing various machining operations. These stations will be replaced by “virtual applications”, i.e. applications that simulate the behavior of the real stations. The working stations will publish, through an UPDATE SPARQL, their availability status and the machining operations that can be performed. Again for these information we have to understand if GOIoTP represents the information to model them or a proper domain ontology must be used.
- Virtual carriages (simulating the real carriages) each one transporting a pallet. They consume data published by the simulator in order to follow a specific route.

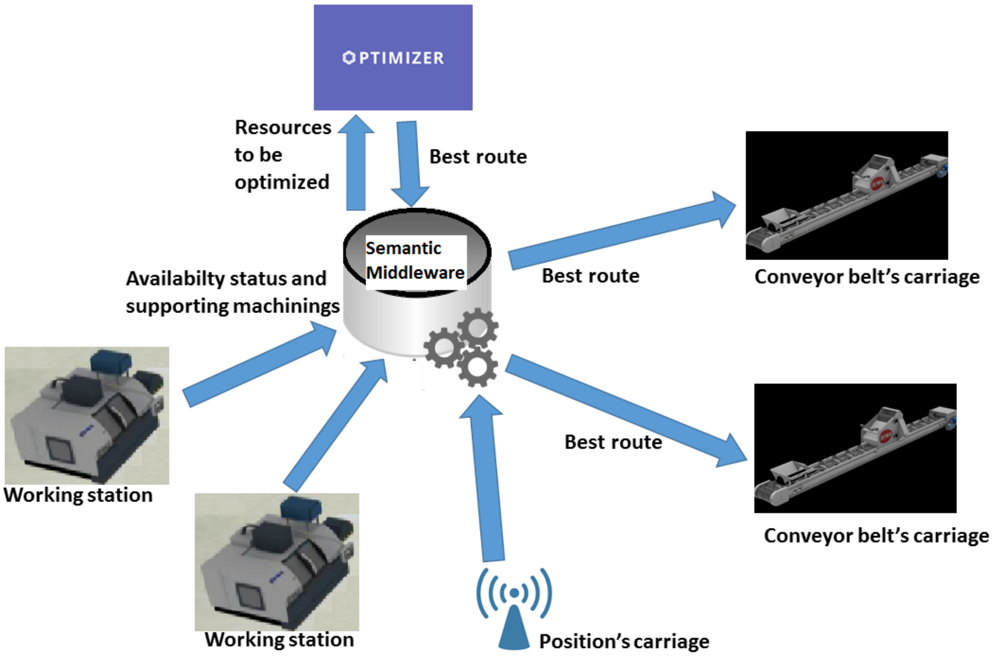


Figure 65: Overall workflow of the test case

### 6.2.12 SecurIoTy

Assuming that the DocRAID crypro proxy will be set up as a sensor gateway (compare Figure 25), we will test functional and non-functional parameters like

- availability and response times
- the gateway (i.e. controller)
- the storages
- optionally the key store, depending on the use case details

The envisioned scenario would include these steps (see Figure 66):

(1) Sensors will deliver data to sensor hubs
(2) sensor hubs will aggregate this data and send the aggregate to the DocRAID controller which is acting as a sensor gateway
(3) the DocRAID controller will fragment, encrypt and distribute the fragments for storage purposes
(4) the “big data” repository (compare Figure 25) will either pull data from the DocRAID controller or data is being pushed to the repository

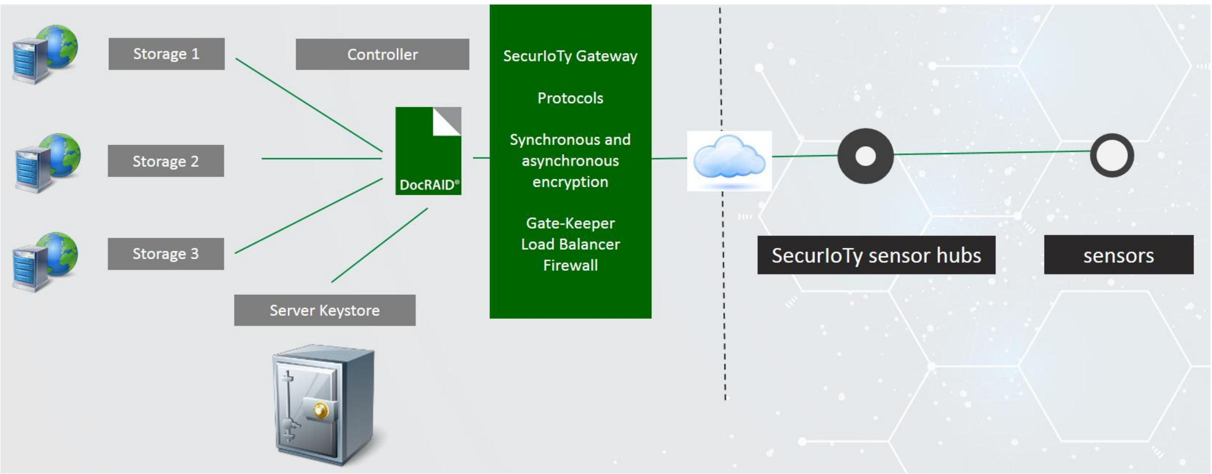


Figure 66: Integration setup

# 7 Related document deliverables

In the following table the related FAT and SAT document overview is shown:

| # | Pilot | Doc | WP | Mentor |
|---|---|---|---|---|
| 1 | INTER-LogP | FAT | D6.x | VPF |
| 2 | INTER-LogP | SAT | D6.x | VPF |
| 3 | INTER-HEALTH | FAT | D6.x | Sabien |
| 4 | INTER-HEALTH | SAT | D6.x | Sabien |
| 5 | sensiNact integration | FAT | D6.x | Prodevelop |
| 6 | sensiNact integration | SAT | D6.x | Prodevelop |
| 7 | INTER-OM2M | FAT | D6.x | Prodevelop |
| 8 | INTER-OM2M | SAT | D6.x | Prodevelop |
| 9 | INTER-HARE | FAT | D6.x | Neways |
| 10 | INTER-HARE | SAT | D6.x | Neways |
| 11 | Mission Critical operations based on IoT analytics | FAT | D6.x | VPF |
| 12 | Mission Critical operations based on IoT analytics | SAT | D6.x | VPF |
| 13 | Interoperable Situation-Aware IoT-Based Early Warning System | FAT | D6.x | VPF |
| 14 | Interoperable Situation-Aware IoT-Based Early Warning System | SAT | D6.x | VPF |
| 15 | SENSHOOK | FAT | D6.x | Neways |
| 16 | SENSHOOK | SAT | D6.x | Neways |
| 17 | SOFOS | FAT | D6.x | TU/e |
| 18 | SOFOS | SAT | D6.x | TU/e |
| 19 | E3Tcity Smart City Platform and Devices Integration | FAT | D6.x | Unical |
| 20 | E3Tcity Smart City Platform and Devices Integration | SAT | D6.x | Unical |
| 21 | ACHILLES | FAT | D6.x | XLab |
| 22 | ACHILLES | SAT | D6.x | XLab |
| 23 | INTER-HINC | FAT | D6.x | Unical |
| 24 | INTER-HINC | SAT | D6.x | Unical |
| 25 | A Semantic Middleware | FAT | D6.x | SRIPAS |
| 26 | A Semantic Middleware | SAT | D6.x | SRIPAS |
| 27 | SecurIoTy | FAT | D6.x | XLab |
| 28 | SecurIoTy | SAT | D6.x | XLab |

*Table 8: Related document deliverable*

# 8 Integration ethics and security

## 8.1 Introduction

During the Ethical Review it was requested that ethics had to be included within the project scope. This section addresses recommendations and proposed solutions to the ethical and security points of the integration. For each pilot the ethics is discussed in paragraphs 8.2 until 8.5. The security aspects of each layer is discussed in paragraph 8.7 and 8.8.

The information for the pilots for both ethics and security comes from the partners and may be included in other documents as well.

## 8.2 Pilot INTER-HEALTH

INTER-HEALTH pilot will be divided in two groups: a Control Group – CG (that will perform the Traditional Nutritional Counseling) and Experimental Group – EG (that will perform the Experimental Nutritional Counseling).

During Nutritional Counseling (First nutritional counseling and Subsequent Checks) at the Nutrition Outpatient both for the Control and the Experimental Groups will carry out detections of both, objective and subjective data that will be entered by healthcare professionals into the Professional Web Tool. Collected data will be the following:

- Objective data:
    - Personal data.
        - gender, age, address;
    - Personal data.
        - civil status, educational level, social and economic status;
    - Anthropometric data.
        - weight, height, Body Mass Index -BMI, blood pressure, waist circumference;
    - Hematochemical data
        - blood glucose, blood insulin, hemoglobin, glycated hemoglobin, cholesterol level, HDL, LDL, blood triglyceride, blood nitrogen, AST, ALT, GGT, blood creatine, urea, blood urea, blood albumin, prealbuminemia, TSH, T3, T4, erythrocyte sedimentation rate;
- Subjective data:
    - Food anamnesis.
        - breakfast, main meals, vegetables consumption, fruit consumption frequency, red and / or white meat consumption frequency, processed meat consumption frequency, egg consumption frequency, cheese consumption frequency, fish consumption frequency, legumes consumption frequency, bread and pasta and substitutes consumption frequency, dry fruit consumption frequency, oil consumption frequency, animal fats consumption frequency, salt consumption frequency, herbs and spices consumption frequency, sugar and / or honey consumption frequency, sweetening consumption frequency, sweet consumption frequency, water consumption frequency, alcohol consumption frequency, sugary drink consumption frequency;
    - Physical activity practice.
        - daily physical activity, organized physical activity;

Subjects participating in the Experimental Group will receive a medical kit containing a weight scale, sphygmomanometer and bracelet for physical activity. Furthermore they will install a mobile application for gathering medical measurements and a questionnaire on eating habits and the physical activity.

Medical kit will allow to take objective data with the following:

- **Blood pressure**, every day in the morning and evenings. Only for subjects with Normal-High pressure values (systolic pressure ≥130 and / or diastolic pressure ≥85). It is determined at the first nutritional counseling (t0) at Nutritional Outpatients;
- **Weight** every week;
- **Physical activity**, every day. Number of steps and duration of physical activity. Subjects will have as objective to achieve at least to 10.000 steps per day and 150 minutes of physical activity per week.

Subjective data will be taken through the online questionnaire biweekly.

## 8.3 Pilot INTER-LogP

In INTER-LogP pilot, sensors and devices attached to port infrastructures, cameras, machines, cranes, containers, trucks and other vehicles will provide most of the monitored data. Some scenarios in the pilot foresee also the use of wearable devices, mainly smart phones, to support logistics and transport operations as well as to enhance security and safety measures.

The recording of images and the use of other devices in the port area can potentially capture human-related data. Compliance of the ISPS code and other security measures can even need the storage of human-provided-data and even biometric data (i.e. fingerprints, face recognition parameters). The ISPS (International Ship Port Security) code is an amendment to the Safety of Life at Sea (SOLAS) Convention (1974/1988) on minimum security arrangements for ships, ports and government agencies. This code came into force in 2004 and it prescribes responsibilities to governments, shipping companies, shipboard personnel, and port/facility personnel to "detect security threats and take preventative measures against security incidents affecting ships or port facilities used in international trade”[8]. Port Authorities and port operators duly inform the people accessing to the port protected area that they are entering in a surveilled area where they can be recorded or other data about them can be captured or requested for the compliance of ISPS and security protection. The information given to the people also includes the mentions to the applicable legislations and what are their rights under these circumstances.

Although the port authority and port operators can be managing human-related-data and human-provided-data, the personal data registered by these entities will not be considered necessary for the execution of the pilot. When it will come to evaluate different scenarios where people is involved, data considered in Inter-LogP will only include the identifiers of the devices carried by the people (i.e. mobile phone number) but it will not be able to retrieve the human identity from the pilot data and the disclosure of any other human-related or human-provided data to these entities. In conclusion, the data handled by the pilot for the experimentation and evaluation will not disclose any personal data registered on the systems of the companies and authorities as they need to respect the regulations regarding personal data protection.

In the case that any scenario in the pilot introduce sensors on equipment, cranes or vehicles able to geolocate these assets and the workers using these assets, the people that will use these assets will be also duly informed about the recording of such information, the mentions

[8] ISPS Code, Part A

to applicable legislation on personal data protection and the rights these workers have for the access, rectification, cancellation and opposition in front of the capture of this data.

Additionally, in those scenarios where it is planned to use data coming from wearable devices owned by a person, it will be always requested and registered the consent of the person to use the data captured by the device. The person will also be informed about the nature of the data shared.  In some scenarios of the pilot the data shared will be the name and person identification document number, the mobile phone and the e-mail, its geographic location when they are in certain areas, the distance/time to destination or the speed. This data will not be transmitted and recorded unless the person gives its consent. This person will be able to deny the consent to use his data in the pilot at any time and existing data will be deleted. Data collected from personal wearable devices during the pilot will be anonymized after the finalization of the pilot.

## 8.4 Pilot INTER-DOMAIN:

Regarding INTER-DOMAIN, which will be the result of the integration of the open call third parties collaborations and the existing INTER-IoT pilots, no proposal coming from the Open Call will provide any data set from humans in the INTER-HEALTH pilot. The only exception is collaboration from University of Twente (Interoperable Situation-Aware IoT-Based Early Warning System) that plans to address the emergency scenario, in which data traces from potential victims of an accident will be monitored. In this case the data sets will be related with those used in INTER-HEALTH pilot and will be managed in the same way as the data sets produced in INTER-HEALTH:

- Objective data:
  - Personal data (name, surname, gender, age, address);
  - Anthropometric data  (weight, height, blood pressure)
- Subjective data:
  - Potential injuries during the emergency event
  - Activity developed by the subject at the time of the emergency
  - Medical track of the individual

However, as the emergency event will be a simulated situation, the selected subjects will provide their consent and the data will be anonymized INTER-IoT servers.

## 8.5 Ethics of each layer

For all pilots the ethics advisory board will be involved in order to avoid ethical issues within the project activities.

In the lower parts of the INTER-IoT system, data is in no way linked to a person or entity yet. Only sensor and device ID's are linked to the data. The only concern is to make sure, through the appropriate mechanisms, that these lower layers are secure enough so that they cannot be configured, utilized, or reverse-engineered in any way that could compromise the privacy and security, both digital and physical, of the users, the system itself, and any other entity within its effects.

In the MW layer, data is only identified by an ID of the gateway and device. Only once a person or entity is linked to the data it is necessary to analyze how the system treats its private information from an ethical standpoint. The layers in which this has to be analyzed are the Data&Semantics (DS2DS) layer and the application & services (AS2AS) layer. In these layers, the data is coupled to represent something. In the DS2DS layer the data is coupled to a property (i.e. weight, pressure, location, etc.) while in the AS2AS layer also the quantified data is coupled to entities or persons.

## 8.6 Security of each layer

Security of the INTER-IoT will be implemented in each layer. Even a simple GW can be attacked and taken over to work for the hacker to execute for instance DDOS attacks.

Recent attacks in IoT systems have proven that especially these simple stand-alone IoT devices are vulnerable to these attacks.

From the architectural point of view already security has been taken into account, the layered setup will make it practically impossible to take over a whole chain (top to bottom). However each separate device or function will still be vulnerable to an attack.

## 8.7 Security aspects of INTER-HEALTH

Regarding data protection strategies different actions will be provided by the different participants in the pilots. In the case of INTER-HEALTH as indicated in D2.5 will meet according to EU laws and regulations, especially Reg. 2016/679, including directive 95/46/EC. And others such a directive 93/42/CEE. Specific measures will be guaranteed mainly by UPV-SABIEN (after TIM withdrawal) and UniCal as technical partners in INTER-HEALTH pilot, but supported by the other partners developing components within the project. Specific technical measures to guarantee data protection and personal data treatment:

- The data communication protection between biomedical wireless devices and the developed mobile gateway application installed on the patient's Smartphone, will be guaranteed by proprietary mechanisms supported by device manufacturers; it is worth noting that such mechanisms will not be altered in any way by the partner;
- Data collected by the mobile gateway application will temporarily reside in the memory buffer of the patient smartphone waiting to be transmitted to the ASL TO5 server; it is worth noting that the procedure for transmitting data to the server implies the automatic delete of the local data copy stored, once the reception and the correct feedback have been made;
- During the registration of a subject into the system, (s)he will have associated a unique alphanumeric code to be identified; Objective data (name and phone number) will be stored
- The bi-directional communication between the mobile gateway application, installed on the patient's Smartphone, and the remote server located at the ASL TO5 structure, will be guaranteed by using login mechanisms and the HTTPS secure communication protocol to secure server authentication, privacy protection, and encryption and integrity of the data exchanged between the communicating parties;
- All subjects will be identified by a unique random number. A list will exist that allows the identification of the subject by means of that random number and will be archived in a secure way in a room or closet with restricted access;
- The informed consent of the subject, which bears his name and the printed signature, must be filed independently and properly in a file with restricted access;
- Personal data may be accessed by authorized personnel, but under no circumstances may they make copies of the identification list or informed consent. Also biomedical data and online questionnaires will travel in anonymous form as they will be associated only with the previously mentioned alphanumeric code but never with subject's identification data;
- Personal data will be separated from other information, which will be stored in a database management system;
- The access to the database management system will be restricted by means of login (user and password);
- Data files will be encrypted and no one will be able to open that file without the password;

- Sensible columns of the database (password) will be encrypted by using HASH and SHA2 algorithms;
- Software involved in INTER-HEALTH pilot will be to be validated according to the state of the art taking into account the principles of development lifecycle, risk management, validation and verification to avoid errors and/or malfunctions of operation.

## 8.8 Security aspects of INTER-LogP and INTER-DOMAIN

In INTER-LogP and INTER-DOMAIN, data coming from the different IoT Platforms of the Port Authority of Valencia and Noatum will not include any personal data able to identify people to respect the data protection obligations from these organizations. In the case of Noatum terminal, some data could be traced to machine operators like Terminal Truck location and speed, or crane operator joystick activity. The technical measures to ensure data protection and personal data treatment is the following:

- The assignment of operators to machines (terminal trucks, cranes, container handlers, etc) is managed by an internal application called GESTIR, which is not accessible from Noatum's IoT platform.
- The information flowing through Noatum's IoT Platform is completely anonymized in terms of personal data. No access to GESTIR is done within the IoT Platform. Device measurements are handled with the only identification of the machine name (e.g. 'RTG049'), with no attribute or reference to operators.

The access to InterLogP will be done exclusively to the IoT Platform, so no personal data will be involved.