Symbiosis of smart objects across IoT environments

688156 - symbIoTe - H2020-ICT-2015

symbIoTe Trials, Deployments & Assessment

The symbIoTe Consortium

Intracom SA Telecom Solutions, ICOM, Greece
Sveučiliste u Zagrebu Fakultet elektrotehnike i računarstva, UNIZG-FER, Croatia
AIT Austrian Institute of Technology GmbH, AIT, Austria
Nextworks Srl, NXW, Italy
Consorzio Nazionale Interuniversitario per le Telecomunicazioni, CNIT, Italy
ATOS Spain SA, ATOS, Spain
University of Vienna, Faculty of Computer Science, UNIVIE, Austria
Unidata S.p.A., UNIDATA, Italy
Sensing & Control System S.L., S&C, Spain
Fraunhofer IOSB, IOSB, Germany
Ubiwhere, Lda, UW, Portugal
VIPnet, d.o.o, VIP, Croatia
Instytut Chemii Bioorganicznej Polskiej Akademii Nauk, PSNC, Poland
NA.VI.GO. SCARL, NAVIGO, Italy
Universität Zürich, UZH, Switzerland

For more information on this document or the symbIoTe project, please contact:
Sergios Soursos, INTRACOM TELECOM, souse@intracom-telecom.com

Document Control

Title: symbIoTe Trials, Deployments & Assessment

Type: Public

Editor(s): João Garcia

E-mail: jmgarcia@ubiwhere.com

Author(s): João Garcia (UW), Matteo Pardi (NXW), Raquel Ventura Miravet (S&C), Luca De Santis (NAVIGO), Lorenzo Neri (NXW), Reinhard Herzog (IOSB), Michael Jacoby (IOSB), Szymon Mueller (PSNC), Karl Kreiner (AIT), Aleksandar Antonić (UNIZG-FER), Martina Marjanović (UNIZG-FER), Svenja Schroeder (UNIVIE), Mario Drobics (AIT), Antonio Paradell (WLI)

Doc ID: D5.6-v1.0.doc

Amendment History

Version	Date	Author	Description/Comments
v0.1	12/11/2018	João Garcia (UW), Matteo Pardi (NXW)	ToC
v0.2	30/11/2018	Raquel Ventura Miravet (S&C), Luca De Santis (NAVIGO), Lorenzo Neri (NXW)	Smart Indoor, Smart Home and Smart Yatching Use cases
v0.3	07/12/2018	João Garcia (UW), Reinhard Herzog (IOSB), Michael Jacoby (IOSB)	Executive Summary, Introduction, EduCampus
v0.4	11/12/2018	Szymon Mueller (PSNC), Raquel Ventura Miravet (S&C), Lorenzo Neri (NXW), Karl Kreiner (AIT), Luca De Santis (NAVIGO), João Garcia (UW), Aleksandar Antonić (UNIZG-FER), Martina Marjanović (UNIZG-FER), Svenja Schroeder (UNIVIE), Mario Drobics (AIT)	symbIoTe Trials Preparation, Smart Indoor updates, Smart Home updates, Smart Mirror, Smart Yatching updates, SMEUR
V0.5	12/12/2018	Antonio Paradell (WLI), João Garcia (UW), Aleksandar Antonić (UNIZG-FER), Michael Jacoby (IOSB), Reinhard Herzog (IOSB)	Smart Stadium, SMEUR updates, Introduction, Conclusion, SMEUR Test Cards, EduCampus
V0.6	24/12/2018	João Garcia (UW), Mario Drobics (AIT), Roman Lapacz (PSNC), Karl Kreiner (AIT), Ivana Podnar Zarko (UNIZG-FER)	Document review and compilation
V1.0	30/12/2018	Roman Lapacz (PSNC), Antonio Paradell (WLI)	Updates to trials preparation and Smart Stadium

Table of Contents

1 Executive Summary

The symbIoTe middleware has been designed in order to accommodate various types of use cases revolving around IoT, where multiple platforms need to interoperate to offer valuable services to users. As such, to validate the developed system, the various use cases involved in the project planned and performed trials with real users in real locations. The assessment of these trials allows for the evaluation of the symbIoTe middleware as a whole, and its specific use cases.

The current document, entitled “symbIoTe Trials, Deployments & Assessment”, aims at giving an overview on how the various trials were designed and evaluated and explains the deployment of the symbIoTe components needed for the various different trials, as well as the specific components used in each use case, and assess the results of the performed trials.

The document starts by describing how the core components were deployed and tuned to handle the various trials to be performed. A special synchronization process was deployed after some issues regarding desynchronization between Core and Cloud components were identified. The biggest problems that occurred during the trials was a concurrency problem that occurred during the Smart Stadium trial due to reregistration of resources, but that was solved and did not affect future trials.

Each use case designed one or more test cards in order to define goals, both technical and non-technical, to be achieved during the trials. The results allowed a relevant assessment of the trials.

The *Smart Residence* use case involved three different trials within the same domain. The Smart Home use case showed how a user can access his/her home devices, operated by different platforms, under the mediation of the symbIoTe Smart Space Middleware. It was possible to see the good responsiveness of the system and how symbIoTe has a general good performance when a large set of resources are registered in the environment. The Smart Indoor Air Quality trial was able to test the integration of the platform with symbIoTe, although a bigger trial would be needed to prove that the developed application improved the users’ quality of life. The Smart Mirror trials were successful, being considered good for real-life deployment, although it is considered that there is room to improve symbIoTe’s response time in some cases.

The *Smart Mobility* use case was able to reach a large number of participants (more than 200), obtaining good user feedback, and where most of the issues were solved and generally unrelated to symbIoTe Core system (e.g. firewall problems impeding communication with the enabler).

Due to some problems with the external platforms involved, the *EduCampus* use case was not evaluated through a trial but rather extensive tests of the expected major outcome of the trial, the semantic mapping, were preformed and acceptable efficiency results were obtained.

The *Smart Stadium* trial suffered due to some efficiency problems from the earlier symbIoTe deployment, but it was able to adapt in order to provide quality of service to the users, who found the experience enjoyable.

The results of the *Smart Yachting* trials were positive, with good user feedback, with hopes of being to involve a more significant number of users in the future.

All in all, the symbIoTe solution was extensively tested under different types of use cases, with different requirements, with real-world settings, involving real end users and real-life environments. The obtained results demonstrate the usefulness of symbIoTe in the aforementioned use cases. The involvement of commercial IoT platforms in some cases, or close-to-market IoT prototypes in other cases, further support the technical achievements of the project, since the developed Open Source middleware was able to interact and facilitate all the different types of involved platforms.

2 Introduction

This section aims at providing information regarding this document, which encompasses all the work done in tasks T5.4 and T5.5.

2.1 Purpose of the Document and Scope

Being the final document of WP5, this document concludes the work done in regards to the various use cases. It mainly focuses on the effort performed for tasks T5.4 and T5.5, although they are dependent on previous WP5 tasks. As such, the document aims at finalizing the use cases, more specifically, their trials planning, deployments and corresponding assessment.

2.2 Document Structure

Section 3 describes the preparations done to the Core components of symbIoTe in order to be able to handle the demand of the different use cases. Section 4 includes the description of all the uses cases, their tests and questionnaires, deployment details, defined KPIs, their issues and risks, the results of functional tests and trials, and their conclusions. Finally, Section 5 concludes the document.

3 symbIoTe Trial Preparation

3.1 symbIoTe middleware

The implementation focus in the months preceding trials was put to prepare and facilitate functionalities required by the trials and use cases. The result of this work has been release *version 2.0.0* of the symbIoTe middleware[1]. It was the first public release that offered L2-compliance software modules of the symbIoTe Cloud as well as support for L3- and L4-compliance usage in symbIoTe SmartSpace environment. It also provided bug fixes and other improvements to the existing L1 components.

Moreover, middleware documentation with a detailed description of the integration process with symbIoTe middleware as well as examples has been provided in the form of github wiki pages:

- symbIoTe Cloud:
  - https://github.com/symbiote-h2020/SymbioteCloud/wiki
- symbIoTe Enabler:
  - https://github.com/symbiote-h2020/SymbioteEnabler/wiki
- symbIoTe Smart Space:
  - https://github.com/symbiote-h2020/SymbioteSmartSpace/wiki

3.2 Deployment of Core services

3.2.1 Technical specification

symbIoTe Core services are hosted on a virtual machine running on PSNC's Open Stack cluster. It uses 8 virtual CPUs (Intel Xeon E312xx series), 16GB of RAM and 160GB of HDD storage. The machine is running the Linux operating system (Ubuntu 16.04 LTS). Each Core Service component runs within a separate instance of the Apache Tomcat servlet container.

3.2.2 Deployment and configuration

Version 2.0.0 of symbIoTe Core components have been deployed on the above machine and configured to provide a central access point for all trial participants to register their platforms, services and resources.

The two external interfaces are available:

- Core Interface, which provides security, search and access request functionalities https://www.symbiote-open.man.poznan.pl/coreInterface

[1] https://www.symbiote-h2020.eu/blog/2018/05/24/new-symbiote-middleware-release-v-2-0-0/

- Cloud Core Interface, which provides platform and resource management functionalities, including resource registration, modification and deletion, passing monitoring information etc.

  https://symbiote-open.man.poznan.pl/coreInterface/

3.3 Feedback from trials

3.3.1 Initial feedback

The initial feedback from the trial participants was positive with minor issues concerning the integration with the symbIoTe framework, connectivity (setting up interworking interface service properly to connect Core and Cloud layers, proper registration of the platform) and security (security configuration, setting up certificate) issues. Also some bugs have been reported using the established communication channels: Jira, slack and weekly status calls with developers. One of the problems that occurred was an issue with desynchronization between the core and cloud. Sometimes due to communication problems (connection issues, timeouts), there were inconsistencies between the status of resources on the platform side (Cloud) and on the central side (Core). To solve this problem, a special synchronization method has been introduced that allows a platform side to update its status based on the status persisted on the core side.

3.3.2 Concurrency issues

The biggest issue that occurred during the initial run of the trial was a concurrency problem that appeared during the trial in June 2018 of the Smart Stadium use case. During the tests preceding the trial, no issues were reported. However, during the trial the high number of requests (especially resource update requests) that have been generated resulted in the Core services to be overloaded and in most of the requests to fail due to timeouts. At peak times Smart Stadium trial was generating over 15 update requests and over 20 search requests per minute. Live assistance has been provided in clearing the message queues and restarting the failing services. Afterwards the logs and statistics have been analysed and issues described below have been found. A plan for improving the performance have been created and implemented in the following weeks, as can be seen in section 3.4.

3.4 Performance improvements

Investigation of the issues that occurred during the Smart Stadium trial resulted in finding out several problems that combined resulted in poor responsiveness of the Core services.

Problem description	Status	Solution
Resource registration (and update) request executed a remote sparql query (from Semantic Manager to Search) to retrieve a potentially existing location definition in the Jena triplestore used in the Search component.	Fixed	A caching mechanism has been introduced to the Semantic Manager component, which holds the information about the locations registered by the platforms and their resources in the repository which can be accessed directly.

Security checks were done per resource and not per result set which resulted in repeated validation of credential requests sent to the Authentication and Authorization Manager component.	Fixed	New methods have been introduced to the SymbIoTeSecurity library, which allows the component using the library to store partial map of validated credentials locally and to pass it to the validation method. This way no extra messages are exchanged between Search and AAM components above the necessary ones.
Sub-query used for observed property checks during the search operation was executed on secured graph (with low performance).	Fixed	Replacement of secure with unsecured graph for the sub-query, since security checks are done on the resource level and are not needed on the reading property level.
Concurrent query support on the Jena triplestore have not been properly configured and implemented.	Fixed	Apache Jena provides multiple-reader/single-writer concurrency support (MRSW). Concurrent access to stored model has been introduced and implemented in the Search component, allowing thread pool with configurable size to serve multiple read requests. However, due to MRSW, in situations where a use cases requires many write access operations on their resources additional solutions are recommended such as load balancing of multiple Search instances.

4 Use Cases

Each Use Case performs one or more trials in order to test the integrated symbIoTe system and use-case specific components on a technical level, as well as the developed applications from an end-user point of view. In this section for each use case the actual deployment, the KPIs being defined, the observed issues and risks, as well as the actual testing and received outcomes are described in detail.

The various Use Cases' trials involved a wide scope of different type of activities, from software and system integration of the the symbIoTe components, testing and validation, to the actual performance of the trials and the assessment of the results.

As such, a validation procedure based on test cards is followed. From a developer's perspective, test cards simplify the system integration activities, identifying small functionalities to be tested independently and progressively merged together to validate the overall workflow and the expected behaviour of the entire system. From a platform owner point of view, test cards allow the verification of the configuration of the different components of the trials and the correct interaction of the symbIoTe middleware with the IoT platforms, so that the expected functionalities and services can be provided as expected. Finally, test cards can also provide guidelines to verify the effectiveness of the audience interaction with the system through the interface proposed for each trial.

4.1 Smart Residence

The Smart Residence use case aims to demonstrate interoperability across different smart home IoT solutions: symbIoTe is used to dynamically combine different services and sensors and actuators so as to provide a natural and homogeneous user experience for smart homes across platforms.

Three applications are implemented to offer comfort, automation, security, energy efficiency and healthcare services. All of them use advanced and ubiquitous technologies including sensors and other devices integrated in the residential infrastructure.

4.1.1 Smart Home (Pisa)

The Smart Home scenario wants to showcase how to control IoT devices located in a physical space. With that goal in mind, two applications have been developed: Smart Area Controller and Home Comfort.

Smart Area Controller

With Smart Area Controller, users interact with the resources of the Smart Space (SSP) with an application installed in their smartphones. The application connects to the Smart Space and gathers data from resources through an Enabler, that is able to search and filter resources on the Core based on their location.

Home Comfort

Home Comfort scenario aims to showcase a situation where devices are automatically controlled in order to keep user's predefined comfort values. A user pre-configures the desired targets in the application, and the system will automatic control of the

environment. For instance, lux levels have been pre-set to a desired value: the system acts on dimmer lights and on automated curtains to control luminance. Moreover, it also sets the fan coils to reach and maintain the requested temperature.

From a symbIoTe viewpoint both use cases are compliant to L3 level: homes and buildings are seen as Smart Spaces (running the symbIoTe Smart Space Middleware, S3M) and thus home resources can be registered from different Platforms or can be Smart Devices.

4.1.1.1 Definition of the Test cards and Questionnaires

The procedure followed to evaluate Smart Area Controller and Home Comfort applications is articulated in three phases described in three different main Test Cards:

Platform Registration

This test card illustrates the procedure to evaluate the correct registration of the SSP resources and locations provided by two separated symbIoTe L3 compliant IoT platforms.

The first one organizes its devices in locations modelled as Rooms, the second one in locations modelled as Floors. The user should then be able to define the topology of the environment and the consequential hierarchy between locations through the Enabler.

The test verifies the correct operation of the basic registration process and the feasibility of interoperability between resources registered by different platforms. One further objective is to verify the capability of the symbIoTe Enabler of correctly managing the topology.

This test is run by a unique tester, producing a single test card, since the registration procedure needs to be run once, and users are not intended to face it.

Smart Area Controller

This card evaluates the correct functioning of Smart Area Controller App developed for Android devices to control all the devices registered in the SSP. It should be possible for a user to navigate through the hierarchical tree of locations to reach any registered device, while remaining unaware of a platform that registered it.

The application leverages on a specific enabler, which gives the possibility to filter the devices in the space, based on their position (building, floor, room, etc.). In this way, the application has to query the symbIoTe enabler in order to retrieve the list of smart devices (sensors and actuators) in a selected area and then to allow a user to control them.

The tests have been run by 10 end users (Nextworks employees) producing functional evaluation Test Cards. The 10 resulting Test Cards has been summarized in a general one. After completing the functional tests, the users were asked to complete a questionnaire about usability and appreciation of the application.

Home Comfort

This card evaluates the correct functioning of Home Comfort App, consisting of a Back-end, running on a server (that monitors the sensors and activates the actuators) and a Front-end (accessible via a web interface for creation of scenarios and configuration).

The tests verify the interoperability between resources registered by different platforms and their capability to cooperate in a scenario.

With Home Comfort App, a user interacts with any resource and navigates through locations being completely unaware of the platform that registered it.

Similarly to the Smart Area Controller App, the tests have been run by 10 end users (Nextworks employees) producing a functional evaluation Test Cards. The ten resulting Test Cards have been summarized in a general one. After completing the functional tests, the users were asked to complete a questionnaire regarding the usability and appreciation of the application.

4.1.1.2 Deployment

The tests have been executed within the Nextworks' offices in Pisa. The location does not need a special preparation since the sensors and actuators are already installed. Nevertheless, the commercial IoT platform from Nextworks, called Symphony, was made symbIoTe compliant, in order for the smart devices to be exposed.

The devices involved in the trial are sensors and actuators controlled by two instances of the same IoT platform, i.e., Symphony, and, as specified in the previous paragraph, they were organized in locations modelled as Floors and Rooms.

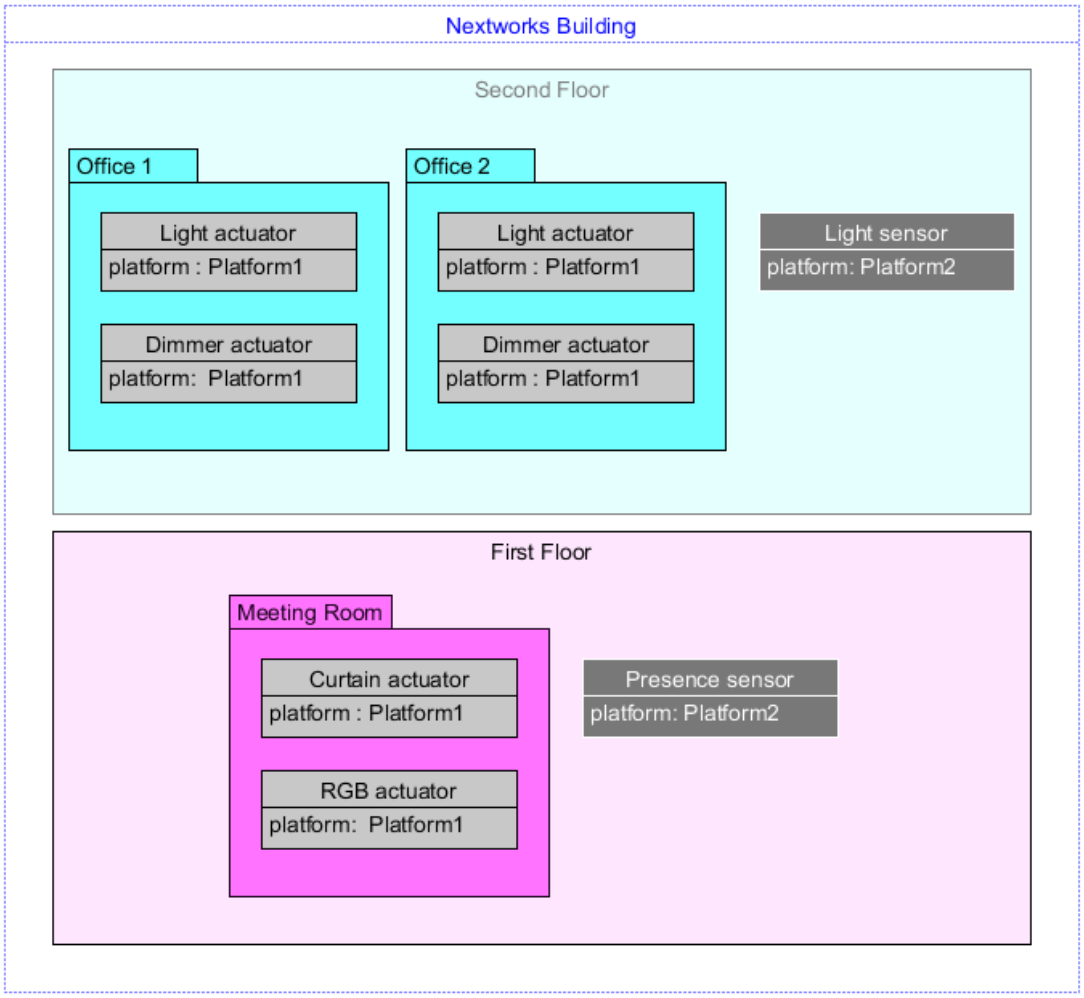


Figure 1: Topology

From the architectural point of view the deployment of the system is shown in Figure 2:

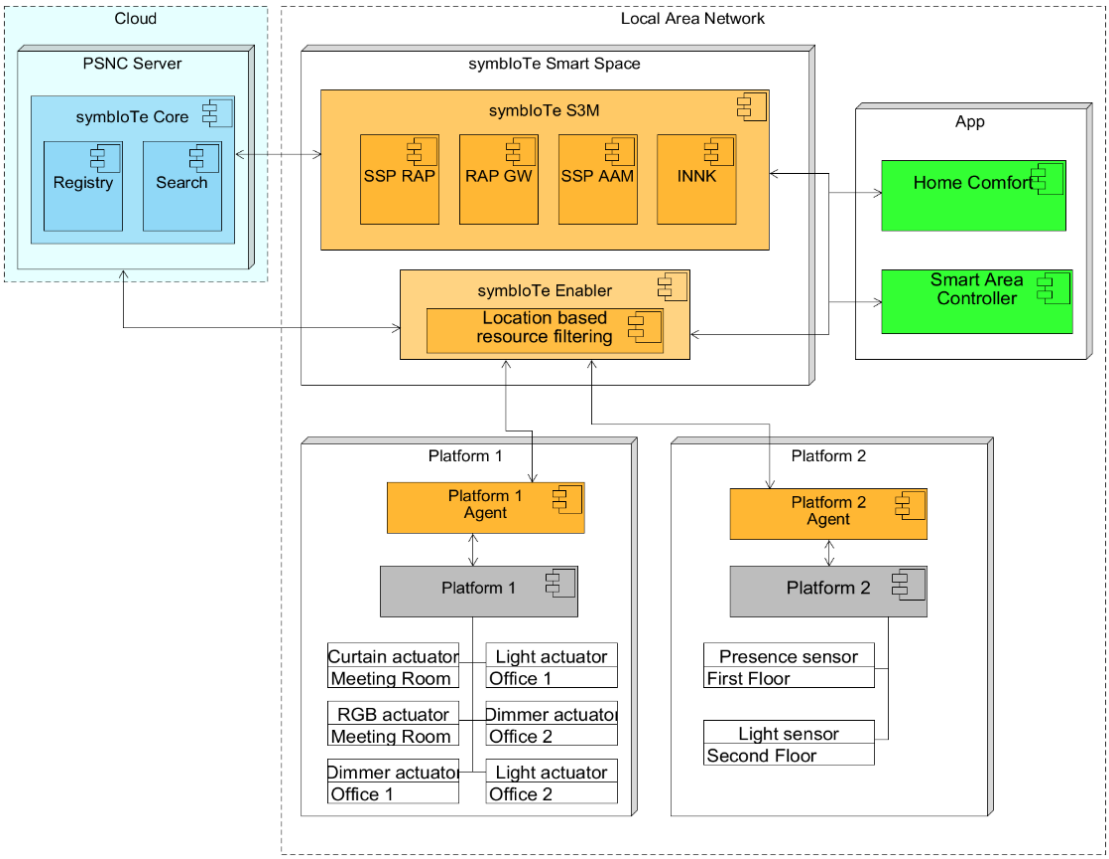


Figure 2: Architecture

4.1.1.3 KPIs

KPIs have been set for both the Home Comfort and Smart Area Controller app.

The procedure shown in **Platform Registration** refers to the preliminary work done to have Home Comfort and Smart Area Controller running.

Application	No. of Users	Description	Target Value
Smart Area Controller	10	Percentage of showed devices (in the right location);	
		Percentage of accessible/controllable devices (depending on the device);	>80%
		Percentage of locations showed, correctly positioned in Hierarchical tree and navigable.	
Home Comfort		Percentage of showed devices (in the right location);	

		Percentage of locations showed, correctly positioned in Hierarchical tree and navigable;	>80%
		Percentage of availability of devices in creating a scenario;	
		Percentage of respect of scenarios rules.	

4.1.1.4 Issues & Risks

It was necessary for this use case to manage all the symbolic locations defined into the SSP and to classify them hierarchically to allow the enabler to understand the actual SSP's topology.

This operation is provided by a manual configuration made by the SSP administrator that must be able to rearrange all the registered areas to specify whether one *contains* or *is contained* by another (e.g. floors in buildings, rooms in floors, etc.).

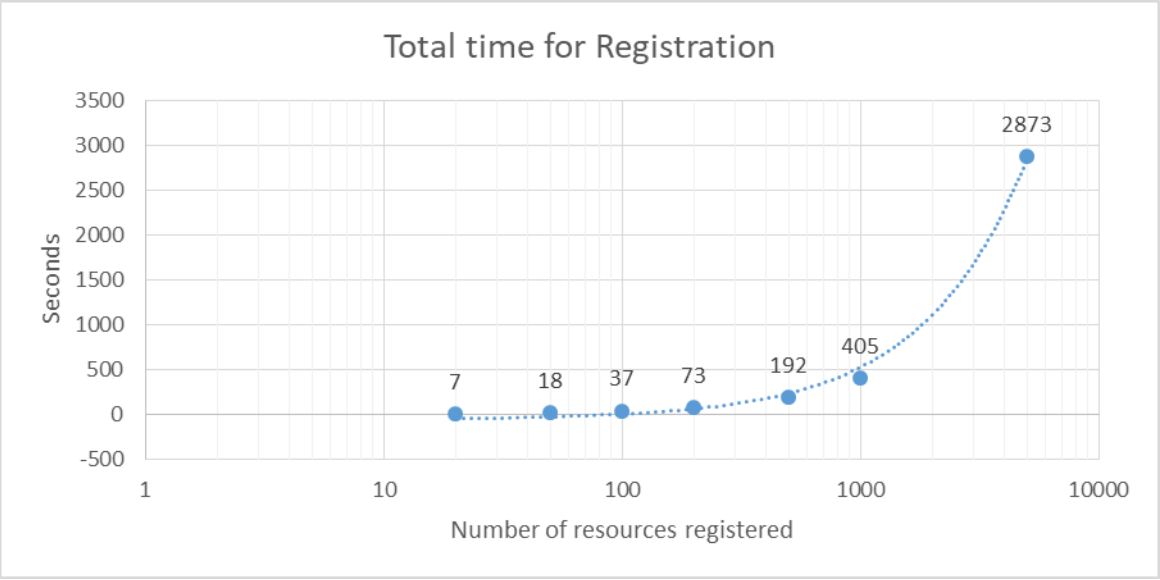
4.1.1.5 Functional tests – Status and Results

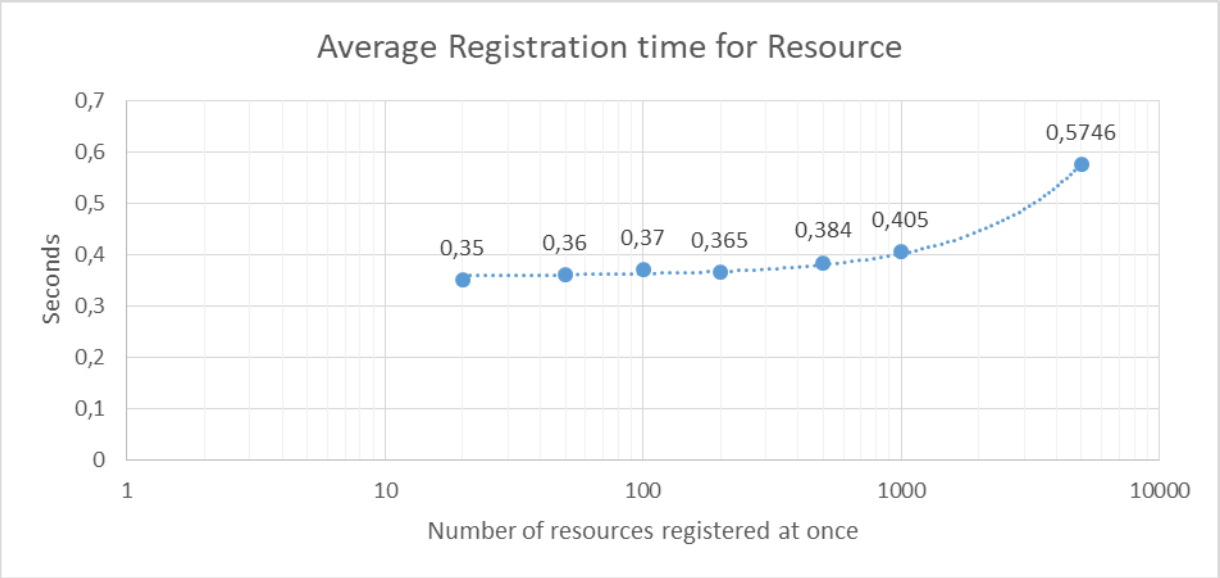
With the **Platform Registration**, a test is done to verify the correct functionality of the basic registration process and the possibility of resources registered by separated platforms to interoperate in the same SSP. It is also verified the capability of the symbIoTe Enabler of correctly managing the topology.

Test Description	Status	Results
Registration of the first Platform (Symphony):	Passed	No errors during registration process in symbIoTe Agent logs
Inspect registered resources and locations on SSP from the first Platform (Symphony)	Passed	All Platform 1 resources and locations are correctly registered, searchable and their information accessible:
Inspect registered resources and locations on SymbIoTe Core	Passed	All Platform 1 resources and locations are correctly registered, searchable and their information accessible
Registration of the second Platform (Symphony):	Passed	No errors during registration process in symbIoTe Agent logs
Inspect registered resources and locations on SSP from the second Platform (Symphony)	Passed	All Platform 2 resources and locations are correctly registered, searchable and their information accessible:

Inspect registered resources and locations on symbIoTe Core	Passed	All Platform 2 resources and locations are correctly registered, searchable and their information accessible
Verify the SSP registered locations on the Enabler: All the locations registered by both of the platforms should be displayed in the Enabler interface. The enabler is completely agnostic about which Platform has registered a specific location.	Passed	All the locations registered by both of the platforms are displayed in the Enabler interface.
Check of the SSP registered locations hierarchy: The Enabler provides a GUI that allow the user to create a hierarchical tree between the locations registered by both of the platforms. This tree must be navigable.	Passed	After the hierarchy has been created it is tested to be actually saved in the SSP database.

A Stress Test has also been run over the system, registering a growing number of resources within the SSP.





It is possible to see that the number of resources do affect the average registration time of devices but it still remains acceptable (about 0,5 s) even when up to 5000 registrations are carried out at the same time.

4.1.1.6 Execution of the trials with users – Status and Results

For Smart Area Controller

Test Description	Status	Results
Navigation through Locations: Verify that every location that appears in the topology is displayed and selectable in Smart Area Controller APP (for each correctly displayed location one point has been given by each operator).	Passed	50/50
Navigation through Locations, hierarchy verification: Verify that the hierarchical order of locations shown in the topology is reflected into the Smart Area Controller APP interface (for each location correctly collocated in the hierarchical tree one point has been given by each operator).	Passed	50/50
Navigation through Locations up to resources: Every resource in the Smart Area Controller APP must be accessible and correctly reachable through any locations it is registered in (for any correctly displayed and collocated resource one point has been given by each operator).	Passed	80/80

Navigation through Locations, resource functioning: Verify that every resource that appears in the topology is readable (sensors) or writable (actuators) in Smart Area Controller APP (for each correctly behaving resource one point has been given by each operator).	Passed	80/80

For Home Comfort

Test Description	Status	Results
Navigation through Locations: Verify that every location that appears in the topology is displayed and selectable in Smart Area Controller APP (for each correctly displayed location one point has been given by each operator).	Passed	50/50
Navigation through Locations, hierarchy verification: Verify that the hierarchical order of locations shown in the topology is reflected into the Smart Area Controller APP interface (for each location correctly collocated in the hierarchical tree one point has been given by each operator).	Passed	50/50
Navigation through Locations up to resources: Every resource in the Smart Area Controller APP must be accessible and correctly reachable through any locations it is registered in (for any correctly displayed and collocated resource one point has been given by each operator).	Passed	80/80
Scenarios creation: At the moment of a new scenario creations rules must be empty and every possible condition/action combination (depending only on sensors and actuators in the location) must be selectable. (Each operator defines an Example Scenario setting conditions on any available sensor and actions on any available actuator, one point is given for any correctly set scenario).	Passed	10/10
Scenarios rules check: Once scenario's rules are set their correct execution must be checked creating the desired conditions on sensors and verifying the correct reactions of actuators	Passed	10/10

(one point is given for any correctly reacting scenario).		

4.1.1.7 Conclusions

The Smart Home trial aimed to showcase how a user can access with a simple application to all his/her home devices operated by different platforms when they are co-located in the same space, under the mediation of the symbIoTe Smart Space Middleware.

The primary goal of the trial was to evaluate the stability of the system, testing the control-loop (monitoring/actuation/monitoring) whether it works well, actually following the user configurations. The second goal was to evaluate the usability of the system, with a view to end-user concrete benefits.

In conclusion, the results of the tests highlighted good system responsiveness: the behavior was the one expected, without any significant error or missing feature. Although the limited number of users involved in the trial didn't allow relevant conclusions with respect to performances under stress conditions, the evaluation has also involved several tests to show what happens when a large set of resources is registered in the environment. In this case the performances have demonstrated a general good quality of the framework as well.

Apart from the aforementioned trial results, it is worth mentioning that the Smart Home scenario was further tested and demonstrated by involving two additional platforms, i.e., the OpenHAB platform provided by UNIZG-FER as well as the OM2M-based solution from Sensinov, a winner from symbIoTe's second Open Call. The former case was tested between the NXW and UNIZG-FER partners and provided to the hackathon participants during the IoT Solutions World Congress 2018. The latter case, i.e., the interoperability achieved via symbIoTe between the Symphony and the Sensinov platforms, was demonstrated at the ETSI IoT Week 2018.

4.1.2 Smart Healthy Indoor Air

This application is based on the indoor/outdoor air quality monitoring and pursues to improve indoor air quality (IAQ). The IAQ refers to the quality of the air inside buildings as represented by concentrations of pollutants and thermal (temperature and relative humidity) conditions that affect the health, comfort and performance of occupants. Exposure to indoor air pollution has been linked to the development of different diseases from infections to asthma or to poor sleep. It can also cause less serious side effects such as headaches, dry eyes and nasal congestion[2]. The Smart Healthy Indoor Air application monitors the IAQ and outdoor pollution and provides recommendations about how and when to ventilate the house. Given the home location through GPS coordinates, the interpolation function provided by symbIoTe gives more precise values of outdoor air quality.

[2] Quantifying the Performance of Natural Ventilation Windcatchers. Jones, B; (2010) Quantifying the Performance of Natural Ventilation Windcatchers. Doctoral thesis , Brunel University

4.1.2.1 Definition of the Test cards and Questionnaires

This section lists the instruments and procedures used to measure the performance and usability variables to evaluate the system. We have defined two different types of test cards:

- Symbiote-testcard-SHIA-Application: The aim of this evaluation is to verify the correct functioning of the user application.
- Symbiote-testcard-SHIA-Integration: The aim of this evaluation is to verify the correct registration of all the indoor and outdoor resources and locations provided by the nAssist platform.

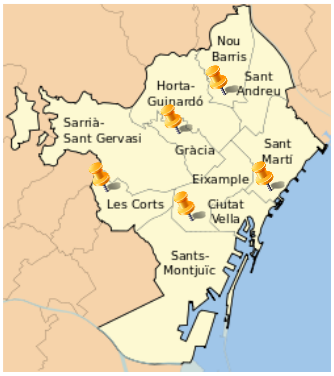
After pilots, users were asked to complete two types of questionnaires. The questionnaires were paper-based that were delivered to the users after trials:

- The System Usability Scale (SUS) is an easy to use, efficient and well-tested scale to assess the overall usability of a system. It features 10 items which can be ranked on a Likert scale of 1-5 or 1-10. According to the outcomes an overall SUS score between 0 and 100 can be calculated. The scale features the following items (whereas “system” can be substituted by the name of the technology/app/program to be assessed):
    - I think that I would like to use this system frequently.
    - I found the system unnecessarily complex.
    - I thought the system was easy to use.
    - I think that I would need the support of a technical person to be able to use this system.
    - I found the various functions in this system were well integrated.
    - I thought there was too much inconsistency in this system.
    - I would imagine that most people would learn to use this system very quickly.
    - I found the system very cumbersome to use.
    - I felt very confident using the system.
    - I needed to learn a lot of things before I could get going with this system.
- Quality of Life Questionnaire in Indoor Air Quality: This questionnaire measures the performance of the system in terms of improving participants’ quality of life by measuring the reduction of the most common symptoms of indoor air pollution, such as coughing, sneezing, watery eyes, fatigue, dizziness, headaches, upper respiratory congestion. This questionnaire is based on ten dichotomous questions asked for a Yes/No. It is focused on the participants’ experiences, feelings, beliefs, perceptions and convictions concerning their health-related quality of life during the pilots.

4.1.2.2 Deployment

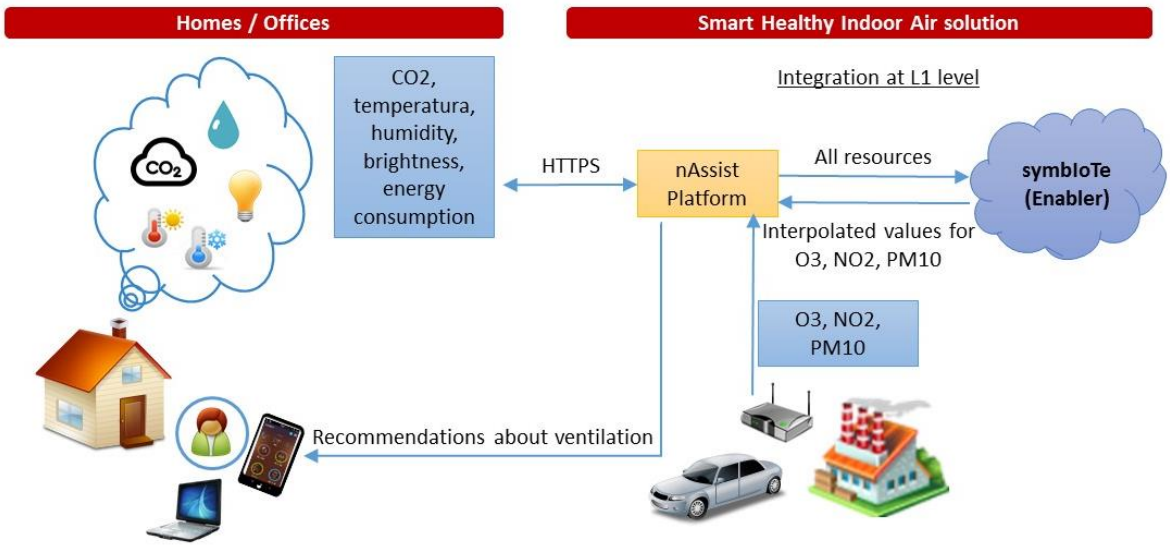
Five different locations inside Barcelona city were evaluated. The indoor equipment available for each location include the following products:

- 1 gateway for communication
- 1 multi-sensor 4x1 (motion, temperature, humidity, and luminosity)
- 1 CO2 sensor
- 1 Electricity energy consumption meter
- 2 smart energy plugs (electricity measuring and on/off for towel heater and TV)



In addition, the interpolation function from the Enabler provides interpolated values for O3, NO2 and PM10 for a specific house location given the GPS coordinates of the fixed stations. The platform registers a set of sensors (O3, NO2, PM10) located in seven different outdoor fixed stations in Barcelona: Ciutadella, Eixample, Gracia, Palau Reial, PobleNou, Sants, and Vall Hebron. The next diagram shows the overall architecture deployed by each house.

Participants were recruited among those who were familiar with the enControl system and live in different areas of Barcelona city. We evaluated the same five subjects from April to November 2018. Since the five subjects are familiar with the system, it was not necessary to organize a training session to explain them the system to them. We asked them to carry out their normal routines.



4.1.2.3 KPIs

We have defined KPIs to measure the functionality of the application from the user point of view and from a technical point of view:

Type of KPIs	Description	Target Value
Integration	Seven resources registered by each indoor installation (five houses)	35 resources registered
	Three resources registered by seven outdoor fixed stations	21 resources registered
	Interpolated values for each installation (O3, NO2 and PM10) being updated hourly	Values from 15 resources being updated hourly from enabler
	Generation of recommended ventilation strategies when indoor air quality is low	Creation of messages by the notification system. The rule engine detects that the indoor air quality is lower than a threshold, then notification system creates a message with a recommended ventilation strategy
	Values from all sensors are being updated	Updated values hourly
Application	State of sensors	On-state for all sensors and actuators
	Connectivity of the gateway	Ping received every 4 minutes
	State of the database	All resources stored and values updated according to the sending frequency for each type of sensor
	Functioning of the rule engine	Recommendations generated when indoor air quality level is bad
	Functioning of the notification system	Recommendations visualized in the user application
	System usability assessment	Usability assessment above 68 according to the SUS tool and two or more affirmative answers to indicate that the system can increase the quality of life.

4.1.2.4 Issues & Risks

The main issue during pilots was the sensors monitoring to make sure that all of them were reporting correctly. The battery management was the main constraint for the indoor

sensors. On the other hand, some outdoor sensors ever stopped working for some days but we did not have any direct control over them to fix the problem.

4.1.2.5 Functional tests – Status and Results

The following tests were carried out in order to validate the integration of the nAssist platform with the symbIoTe Core:

Test Description	Status	Results
Registration of the nAssist platform	Passed	nAssist platform listed
Registration of the resources	Passed	A total of 64 resources were registered: • Three sensors (O3, NO2, PM10) for 7 fixed stations • Eight sensors (CO2, temperature, humidity, energy consumption, brightness, O3, NO2, PM10) for 5 houses • Three actuators (tower heater, TV, wall plug) for 2 houses
Interpolation function updates hourly values for O3, NO2, PM10 for each house by taking into consideration the GPS coordinates and values of the fixed stations and GPS coordinates of the houses	Failed	• Interpolation function finds fixed stations through a search in the core • Checking manually the output of the interpolation function
Data exchange between the interpolation function and nAssist platform. Resources are required to be found through a search in the core	Pending	• Interpolation function finds fixed stations through a search in the core • Output from the interpolation function matches with the data visualised in the user application

4.1.2.6 Execution of the trials with users – Status and Results

The following tests were carried out in order to evaluate the functionality of the application and the user experience:

Test Description	Status	Results
To guarantee the correct functioning of sensors and actuators through the application.	Passed	Monitoring of the list of sensors and actuators through the Devices functionality. All values were updated according the sending frequency configuration of each sensor.

To guarantee the correct functioning of the gateway. The gateway sends a ping to the cloud every 4 minutes to indicate that it is working fine. If there is not any ping received after 10 minutes, the notification system sends an alert.	Passed	A ping command sent from the gateway every 4 minutes. When the gateway lost its connectivity, an automatic notification was sent to the application.
To validate the correct storage of the data coming from sensors in the database of nAssist platform.	Passed	Database updated with values from all resources.
To validate the correct functioning of the rule based reasoning engine. The rule engine generates the recommendation to the user when the indoor air quality is not good according to some rules.	Passed	Generation of recommendations when the indoor air quality was modified manually to check the rule engine functionality.
To validate the notification system which will send a recommendation to the user through the application when the rule engine detects a bad indoor air quality.	Passed	Recommendations received by the user when a bad indoor air quality was detected.
To measure the user acceptability. Two different questionnaires were provided to the user to complete after the trial. One questionnaire is for measuring the usability of the system (System Usability Scale, SUS) and the other one is to measure the effectiveness of the recommendations to improve the user's quality of life.	Passed	• The SUS score was calculated for each user. Three participants got a score above 68 and 2 participants below 68. • Two or more affirmative answers from the dichotomous questionnaire indicates that the system helps to improve the quality of life. Two participants got two or more affirmative answers and three participants had less than two affirmative answers.

4.1.2.7 Conclusions

Two different sets of tests were carried out to evaluate the integration of the nAssist platform into the symbIoTe system and the functionality of the application, from a technical and a user experience point of view. The use of the user questionnaires were intended to measure perceived ease-of-use and perceived improved quality of life, distinguishing between unusable and usable systems. One critical point was the small number of

participants and, therefore, the limited number of trials that does not allow us gain conclusions about the user acceptability and the functionality of the application in terms of improving the participants' quality of life. However, the evaluation of the integration of the platform into the symbIoTe system did not depend on the number of trials.

4.1.3 Smart Mirror

The general idea of the trial was to test the smart mirror application (SMILA) over the course of one week (per participant) in an unsupervised setting, in a lab environment at the AIT offices, with participants from the AIT workforce. Participants were asked to wear a Fitbit Ionic smart watch (paired with an Android-based smart phone) on a daily basis and to interact with the mirror at least once a day performing (a) measurement of weight and (b) recording of personal wellbeing following a measurement schedule provided by the KIOLA health platform. The overall workflow can be summarized as follows:

(1) The participant stands in front of the smart mirror
(2) The device ID of the Fitbit smart watch is resolved into a symbIoTe resource ID using a search on the core interface
(3) Accessing the RAP interface of KIOLA data on physical activity done in the current week is retrieved from KIOLA and displayed on screen with a personal greeting of the user.
(4) The user is asked to step on a Bluetooth-enabled scale, taking a weight measurement
(5) The user is then asked to state his/her personal well-being using speech recognition
(6) Weight and personal well-being are transferred back to the KIOLA platform.

The trial is an extension to the 2 living lab evaluations done in November 2017 and March 2018.

4.1.3.1 Definition of the Test cards and Questionnaires

Test cards have been developed around three categories: Key performance indicators in the area of integration issues (data retrieval and submission to the KIOLA IoT platform), the smart mirror app itself (e.g. speech recognition accuracy) and usability and user acceptance. A custom questionnaire was designed to evaluate perceptions in the field of fitness apps as well as a standardized usability questionnaire (SUS) was used in order to evaluate the usability of the system.

4.1.3.2 Deployment

SMILA was deployed in AIT offices in the Smart Business Center in Graz, Austria for a period of one month starting with Oct 5th, and the trial ended on Oct 22nd. 7 people (3 female, 4 male) were recruited from offices in the Smart Business Center over the course of two weeks. Participants signed an informed consent in agreement with European General Data Protection Regulation (GDPR) following a personal information conversation. Moreover, participants were handed out smart phones with de-identified google accounts for privacy reasons. Actual weight measurements were de-identified as well before data got stored in the database. Following inclusion and exclusion criteria where used for interested participants:

Inclusion criteria:

1. Participants are older than 18 years

Exclusion criteria:

1. Participants wears active or passive implants (e.g. pacemaker, stents, insulin pumps)
2. Participants suffering from cardiac arrhythmia
3. Pregnancy
4. Participants suffering from EMF syndrome
5. Participants taking beta blockers or antiarrhythmic medication
6. Participants suffering from rheumatism
7. Known allergies against product materials as found in common fitness trackers (especially in wristbands)

User interaction was not explained or shown to end-users. Finally end-users were asked to keep a personal “interaction diary” in order to document any successful or failed interaction.

4.1.3.3 KPIs

We have defined KPIs to measure the functionality of the application from the user point of view and from a technical point of view:

Type of KPI	Description	Target
Integration	Number of correctly recorded weight measurements	>80%
Integration	Number of correctly recorded measured personal well-beings	>80%
App	Average confidence for speech recognition	>75%
App	Number of repeated weight measurements	< 20 %
App	Number of repeated questions for personal wellbeing	< 20 %
Usability	User compliance	>50%
Usability	User acceptance (for use in daily life)	>50%

4.1.3.4 Issues & Risks

The main issue during the trial were the smart watches. End-users reported problems with Fitbit Ionic watches properly synchronising data with the backend. The smart watch was supposed to synchronize data with the Fitbit cloud and KIOLA would then synchronize this data with its own database. From there, it would be transferred back to the smart mirror app using the symbIoTe RAP interface. However, the smart watch stopped synchronizing data into the Fitbit cloud in some cases resulting in missing data up to 2 days.

4.1.3.5 Functional tests – Status and Results

The following tests were carried out in order to validate the integration of the KIOLA platform with the symbIoTe core:

Test Description	Status	Results
Registration of the KIOLA platform	Passed	KIOLA is visible as IoT platform in symbIoTe core (open server)
Registration of the resources	Passed	A total of 4 resources (Fitbit Ionic smart watches) were registered with symbIoTe core for the trials.
Successful data retrievals via KIOLA symbIoTe RAP implementation	Passed	27/35 successful interactions

The following table provides statistics on response times of the overall system. “User interface active” denotes the time it takes, until the user interface is active for end-users, meaning, the Bluetooth device ID is resolved into a symbIoTe sensor ID, data is retrieved from KIOLA and the personal greeting is performed on the successful interactions with the mirror:

	User interface active (ms)	symbIoTe sensor recognition complete (ms)	KIOLA data collection complete (ms)
Median	7481	6252	1166
Average	7565	6194	1371
Max	13811	11496	2315
Min	3193	2657	536

4.1.3.6 Execution of the trials with users – Status and Results

A total of 35 interactions with the mirror were recorded over the time of four weeks. 28 (80%) interactions were successful (meaning the whole workflow could be completed) and 7 interactions failed. Compliance with respect to daily measurements was 94.28%.

Test Description	Status	Results
Number of correctly recorded weight measurements	Passed	28/35 (80%)
Number of correctly recorded measured personal well-beings	Passed	22/28 (78,5%)
Average confidence for speech recognition	Passed	0,83 (±0,07)
Number of repeated weight measurements	Not evaluated	KPI was initially included in the list of performance criteria but could not be evaluated after the final trial.
Number of repeated questions for personal wellbeing	Not evaluated	KPI was initially included in the list of performance criteria but could not be evaluated after the final trial.
User compliance	Passed	94,28%
User acceptance (for use in daily life)	Passed	SUS Score 74.17 (B-)

4.1.3.7 Conclusions

The SMILA trials were carried out over the period of roughly one month. 7 persons participated in the trial and 80% of all interactions were successful. However, due to the somehow complicated trial protocol (i.e., the actions/tasks needed to be performed by the participants in order to test the various aspects of the use case), some complaints have been received, with 2 people aborting the trial early due to a lack of time to perform the required tasks. Speech recognition and interaction proved to be reliable enough for real-word deployments. While technical integration was successful, an evaluation of interactions with the Core symbIoTe components revealed high response times up to 11 seconds. While the user interface is designed to show a progress bar during this time, indicating that the system is retrieving data, it can be argued that there is room for improvement in terms of system performance.

4.2 Smart Mobility & Ecological Routing

The Smart Mobility and Ecological Routing Use Case addresses the problems regarding environment pollution and air quality in the major European cities. It does so by collecting air quality data from multiple IoT platforms in different countries and uses such measurements for runners, joggers and cyclists in order to plan the best routes to their destination.

Through symbIoTe, air quality measurement are obtained from different platforms. This data is obtained from fixed stations already existing in the cities and from specifically deployed mobile sensors which were carried by the users.

This air measurement data is aggregated and interpolated using a symbIoTe enabler and associated to the map’s street segments. With the streets correctly classified by their air quality, routing engines can take that information into account when computing the most ecological routes to the application’s users. These paths can also benefit from other

factors such as traffic and available parking, in case the platforms have access to these kinds of sensors.

Finally, users are able to search for Points of Interest (POIs) following certain criteria, including data from sensors such as available parking or noise levels. Routes for the selected POI can be computed using the previously mentioned service.

4.2.1 Definition of the Test cards and Questionnaires

For the end-user satisfaction evaluation, we evolved the questionnaire used in the Zagreb pilot test (see deliverable D5.3) in a combined effort with all involved partners. Some questions were added, especially the System Usability Scale (SUS)[2]. Similarly to the pilot test in Zagreb, participants had to fill out the questionnaire after the successful completion of the trial. The questionnaire was provided via Google Forms and thus could be filled out online.

As already stated in D5.3, the questionnaire is divided in five sections:

- General demographic information about end user and their equipment,
- Questions related to the wearable sensor, the data acquisition process and the CUPUS (sensing) app (including SUS for this app),
- Questions related to the routing, the PoI search services and the SMEUR (routing) app (including SUS for this app),
- Questions related to the overall Quality of Experience (QoE), and
- other comments

The System Usability Scale (SUS), as described in section 4.1.2.1, is an easy to use, efficient and well-tested scale to assess the overall usability of a system. The questionnaire thus features quantifiable feedback, as well as qualitative feedback through text boxes with free text input. With this free text input we hoped to gain some deeper insights into positive as well as negative experiences with the SMEUR ecosystem.

4.2.2 Deployment

The use case is deployed across three consortium members (AIT, UNIZG-FER and UW), where each member is responsible for its own components and maintains its own platform. The SMEUR Enabler is running at the UNIZG-FER site, but it is a joint work of all three involved partners (UNIZG-FER, AIT and UW). The overall architecture is depicted in Figure 3.

The use case deployment relies on the two mobile applications published in Google Play store: 1) CUPUS crowdsensing application[3] that gathers data from mobile sensors (UNIZG-FER sensors and DunavNet sensors); and 2) the symbIoTe SMEUR routing application[4] which offers POI search and green routing information. The CUPUS app is a pure OpenIoT application which uploads sensor data acquired by mobile phones and wearable air quality sensors to the OpenIoT platform. It exposes collected sensor data using symbIoTe L1 components (mainly RAP). Similarly, AIT created an extension for the

[3] https://play.google.com/store/apps/details?id=hr.fer.tel.cupusmobileapp
[4] https://play.google.com/store/apps/details?id=com.ubiwhere.symbiote

openUWEDAT platform to expose data collected from stationary stations across Austria and Croatia) using symbIote middleware, and UW's MoBaaS platform exposes sensors deployed in Porto. For Vienna, the routing service used was developed by AIT and, for Zagreb and Porto, MoBaaS routing engine, developed by Ubiwhere, was used. symbIoTe core services are running on PSNC's servers, supporting the use case with services for registration and discovery of resources (i.e., data sources).

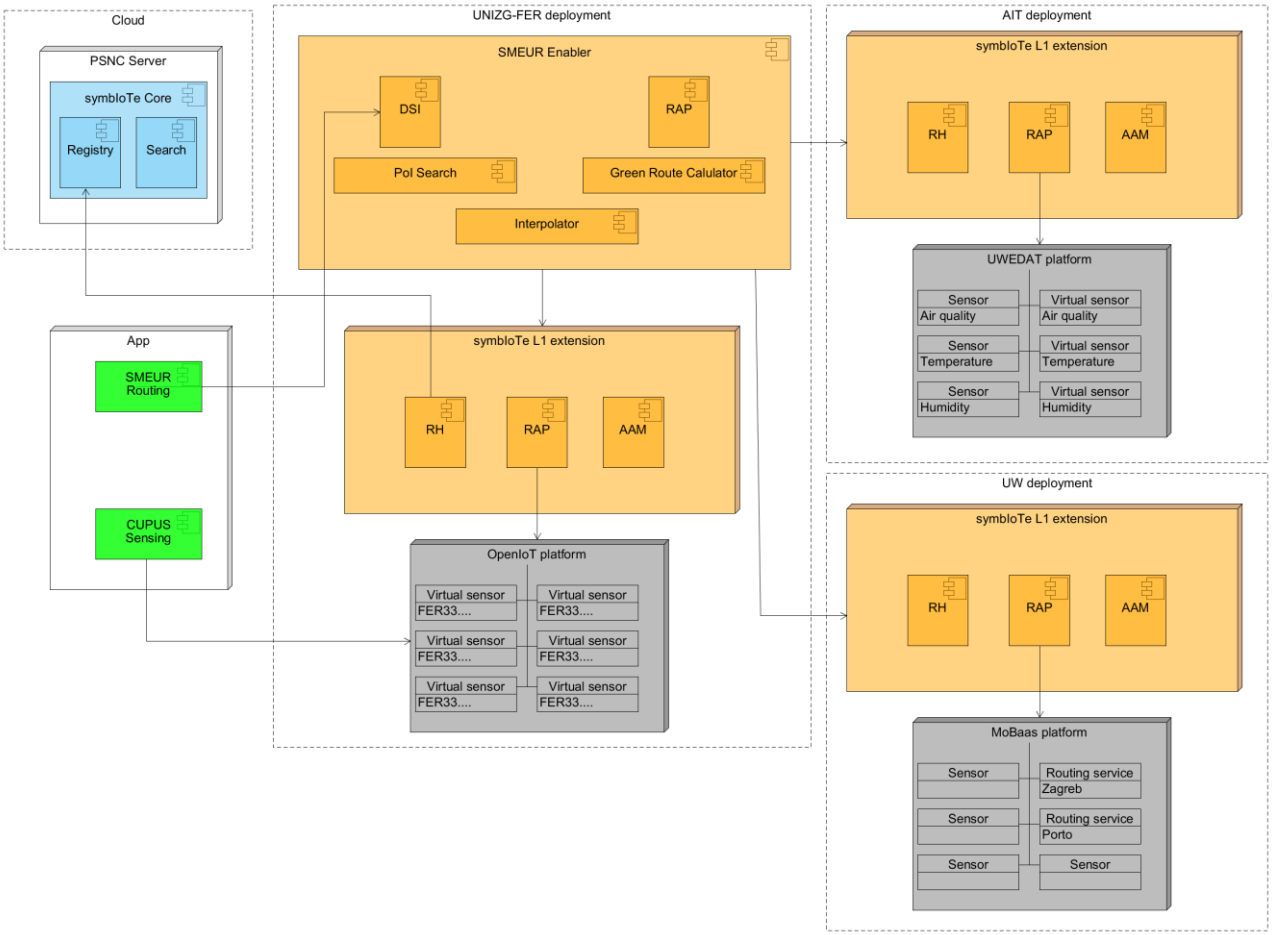


Figure 3: SMER use case deployment

Vienna

To obtain official air quality data for Vienna, the official source for air quality data in Austria provided by the UBA was used. The UBA data was pulled from their server and stored in a system, hosted by AIT (openUwedat). This system was integrated as a L1 platform in symbIoTe and the 17 sensors were registered in the core. The routing enabler was then able to find these sensors and provider their data for the interpolator. To provide smart ecological routing for Vienna, AIT's routing engine was updated to use the air quality data provided by this interpolator as an additional parameter for the route calculation.

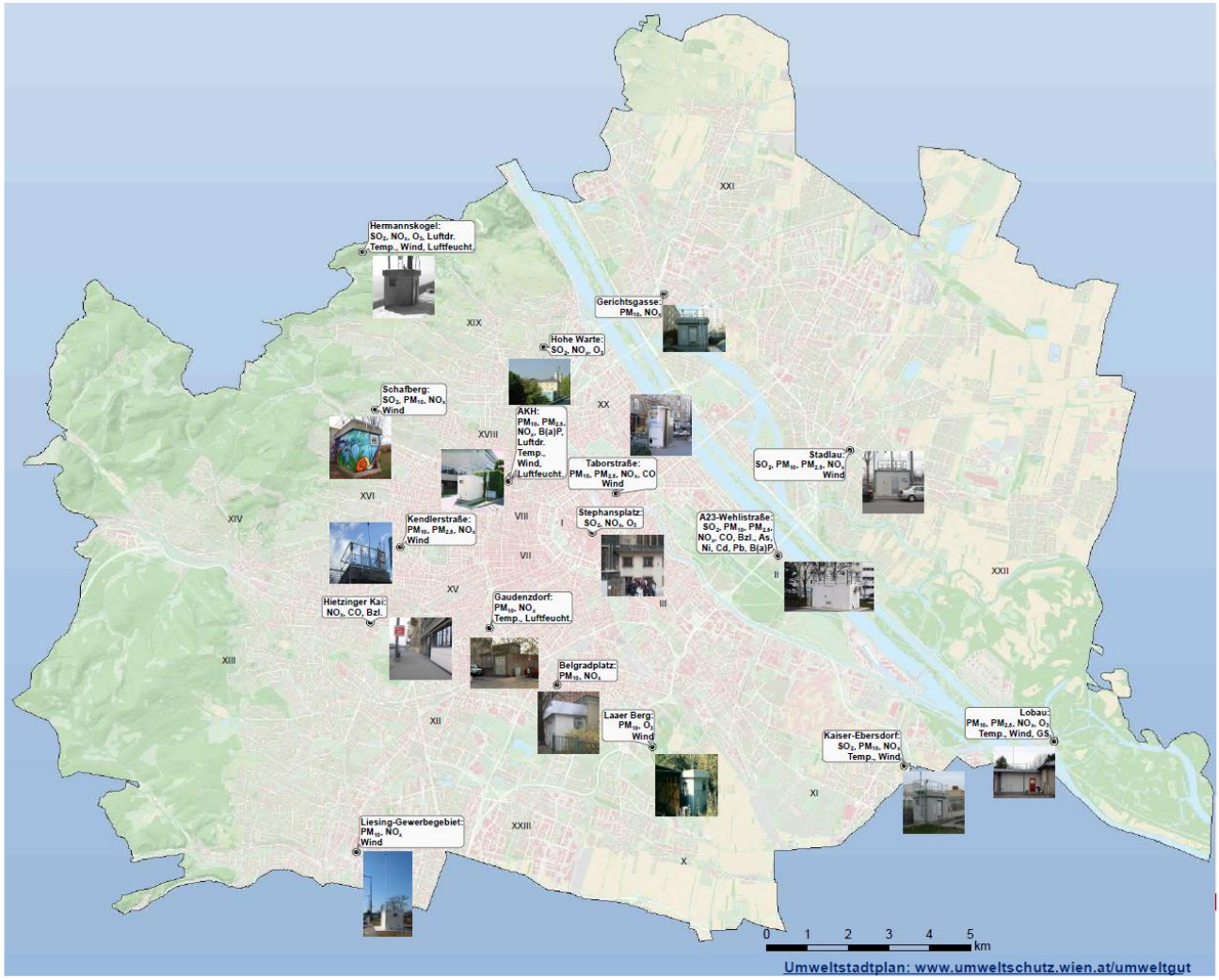


Figure 4: Stationary air quality stations in Vienna

Wearable sensors for the trial participants were built by DunavNet, based on the specification of AIT and FER. Overall 20 mobile sensors were used in the trial, from which ten were provided to the participants, and the other 10 sensors were used for testing and stationary measurements.

Figure 5: Mobile air quality sensors used in Vienna

Zagreb

The city of Zagreb is covered by observations gained from fixed environmental stations (integrated through the openUWEDAT platform) and wearable sensors that we carried by end users. Wearable sensors were connected to mobile phones acting as gateways to transmit air quality readings to the OpenIoT platform. The mobile sensors used in Zagreb, both by UNIZG-FER students and volonteers recruited by our OC2 winner Sindikat Biciklista are shown in **Figure 6**. The trials in Zagreb started in June (organized by UNIZG-FER) and were organized by Sindikat Biciklista from September to November 2018). Wearable sensors sense CO and NO_2 gas levels, temperature, humidity and atmospheric pressure, and end user mobile phone gathers the noise level and luminescence. On average, 10 to 20 volunteers were actively acquiring air qulity data during the trials, and each volunteer was carring a wearable sensor for a duration of one to two weeks, depending on his/her availability. Volunteers were asked to acquire sensor data during their normal daily activities.

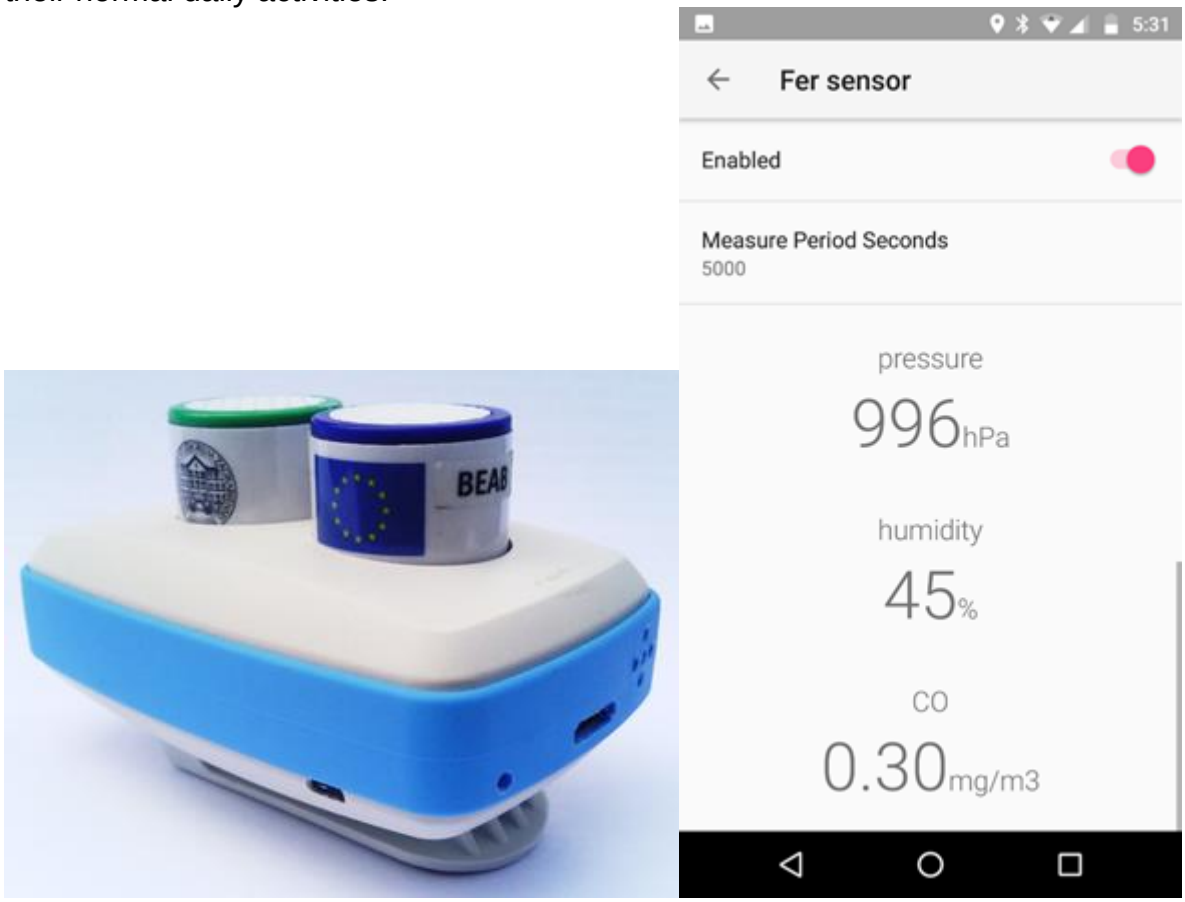


Figure 6: Wearable sensors used in the trials in Zagreb

Volunteers have collected air quality readings covering almost the whole city of Zagreb, as depicted in Figure 7, and the total of 56 stationary FER sensors were registered to convey readings from wearable sensors during the trials.

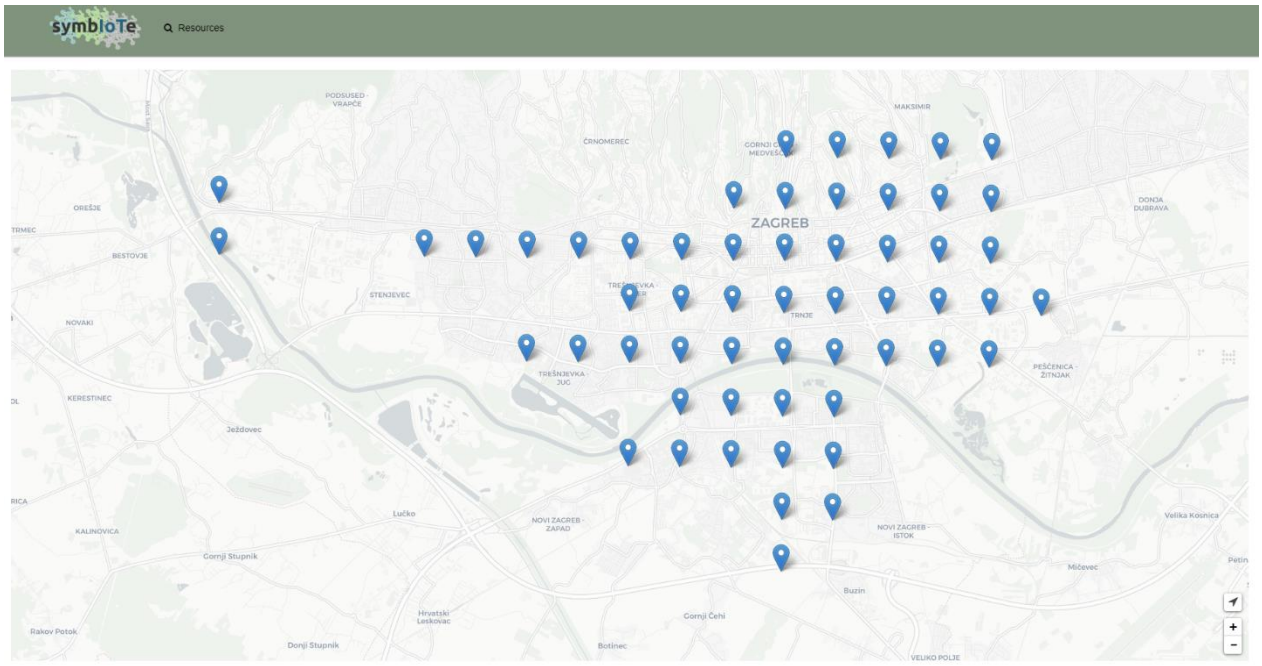


Figure 7: OpenIoT sensors registered in symbIoTe for the city of Zagreb

Porto

The air quality readings for the city of Porto comes from the MoBaaS platform, which gathers data from fixed air quality stations and wearable sensors (mainly the ones used in the OC2 extension for trials by Monitar). The sensors gather data about temperature, precipitation, noise levels, carbon dioxide, nitrogen dioxide, ozone, humidity, wind speed, light, solar radiation, carbon monoxide, PM1, PM2.5 and PM10. Trials were run during November and December, where 20 users used the developed application and reported their experiences.

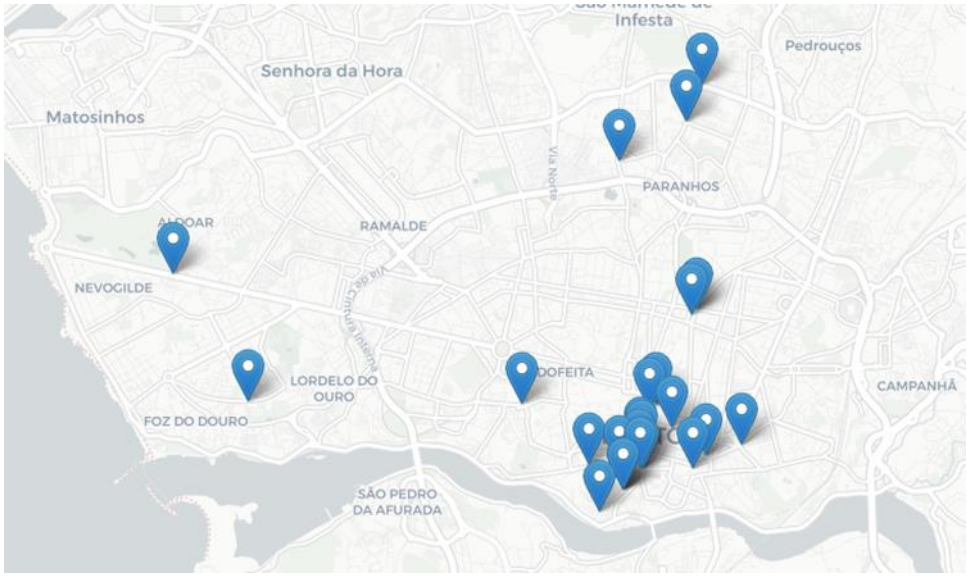


Figure 8: Air Quality Sensors in Porto

The trials were organized in a way that they could be done at the same time as the OC2 extension winners MONITAR trials. This way, more air quality data could be gathered at the same time, both from fixed and mobile sensors.

Figure 9: Air Quality sensor in Porto

4.2.3 KPIs

Type of KPIs	Description	Target Value
User	User Satisfaction Score	Above 68
Technical	Ecological Route response time	Less than two seconds
	Point of Interest request response time	Less than two seconds
	RAP plugin response time	Less than one second
	Core services number of requests	6 per hour
	Number of requests made to RAP plugin	30 per hour per platform

Logging on the OpenIoT extension was implemented at two levels: 1) inside the RAP plug-in to measure processing time of the RAP plug-in, i.e., to measure time to acquire data from the underlying database (Virtuoso) and to serve it to the generic RAP component; and 2) at the Interworking Interface to monitor all connections between symbIoTe extension and the outside world (including the symbIoTe core services).

The RAP plug-in processing time was measured by putting timers at the beginning and at the end of the requests made to the plug-in, and the total processing time is the difference between two timestamps.

The Interworking Interface is the nginx reverse proxy server, which enables monitoring of each connection routed through it. The OpenIoT extension is configured to use the Interworking Interface for all incoming and outgoing connections (e.g., to the symbIoTe core), and even for the communication within the same machine (e.g., RAP checks the token validity at the local AAM). Thus, it was enabled to monitor all parameters in the nginx log. The log contains the following data: IP address of request, HTTP method and path, response time and upstream connect, header and response time. Additionally, number of log files for the nginx server was increased to save up to last 70 days of log, in order to process performance related data during whole length of the trial. To process the nginx log file, we wrote a simple Java application that calculates the probability mass function for each request type (e.g., request to AAM, or request for resource data using the OData interface)

A similar approach, used in the OpenIoT extension at UNIZG-FER, was used at the enabler machine. The nginx server was configured in the same way, measuring the same parameters, but processing of the log was modified to measure enabler specific requests, rather than the generic L1 extension components.

For MoBaaS, a similar approach was used to the one implemented for OpenIoT's nginx, only making an addition to the script to also take into account requests to services (or, more specifically, the routing service).

4.2.4 Issues & Risks

Vienna

The preparation of the trial took longer than expected, as the sensor boxes were delivered with a significant delay by the external partner (end of June). Thus, all related tests of the app and the actual roll-out to the test-users where delayed by about two months.

Due to the late availability of the sensors boxes, the initial user tests were also compromised by bugs in the app used for reading out the sensor measurements. This was caused by a different implementation of the Bluetooth stack in the boxes, as initially implemented in the application. In the later test groups, this problem was eventually resolved. Furthermore, the trial could not be carried out with students, as initially planned, as the testing had to be shifted to the summer holiday season.

Zagreb

The first trial organized by UNIZG-FER was held in June 2018 after the initial testing of individual components, so this was the first run of all integrated components in a real-world deployment. Overall, the system worked well but some problems with connectivity with sensor node were registered. Due to a large number of components involved in execution of the SMER use case, a limited risk is present for the service unavailability due to a single component failure.

The second trial, organized in autumn 2018 (September – November) by Sindikat Biciklista raised different risks, with respect to the recruitment of volunteers and volunteer activity. Risk related to a low number of recruited volunteers was managed by organizing multiple workshops and promo events at UNIZG-FER and HUB385, where the developed apps were presented and flyers for possible volunteers were disseminated. Additionally, the risk of volunteer inactivity during trial was present because volunteers could forget to turn on a sensor node while moving around the city. This risk was managed by checking

the number of data readings regularly (on daily basis) so that a reminder can be sent to all volunteers timely.

Due to the physical limitations and characteristics of sensor nodes, a risk related to the weather was present during trials (sensor nodes should not get wet) and volunteers were advised not to use sensor nodes while raining. In case of longer periods of raining weather, it is expected to gather only a low number of readings.

Porto

A major problem that affected the Porto trial was that the development of the wearable sensors got delayed, which resulted in the trials sarting only using fixed stations. The same users were able to use the app again with the wearable sensors working.

Secondly, at some point during the trials, the routing server was taking some time to return results to the users. Although the users thought they were not too much affected by the delay, it is still a problem Ubiwhere put some effort to fix and it was solved within two days.

4.2.5 Functional tests – Status and Results

Test Description	Status	Results
User Satisfaction (SUS)	Passed and Failed	62.5% (see explanation following this table)
Green Route response time	Passed and Failed	More than 64% of requests were executed in less than 2 seconds. Total average of 2,996 seconds.
PoI Response time	Passed	More than 93% of requests were executed in less than 2 seconds. Total average of 1,04 seconds
RAP plug-in response time	Passed	More than 67% of requests were executed in less than 1 second. An average of 0.48 seconds.
Core services request	Passed	On average 23 requests per hour
Number of requests for RAP	Passed	On average 1110 requests per hour

There are two entries in this table that stand out, but in the end are not very worrying. The first is the user satisfaction score, which is a bit low than what was desired (68%). This is due that the results from the first users, that tested the app around August and September, were not very positive. In the meantime, the app was fixed and improved, and later feedback was much more positive.

Additionally, the response times for green routes, although the majority was below 2 seconds, our target goal, the average was a bit higher. One of the main factors that contributed to this value was that, during 2 days, MoBaaS servers were having problems and route requests were taking close to 20 seconds. This skewed the number by quite a bit. After this was fixed, the process run much smoother and users didn’t complain about this issue.

4.2.6 Execution of the trials with users – Status and Results

4.2.6.1 Test Flow

Vienna

For the end-users, weekly training sessions where organized to introduce the system and to obtain their consent for participation and data handling. Prerequisites for taking part in the trial were regular walks by foot or commute by bike in the inner ring of Vienna. Participants then were asked to install both CUPUS as well as SMEUR app on their private smartphone and receive a mobile sensor provided by the AIT. Related material (user manual, GDPR consent, trial consent form, rental agreement) were prepared and handed out to the users. The trainings took place at AIT and UNIVIE between 20. August and 17. September.

Due to the delayed start of the trial, the evaluation couldn't be carried out with students from the university, as initially planned. The testing was instead done by colleagues and volunteers from AIT and UniVIE. Each participant received a compensation in the form of an external USB battery pack. All of these users had a sensor box for the duration of the trial and installed the SMEUR app, as well as the smart ecological routing app installed on their own mobile phones.

Zagreb

As already stated, the second trial in Zagreb was organized by the OC2 winner Sindikat Biciklista. In order to encourage volunteers, mostly cyclists, from different parts of the city to participate in a real-world trial, Sindikat Biciklista organized four workshops and three promo events in coordination with UNIZG-FER to inform users about the project and demonstrate the usage of both application. The organized events had a high impact on the audience and resulted with an increased number of both CUPUS as well as SMEUR application downloads in the following days. In total 53 volunteers from Zagreb downloaded the CUPUS Crowdsensing App, and 189 volunteers downloaded the SMEUR Routing App. Altogether, the recruited volunteers collected 123,309 different measurements about their environment during the three months period as shown in Table 1. The most measurements were obtained during October which confirms that the organized promo events had a great success among volunteers in Zagreb.

Table 1 Number of measurements which were collected during the trial period

Measurement type	September	October	November
ambientLight	2840	15413	9549
noise	8536	15413	9687
CO	3596	6410	2593
humidity	3683	6084	2813
NO2	3353	5898	2585
pressure	3696	6544	2734
temperature	3697	5471	2714

Porto

Due to the opportunity to organize the trials alongside OC2 winners MONITAR, which would create a much larger engagement and concentration of air quality sources, the trials were delayed. Unfortunately, due to several problems, MONITAR trials were delayed more than expected, and the process only begun in November. This led to less users being available at the time, but still provided good results and valuable feedback, to the developed application and enabler, and symbIoTe in general.

Users either used their personal mobile phones or were provided a phone by Ubiwhere, and used the app between the 5th of November and 7th of December.

In overall 48 participants across all sites filled out the questionnaire. While in Zagreb as well as in Vienna participants used both apps and the sensor pack, in Porto only the routing app was tested:

Table 2 - Trials' Participants

Location	Participants	Questionnaires obtained	Apps
Zagreb	~40	8	CUPUS + Routing
Vienna	26	26	CUPUS + Routing
Porto	~20	14	Routing

The majority of participants were 25-34 years old, some were older (35-44), while only a small percentage was either under 25 or over 45. All of the participants were skilled Android users, since owning a private Android phone was a prerequisite. Most of the participants did walk or bike regularly in their respective cities (see Figures 10 - 12).

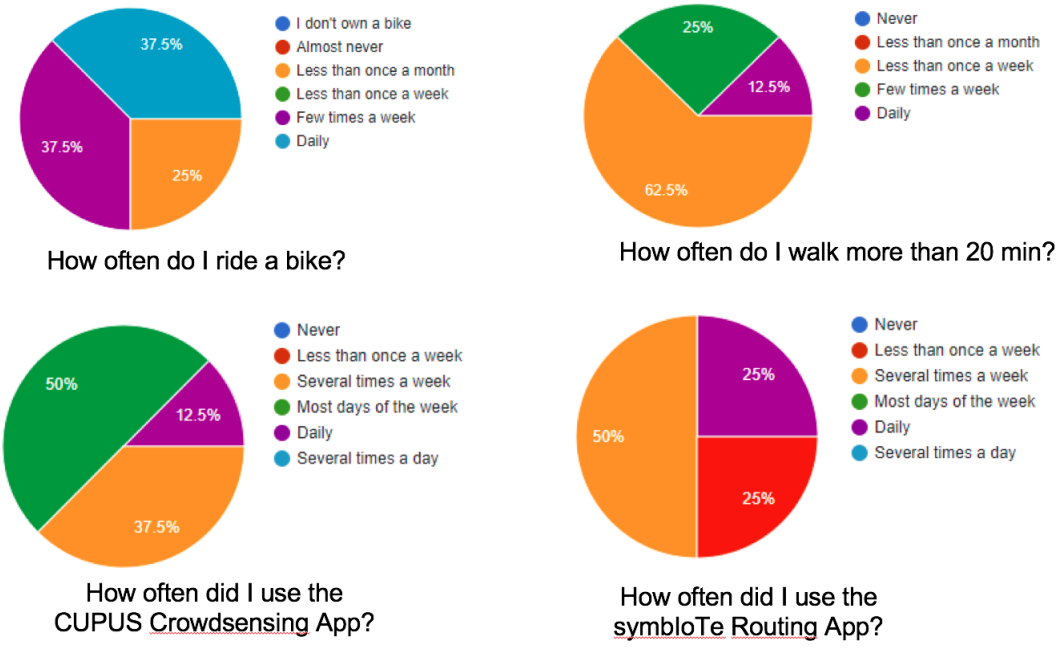


Figure 10: Commuting behaviour and app usage in Zagreb

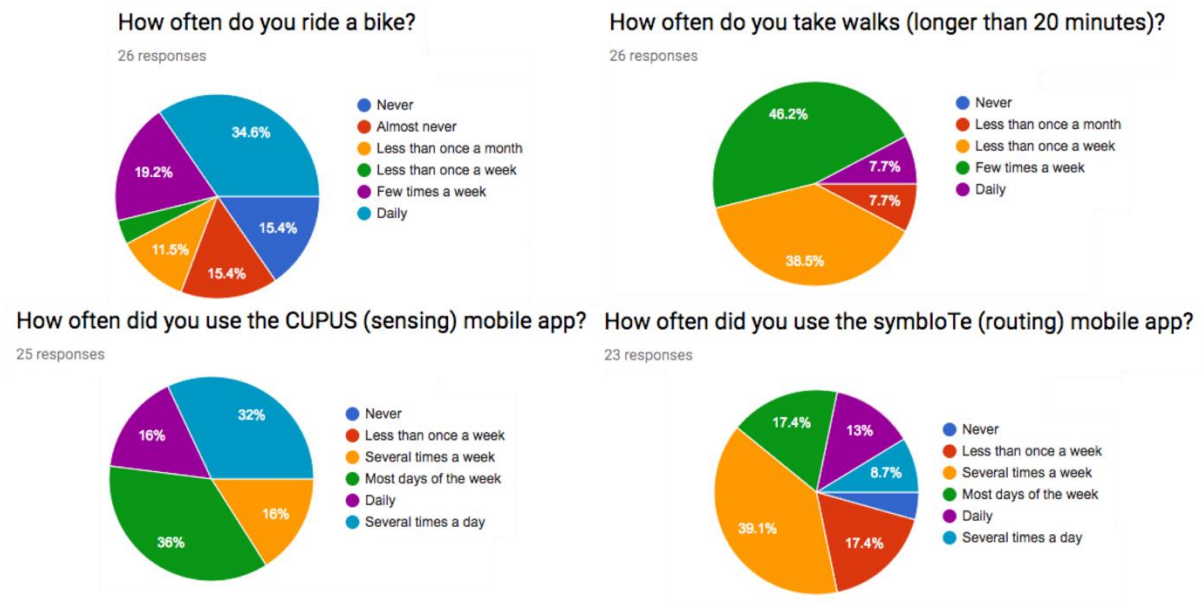


Figure 11: Commuting behavior and app usage in Vienna

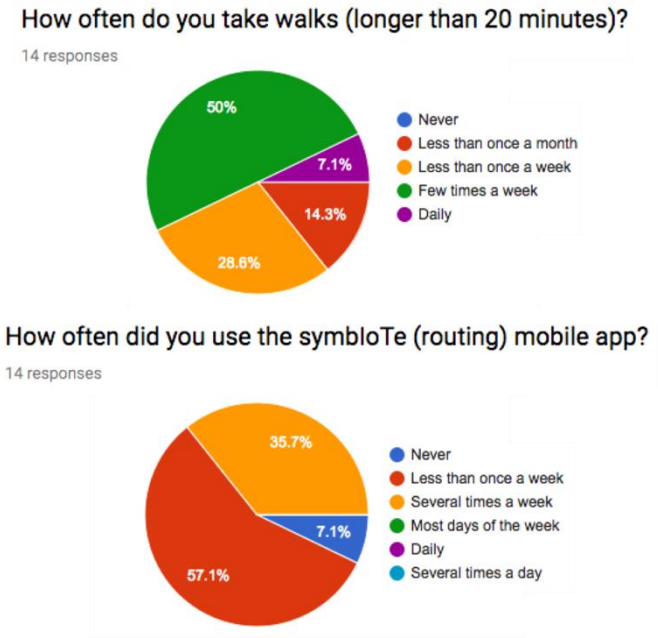


Figure 12: Commuting behaviour and app usage in Porto

In Vienna, 25 participants said they've used the CUPUS (sensing) app and 23 used the SMEUR (routing) app. From the 14 participants in Porto, 13 used the routing app. In Zagreb, 8 of the 8 participants used the CUPUS app and 4 used the routing app.

For each of the apps, the participants filled out a SUS scale, as explained previously. The SUS scores are slightly below the target KPI, but nonetheless the outcome is promising:

- Vienna CUPUS: m = 60.6, sd = 9.6
- Vienna SMEUR: m = 60.8, sd = 8.6

- Porto SMEUR: m = 67, sd = 10.6
- Zagreb CUPUS: m = 65.3, sd = 3.3
- Zagreb SMEUR: m = 62.2, sd = 8.5
- **CUPUS overall: m = 61.7, sd = 8.7**
- **SMEUR overall: m = 63, sd = 9.6**
- **Both apps overall: m = 62.5, sd = 9.2**

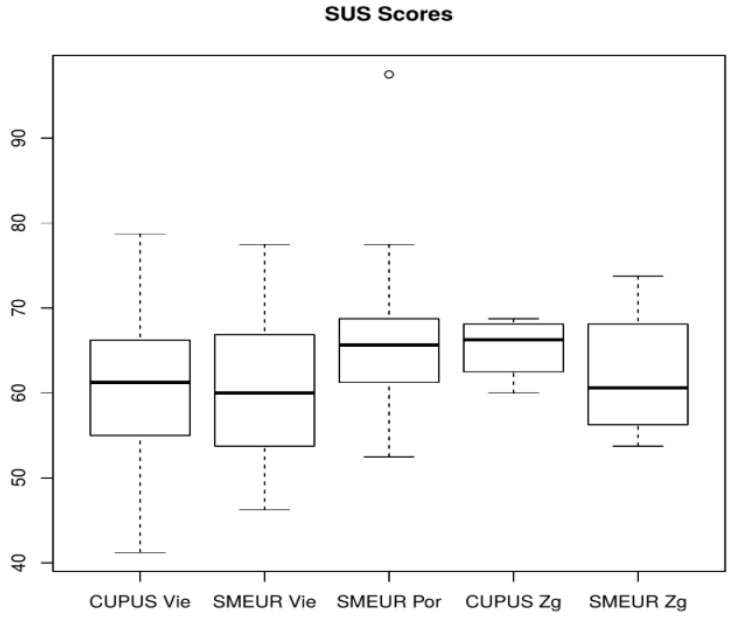


Figure 13: Boxplots of the SUS scores across all testing sites (scores can range from 0 – 100)

In the last part of the questionnaire, participants were being asked in open questions what they liked and disliked about the system, or if they encountered any problems.

4.2.6.1.1 Dislikes

One major issue during the trial were **usability issues and instability** of both apps. Since for the trial high-end prototypical apps and sensors (instead of a production-ready system) were used, the users encountered several issues – many of which were being instantly fixed during the time of the trial. After these bugfixes updated versions were spread to the participants. Another major issue was battery drainage, which in Vienna we tried to address by giving out battery packs as compensation for participation, which proved to be helpful.

In Vienna, the **CUPUS app** had to be fixed shortly after the trial started, since the implemented Bluetooth stack didn't work with the newest version of the sensors from DunavNet**.** After the successful re-implementation of the connection to the sensor worked, but participants complained about an unreliable connection. Sometimes sensor readings weren't displayed and participants were unsure whether data has been acquired and been sent to the server. Also, the end users perceived the prototypical sensor box to be cumbersome and noted that future, production-ready versions should be smaller.

The **SMEUR routing app** initially didn't load the local maps due to blocking of certain IP addresses (e.g. from the university). After this issue has been fixed, the app generally worked smoothly for routing. The only downside for the participants was the intransparent indication of air quality along the given route, since a visual indicator or measurement information weren't displayed on the map. Often participants couldn't tell why a certain route was chosen.

4.2.6.1.2 Likes

On the other hand, participants **liked many aspects of the system**. Especially the idea behind the use case, namely ecological routing, was something many end users were interested in and found it highly valuable. Participants saw a high value in routes through their city with good air quality and said that not only routes, but also other higher-impact decisions would profit from better knowledge about air quality. One Participant stated: *"I think just having a detailed map of air quality data would be more interesting though, seeing as I would probably want to choose destinations based on air quality"*. Another end user said that air quality maps often contain large gaps, and that portable sensors could fix this issue at the root. Other use cases like the real estate market, city-wide traffic routing and environmental monitoring in general were mentioned.

Despite certain usability issues, participants praised the simplicity and the clean design of the **SMEUR routing app**. One participant stated that *"finding routes between current location and POIs is very easy to do"*. Also fast response and non-existent delay of route calculation and presentation on the smartphone was mentioned. Participants also said that they found the POIs nearby interesting and that they learned something about their surroundings.

For the **CUPUS sensing app** the end users especially liked that they could directly observe the gathered data in real time. Thus, they could monitor the air quality on their daily routes and gain insights into the air pollution they encountered during their day: (I like to…) *"Get (almost) instant feedback on air quality for the routes I take quite often"*. It seemed to appeal their scientific curiosity.

Overall, the majority of participants tend to recommend the system to friends and family (see Figure 14). Trials performed later in the project tended to have more positive results. This is due to issues being solved since the first use of the app (Vienna trial), and the last users tended to be presented with an app more reliable and adjusted following feedback.

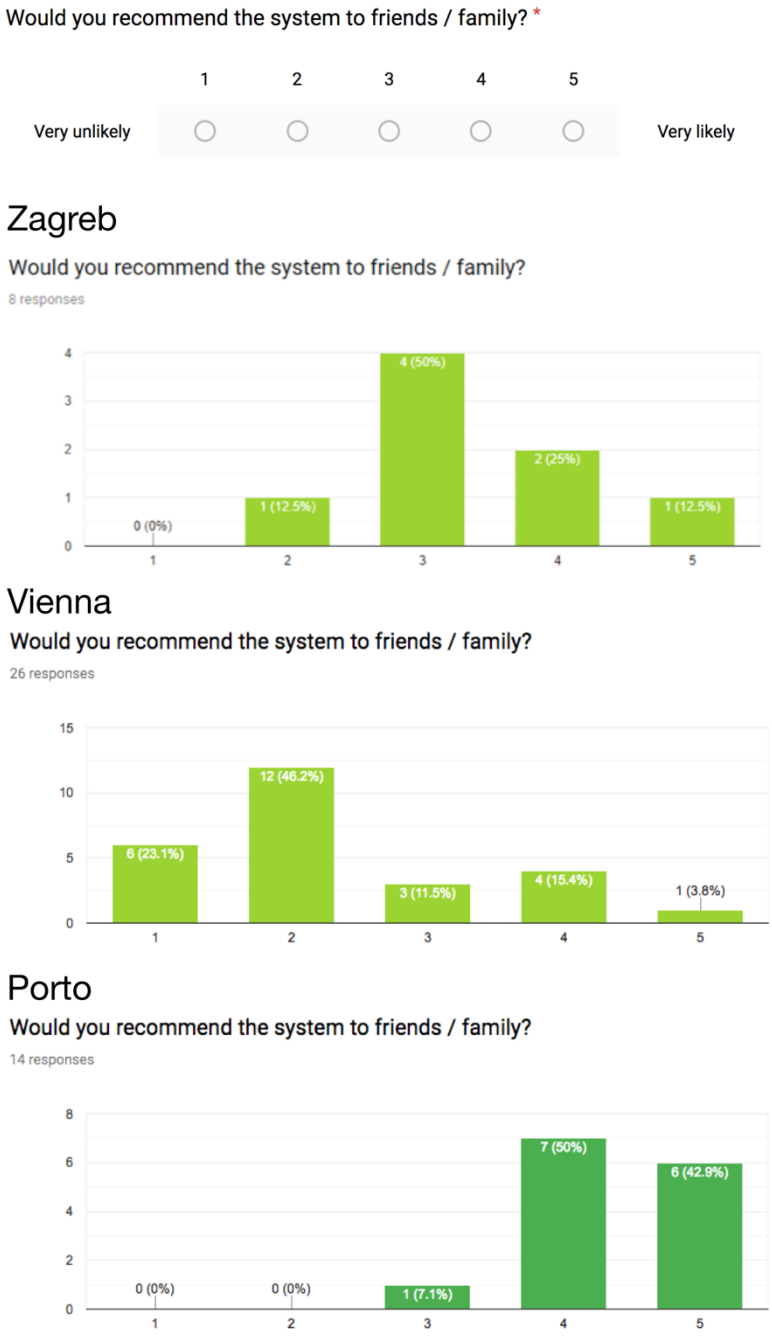


Figure 14: Outcome of the question "Would you recommend the system to friends / family?" across all testing sites

4.2.7 Conclusions

The SMEUR use case was developed by UNIZG-FER, AIT and UW in cooperation with their local communities and OC2 winners, Sindikat Biciklista and MONITAR. The main goal was to address the problem of urban pollution in European cities and offer “green” ecological routes to the end-users. To achieve this goal, citizens of Zagreb, Vienna and Porto were utilized to participate in real-world trials, and collect dense spatio-temporal

data about their environment, as well as to use the routing application to their destinations. In total, more than 80 users actively participated in the trial, and more than 230 users at least once tried the routing application. After the trial, participants were asked to respond to a questionnaire and provide their feedback regarding both applications. Overall user satisfaction on a SUS scale was 62.5 which is slightly below the targeted KPI, but the users still see the benefits of the applications and would recommend them to their family and friends.

4.3 Smart Campus

The EduCampus use case is inspired by the eduroam (EDUcation ROAMing) initiative, an international roaming service for users in research and education. The key idea behind both concepts is to agree on a common framework to harmonize infrastructure services, in order to provide researchers, teachers and students easy and secure access to campus services when visiting campuses other than their own. While eduroam focuses on network access, the EduCampus use case aims for IoT middleware services.

The vision behind the EduCampus is following. When looking at the rapidly growing market for sensors included in smart devices, used in or attached to smart buildings, establishing smart campus infrastructures, there will be rich offering of services based on IoT middleware installations on a campus. Examples are climate control systems in workplaces, electronic access control systems, indoor location and navigation support, guidance systems for handicapped people, location-based collaboration support, or room information and reservations systems as used in the EduCampus use case.

The idea of this use case is to integrate two exemplary campus IoT platforms, one from KIT and one from IOSB, using different information models to describe the same domain. Both platforms provide legacy mobile applications that only work within the campus of KIT respectively IOSB, i.e. with data from their original platform. Goal of the use case is to enable both apps to work on both campuses, i.e. being able to consume data from both platforms, with no modifications to the apps and only minimal modifications to their backend services. As they do not use the same data model, this requires semantic mapping to translate data and queries between the two models.

4.3.1 Definition of the Test cards and Questionnaires

No formal interviews based on test cards have been done, because the use case could not be integrated with the symbIoTe framework. The encountered problems are described in more detail in Section 4.3.4. Experiences with the conceptual integration of the KIT information models are described in the deliverable D5.3.

4.3.2 Deployment

The backend application for the IOSB room reservation application and the KIT campus navigation application has been deployed on virtual machines within the DMZ areas in IOSB and the KIT data center.

Figure 15: EduCampus deployment shows the two virtual machines with the main components. For both installations, the dockerized symbIoTe Cloud deployment was used.

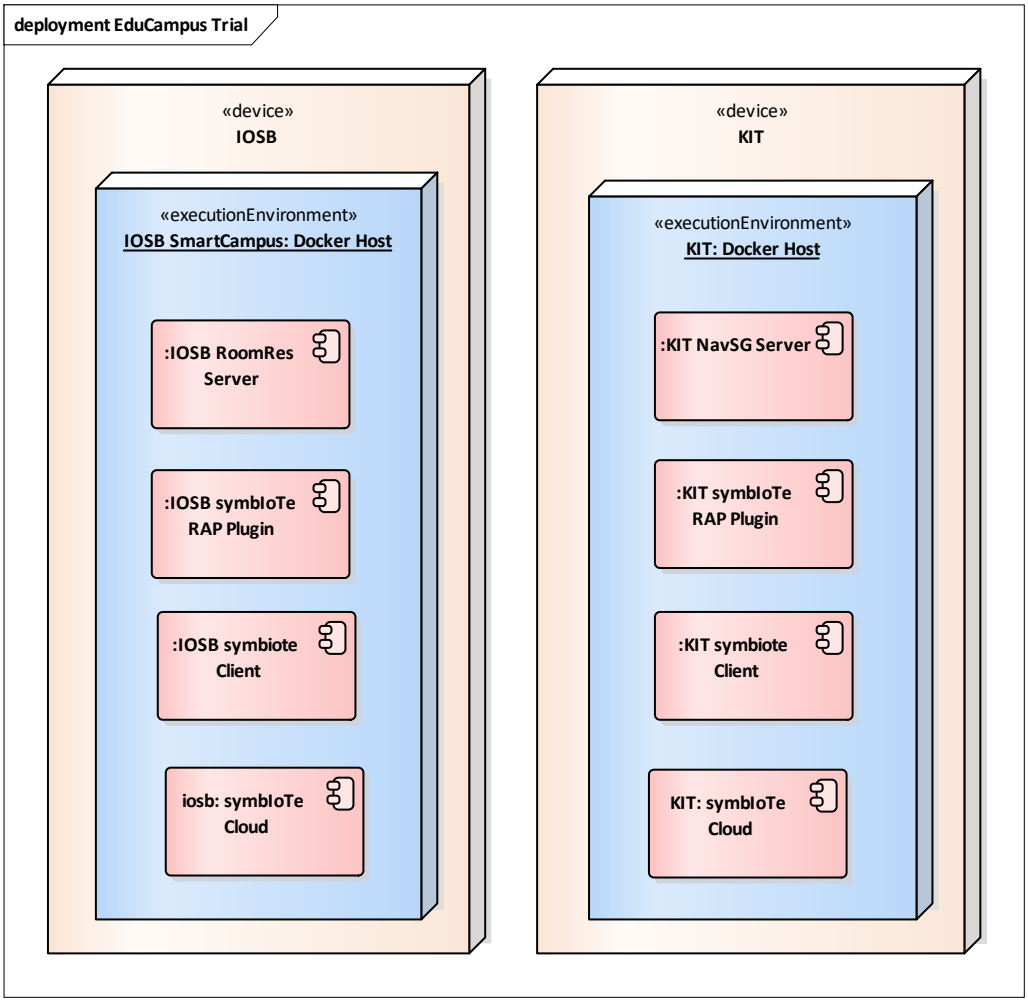


Figure 15: EduCampus deployment

It turned out, that the requirements on the virtual machine in terms of memory and CPU usage, was rather high. The KIT administration suffered from long loading times (up to 30 min) and very high RAM utilization, which caused problems in system response times for the server administration.

The KIT navigation application is based on an Android App, which is shown in Figure 16. It provides information on the current location, which either is derived on GPS location in case the user us outside a building or based on BLE triangulation if the user is inside a building.

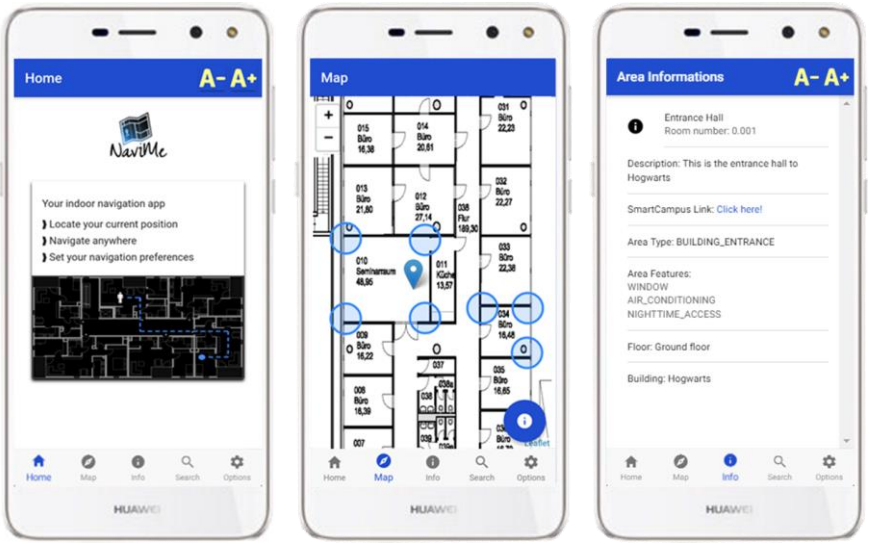


Figure 16: KIT Mobile Navigation Application

The IOSB room reservation application shown in Figure 17 is also based on an Android App. It gives information on nearby rooms, based on BLE signal strength. The reservation feature is based on a calendar that allows reservation of room and the creation of meetings with other users.

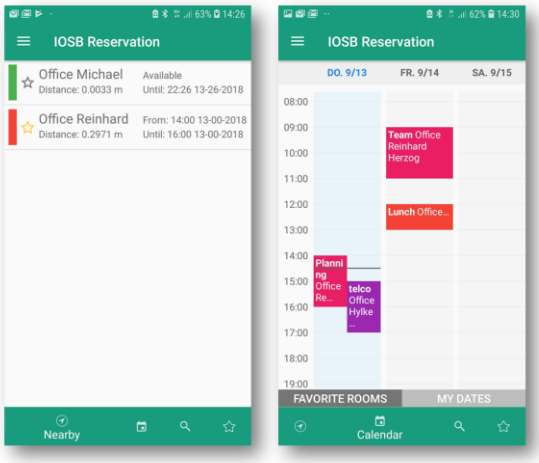


Figure 17: IOSB Mobile Reservation Application

In general, both applications and their backend services are ready to be integrated via symbIoTe but as the needed functionality for semantic mapping in symbIoTe is not released at time of writing they remain separate applications for now.

4.3.3 KPIs

The following functional requirements should be evaluated:

- BLE beacons which are assigned to some geo information shall be discoverable
- A campus platform shall provide information about room on the campus

The following non-functional requirements should be evaluated:

- Transformation between different data models < 50 ms
- Response time of federated resource lookup < 200 ms

4.3.4 Issues & Risks

The platforms involved in the EduCampus use case were defined to use L2 of symbIoTe. Due to time constraints, it was decided to not support Platform-Specific Information Models (PIMs) in L2 and that the use case will be realized using L1. Additional constraints and workload when performing other trials and supporting the Open Call 2 winners, as well as several issues with PIM support on L1, resulted in readapting the plans for the trials of this use case: instead of running the trials, we focused on doing extensive tests so as to evaluate the correctness and performance of the semantic mapping component. The results of these tests are presented in the following section. At time of writing, PIM and semantic mapping support on L1 are implemented and working in a development stage and will be included in the final release of the symbIoTe framework.

4.3.5 Functional tests – Status and Results

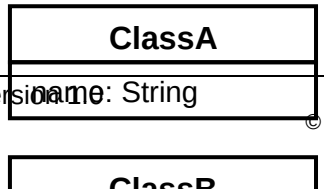
The functional tests were divided into three parts:

- Performance test for data translation using the symbIoTe client API (mocking all external requests)
- Performance test for data transformation using different sizes of data
- Performance test for SPARQL Query rewriting

All tests were executed as a JUnit tests on a laptop with Intel Core i7-4720HQ CPU with 16GB RAM running Win10 Enterprise Edition and Netbeans IDE. Each test was run 50 times. Calculated times are the median execution time in milliseconds with a drop of 20% (i.e. 10% fastest and 10% slowest runs were dropped).

Performance test for data translation using the symbIoTe client API

Goal of this test is to validate functionality of the semantic client library and performance of the semantic mapping in a way it would be used in any real application. All calls to other symbIoTe components (namely Search and ResourceAccessProxy) were mocked. The scenario for the test was as follows. There are two L1 platforms registered in the Core, platformA and platformB. Each of the platform uses its own PIM, i.e. platformA uses modelA and platformB uses modelB. Both PIMs have an overlap in domains, i.e. there are classes in one of the PIMs that semantically align with the other PIM and vice versa. They are referred to as classA and classB thereafter (see Figure 18). Furthermore, a mapping between modelA and modelB is registered in the Core mapping classA to class B and vice versa (see Listing 1 for an example). PlatformB offers a service that takes an instance of classB as parameter and returns an instance of classB, i.e. the method signature looks like service(classB) → classB. Now platformA (or someone else only knowing modelA) wants to use the service but without knowing (and caring), that it uses modelB. Furthermore, the caller of the service wants to use Java classes as in- and output of the service to deliver himself of the burden to manually map/decode them to/from String (which is the only format RAP supports).




```
BASE <http://example.org/A#>
PREFIX b: <http://example.org/B#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

RULE
  CONDITION
    CLASS :ClassA
```

Listing 1:
Mapping from modelA to modelB

```
ClassA result = semanticRapClient.invokeServiceWithMapping(
      RESOURCE_ID_SERVICE,        // symbIoTe ID of service to call
      new ClassA(),               // service input, class must use
                                     provided JSON-LD annotations
      ClassA.class,               // result type, also needs
                                     JSON-LD annotations
      true,                       // server validation (from RAP client)
      HOME_PLATFORM_IDS);         // home platform IDS
```

Listing 2: Example call to symbIoTe semantic client library.

In the following, we explain in detail what steps happen in the semantic client library when executing this function call.

1. The ID of the PIM used for the input parameter is determined by either HOME_PLATFORM_IDS (if possible) or by finding a PIM containing the used semantic classes and properties.
2. The ID of the PIM used for the service is resolved based on the resource ID of the service.
3. Both mappings (forward, i.e. from modelA to modelB, and backward, from modelB to modelA) are fetched from the Core.
4. The actual input (i.e. new ClassA()) is converted to JSON-LD based on annotations in ClassA.
5. The forward mapping is parsed and executed.
6. The actual service is called via RAP from platformB with the result from previous step as payload.
7. The backward mapping is parsed and executed.
8. The resulting JSON-LD is parsed to an instance of ClassA and is returned.

Steps 1 to 3 are executed as SPARQL queries to the Search component. For the test they are mocked, i.e. no real connection to the Search happens but the result is provided locally. The actual resource access in step 6 is also mocked. The semantic mapping library is used in step 5 and 7 for parsing and execution.

Two different scenarios were tested. The simple scenario uses very simple PIMs with only one class and the most trivial mapping (renaming a property). The EduCampus scenario works with the information models prepared in the use case for KIT and IOSB. They contain multiple classes and three mapping rules including a custom JavaScript transformation function.

Table 3: Performance results for the simple scenario (in ms).

find PIMs		Access	Mapping						Overhead	Total
service (by resource ID)	client (by platform ID)	invoke service	client -> service			service -> client				
			get	parse	execute	get	parse	execute		
0,53	0,60	0,08	0,50	0,94	22,84	0,70	0,98	23,40	7,37	58,02
1,13			49,36							

Table 4: Performance results for the EduCampus scenario (in ms).

find PIMs		Access	Mapping						Overhead	Total
service (by resource ID)	client (by platform ID)	invoke service	client -> service			service -> client				
			get	parse	execute	get	parse	execute		
0,46	0,56	0,08	0,50	3,83	58,21	0,66	3,37	46,04	9,76	124,43
1,02			112,61							

There are a few things to keep in mind when interpreting these results. Finding and getting PIMs as well as invoking the service are mocked. This explains why they only take less than 1ms. In reality, such a SPARQL query takes around ~600ms as tested manually. The most interesting results are the time for actually transform the data (highlighted in red) which is roughly ~20ms for the simple scenario and ~50ms for the EduCampus scenario. As a RAP call takes around 800-1200ms (based on manual tests), this is reasonable cheap to be used in real-world scenarios.

In reality, each call to find a PIM as well to get a mapping is a SPARQL query to the Search that takes ~600ms. This has an important impact on performance in reality. Calls to find PIMs are only added for convenience and can be left out (by manually providing the PIM IDs) wherever more performance is needed. The call to get mappings from the Core is essential but caching could be implemented to reduce the costs.

It is to note, that execution time of mappings will increase with the amount of data to transform but the same applies to sending/fetching data to/from RAP. The following figure shows how execution time for executing one-way mappings scale with size of the data.

Performance test for data transformation using different sizes of data

Figure 19 shows how execution time of data transformation increases with size of input data. For each of the two scenario multiples of the original test size is used. It is shown, that for the simple scenario the execution time is almost linear but in fact, it is just increasing very slowly. For the complex mapping, it shows that execution time is scaling linear with the size of the input. These results indicate that semantic mapping through data transformation is feasible in the real world.

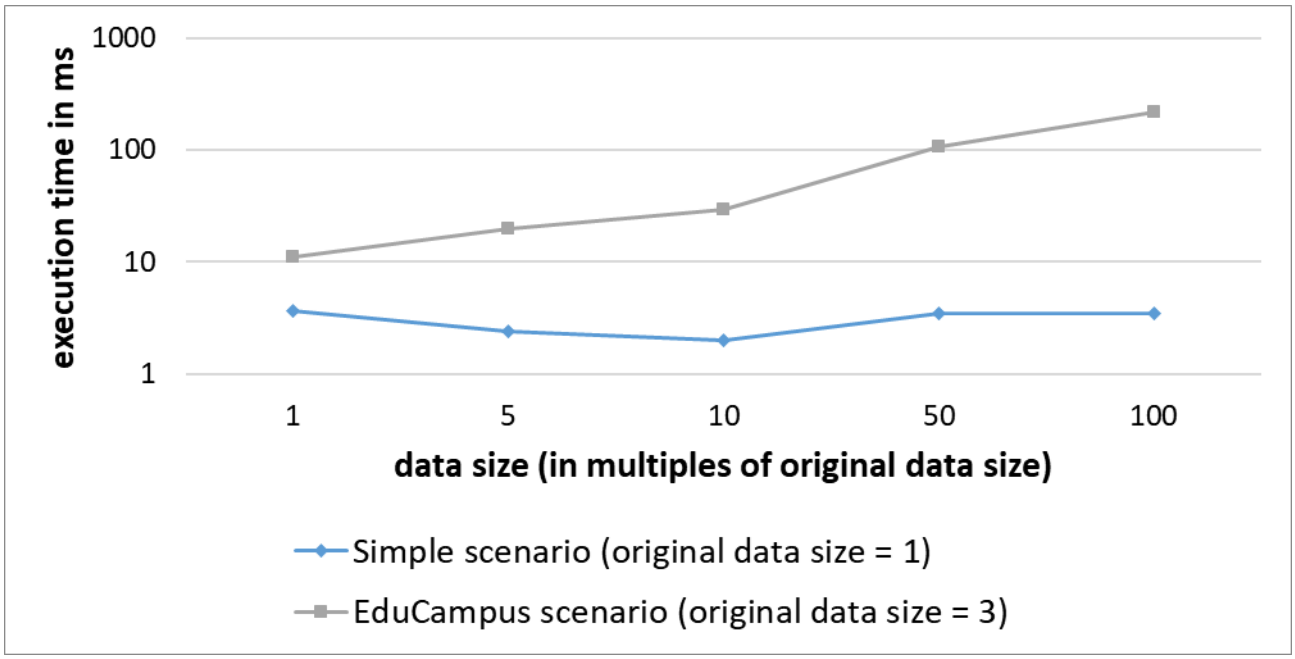


Figure 19: Performance for executing semantic mapping with varying data size.

Performance test for SPARQL Query rewriting

For federated resource lookup, symbIoTe Search needs to identify available mappings and for each mapping found, rewrite the original search query, execute it and merge the results. Tests show, that it takes 10.6 (mean) ± 3,6 (st.dev) ms to do such a query rewriting for the EduCampus scenario. Performance of query rewriting only depends on query and mapping length/complexity. As the rewritten queries can the executed in parallel in symbIoTe Search, this approach seems very promising.

4.3.6 Execution of the trials with users – Status and Results

Trials with users could not be executed, as the integration of the PIM model support was not completed within the project time.

4.3.7 Conclusions

The performance of semantic mapping is good enough to be integrated into real-life systems with adding roughly an overhead between 0.5% (simple scenario) and 2% (EduCampus scenario) to regular RAP access (for data size of 10). Currently, finding used PIMs/mappings, which are just syntactic sugar, introduces overhead. However, with some minor additional work, this can easily be reduced dramatically e.g. by introducing caching.

It is still planned to work on semantic interoperability using mapping in the context of EduCampus as part of the ongoing cooperation between KIT and IOSB. We will also consider using the whole framework or only the semantic mapping library as an MVP.

4.4 Smart Stadium

The Smart Stadium use case aims at showcasing the feasibility, advantages and added value of smart objects for a solution that, in this particular case, is focused on retailer services for the visitors of a stadium. The use case is based on three main applications, the visitor application, the retailer application and the promowall application, all of them discovering and taking advantage of the different symbIoTe enabled smart objects deployed throughout the stadium (remote ordering devices for retailers, symbIoTe-enabled visitor mobile devices and promowall screens).

All smart objects have also indoor location capabilities by means of BLE beacons that have been deployed throughout the stadium. The number of beacons, density, transmission power and indoor location accuracy are adapted in each case to the specific needs of the stadium manager.

Visitors are identified through their smartphones, provided they have installed the corresponding smart stadium visitor app, while remote ordering devices and beacons allow identifying retailers (both moving carts and physical shops). From the visitor's point of view, Smart Stadium brings the opportunity to detect the closest retailers, place orders independent of where they are, or receive products that they bought directly in their seat.

Additionally, retailers can broadcast their offers and promotions to all visitors inside the stadium, or those that are moving near specific areas inside the stadium. Retailers can send their promotions to large SmartTVs (Promowalls) strategically located throughout the stadium.

4.4.1 Definition of the Test cards and Questionnaires

This section lists all designed tests, KPIs and questionnaires to measure the Smart Stadium solution from visitor's point of view.

Test cards are divided into four categories: one per end user application, plus a generic one:

- Generic: SS-TESTCARD-Generic-01
- Visitor application: SS-TESTCARD-VisitorApp-01
- Retailer application: SS-TESTCARD-RetailerApp-01
- Promowall application: SS-TESTCARD-Promowall-01

An interactive questionnaire was accessible through the application, with a few numeric and textual questions. Users were available to answer it only once.

These were the provided questions:

- Are you a partner? Yes/No
- Did you find the application of Atlètic useful? 1-5
- Did you like application design? 1-5
- Is it easy to use? 1-5
- Will you continue using it in the future? 1-5
- Would you recommend the app to other people? 1-5

- Comments. Free text.

4.4.2 Deployment

The Smart Stadium trials took place at the premises of **Atlètic Terrassa Hockey Club** (ATHC, http://athc.cat/), a sports club located just outside the city of Terrassa, about 30 km NW from the city of Barcelona.

Figure 20: Atlètic Terrassa Hockey Club logo

The picture below shows a map of ATHC premises.

Figure 21: Map of ATHC premises

ATHC is currently running a team of field hockey that is participating in the Spanish national league, and as winner of last year Spanish league, is also participating in the European championships, the Euro Hockey League.

It is expected (still to be agreed with ATHC) that some of the demonstrations will take place taking advantage of the most important matches in which ATHC will be playing, in order to involve as many visitors as possible during the trials.

4.4.3 KPIs

There were three groups of KPIs, one per application. They are described below.

Application	Type of testing	KPIs
Visitor Application	Visitor registers with its location Visitor updates location while moving through stadium Searches closest Carts & Shops and gets Offering Places Order	Number of users that have downloaded the app > 100 More than 20% of users answer the questionnaire More than 80% of answers are positive as a general score Percentage of shops showed and ordered by proximity > 80% Percentage of orders successfully completed > 80% Percentage of received promotions from retailers > 80%
Retailer Application	Shop / Cart register with its location Cart updates location while moving through stadium Get new orders Sends promotions	Percentage of orders sent by users and received by retailer > 80% Percentage of orders successfully completed > 80% Percentage of promotions successfully provided to users and promowalls > 80%
Promowall	Display promotions	Percentage of received promotions from retailers > 80%

4.4.4 Issues & Risks

The users participating in the trials were both employees from ATHC (and related third parties such as retailers), and public attending the selected events.

Employees and retailers were using the remote ordering devices, which only deals with commercial information (name of the brand, products offered, or promotions) and do not collect or store any personal data at all.

Smart Stadium features were included in existing ATHC official application, so many potential users could start using it even without marketing effort. Furthermore, ATHC spent resources in make the new app known by supporters so they started downloading and using it.

4.4.5 Functional tests – Status and Results

Test Description	Status	Results
Review Google Play and App Store	Passed	More than new 250 installations

metrics to check how many app installations occurred during trials		joining iOS and Android users.
Review Google Forms answers to determine % of active users that answered it.	Passed	29.07% of users answered the questionnaire.
Review Google Forms answers to determine % of positive scores.	Passed	85.71% of positive scores.
Check number of shops showed and ordered by proximity.	Passed	100% of retailers.
Check number of orders successfully completed.	Passed	100% of completed orders.
Check number of promotions received from retailers.	Passed	100% of received promotions.
Validate the quality of ordering service by checking the amount of orders sent by visitors and the amount of received ones by retailers.	Passed	100% or user orders were received by customers.
Check number of orders successfully completed.	Passed	100% of successfully completed orders.
Check the number of promotions successfully provided to users and Promowalls.	Passed	100% of promotions.
Validate that retailers send promotions to be displayed in Promowalls across the stadium.	Passed	83.33% of received promotions

4.4.6 Execution of the trials with users – Status and Results

Trial took place during a special event in ATHC facilities with several matches from different categories (men and women, child and senior). More than 1000 people met at the stadium and more than 250 of them downloaded ATHC official application.

Application is considered a short-period usage application, as its purpose is to provide sports information, location information and retail information. Due to that, user session lives for a short period of time (2 to 3 minutes average at most).

Concurrency is not a key problem in a regular app usage, unless a promotion is sent to all users and they feel curious to read it as soon as they receive the app.

Beside this, the most challenging feature is outdoor location. Mobile app is programmed to send location updates to the symbIoTe Core services every 30 seconds, which causes resource updates and subsequent updates of the Jena triplestore. The most devices running at the same time, the most overloaded is server. When multiple devices are sending these updates at the same time, the server overloaded.

This was actually the problem we faced during trials. Server experienced an overload and its responses delayed in time, until it was necessary to reboot it.

Core developers needed to implement performance improvements that became available in later release 3.0.0. This can be seen in section 3.3.

Beside these performance issues, users that were able to use symbIoTe-powered features felt impressed because of the power it provided to them, being remote ordering and location the most appreciated new features.

4.4.7 Conclusions

ATHC and its supporters were satisfied enough with the three applications (visitor, remote ordering and promowall) and what symbIoTe provided to them in the form of location and remote ordering features. The visitor mobile app (Atlètic Terrassa Oficial) is available both on Play Store (Android) and App Store (iOS). However, ATHC is a relatively small sports club which cannot maintain the devices (beacons, remote ordering devices and promowalls) available during the whole year. Hence, symbIoTe capabilities are only enabled when the corresponding devices have been deployed in the stadium, a decision that the Stadium Manager will usually take only when a significant enough event takes place in the stadium.

The trials that took place in the stadium faced some performance issues when still using symbIoTe release 2, but we could make a workaround and users could continue using the app, even if with less features. Once release 3 became available, all those issues were resolved and users could take full advantage of the applications.

The overall user and team satisfaction proves that the pilot was a complete success. The trials took place during events attended by more than 1000 people, from which more than 250 downloaded the visitor app. From these users, almost 70 answered the questionnaire, and regardless the performance issues they gave a qualification of more than 85% of positive scores. symbIoTe provided the ability to discover and dynamically access new devices ordered by proximity, as well as known and preregistered ones, and it was used to develop next evolution of customer immersion in stadiums.

4.5 Smart Yachting

Smart Yachting includes two showcases:

- Smart Mooring (SM), whose goal is to automate the mooring workflow through events recognized by sensors and to transfer yacht data to the mooring application (Portnet)
- Automated Supply Chain (ASC), to detect the maintenance and consumable needs of the yacht and to perform an automatic match-making with the suppliers in the port area.

They both exploit data from IoT sensors on board of the yacht; two Enablers (L1) mediate the interactions of the applications with symbIoTe resources.

The yacht is seen as a Smart Roaming Device (L4) accessing the Smart Space of the Port (L3/L4).

4.5.1 Definition of the Test cards and Questionnaires

Two different test cards have been defined for the Smart Yachting use case, one for each of its two showcases. They include the general objective of the tests, the steps to take to perform validations, a description of the software components deployed for the tests and the KPIs to measure. Their complete description is available in two separate documents:

- For Smart Mooring: SY-TESTCARD-Smart_Mooring
- For Automated Supply Chain: SY-TESTCARD-Automated_Supply_Chain

After pilots, users were asked to complete an on-line questionnaire based on Google Form. The target users were the Port operators which were involved in the trials (they also "simulated" the role of Yacht owner, since the Yachts that have been used belong to the Ports).

The Questionnaire consists of 11 questions which center around the users' expectations and experience during the trial. Users were also asked to express their opinions about the possibility offered by the new technologies introduced by symbIoTe. It was possible to collect the feedbacks of 5 out of the 6 users involved in the trial.

4.5.2 Deployment

Smart Yachting's trial took place in two Italian ports, Viareggio and Marina Cala De' Medici, on the Tuscany coast. The execution of the trial has been influenced by the availability of the symbIoTe Smart Space Middleware (S3M), since Smart Yachting was a Level 3/Level 4 compliant use case and relied on this component. General validation started in late Spring/early Summer 2018, initially assuming that the Yacht was a "normal" L1 platform, and continued later on, when S3M has been released, till the month of October 2018.

From a technically viewpoint, each port exposed a S3M: their applications – Navigo's Portnet and Centrale Acquisti – were integrated into the symbIoTe ecosystem through two dedicated Enablers. For simplicity's sake, the same deployment – shown in the diagram below – has been used for the two Ports, with minimal reconfigurations when the trial was performed in the second harbour. This did not influence the generality of the trial, since all the tests of the use case involve only a single Port at the time.

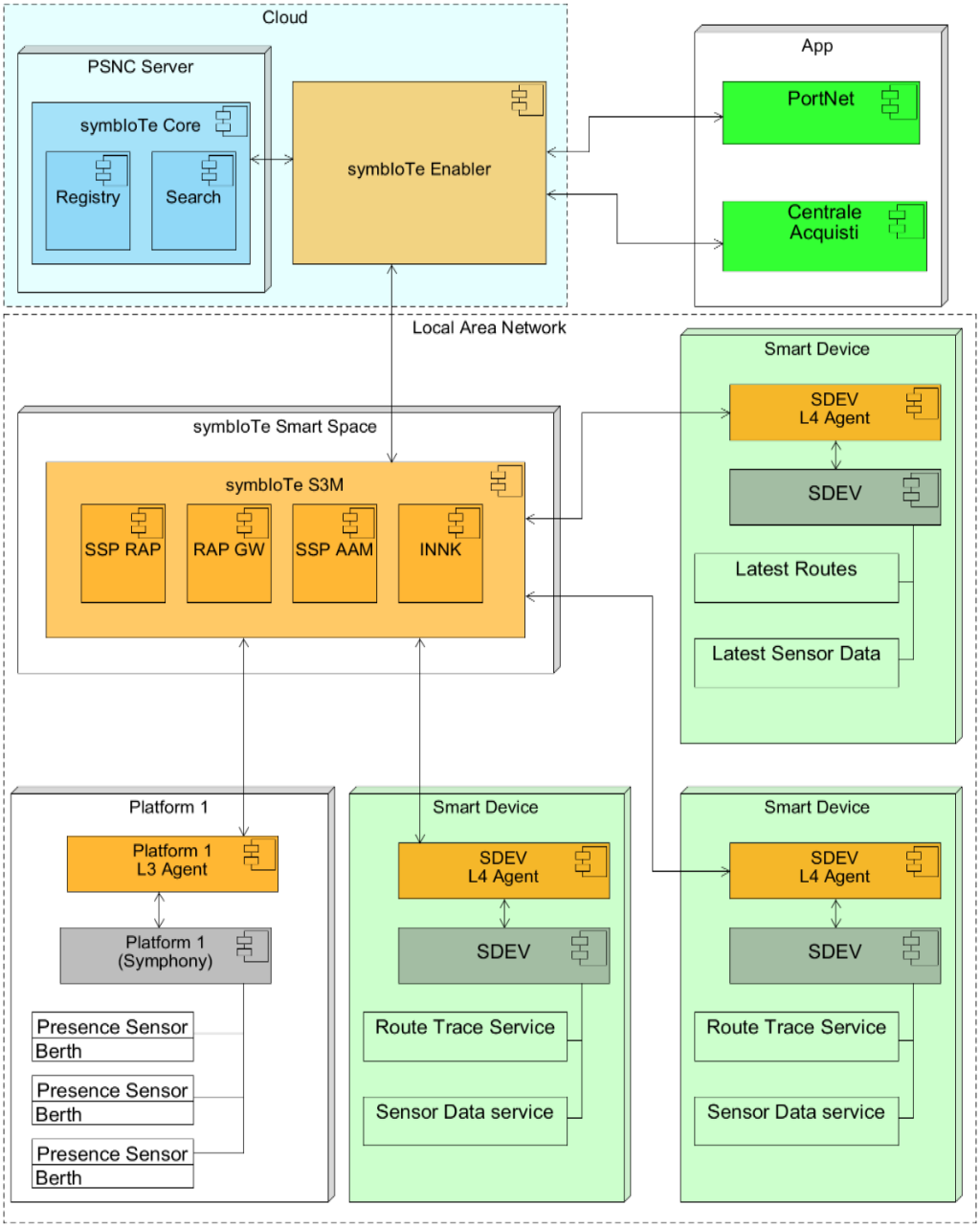


Figure 22: Smart Yachting trial deployment diagram

Separate set-ups, for each of the two showcases foreseen for the use case – Smart Mooring and Automated Supply Chain (ASC) – have been arranged.

For Smart Mooring the trial involved a real yacht to navigate towards Viareggio and berth in its port, while the Portnet mooring management application can automatically advance its workflow through the acquisition of data from several sensors.

The assessment involved:

- one Yacht

- 2 Port operators, one in the office, and one on the pier. The former supervised the correct advancement of the mooring workflow in the Portnet App, while the latter was expecting the Yacht arrival on the destination Pier and could testify when it finally berthed (in order to verify if the presence sensor correctly detected the arrival of the boat).

The set-up included a LoRaWAN antenna installed in the Viareggio port, a LoRaWAN tag and various machine sensors on board of the yacht and LoRaWAN presence sensors on the pier. Wi-Fi connectivity was also available for the berthing Yacht.

Automated Supply Chain required a simpler set-up, since its aim was to automatically identify the maintenance and supply needs on board of a Yacht, berthed on the Port and connected to the Internet via the Port's Wi-Fi. Services were exposed by Navigo's Centrale Acquisti market-place web application, where the services of 10 local companies are offered.

We arranged two separate trials, one in Viareggio and one in Marina Cala De' Medici, each one with a Yacht and two users (a Yacht owner and a Centrale Acquisti operator).

4.5.3 KPIs

Separate KPIs have been defined for the two showcases, as described in the table below.

Type of KPIs	Description	Target Value
Smart Mooring	Approval rating regarding the ease of use of the solutions prepared for Smart Yachting, as experienced by port personnel	> 90%
	Average global time spent by the port authority operators in managing a mooring procedure through symbIoTe	< 5 min (at present for the manual procedure it's more than 30 min)
	Percentage of data automatically acquired from the boat respect those managed manually	100%
	Percentage of extra data automatically acquired from the boat	> 50%
	Approval rating regarding the ease of use of the solutions prepared for Smart Yachting, as experienced by port personnel (as assessed through the questionnaires)	>90%
Automated Supply Chain	Number of possible suppliers to fulfil the yachts needs in the two ports	At least 10
	Percentage of the most common	>60%

	resupply yacht's needs that can be mapped through IoT sensors	

4.5.4 Issues & Risks

Smart Yachting was not an easy use case to assess, since for its actual execution a lot of different things had to be prepared.

First and foremost, the availability of a working implementation of S3M in late Summer pushed forward the timing initially foreseen for the trial. This wasn't a major issue anyway, since earlier technical tests were performed assuming the Yacht was a "normal" symbIoTe L1-compliant platform. This allowed the team to identify all the issues regarding the development of the Enablers and the actual interactions amongst the Yacht's IoT platform and the Navigo's Applications.

The yachts in the two ports had to be booked in advance and especially during touristic season their availability was very limited. In Summer also the personnel of the two Ports was often too busy to be involved in the trial. This forced to reschedule the final tests of the trial in late October.

Despite these issues, the Smart Yachting trial was successfully completed.

4.5.5 Functional tests – Status and Results

Several tests were carried out in order to validate the technical components of Smart Yachting. Some of the most significant validations carried out for the trial follow.

For Smart Mooring

Test Description	Status	Results
The mooring procedure is started by PortNet APP. PortNet APP expects a specific Smart Device registration over port SSP and it contacts the Enabler, which in turn starts polling periodically the Core for the Smart Device to connect to the SSP and register the services: • Latest routes • Latest sensor data	Passed	After Mooring procedure is started in the PortNet APP, the port SSP starts to receive periodic requests for Smart Device presence through its enabler.
As soon as the Smart Device is connected to the Port Wi-Fi (meaning that the Yacht has reached the harbour) it automatically starts the registration of two services: • Latest routes • Latest sensor data In order to verify this the SSP innkeeper is checked through a REST GET.	Passed	The Smart Device services are correctly registered and the Latest routes and Latest sensor data services are available and can be read by the enabler.
The innkeeper has also to register the new services on symbIoTe core. The	Passed	All Smart Device services are correctly registered in the core, they

core status is controlled through the WebApp.		are also searchable and their information accessible.
The Enabler starts polling for the services, and when it receives data from the Latest routes and Latest sensor data services, it informs the PortNet APP with the newly read data.	Passed	The Enabler received data and updated the PortNet APP with the information read
PortNet APP starts polling the proximity sensor on the berth to detect when the approaching yacht has finally docked	Passed	Proximity sensor status received in PortNet via the Enabler
Once the proximity sensor detects the presence of the expected yacht to the right berth, PortNet APP closes the mooring procedure.	Passed	Proximity sensor triggered and PortNet updated the workflow status

For Automated Supply Chain

Test Description	Status	Results
The procedure is started in the Automated Supply Chain APP by the yacht owner. Automated Supply Chain APP offers a set of goods and services suppliers which the yacht owner can contact for his/her needs and get offers on the requested goods or services. The Yacht (Smart Device) is contacted by the Automated Supply Chain (via the Enabler) and communicates its readings for consumable and/or maintenance needs. The APP shows the yacht owner the data gathered and prepares offer requests to be sent to the suppliers and get offers for them.	Passed	All Yacht's needs are translated into offer requests and correctly registered in the APP. The offer requests have been correctly received by the suppliers.
The IoT platform on board successfully detect the resupply needs on board through sensors	Passed	In the trial the following supply/maintenance needs were detected: • Fuel • Engine faults • Electric faults. Since the yachts used for the trial have limited electronics on board (they were both below 15 mts and quite old), these were the only

		possible needs that could be detected for the trial.

4.5.6 Execution of the trials with users – Status and Results

The following tests were carried out in order to evaluate the functionality of the application and the user experience.

For Smart Mooring

Test Description	Status	Results
Ease of use: the Portnet application, already known by the operators, have been extended to support the use case	Passed	The users didn't need any specific training because the general workflow of the App was respected.
Savings of time: total duration of the workflow is reduced by the introduction of the Smart Mooring functionality	Passed	The tests were repeated twice. The total timing spent for each use case by the operator has been of around 5 minutes, with a significant improvement over the current totally manual procedure.
Accuracy: all the steps of the workflow are taken in the correct order	Passed	All the steps in the Portnet workflow are correctly activated, as the Yacht arrives in the Port.

For Automated Supply Chain

Test Description	Status	Results
Ease of use: the Yacht owners should easily understand how to activate the needs detection	Passed	The Automated Supply Chain application was extended with an intuitive interface. The users involved in the trial didn't need any training nor documentation to use it.
Savings of time: the functionality must significantly reduce the time spent by the yacht owner to deal with local suppliers	Passed	Any need detected on board automatically generated a Request for offer at the right supplier.
Accuracy: As the yacht owner receives offers from the suppliers, he/she can arrange an agreement with the ones he/she chooses and make orders. The suppliers receive the offer requests and reply to them with their offer. The communication goes back and forth between the yacht owner and the suppliers until an agreement is reached	Passed	The offer requests generated are correctly received by the suppliers, which can make offers; the yacht owners are able to create orders out of them.

4.5.7 Conclusions

The Smart Yachting trial aimed to validate the vision of seeing a touristic port as a symbIoTe's Smart Space and the Yachts in transit as Roaming Smart Devices. All the Port's applications can interact with the symbIoTe ecosystem through enablers: in the project two specific kinds of software systems – the Mooring Workflow Management System and the Port Marketplace – have been considered.

Feedbacks from users, both in the form of answers to the questionnaire, feedbacks from functional testing and KPIs analysis, have been all positive.

The limits of the trial are in the use case inherent complexity, that doesn't allow to involve a significant number of users (e.g. Ports, Port operators, Yacht owners, Port Application developers, etc.).

Even with this relatively simplified set-up (that in any case involved 2 Ports, 2 Yachts, 2 Applications and 6 operators) the results of the trial confirmed the soundness of the vision described above, that of course can become commercially viable once a significant number of Ports and Yacht manufacturers embrace this approach and technology.

5 Conclusions

Throughout this document, the performance of the various trials for each use case were described, as well as all the blocks in the roads that impacted their performance. It was possible to compare the results with the expected outcomes, and evaluate the symbIoTe system in different situations.

Initially, performance problems arose with the Smart Stadium trials, where the Core Services weren't able to handle the workload. However, this was fixed, and the Smart Mobility trials were able to prove that the system can handle a large number of users using it at the same time.

The diverse number of trials provided different types of challenges for the system. While the Smart Residence use case trials were performed in contained locations, trials such as Smart Mobility operated through the whole cities. Different levels of integration were tested under different domains, with the Smart Residence and Smart Yachting putting to test the Smart Spaces' components. Semantic Mapping was put to test in order to provide another tool to improve platforms' interoperability.

All in all, the system responded satisfactorily to the workload put upon it, and various symbIoTe functionalities were tested successfully. Not only was the system validated, but the various specific enablers and applications developed for each use case were also put to test, with good results.

Although there is still room for improvement (e.g. regarding the performance of the system) the trials showed how symbIoTe can support real-life use cases in various domains and how this can improve the flexibility of the applications and increase the utilization of existing IoT environments.

6 References

[1] symbIoTe project Deliverable D9.1 – POPD Requirement No.2; February 2016.

[2] Brooke, John. "SUS-A quick and dirty usability scale." *Usability evaluation in industry* 189.194 (1996): 4-7.

7 Abbreviations

AAL	Ambient Assisted Living
GA	Grant Agreement
HCI	Human-Computer-Interaction
ICT	Information and Communications Technology
IoE	Internet of Everything
IoT	Internet of Things
KPI	Key Performance Indicator
POPD	Protection of Personal Data
QoE	Quality of Experience
S3M	symbIoTe Smart Space Middleware
SSP	symbIoTe Smart Space
WTP	Willingness-To-Pay

8 Annexes

8.1 Smart Healthy Indoor Air

Test-card #	Symbyote-TestCard-SHIA-Integration	Execution Status	*Partial PASSED*
Test Name	Registration of resources from nAssist platform and use of interpolation function from symbIoTe core		
Objectives	The aim of these tests is to verify the correct registration of all the indoor and outdoor resources and locations provided by the nAssist platform L1 compliant to symbIoTe architecture by using the symbiote.man.poznan.pl server. • The platform registers a set of sensors located in 7 different outdoor fixed stations in Barcelona: ○ GenCat Ciutadella ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat Eixample ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat Gracia ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat Palau Reial ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat PobleNou ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat Sants ▪ O3 ▪ NO2 ▪ PM10 ○ GenCat Vall Hebron ▪ O3 ▪ NO2 ▪ PM10 • Also, it registers a set of sensors located inside 5 different houses and three values interpolated from the interpolation function (O3, NO2, PM10) taking into consideration the GPS coordinates of each house: ○ symbiote 1 (Madrazo) ▪ Indoor CO2 ▪ Indoor temperature		

- Indoor humidity
- Energy consumption
- Brightness
- O3
- NO2
- PM10

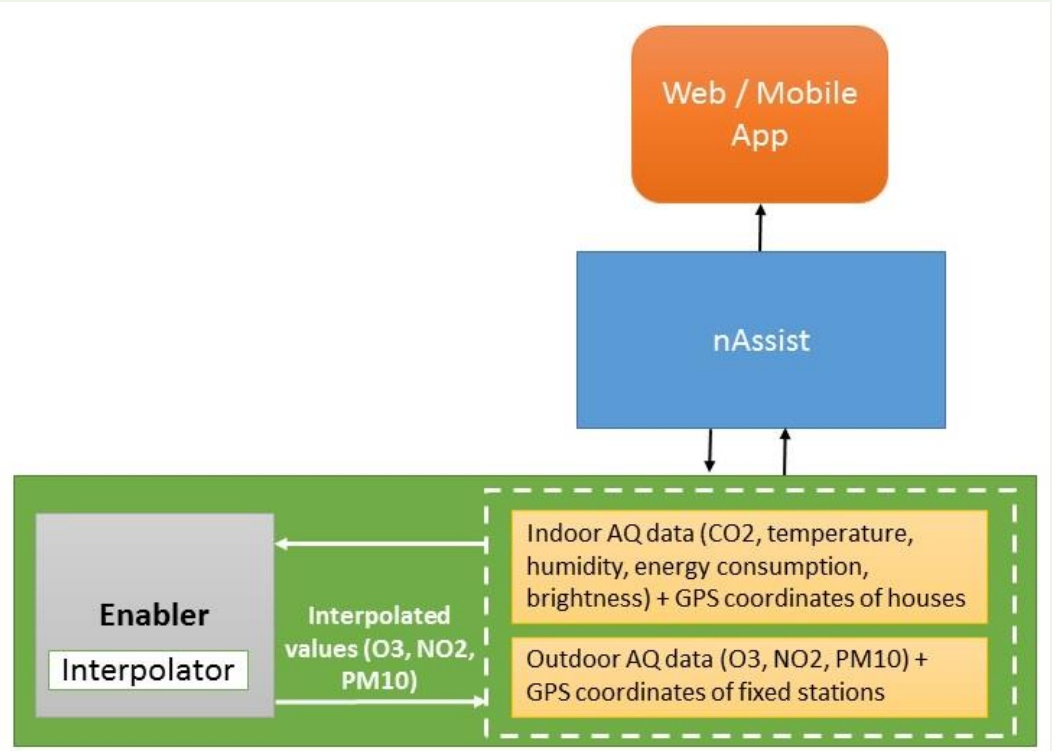
- symbiote 2 (Office dev)
  - Indoor CO2
  - Indoor temperature
  - Indoor humidity
  - Energy consumption
  - Brightness
  - O3
  - NO2
  - PM10
- symbiote 3 (Office usb)
  - Indoor CO2
  - Indoor temperature
  - Indoor humidity
  - Energy consumption
  - Brightness
  - O3
  - NO2
  - PM10
- symbiote 4 (Santa Rosa)
  - Indoor CO2
  - Indoor temperature
  - Indoor humidity
  - Energy consumption
  - Brightness
  - O3
  - NO2
  - PM10
- symbiote 6 (Ursula 4)
  - Indoor CO2
  - Indoor temperature
  - Indoor humidity
  - Energy consumption
  - Brightness
  - O3
  - NO2
  - PM10

- An some actuators from installations symbiote 1 and symbiote 2:
  - symbiote 1 (Madrazo)
    - Tower Heater
    - TV
  - symbiote 2 (Office dev)
    - Wall plug

The objective of the test is to verify the correct operation of the basic registration process

	and the validation of the interpolation function from the Enabler by giving the values of the fixed outdoor stations, their GPS coordinates and the houses' GPS coordinates.		
KPIs to measure	DESCRIPTION	PROCEDURE	TARGET VALUE
	Seven resources registered by each indoor installation (five houses)	1. Create a symbiote client 2. Get the guest token CORE_URL/query?name=*	35 resources registered
	Three resources registered by seven outdoor fixed stations	1. Create a symbiote client 2. Get the guest token CORE_URL/query?name=*	21 resources registered
	Interpolated values for each installation (O3, NO2 and PM10) being updated hourly	https://symbiote.encontrol.io/rap/Sensor/" + sensorId (symbiote.encontrol.io is the nAssist's URL)	Values from 15 resources being updated hourly from enabler
	Generation of recommended ventilation strategies when indoor air quality is low	The rule engine detects that the indoor air quality is lower than a threshold, then notification system creates a message with a recommended ventilation strategy	Creation of messages by the notification system
	Values from all sensors are being updated	https://symbiote.encontrol.io/rap/Sensor/" + sensorId + "/history"	Updated values hourly
Related Use Cases	Smart Healthy Indoor Air (Smart Residence)		
Designers	*Raquel Ventura, Pablo Giménez, Adrián Cañadas*		
Related Test-cards	SYMBIOTE-TESTSCARD-SHIA-Application		
Attachments	None		
Additional Comments	This Test Card validates the registration of the nAssist platform at L1 compliance that includes the registration of the resources necessary for this specific use case and the use of the interpolation function from the Enabler.		

Trial resources and topology	
Required resources	O3, NO2 and PM10 sensors from 7 different fixed stations and the following indoor sensors for 5 houses: CO2, temperature, humidity, brightness. Also, the interpolation function from the Enabler will provide interpolated values for O3, NO2 and PM10 for a specific house locations given the GPS coordinates of the fixed stations.

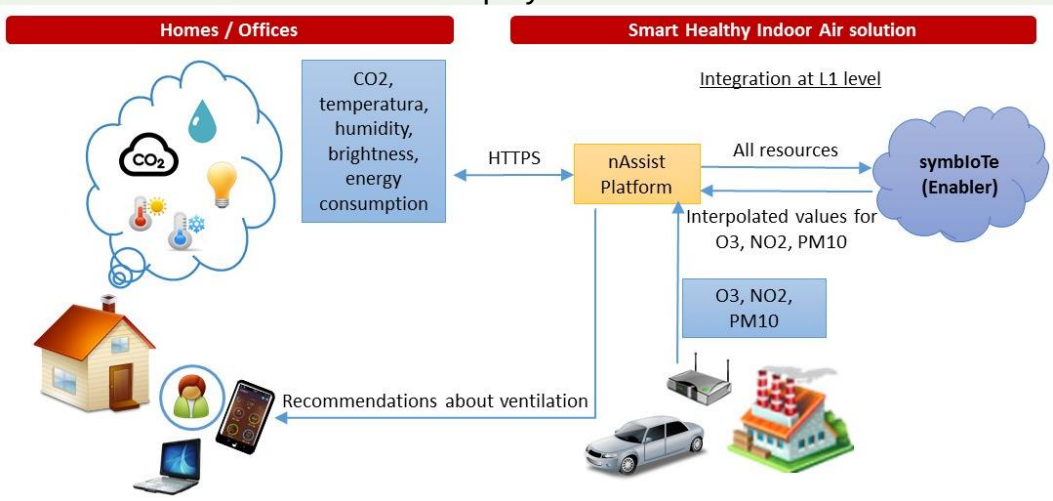
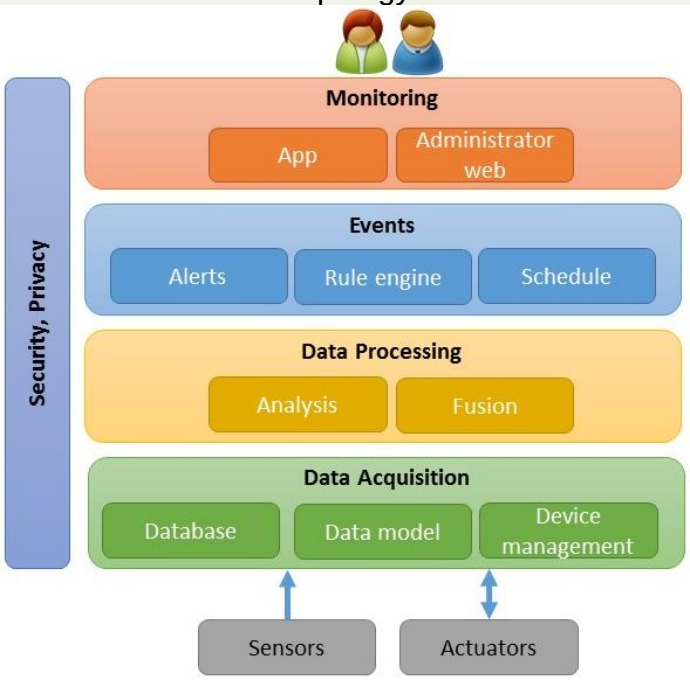
Topology details	

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Registration of the nAssist platform **Expected Results:** • Successful registration process **Measured Results:** • nAssist platform listed **Comments**: The registration of the nAssist platform has been carried out on the server symbiote.man.poznan.pl	*Passed*
2	**Description**: Registration of the resources **Expected Results:** • All resources are correctly registered, searchable and their information accessible **Measured Results:** A total of 64 resources were registered: • Three sensors (O3, NO2, PM10) for 7 fixed stations	*Passed*

	• Eight sensors (CO2, temperature, humidity, energy consumption, brightness, O3, NO2, PM10) for 5 houses • Three actuators (tower heater, TV, wall plug) for 2 houses **Comments**: The registration of the nAssist resources has been carried out on the server symbiote.man.poznan.pl	
3	**Description**: Interpolation function updates hourly values for O3, NO2, PM10 for each house by taking into consideration the GPS coordinates and values of the fixed stations and GPS coordinates of the houses **Expected Results:** • Interpolated values calculated by the interpolation function from the Enabler updating the sensors O3, NO2, PM10 for each house **Measured Results:** • Interpolation function finds fixed stations through a search in the core • Checking manually the output of the interpolation function **Comments**:	*Failed*
4	**Description**: Data exchange between the interpolation function and nAssist platform. Resources are required to be found through a search in the core **Expected Results:** • Visualisation of interpolated data from the user application and use of these data from the Rule-based Reasoning (RBR) engine **Measured Results:** • Interpolation function finds fixed stations through a search in the core • Output from the interpolation function matches with the data visualised in the user application **Comments**:	*Failed*

Test-card #	SYMBIOTE-TESTCARD-SHIA-Application	Execution Status	*PASSED*
Test Name	Smart Residence – Smart healthy indoor air		
Objectives	Verify the correct functioning of the user application. The application consists of: • Back-end running on the nAssist platform to manage devices, store data, events, rule-based reasoning engine, and notification system. • Front-end accessible via web and mobile app for monitoring data, sensors states, configuration and recommendations.		
KPIs to measure	DESCRIPTION	PROCEDURE	TARGET VALUE
	State of sensors	Device management functionality	On-state for all sensors and actuators
	Connectivity of the gateway	The ping command	Ping received every 4 minutes
	State of the database	Access and monitoring of the database in the cloud	All resources stored and values updated according to the sending frequency for each type of sensor
	Functioning of the rule engine	Monitoring and configuration of the rule engine through the rules functionality	Recommendations generated when indoor air quality level is bad
	Functioning of the notification system	Monitoring of the notification system through the application	Recommendations visualized in the user application
	System usability assessment	Two questionnaires completed after trials for each user	Usability assessment above 68 according to the SUS tool and two or more affirmative answers to indicate that the system can increase the quality of life.
Related Use Cases	Smart Residence – Smart Healthy Indoor Air		
Designers	Raquel Ventura, Pablo Giménez, Adrián Cañadas		
Related Test-cards	SYMBIOTE-TESTCARD-SHIA-Integration		
Attachments	User questionnaires: Quality of Life and System Usability Scale.		
Additional Comments	This user application monitors the data acquired by sensors at indoor location and the output air quality given by the interpolated data of O3, NO2 and PM10. The interpolation function is provided by the Enabler by taking into consideration the values of the sensors O3, NO2 and PM10 provided by fixed stations located in different places in Barcelona and their GPS coordinates and GPS coordinates of the user's home.		

Trial resources and topology

Required resources	• O3, NO2 and PM10 sensors from 7 different fixed stations and the following indoor sensors for 5 houses: CO2, temperature, humidity, brightness. Also, the interpolation function from the Enabler will provide interpolated values for O3, NO2 and PM10 for a specific house locations given the GPS coordinates of the fixed stations.
Topology details	Deployment  Topology 

Test description

Step #	Step description and expected results	Status

1	**Description**: The objective of this test is to guarantee the correct functioning of sensors and actuators through the application. **Expected Results:** All sensors and actuators states are on and all values are updated according the sending frequency configuration of each sensor. **Measured Results:** Monitoring of the list of sensors and actuators through the Devices functionality. **Comments**: These results are expected for each installation, both houses and fixed stations.	Passed
2	**Description**: The objective of this test is to guarantee the correct functioning of the gateway. The gateway sends a ping to the cloud every 4 minutes to indicate that it is working fine. If there is not any ping received after 10 minutes, the notification system sends an alert. **Expected Results:** A ping command sent from the gateway every 4 minutes. **Measured Results:** A ping command sent from the gateway every 4 minutes. When the gateway loses its connectivity, an automatic notification is sent to the application. **Comments**: The user will be notified of the loss of connectivity by a message through the application.	Passed
3	**Description**: This test is to validate the correct storage of the data coming from sensors in the database of nAssist platform. **Expected Results:** Database updated with values from all resources. **Measured Results:** Database updated with values from all resources. **Comments**:	Passed
4	**Description**: The objective of this test is to validate the correct functioning of the rule based reasoning engine. The rule engine generates the recommendation to the user when the indoor air quality is not good according to some rules. **Expected Results:** Generation of a recommended ventilation strategy when the indoor air quality is bad. **Measured Results:** Generation of recommendations when the indoor air quality was modified manually to check the rule engine functionality. **Comments**:	Passed
5	**Description**: The objective of this test is to validate the notification system which will send a recommendation to the user through the application when the rule engine detects a bad indoor air quality. **Expected Results:** A recommendation about ventilation is sent to the Notification functionality according to the rule base output.	Passed

	Measured Results: Recommendations received by the user when a bad indoor air quality is detected. **Comments**:	
6	**Description**: This test is based on measuring the user acceptability. Two different questionnaires were provided to the user to be completed after the trial. One questionnaire is for measuring the usability of the system (System Usability Scale, SUS) and the other one is to measure the effectiveness of the recommendations to improve the user's quality of life. **Expected Results:** Five users from the different houses/offices are expected to complete two questionnaires: (i) a 10 item questionnaire with five response options, from strongly agree to strongly disagree, and (ii) dichotomous questions about the reduction of the most common symptoms of indoor air pollution after using the system. **Measured Results:** • The SUS score was calculated for each user. The scores have a range of 0 to 100. An SUS score above 68 is considered above average and, therefore, a good subjective measure of usability. • Two or more affirmative answers from the dichotomous questionnaire indicates that the system helps to improve the quality of life. **Comments**: To calculate the SUS score, first sum the score contribution from each item. For items 1,3,5,7 and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Then, the sum of the scores is multiplied by 2.5 to obtain the overall value of SU.	Passed

	KPI DESCRIPTION	ACTUAL VALUE	TARGET VALUE
Results	State of sensors	All sensors' state on	All sensors' state on
	Connectivity of the gateway	Gateway with connectivity	Gateway with connectivity
	State of the database	Updated database	Updated database
	Functioning of the rule engine	Rule engine working	Rule engine working
	Functioning of the notification system	Notification system working	Notification system working
	System usability assessment	Three users with SUS score above 68 and four users with two or more affirmative answers	Above 68 for SUS score and two or more affirmative answer about the reduction of the health symptoms

8.2 Smart Home

Platform Registration

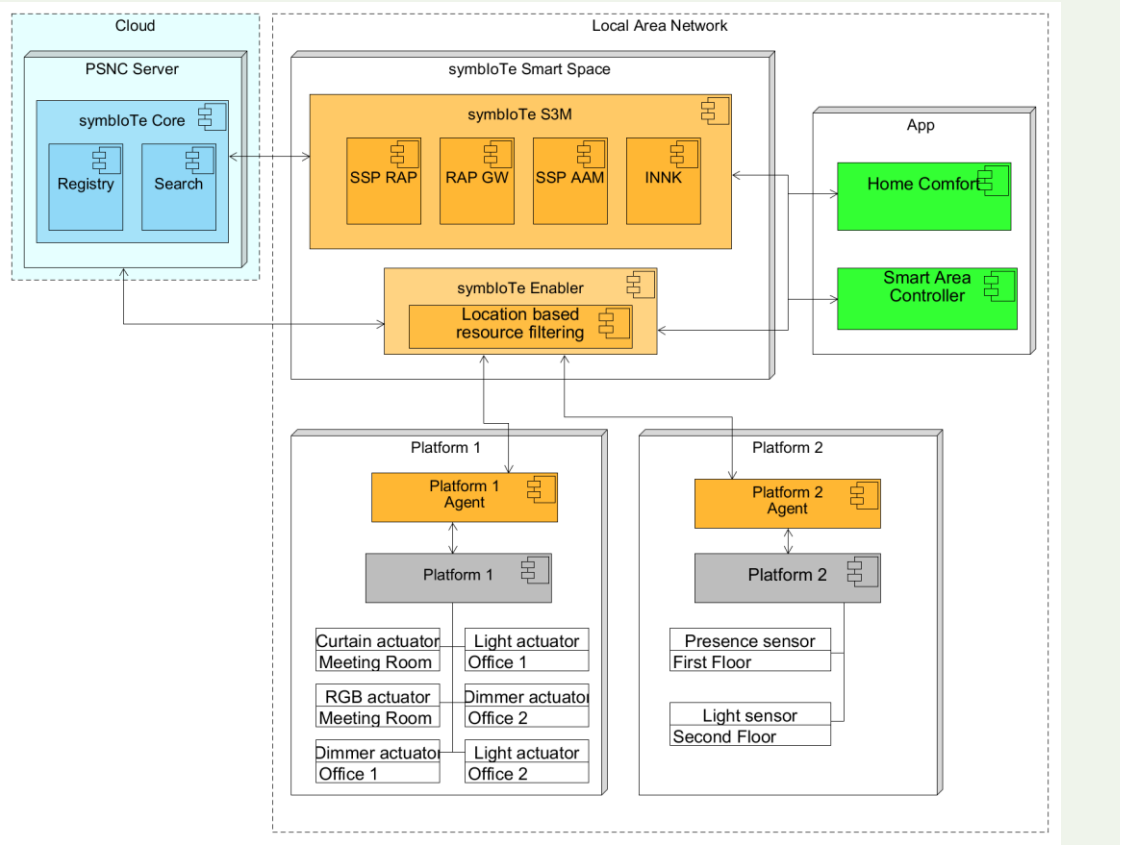
Test-card #	Symbyote_SR_NXW_1	Execution Status	*PASSED*
Test Name	Registration of resources and locations from two IoT platforms		
Objectives	The aim of these tests is to verify the correct registration of all the SSP resources and locations provided by two IoT platforms L3 compliant to symbIoTe architecture. For the test, two separated platforms are used which register resources and locations on the same SSP: • Platform 1 registers a set of actuators located into rooms: o #1 Meeting room ▪ #1 RGB actuator ▪ #1 Curtain actuator o #1 Office 1 ▪ #1 Dimmer actuator ▪ #1 Light actuator o #1 Office 2 ▪ #1 Dimmer actuator ▪ #1 Light actuator • Platform 2 registers a set of sensors located into floors: o #1 First floor ▪ #1 Presence sensor o #1 Second Floor ▪ #1 Light sensor The resources are divided this way in order to emulate the behaviour of two different Platforms controlling different kind of devices and organizing locations on different levels: Platform 1 controls actuator devices and organizes them into locations modelled as Rooms. Platform 2 instead, controls sensor devices and organizes them into locations modelled as Floors.		

	The definition of the topology of the environment and the consequential hierarchy between locations is set by the Enabler. The objective of the test is to verify the correct operation of the basic registration process and the feasibility of interoperability between resources registered by separated platforms. One further objective is to verify the capability of the symbIoTe Enabler of correctly managing locations when a hierarchical tree is created.		
KPIs to measure	DESCRIPTION	PROCEDURE	TARGET VALUE
	Correctly registered resources on SSP	*Rest call to the Innkeeper*	#2 Dimmer actuator #1 RGB actuator #1 Curtain actuator #2 Light actuator #1 Presence sensor #1 Light sensor
	Correctly registered locations on SSP	*Rest call to the Enabler*	#1 Meeting room #1 Office 1 #1 Office 2 #1 First floor #1 Second Floor
	Correctly registered resources on Core	*Check via WebApp*	*#2 Dimmer actuator* *#1 RGB actuator* *#1 Curtain actuator* *#2 Light actuator* *#1 Presence sensor* *#1 Light sensor*
	Correctly registered locations on Core	*Check via WebApp*	*#1 Meeting room* *#1 Office 1* *#1 Office 2* *#1 First floor* *#1 Second Floor*
	Correct hierarchical registration of locations	*Rest call to the Enabler*	*First Floor -> Meeting Room* *Second Floor -> Office 1, Office 2*
	Correct access to resources	*Check via WebApp*	*#2 Dimmer actuator* *#1 RGB actuator* *#1 Curtain actuator* *#2 Light actuator* *#1 Presence sensor* *#1 Light sensor*
Related Use Cases	Smart Residence		
Designers	*Gino Carrozzo, Matteo Pardi, Luca Tomaselli, Lorenzo Neri*		
Related Test-cards	Symbyote_SR_NXW_2, Symbyote_SR_NXW_3		

Attachments	None
Additional Comments	This Test card refers to the preliminary work done to set up the symbIoTe environment (based on two L3 Platforms registered on the same SSP) on which the Home Comfort and Smart Area Controller app are tested. These test aren't meant to be run by users and are illustrated in this Test Card to show the necessary workflow to have Home Comfort and Smart Area Controller running.

Trial resources and topology	

Required resources	Symphony L3 Agent on Platform 1 (Symphony);
	Symphony L3 Agent on Platform 2 (Symphony);
	Controlled devices over both Platforms;
	• Platform 1 resources:
	  ○ #2 Dimmer actuator
	  ○ #1 RGB actuator
	  ○ #1 Curtain actuator
	  ○ #2 Light actuator
	• Platform 1 locations:
	  ○ #1 Meeting room
	  ○ #1 Office 1
	  ○ #1 Office 2
	• Platform 2 resources:
	  ○ #1 Presence sensor
	  ○ #1 Light sensor
	• Platform 2 locations:
	  ○ #1 First floor
	  ○ #1 Second Floor
	
	Home Comfort Application
	Smart Area Controller Application

Topology details	

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Registration of the first Platform (Symphony): **Expected Results:** • Successful registration process **Measured Results:** • No errors during registration process in symbIoTe Agent logs **Comments**:	*Passed*
1.1	**Description**: Inspect registered resources and locations on SSP from the first Platform (Symphony) **Expected Results:** • All Platform 1 resources are correctly registered, searchable and their	*100% Passed*

	information accessible: ○ #2 Dimmer actuator ○ #1 RGB actuator ○ #1 Curtain actuator ○ #2 Light actuator • All Platform 1 locations are correctly registered, searchable and their information accessible: ○ #1 Meeting room ○ #1 Office 1 ○ #1 Office 2 **Measured Results:** • Search and retrieve information for Platform 1 resources via Rest call to SSP: ○ #2 Dimmer actuator ○ #1 RGB actuator ○ #1 Curtain actuator ○ #2 Light actuator • Search and retrieve information for Platform 1 locations via Rest call to SSP: ○ #1 Meeting room ○ #1 Office 1 ○ #1 Office 2 **Comments**:	
1.2	**Description**: Inspect registered resources and locations on symbIoTe Core **Expected Results:** • All Platform 1 resources are correctly registered, searchable and their information accessible: ○ #2 Dimmer actuator ○ #1 RGB actuator ○ #1 Curtain actuator ○ #2 Light actuator	*100% passed*

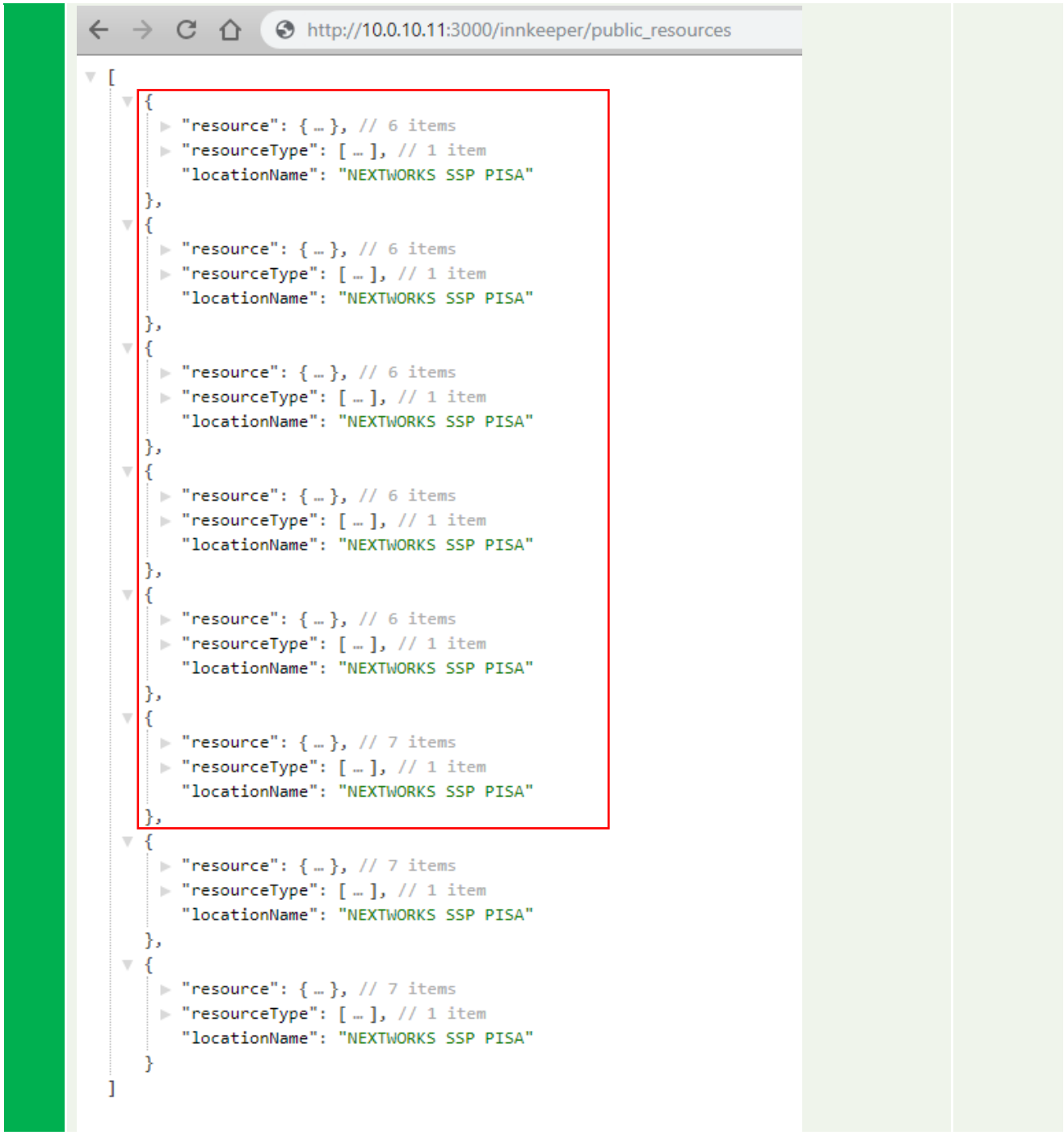
- All Platform 1 locations are correctly registered, searchable and their information accessible:
    - #1 Meeting room
    - #1 Office 1
    - #1 Office 2

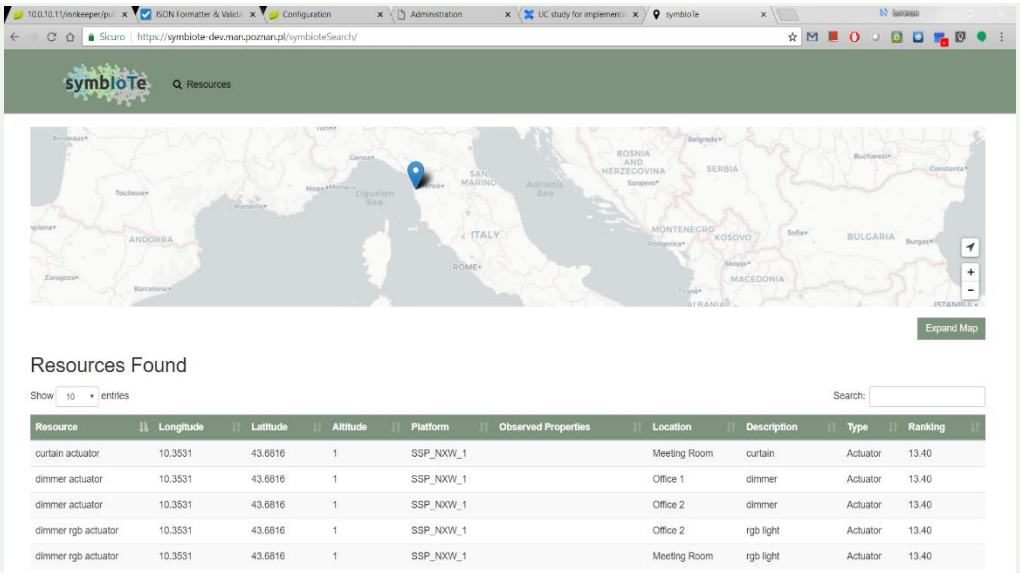
Measured Results:

- All Platform 1 resources are correctly registered, searchable and their information accessible:
    - #2 Dimmer actuator
    - #1 RGB actuator
    - #1 Curtain actuator
    - #2 Light actuator

- All Platform 1 locations are correctly registered, searchable and their information accessible:
    - #1 Meeting room
    - #1 Office 1
    - #1 Office 2

Comments:





Resource	Longitude	Latitude	Altitude	Platform	Observed Properties	Location	Description	Type	Ranking
curtain actuator	10.3531	43.6816	1	SSP_NXW_1		Meeting Room	curtain	Actuator	13.40
dimmer actuator	10.3531	43.6816	1	SSP_NXW_1		Office 1	dimmer	Actuator	13.40
dimmer actuator	10.3531	43.6816	1	SSP_NXW_1		Office 2	dimmer	Actuator	13.40
dimmer rgb actuator	10.3531	43.6816	1	SSP_NXW_1		Office 2	rgb light	Actuator	13.40
dimmer rgb actuator	10.3531	43.6816	1	SSP_NXW_1		Meeting Room	rgb light	Actuator	13.40

2	**Description**: Registration of the second Platform (Symphony) **Expected Results:** • Successful registration process **Measured Results:** • No errors during registration process in symbIoTe Agent logs **Comments**:	*Passed*
2.1	**Description**: Check of the Second Platform (Symphony) registered resources and locations on SSP **Expected Results:** • Platform 2 resources: o #1 Presence sensor o #1 Light sensor • Platform 2 locations: o #1 First floor o #1 Second Floor **Measured Results:** • Platform 2 resources: o #1 Presence sensor o #1 Light sensor • Platform 2 locations: o #1 First floor o #1 Second Floor	*100% passed*

Comments:

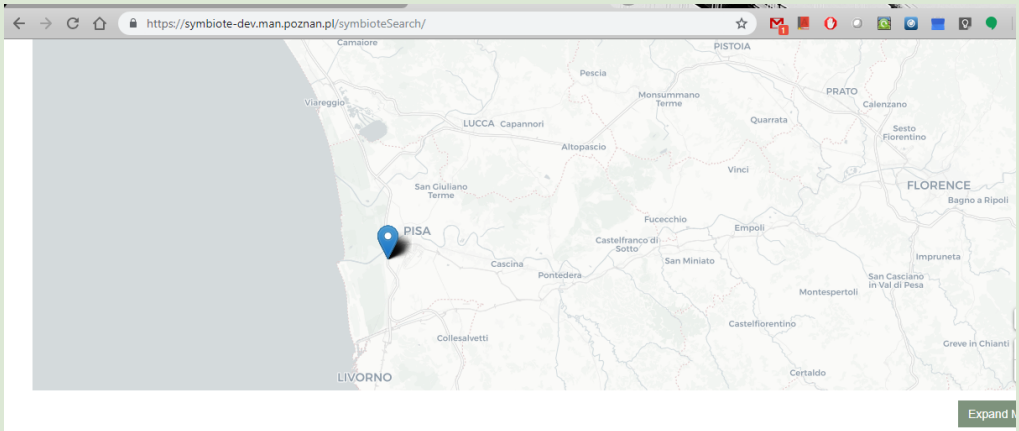
2.2	**Description**:
	Check of the Second Platform (Symphony) registered resources and locations on Core
	Expected Results:
	• Platform 2 resources:
	  ○ #1 Presence sensor
	  ○ #1 Light sensor
		100% passed

- Platform 2 locations:
  - #1 First floor
  - #1 Second Floor

Measured Results:

- Platform 2 resources:
  - #1 Presence sensor
  - #1 Light sensor
- Platform 2 locations:
  - #1 First floor
  - #1 Second Floor

Comments:



Resources Found

Show 10 entries Search:

Resource	Longitude	Latitude	Altitude	Platform	Observed Properties	Location	Description	Type	Ranking
brightness sensor	10.3531	43.6816	1	SSP_NXW_1	illuminance	Second Floor	Sensore luminosita	StationarySensor	13.40
curtain actuator	10.3531	43.6816	1	SSP_NXW_1		Meeting Room	curtain	Actuator	13.40
dimmer actuator	10.3531	43.6816	1	SSP_NXW_1		Office 1	dimmer	Actuator	13.40
dimmer actuator	10.3531	43.6816	1	SSP_NXW_1		Office 2	dimmer	Actuator	13.40
dimmer rgb actuator	10.3531	43.6816	1	SSP_NXW_1		Meeting Room	rgb light	Actuator	13.40
dimmer rgb actuator	10.3531	43.6816	1	SSP_NXW_1		Office 2	rgb light	Actuator	13.40
presence sensor	10.3531	43.6816	1	SSP_NXW_1	surface density	First Floor	sensor	StationarySensor	13.40
sensor sensor	10.3531	43.6816	1	SSP_NXW_1	power	First Floor	sensor	StationarySensor	13.40

3

Description:
Verify the SSP registered locations on the Enabler:
All the locations registered by both of the platforms should be displayed in the Enabler interface.
The enabler is completely agnostic about which Platform has registered a specific location.

Expected Results:

- #1 Meeting Room

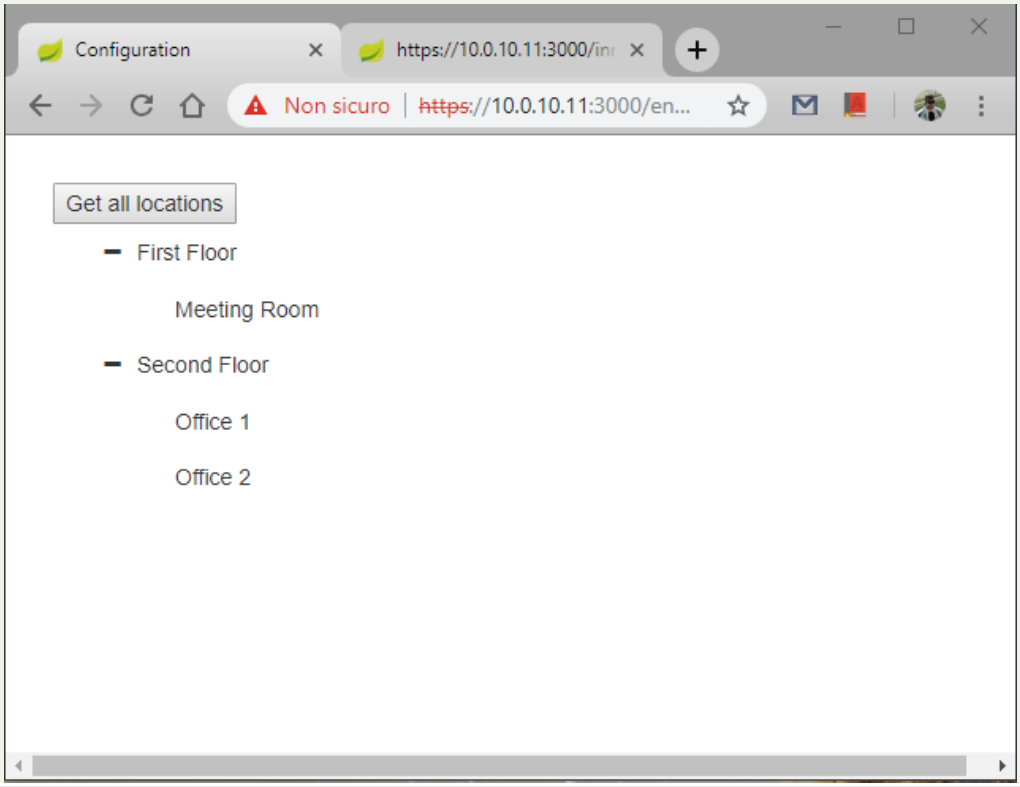
100% Passed

- #1 Office 1
- #1 Office 2
- #1 First Floor
- #1 Second Floor

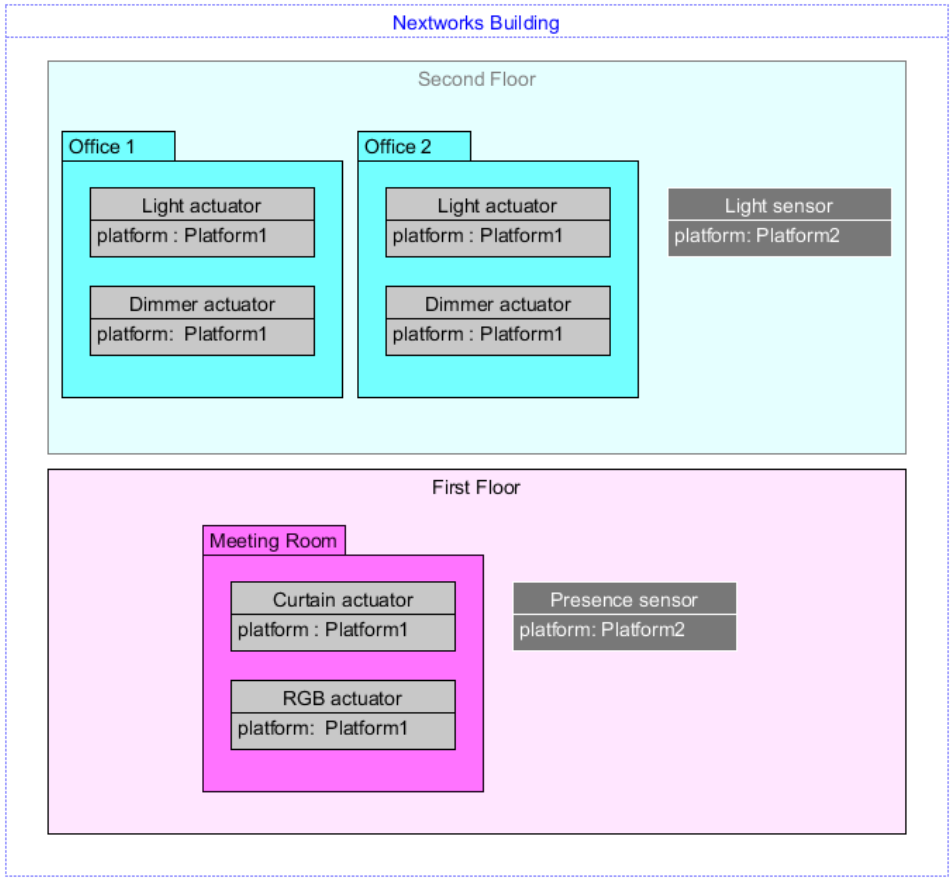
Measured Results:

- #1 Meeting Room
- #1 Office 1
- #1 Office 2
- #1 First Floor
- #1 Second Floor

Comments:



4	**Description**: Check of the SSP registered locations hierarchy: The Enabler provides a GUI that allow the user to create a hierarchical tree between the locations registered by both of the platforms. This tree must be navigable. After the hierarchy has been created through a GET call to the SSP is possible to verify that it is actually saved in the SSP database. The topology set for these test is illustrated in the image below.	*100% Passed*



Expected Results:

- Location's hierarchy illustrated in the topology saved in SSP
  - First Floor -> Meeting Room
  - Second Floor -> Office 1, Office 2

Measured Results:

- Location's hierarchy illustrated in the topology saved in SSP
  - First Floor -> Meeting Room
  - Second Floor -> Office 1, Office 2

Comments:

Smart Area Controller

Test-card #	Symbyote_SR_NXW_3	Execution Status	*PASSED*
Test Name	Smart Residence – Smart Area Controller APP		
Objectives	Verify the correct functioning of Smart Area Controller APP The mobile app should be capable of controlling devices all the devices registered in the SSP. It should be possible for the user to navigate through the hierarchical tree of locations		

	to reach any devices (registered by both the platforms). The application leverages on a specific enabler, which gives the possibility to filter the devices in the space, based on their position (building, floor, room, etc.). In this way, the application has to query the symbIoTe enabler in order to retrieve the list of the CPSs in the selected area and then to allow the user to control them.

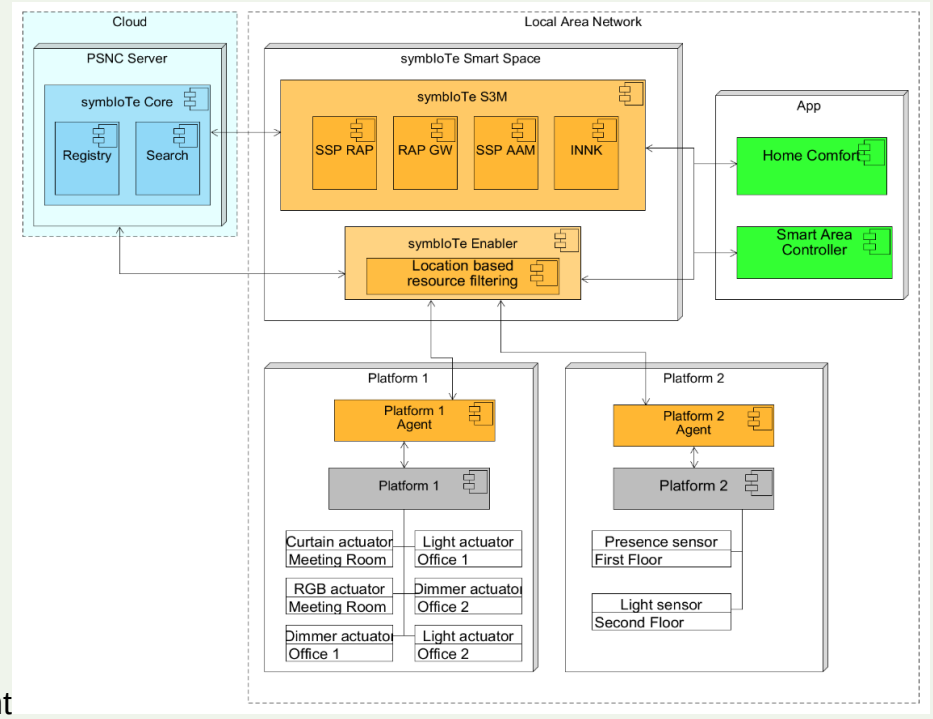
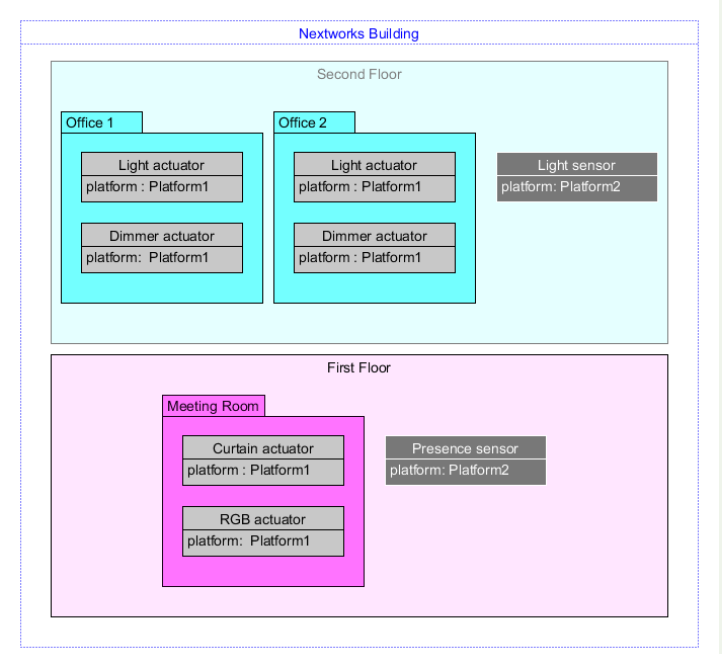
KPIs to measure

DESCRIPTION	PROCEDURE	TARGET VALUE
Correctly displayed locations	Each operator running the test gives one point for any location described in the topology and correctly displayed in the Smart Area Controller APP	50 points (5 locations x 10 operators)
Correct hierarchical positioning of locations	Each operator running the test gives one point for any location described in the topology and correctly collocated in the hierarchical tree in the Smart Area Controller APP	50 Points (5 locations x 10 operators)
Correctly accessible resources through locations	Each operator running the test gives one point for any resource described in the topology that appears and is collocated in the correct location in the Smart Area Controller APP	80 Points (8 resources x 10 operators)
Correctly readable and writable resources	Each operator running the test gives one point for any readable sensor and writable actuator in the Smart Area Controller APP	80 Points (8 resources x 10 operators)

Related Use Cases	Smart Residence
Designers	*Gino Carrozzo, Matteo Pardi, Luca Tomaselli, Lorenzo Neri*
Related Test-cards	Symbyote_SR_NXW_1, Symbyote_SR_NXW_2
Attachments	
Additional Comments	

Trial resources and topology	

Required resources	Symphony L3 Agent on Platform 1 (Symphony);
	Symphony L3 Agent on Platform 2 (Symphony);
	Controlled devices over both Platforms;
	• Platform 1 resources:
	  ○ #1 Dimmer actuator
	  ○ #1 RGB actuator
	  ○ #1 Curtain actuator
	  ○ #1 Dimmer actuator
	  ○ #1 Light actuator
	  ○ #1 Light actuator
	• Platform 1 locations:
	  ○ #1 Meeting room
	  ○ #1 Office 1
	  ○ #1 Office 2
	• Platform 2 resources:
	  ○ #1 Presence sensor
	  ○ #1 Light sensor
	• Platform 2 locations:
	  ○ #1 First floor
	  ○ #1 Second Floor
	
	Smart Area Controller APP installed on 10 android devices
	10 operators to run the tests

Topology details	Deployment  Topology 

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Navigation through Locations: Verify that every location that appears in the topology is displayed and selectable in Smart Area Controller APP, for each correctly displayed location one point has been given by each operator.	50/50

	Expected Results: • #1 Meeting Room • #1 Office 1 • #1 Office 2 • #1 First Floor • #1 Second Floor **Measured Results:** • #1 Meeting Room • #1 Office 1 • #1 Office 2 • #1 First Floor • #1 Second Floor **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_3.1** to **Symbyote_SR_NXW_3.10**	
2	**Description**: Navigation through Locations, hierarchy verification: Verify that the hierarchical order of locations shown in the topology is reflected into the Smart Area Controller APP interface, for each location correctly collocated in the hierarchical tree one point has been given by each operator. **Expected Results:** • First Floor -> Meeting Room • Second Floor -> Office 1, Office 2 **Measured Results:** • First Floor -> Meeting Room • Second Floor -> Office 1, Office 2 **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_3.1** to **Symbyote_SR_NXW_3.10**	50/50
3	**Description**: Navigation through Locations up to resources: Every resource in the Smart Area Controller APP must be accessible and correctly reachable through any locations it is registered in, for any correctly displayed and collocated resource one point has been given by each operator. Expected Results: • First Floor ->	80/80

	○ Meeting Room -> ▪ #1 Curtain actuator ▪ #1 RGB actuator ○ #1 Presence sensor • Second Floor ->: ○ Office 1->: ▪ #1 Dimmer actuator ▪ #1 Light actuator ○ Office 1->: ▪ #1 Dimmer actuator ▪ #1 Light actuator ○ #1 Light sensor **Measured Results:** • First Floor ->: ○ Meeting Room ->: ▪ #1 Curtain actuator ▪ #1 RGB actuator ○ #1 Presence sensor • Second Floor ->: ○ Office 1->: ▪ #1 Dimmer actuator ▪ #1 Light actuator ○ Office 2->: ▪ #1 Dimmer actuator ▪ #1 Light actuator ○ #1 Light sensor **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_3.1** to **Symbyote_SR_NXW_3.10**	
4	**Description**: Navigation through Locations:	80/80

	Verify that every resource that appears in the topology is readable (sensors) or writable (actuators) in Smart Area Controller APP, for each correctly behaving resource one point has been given by each operator. **Expected Results:** • #2 Dimmer actuator - Writable • #1 RGB actuator - Writable • #1 Curtain actuator - Writable • #2 Light actuator - Writable • #1 Presence sensor - Readable • #1 Light sensor - Readable **Measured Results:** • #2 Dimmer actuator - Writable • #1 RGB actuator - Writable • #1 Curtain actuator - Writable • #2 Light actuator - Writable • #1 Presence sensor - Readable • #1 Light sensor - Readable **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_3.1** to **Symbyote_SR_NXW_3.10**	

Results

KPI DESCRIPTION	ACTUAL VALUE	TARGET VALUE
Correctly displayed locations	50	50 points (5 locations x 10 operators)
Correct hierarchical positioning of locations	50	50 Points (5 locations x 10 operators)
Correctly accessible resources through locations	80	80 Points (8 resources x 10 operators)
Correctly readable and writable resources	80	80 Points (8 resources x 10 operators)

Home Comfort

Test-card #	Symbyote_SR_NXW_2	**Execution Status**	*PASSED*

<table>
<tr><td>Test Name</td><td colspan="3">Smart Residence – Home Comfort APP</td></tr>
<tr><td>Objectives</td><td colspan="3">Verify the correct functioning of Home Comfort APP
Home Comfort App consists of
• Back-end running on a server, that manages core operations monitoring the sensors and activating the actuators
• Front-end accessible via web for scenarios creation and configuration

The purpose is to set conditions on comfort environmental parameters like luminosity, temperature etc. in order to reach the desired comfort state.
These tests verify the interoperability between resources registered by separated platforms and their capability to cooperate in a scenario.
The Home Comfort APP interacts with any resource and navigates through every locations being completely unaware of the Platform that registered it.</td></tr>
<tr><td rowspan="4">KPIs to measure</td><td>DESCRIPTION</td><td>PROCEDURE</td><td>TARGET VALUE</td></tr>
<tr><td>Correctly displayed locations</td><td>Each operator running the test gives one point for any location described in the topology and correctly displayed in the Home Comfort APP</td><td>50 points
(5 locations x 10 operators)</td></tr>
<tr><td>Correct hierarchical positioning of locations</td><td>Each operator running the test gives one point for any location described in the topology and correctly collocated in the hierarchical tree in the Home Comfort APP</td><td>50 Points
(5 locations x 10 operators)</td></tr>
<tr><td>Correctly accessible resources through locations</td><td>Each operator running the test gives one point for any resource described in the topology that appears and is collocated in the correct location in the Home Comfort APP</td><td>80 Points
(8 resources x 10 operators)</td></tr>
<tr><td></td><td>Scenarios Creation</td><td>Each operator creates his own scenario setting conditions on sensors and operations on actuators.
The operator gives one point if the scenario is</td><td>10 Points</td></tr>
</table>

		correctly created.	
	Scenario Working	Each operators controls the correct functioning of the created scenario creating on sensors the expected conditions and verifying the correct behaving of actuators. One point is given if the scenario works properly.	10 Points
Related Use Cases	Smart Residence		
Designers	Gino Carrozzo, Matteo Pardi, Luca Tomaselli, Lorenzo Neri		
Related Test-cards	Symbyote_SR_NXW_1, Symbyote_SR_NXW_3		
Attachments	None		
Additional Comments			

Trial resources and topology	

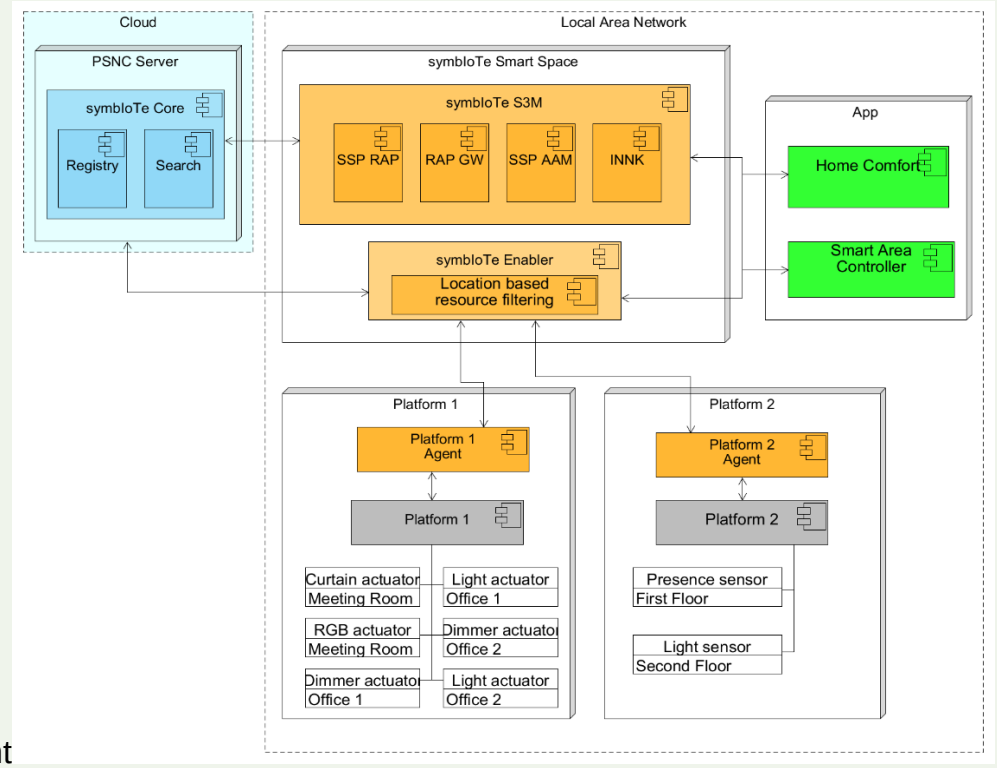
Required resources	Symphony L3 Agent on Platform 1 (Symphony);
	Symphony L3 Agent on Platform 2 (Symphony);
	Controlled devices over both Platforms;
	• Platform 1 resources:
	  ○ #2 Dimmer actuator
	  ○ #1 RGB actuator
	  ○ #1 Curtain actuator
	  ○ #2 Light actuator
	• Platform 1 locations:
	  ○ #1 Meeting room
	  ○ #1 Office 1
	  ○ #1 Office 2
	• Platform 2 resources:
	  ○ #1 Presence sensor

- #1 Light sensor
- Platform 2 locations:
  - #1 First floor
  - #1 Second Floor

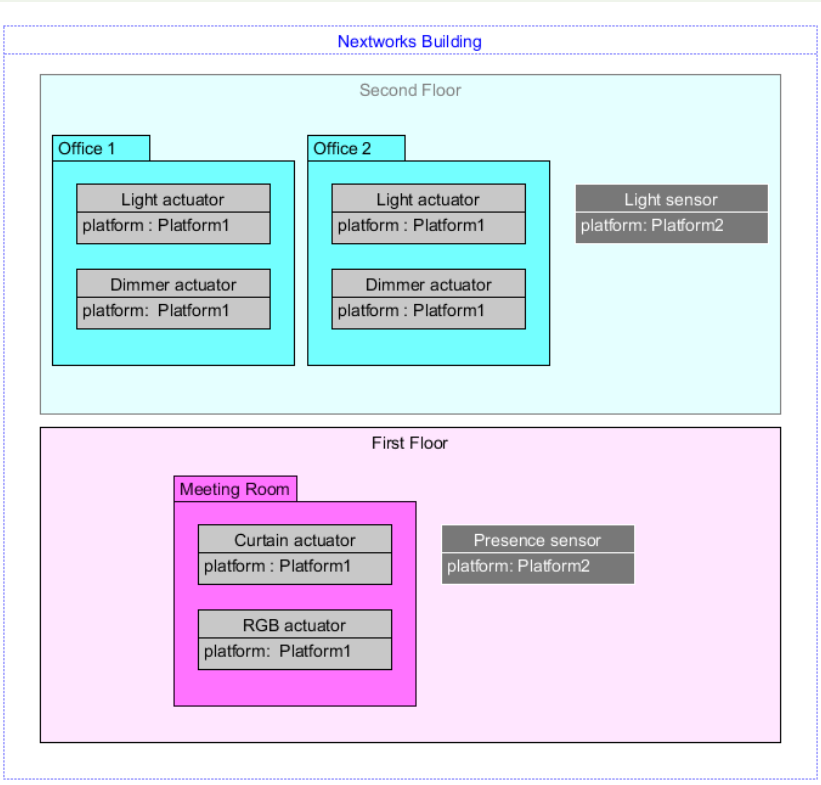
Home Comfort APP installed on 10 devices

10 operators to run the tests

Topology details



Deployment Topology



Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Navigation through Locations: Verify that every location that appears in the topology is displayed and selectable in the Home Comfort APP, for each correctly displayed location one point has been given by each operator. **Expected Results:** • #1 Meeting Room • #1 Office 1 • #1 Office 2 • #1 First Floor • #1 Second Floor **Measured Results:** • #1 Meeting Room • #1 Office 1 • #1 Office 2 • #1 First Floor	50/50

	• #1 Second Floor **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_2.1** to **Symbyote_SR_NXW_2.10**	
2	**Description**: Navigation through Locations, hierarchy verification: Verify that the hierarchical order of locations shown in the topology is reflected into the Home Comfort APP interface, for each location correctly collocated in the hierarchical tree one point has been given by each operator. **Expected Results:** • First Floor -> Meeting Room • Second Floor -> Office 1, Office 2 **Measured Results:** • First Floor -> Meeting Room • Second Floor -> Office 1, Office 2 **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_2.1** to **Symbyote_SR_NXW_2.10**	50/50

3	**Description**:
	Navigation through Locations up to resources:
	Every resource in the Home Comfort APP must be accessible and correctly reachable through any locations it is registered in, for any correctly displayed and collocated resource one point has been given by each operator.
	Expected Results:
	• First Floor ->
	  ○ Meeting Room ->
	  ▪ #1 Curtain actuator
	  ▪ #1 RGB actuator
	  ○ #1 Presence sensor
	• Second Floor ->
	  ○ Office 1-> ▪ #1 Dimmer actuator
	  ▪ #1 Light actuator
	  ○ Office 1-> ▪ #1 Dimmer actuator
		80/80

	▪ #1 Light actuator o #1 Light sensor **Measured Results:** • First Floor ->: o Meeting Room ->: ▪ #1 Curtain actuator ▪ #1 RGB actuator o #1 Presence sensor • Second Floor ->: o Office 1->: ▪ #1 Dimmer actuator ▪ #1 Light actuator o Office 2->: ▪ #1 Dimmer actuator ▪ #1 Light actuator o #1 Light sensor **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_2.1** to **Symbyote_SR_NXW_2.10**	
4	**Description**: Scenarios creation: At the moment of a new scenario creations rules must be empty and every possible condition/action combination (depending only on sensors and actuators in the location) must be selectable. Define an Example Scenario setting conditions on any available sensor and actions on any available actuator. **Expected Results:** • Every possible condition/action combination available (given the topology) • Scenario correctly created with the rules chosen by the operator: scenario 'ExampleXXX' appears with the expected rules in the scenario's list of the app **Measured Results:** • Every possible condition/action combination available (given the topology)	10/10

	• Scenario correctly created with the rules chosen by the operator: scenario 'ExampleXXX' appears with the expected rules in the scenario's list of the app **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_2.1** to **Symbyote_SR_NXW_2.10**	
5	**Description**: Scenarios rules check: Once scenario's rules are set their correct execution must be checked creating the desired conditions on sensors and verifying the correct reactions of actuators. **Expected Results:** • Respect of scenario's rules **Measured Results:** • Scenario rules respected **Comments**: These results are expected for every operator and registered in test cards from **Symbyote_SR_NXW_2.1** to **Symbyote_SR_NXW_2.10**	10/10

Results	KPI DESCRIPTION	ACTUAL VALUE	TARGET VALUE
	Correctly displayed locations	50	50 points (5 locations x 10 operators)
	Correct hierarchical positioning of locations	50	50 Points (5 locations x 10 operators)
	Correctly accessible resources through locations	80	80 Points (8 resources x 10 operators)
	Scenarios Creation	10	10 Points
	Scenario Working	10	10 Points

8.3 Smart Mobility

8.3.1 End-user satisfaction evaluation

<questionnaires as provided to test users (UNIVIE)>

Vienna

Test-card #	MD-SMEUR-01	Execution Status	*passed or failed*
Test Name	Overall end-user satisfaction of mobile sensors and symbIoTe SMEUR Routing App		
Objectives	40 different people are testing the mobile sensors together with the routing app for four weeks (probably 10 people per week) in Vienna. We expect an overall satisfaction of 80% since the apps are not perfectly developed yet. We expect that the user would use the symbiote SMEUR Routing App rather than any other routing app in the future with a chance of 50%.		
KPIs to measure	Description	Current value	Target value
	SUS		> 68/100
Related Use Cases	Smart Mobility and Ecological Routing		
Designers	Drobics M., Lederer I.		
Related Test-cards	MD-SMEUR-02		
Attachments	Questionnaire		

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Preparation of material **Expected Results:** • All mobile sensors arrive on time • Register all sensors (contact Aleksandar Antonic) • Create and print documents (e.g. introduction, tutorial) for every trial participant on time • Charge enough portable battery chargers (every participant gets one) **Measured Results:** **Comments**:	*passed*
2	**Description:**	*passed*

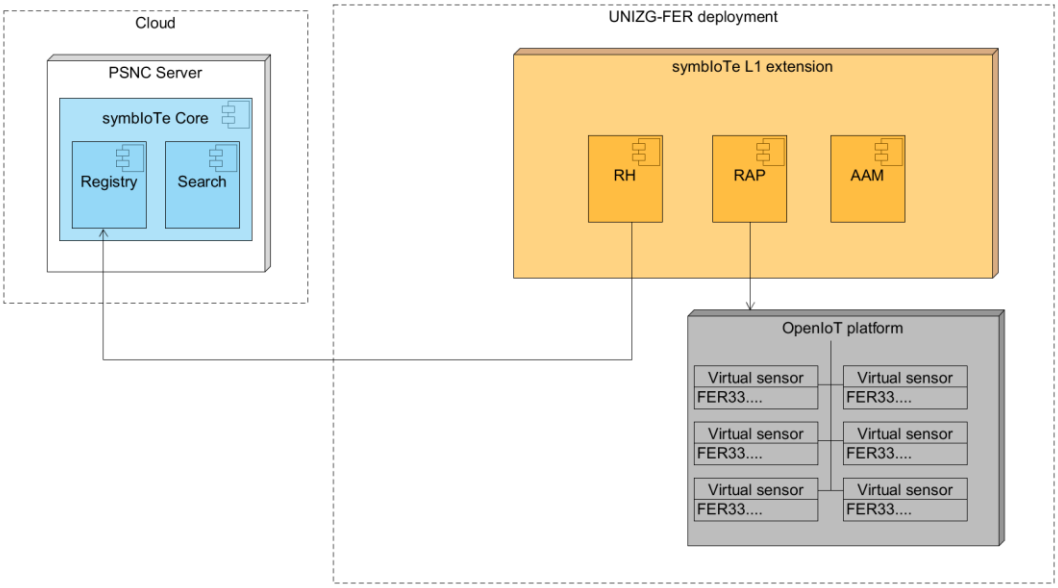
	Introduction **Expected Results:** • No issues with connecting sensors to Smartphone • No issues with usage of app and sensor • No unexpected rejection from volunteers **Measured Results:** **Comments:** First Introduction will take place on 20/08/2018 at 4pm	
3	**Description:** Field test **Expected Results:** • Participants activate the sensor outdoors and deactivate it when they go inside • Participants search for a route in the Routing App minimum twice a day **Measured Results:**	*passed*
4	**Description:** Final questionnaire **Expected Results:** • Full attendance • Outcome fulfills KPIs of MD-SMEUR-01 **Measured Results:**	*passed*

Zagreb

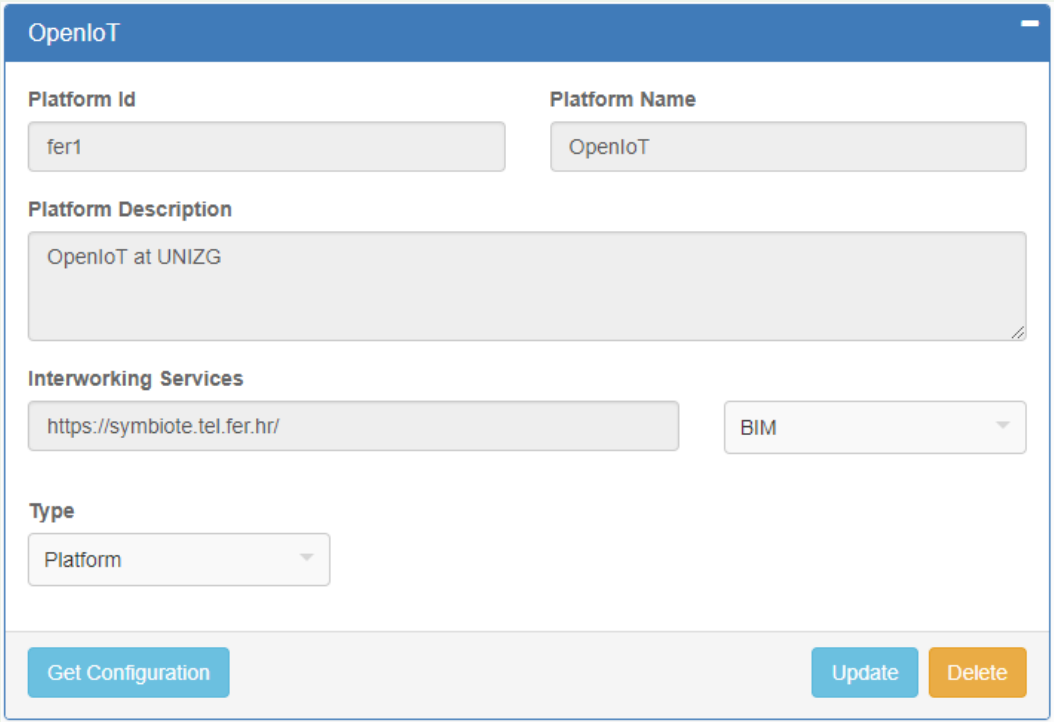
Test-card #	**SymbIoTe_OpenIoT_1**	**Execution Status**	***PASSED***
Test Name	Registration of resources and data acquisition from OpenIoT platform		
Objectives	The aim of these tests is to verify the successful registration of all L1 resources provided by OpenIoT platform installed at UNIZG-FER. Additionally, tests verify the availability of data through symbIoTe middleware provided by OpenIoT platform. The test validates L1 functionality of the symbIoTe middleware. It provides functional example of extension of an IoT platform (i.e., OpenIoT platform) to become compatible with symbIoTe ecosystem. The OpenIoT platform is an open-source middleware that has capability of gathering data from heterogeneous data sources. Extension of the platform with symbIoTe L1 cloud components will provide additional data endpoint for all users and it will integrate all OpenIoT resources in the symbIoTe ecosystem, so that all resources can be discovered through symbIoTe cloud services. OpenIoT instance at UNIZG-FER hosts data from 100 virtual sensors where each sensor is covering area of 1 square kilometre, making total coverage of at least 100 km^2 across urban areas of Zagreb (Croatia) and Vienna (Austria). A virtual sensor covers some pre-defined geographical area and publishes data from wearable sensors located in that area, thus anonymizing contributor and supporting mobility of wearable sensors. The objective of the test is to verify the correct operation of the L1 registration process (i.e., platform registration and sensor registration		

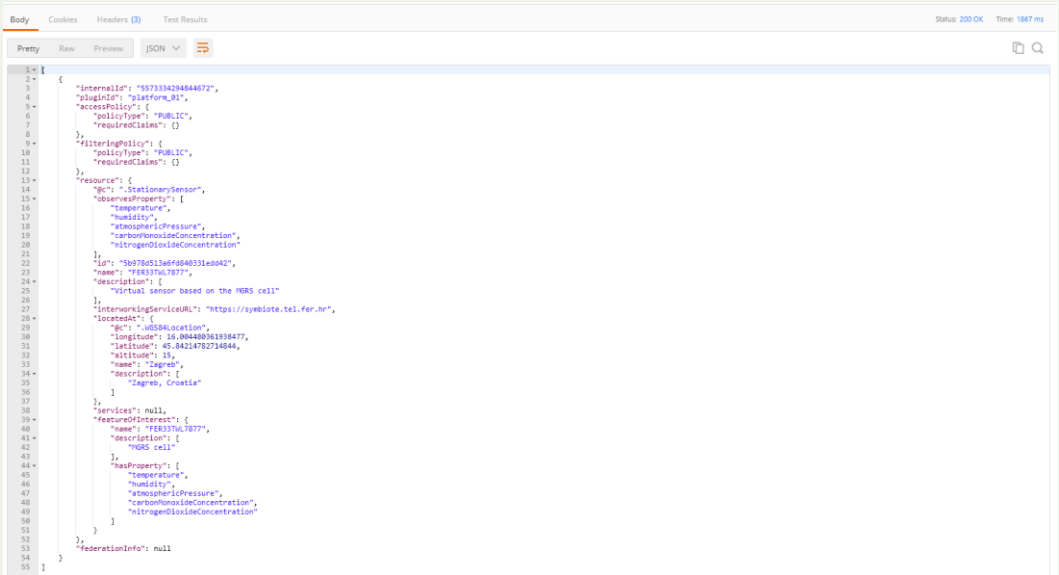
	process). Also it is tested feasibility of L1 interoperability using symbIoTe cloud components, i.e., a uniform way to access data no matter on underlying platform and used information model.
KPIs to measure	• IoT platform registered in symbIoTe core → passed • *Percentage of correctly registered resources → target value 100%* • *Number of correctly registered resources → target value 100* • *Number of searchable resources → target value 100* • *Percentage of correctly accessible data from registered sensors (via internet) → target value 100%* • RAP plug-in response time → target value < 1 sec
Related Use Cases	Smart Mobility and Ecological Routing
Designers	*Aleksandar Antonić*
Related Test-cards	SMEUR-ZAGREB-01
Attachments	None
Additional Comments	

Trial resources and topology	
Required resources	OpenIoT deployment (UNIZG-FER) symbIoTe core services – L1 100+ virtual sensors with data available in UNIZG-FER deployment of OpenIoT symbIoTe extension of an IoT (OpenIoT) platform

Topology details	

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: IoT platform registered in symbIoTe core; i.e., registration of the OpenIoT platform instance at UNIZG-FER: **Expected Results:** • Successful registration process **Measured Results:** • No errors during registration process in symbIoTe core **Comments**:	*Passed*



2.1	**Description**: Successful registration of resources in OpenIoT; percentage of correctly registered resources is 100% **Expected Results:** • All registration requests finish with 200 OK response **Measured Results:** • All registration requests return 200 OK response and it contained symbIoTe Id for each resource. **Comments**: 	100% passed
2.2	**Description**: Registration of resources in OpenIoT; number of correctly registered resources is at least 100	*Passed*

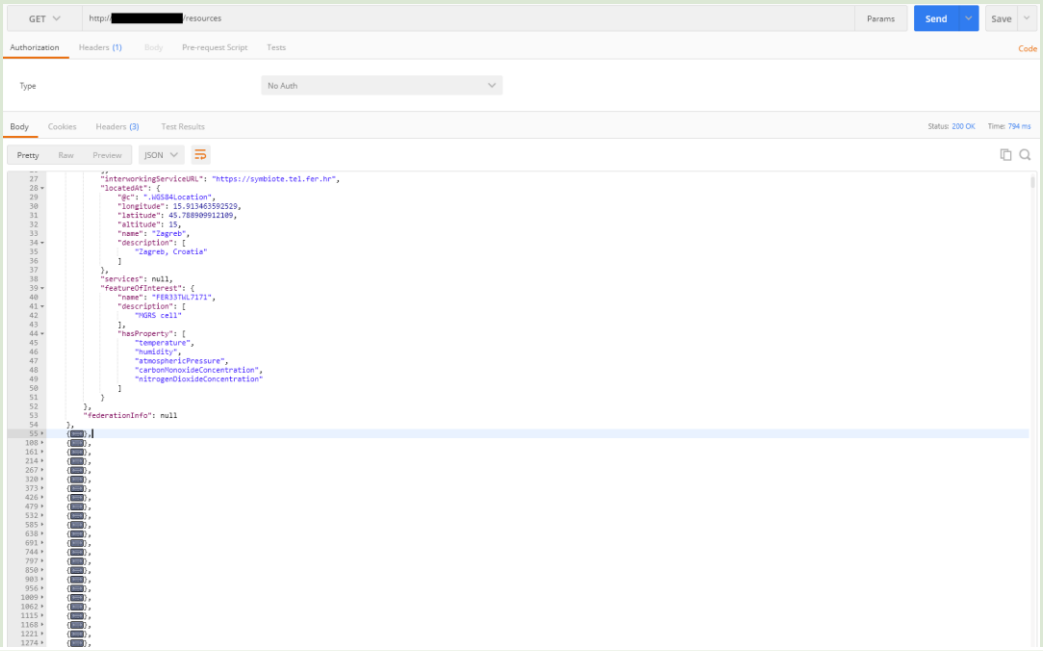
Expected Results:

- At least 100 virtual sensors from OpenIoT registered at Registration Handler (RH) in symbIoTe extension of the OpenIoT platform. Each virtual sensor represents geographical area (naming and grid is defined by using NATO standard Military Grid Reference System – MGRS) which publishes data from physical sensor nodes in corresponding location. Each virtual sensor is created on-the-fly when an end user publishes data for the first time for his/her location.

Measured Results:

- Registration handler returned 146 registered virtual sensors with their description.

Comments:



3	**Description**: Number of searchable resources; resources from OpenIoT platform visible at symbIoTe core services **Expected Results:** ○ At least 100 virtual sensors from OpenIoT registered in symbIoTe core. Since RH registered 146 sensors from OpenIoT (end users covered more urban area than we expected) we expect that 146 resources are visible in symbIoTe core. **Measured Results:** • Search and retrieve information from symbIoTeSearch web application which presents all available registered resources ○ 146 virtual sensors registered (covering in total 146 km2 in cities of Zagreb and Vienna) **Comments**:	Passed

Resources Found

Show 10 entries

Resource	Longitude	Latitude	Altitude	Platform	Observed Properties
FER33TWL6573	15.836546897888184	45.807498931884766	15	OpenIoT	humidity,atmospheric pressure
FER33TWL6574	15.836682319641113	45.816497802734375	15	OpenIoT	humidity,atmospheric pressure
FER33TWL6973	15.888021469116211	45.80710983276367	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7073	15.90088939666748	45.807010650634766	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7171	15.913463592529	45.788909912109	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7173	15.91375732421875	45.806907653808594	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7271	15.9263277053833	45.7888069152832	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7273	15.926626205444336	45.80680465698242	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7369	15.938889503479004	45.77070236206055	15	OpenIoT	humidity,atmospheric pressure
FER33TWL7371	15.939191818237305	45.788700103759766	15	OpenIoT	humidity,atmospheric pressure

Showing 1 to 10 of 146 entries (filtered from 1,159 total entries)

4	**Description**: Percentage of correctly accessible data from registered sensors (via internet); latest readings obtained from registered sensors **Expected Results:** • Successful data acquisition **Measured Results:** • No errors during data acquisition and received latest measurement values **Comments**:	*100% passed*

FER33TWL7572 data

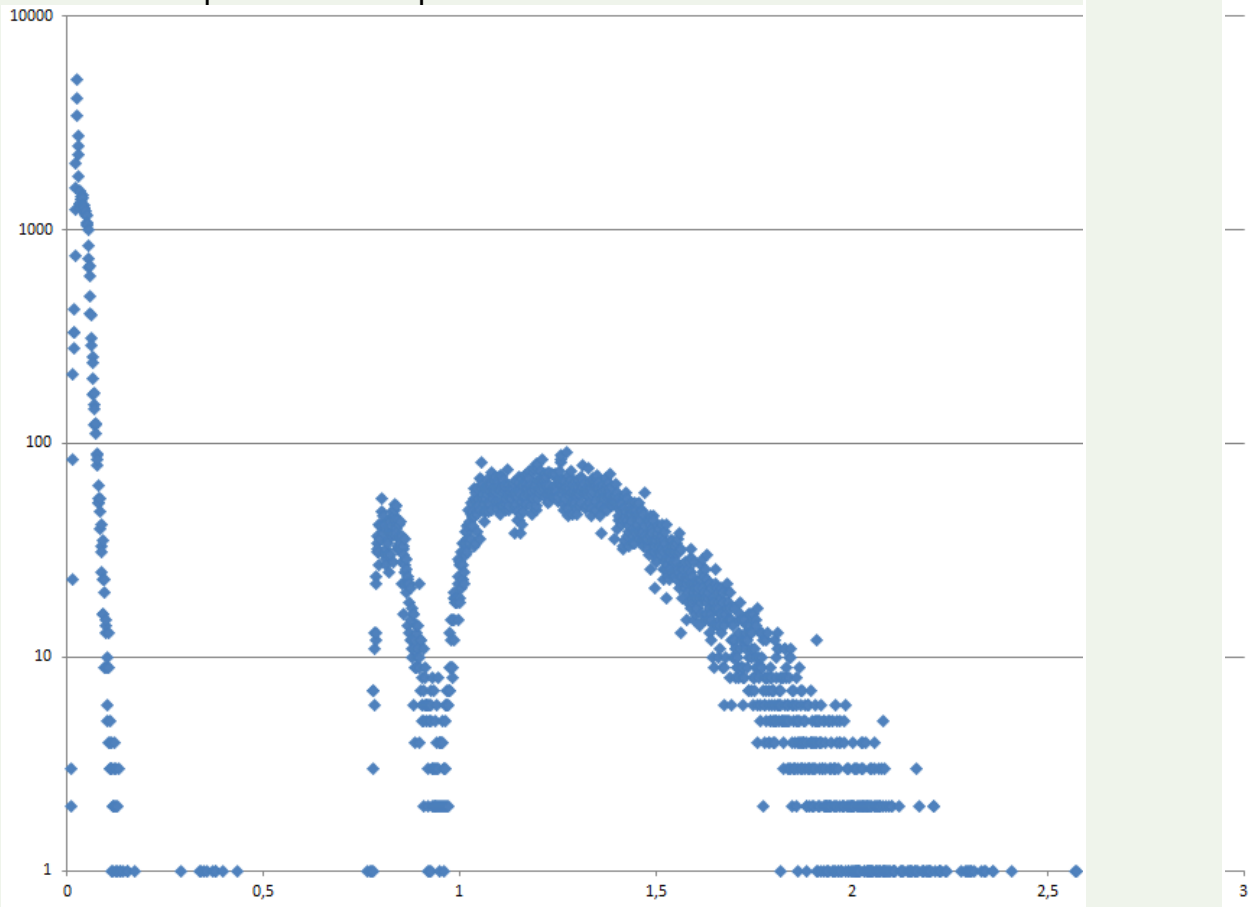
Show 10 entries Search:

Measurement	Observed Property	Unit	Latitude	Longitude	Observation Time
28.0	temperature	C	45.801265	15.9712231	2018-09-11, 14:35:52
0.12233549999999999	co	mg/m3	45.801265	15.9712231	2018-09-11, 14:36:02
44.0	humidity	Percent	45.801265	15.9712231	2018-09-11, 14:36:13
16.938	no2	ug/m3	45.801265	15.9712231	2018-09-11, 14:36:34
1007.5	pressure	hPa	45.801265	15.9712231	2018-09-11, 14:36:44
0.1200435	co	mg/m3	45.801265	15.9712231	2018-09-11, 14:36:54
13.644499999999999	no2	ug/m3	45.801265	15.9712231	2018-09-11, 14:37:05
1007.0	pressure	hPa	45.801265	15.9712231	2018-09-11, 14:37:25
1007.25	pressure	hPa	45.801265	15.9712231	2018-09-11, 14:37:46
0.0873825	co	mg/m3	45.801265	15.9712231	2018-09-11, 14:37:57

Showing 1 to 10 of 31 entries Previous 1 2 3 4 Next

temperature Generate Graphical Report Close

5	**Description**: RAP plug-in response time; measured time needed to serve one data	

	request, i.e., latency of the RAP plug-in for OpenIoT platform **Expected Results:** • RAP plug-in for the OpenIoT platform fetches data from OpenIoT platform in less than 1 second, and forwards readings to generic RAP **Measured Results:** • More than 67% of requests were executed in less than 1 second. An average of 0.48 seconds. **Comments**: The mass density function for RAP plug-in response time. Note: y-axis is presented in the logarithmic scale and represents number of occurrences, while x-axis represent the response time.  	*Passed*

Porto

Test-card #	SMEUR-PORTO-01	**Execution Status**	*passed*
Test Name	Overall end-user satisfaction of symbIoTe SMEUR Green Routing App and SMEUR Enabler		
Objectives	20 different people are testing the the SMEUR Green Routing App for four weeks in Porto. We expect an overall satisfaction of 75% since		

	the apps are not perfectly developed yet. We expect that the user would use the symbiote SMEUR Routing App rather than any other routing app in the future with a chance of 25%.
KPIs to measure	• Preparation of trial material→ passed • *Field trial→ passed* • *Final questionnaire→ fill in by 75% participants* • Questionnaire: • Overall recommendation of the use case → at least 50% of all participants
Related Use Cases	Smart Mobility and Ecological Routing
Designers	Drobics M., Lederer I., Aleksandar Antonić, Ivana Podnar Žarko, João Garcia
Attachments	Questionnaire

Test description		
Step #	**Step description and expected results**	**Status**
1	**Description**: Preparation of trial material **Expected Results:** • All sensors available • SMEUR Green Routing application tested and ready to be used • Register sensors for Porto **Measured Results:** • Applications ready to be used • Sensors registered	*passed*
2	**Description:** Field trial **Expected Results:** • Participants search for a route in the Routing App **Measured Results:** • Several thousand route requests were made using the routing app	*passed*
3	**Description:** Final questionnaire **Expected Results:**	*passed*

	• 75% of participants fill in the online questionnaire **Measured Results:** • 77% of participants fill in the online questionnaire	
4	**Description**: Questionnaire; overall user satisfaction with the use case investigated through post-trial questionnaire **Expected Results:** • At least 50% of all participants would recommend the use case to their family and friends **Measured Results:** • More than 92% voted at least 4 or more (in a scale of 5) that they would recommend the app to their families and friends	*passed*

8.3.2 Technical Evaluation of Field Trial

Test-card #	MD-SMEUR-02	**Execution Status**	*passed*
Test Name	Overall technical evaluation of symbIoTe's SMEUR use-case		
Objectives	Validate the technical stability and quality of service of symbIoTe components based on the requests generated by the test participants.		
KPIs to measure	• Route response → 2 sec • PoI response → 2 sec • RAP plug-in response → < 1 sec • Core services request → > 6/hour • Number of requests for RAP → > 30/hour /per platform		
Related Use Cases	Smart Mobility and Ecological Routing		
Designers	Drobics M., Lederer I., Aleksandar Antonić, Ivana Podnar Žarko, João Garcia		
Related Test-cards	MD-SMEUR-01		

Test description		
Step #	**Step description and expected results**	**Status**

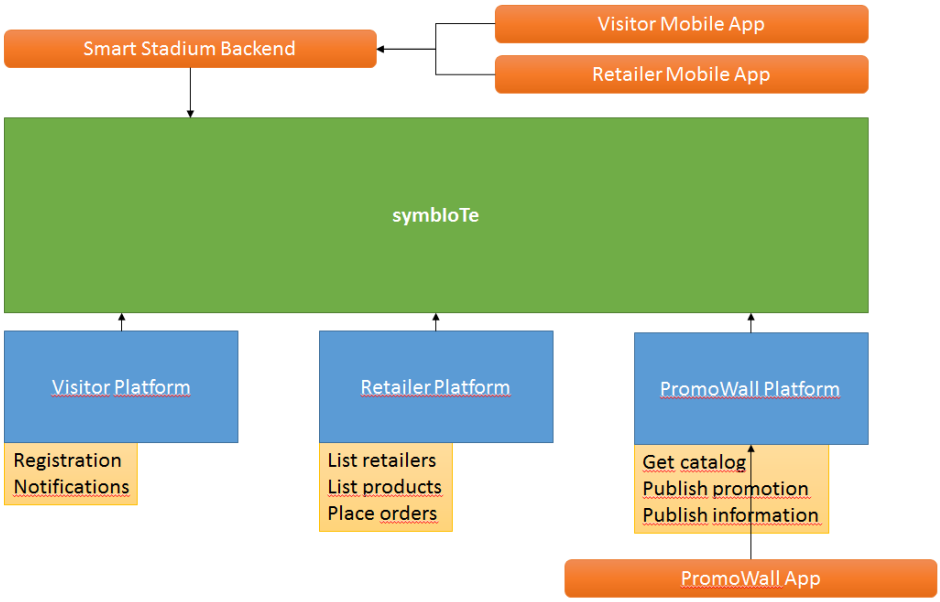
1	**Description**: Route response **Expected Results:** • Response times of less than two seconds **Measured Results:** • More than 64% of requests were executed in less than 2 seconds. Total average of 2,996 seconds. **Comments**: The response times for green routes, although the majority was below 2 seconds, our target goal, the average was a bit higher. One of the main factors that contributed to this value was that, during 2 days, MoBaaS servers were having problems and route requests were taking close to 20 seconds. This skewed the number by quite a bit. After this was fixed, the process run much smoother and users didn't complain about this issue.	*passed and failed*
2	**Description:** PoI response **Expected Results:** • Response times of less than two seconds **Measured Results:** • More than 93% of requests were executed in less than 2 seconds. Total average of 1,04 seconds	*passed*
3	**Description:** RAP plug-in response **Expected Results:** • Response times of less than one seconds **Measured Results:** • More than 67% of requests were executed in less than 1 second. An average of 0.48 seconds.	*passed*
4	**Description:** Core services request **Expected Results:** • At least 6 requests per hour **Measured Results:** • On average 23 requests per hour	*passed*
5	**Description:** Number of requests for RAP **Expected Results:** • At least 6 requests per hour **Measured Results:** • On average 1110 requests per hour per platform	*passed*

8.4 Smart Stadium

8.4.1 Generic evaluation

Test-card #	SS-GN.01	Execution Status	PASSED
Test Name	Generic KPIs validation		
Objectives	The goal is to validate system is registered successfully in symbIoTe Core so it's searchable and accessible. This section will then validate all generic KPIs which are also L1-specific: - Platforms are registered and accessible through symbIoTe - Known resources are registered and accessible through symbIoTe		
KPIs to measure	DESCRIPTION	PRODEDURE	TARGET VALUE
	Number of registered platforms	REST request to	4 (Visitor, Retailer and Promotion and Information and Beacon Platform)
	Resource registration: - Visitors - Retailers - Promowalls - Beacons	REST request to Resource Handler's 'resource' service	Resources: - 5 - 8 - 2 - 0 *(not registered in symbIoTe, the map is managed only by Promowall Platform)*
	Service registration: - Visitor Platform - Retailer Platform - Promowall Platform - Beacon Platform	REST request to Core Interface 'query' service	 - 2 - 6 - 4 - 1
Related Use Cases	Smart Stadium		
Designers	Juan Belmonte Rodríguez		
Related Test-cards	None		
Attachments	Visitor App download links in public stores (Spain only, Catalan language): - App Store: ATLÈTIC Terrassa Oficial* - Google Play: ATLÈTIC Terrassa Oficial* *symbIoTe integration currently deactivated until next social event in		

	ATHC stadium.
Additional Comments	Smart Stadium software was using symbiote-open.man.poznan.pl instance during trials (using release 2.0.0).

Trial resources and topology	
Required resources	*In order to run all tests, the following resources must be available:* - *Mobile application available in stores (Google Play and App Store)* - *Visitor Platform registered and accessible, with registered resources* - *Retailer Platform registered and accessible, with registered resources* - *Beacon platform registered and accessible, with registered resources* - *Promowall platform registered and accessible, with registered resources*
Topology details	

Test description		
Step #	**Step description and expected results**	**Status**
1.	**Description**: Start Smart Stadium backend platforms and check they are all registered in symbIote and searchable. **Expected Results:** • All platforms (Visitor, Retailer, Beacon and Promowall) registers themselves to symbIoTe **Measured Results:** • All four platforms were registered. **Comments**: None.	passed

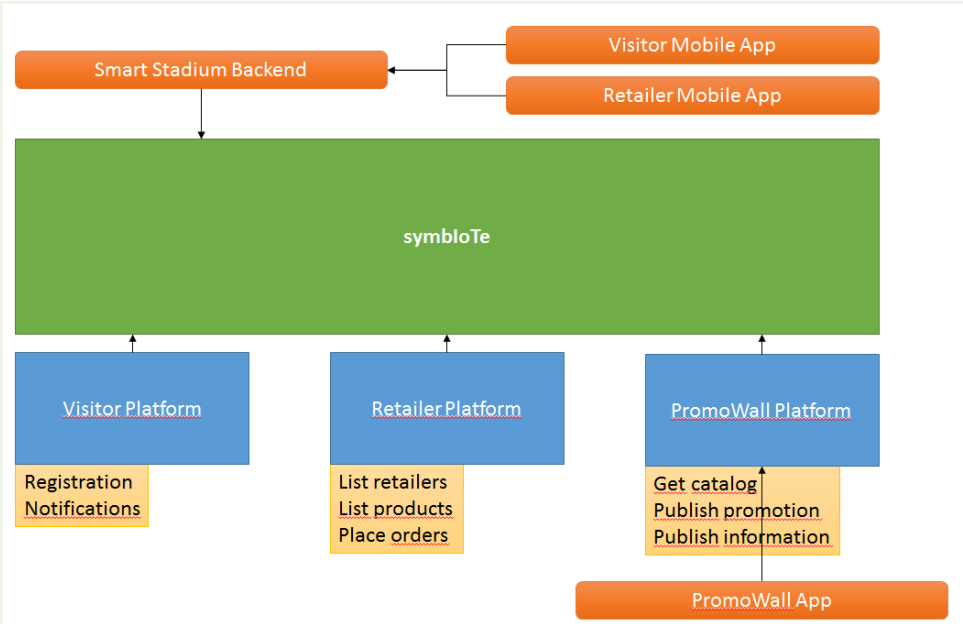
2	**Description**: All initial resources in the whole pilot environment are successfully registered in symbIoTe: - List of retailers - List of beacons to locate everything in the stadium - List of Promowalls spread across the stadium - Demo users to test the system **Expected Results:** • Number of retailers: 8 • Number of beacons: 2 • Number of Promowalls: 2 • Number of beacons: 0 • Demo users: 8 **Measured Results:** • Number of retailers: 8 • Number of beacons: 2 • Number of Promowalls: 2 • Number of beacons: 0 • Number of users: 8 **Comments**: Users involved in this test step belong to development team and a selected group of members of ATHC club. Later on end users were successfully registered. Tests covered in test card SS-VA.01.	passed
3	**Description**: Start Smart Stadium backend platforms and check they all startup resources are also registered and searchable. **Expected Results:** • Number of services: 13 **Measured Results:** • Number of registered resources which refers to services: 13 **Comments**: None.	passed

8.4.2 Visitor app evaluation

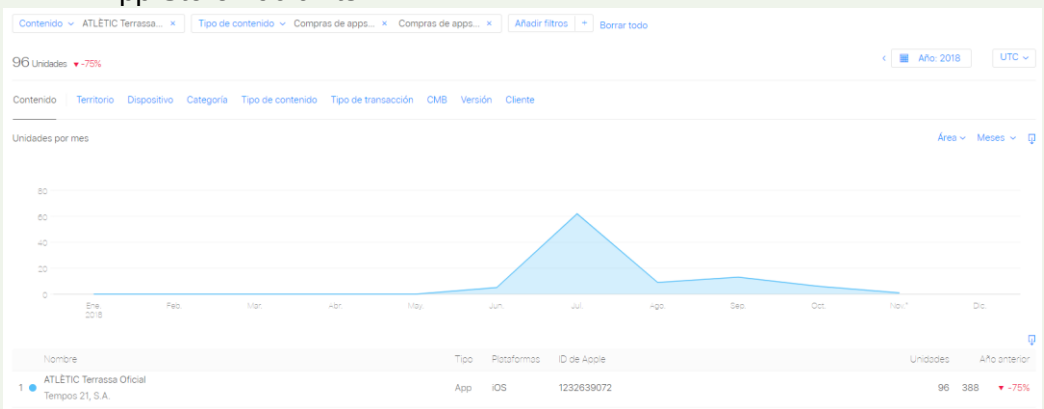
Test-card #	**SS-VA.01**	**Execution Status**	**PASSED**
Test Name	Use case specific KPIs for Visitor App		
Objectives	Validate all use case specific KPIs defined for Visitor App are covered and passed.		
KPIs to measure	DESCRIPTION	PRODEDURE	TARGET VALUE
	Number of app downloads from stores	Check app store specific metrics	More than 100 times
	Number of answers of questionnaire	Check Google Forms response	More than 20% of users
	More than 80% of answers are positive	Check Google Forms response	More than 80% positive answers

	as a general score		
	Percentage of shops showed and ordered by proximity	Check database	More than 80%
	Percentage of orders successfully completed	Check database, as only a few finished orders are hidden from REST responses (bandwidth optimization)	More than 80%
	Percentage of received promotions from retailers	Check database	More than 80%
Related Use Cases	*Smart Stadium*		
Designers	*Juan Belmonte Rodríguez*		
Related Test-cards	*SS-GN.01, SS-RA.01*		
Attachments	Questionnaire and answers: Google Forms answers (App ATLÈTIC Terrassa Oficial.csv)		

Trial resources and topology	
Required resources	*In order to run all tests, the following resources must be available:* - *Mobile application available in stores (Google Play and App Store)* - *Visitor Platform registered and accessible, with registered resources* - *Retailer Platform registered and accessible, with registered resources* - *Beacon platform registered and accessible, with registered resources* - *Promowall platform registered and accessible, with registered resources*

Topology details	

Test description

Step #	Step description and expected results	Status
1.	**Description**: Review Google Play and App Store metrics to check how many app installations occurred during trials **Expected Results:** • More than 100 installations **Measured Results:** • Google Play: 162 units • App Store: 96 units  **Comments**: *Any additional comment*	passed
2	**Description**: Review Google Forms answers to determine % of active users that answered it. **Expected Results:**	passed

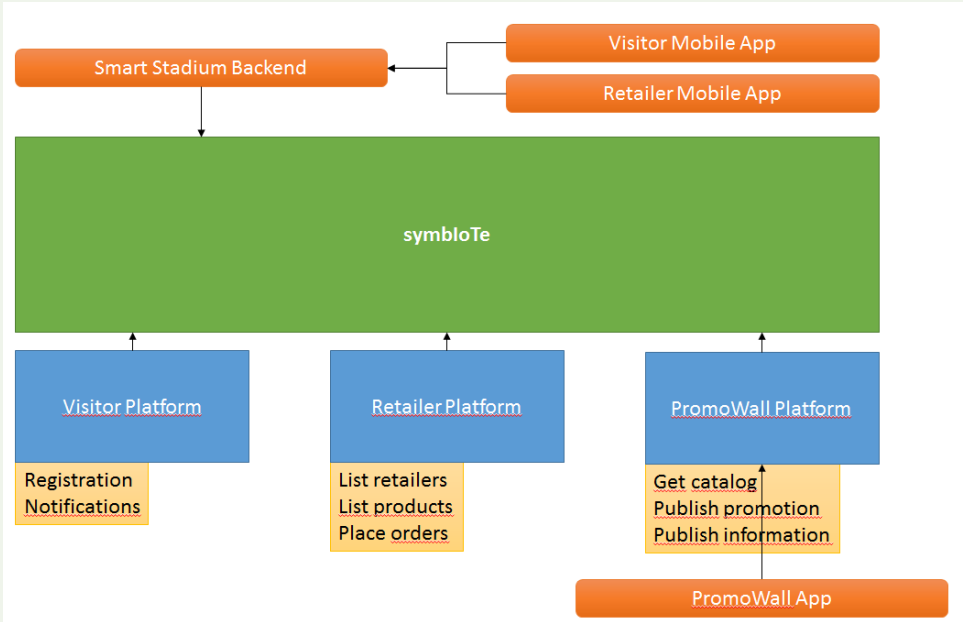
	• More than 20% of users **Measured Results:** • Number of users (from previous test): 258 • Number of questionnaire answers: 75 • Percentage of users that answered: 29,07% **Comments**: Any additional comment	
3	**Description**: Review Google Forms answers to determine % of positive scores. **Expected Results:** • More than 80% of positive answers **Measured Results:** • Number of answers (from previous test): 75 • Number of positive answers (all answers with value 3 or greater): 60 • Percentage of positive scores: 85,71% **Comments**: Any additional comment	passed
4	**Description**: Check number of shops showed and ordered by proximity. **Expected Results:** • More than 80% **Measured Results:** • Number of different users with active BLE scan during event: 42 • Number of registered retailers: 8 • Number of geolocated retailers: 8 • Percentage of geolocated retailers: 100% **Comments**: Any additional comment	passed
5	**Description**: Check number of orders successfully completed. **Expected Results:** • More than 80% **Measured Results:** • Number of orders in retailer's database: 27 • Number of finished orders in retailer's (completed or rejected by retailer): 27 • Percentage of successfully completed orders: 100% **Comments**: Any additional comment	passed
6	**Description**: Check number of promotions received from retailers. **Expected Results:** • More than 80% **Measured Results:** • Number of published promotions in retailer's database: 36 • Number of promotions listed in visitor's database (inbox): 36 • Percentage of promotions received: 100% **Comments**:	passed

	Any additional comment	

8.4.3 Retailer application evaluation

Test-card #	**SS-RA.01**	**Execution Status**	**PASSED**
Test Name	Use case specific KPIs for Retailer App		
Objectives	Validate all use case specific KPIs defined for Retailer App are covered and passed.		
KPIs to measure	DESCRIPTION	PRODEDURE	TARGET VALUE
	Percentage of orders sent by users and received by retailer	Check retailer's server logs and database	More than 80%
	Percentage of orders successfully completed	Check retailer's database	More than 80%
	Percentage of promotions successfully provided to users and Promowalls	Check retailer's and Promowall's databases	More than 80%
Related Use Cases	*Smart Stadium*		
Designers	*Juan Belmonte Rodríguez*		
Related Test-cards	*SS-GN.01, SS-VA.01*		

Trial resources and topology	
Required resources	In order to run all tests, the following resources must be available: - Mobile application available in stores (Google Play and App Store) - Visitor Platform registered and accessible, with registered resources - Retailer Platform registered and accessible, with registered resources - Beacon platform registered and accessible, with registered resources - Promowall platform registered and accessible, with registered resources

Topology details	

Test description

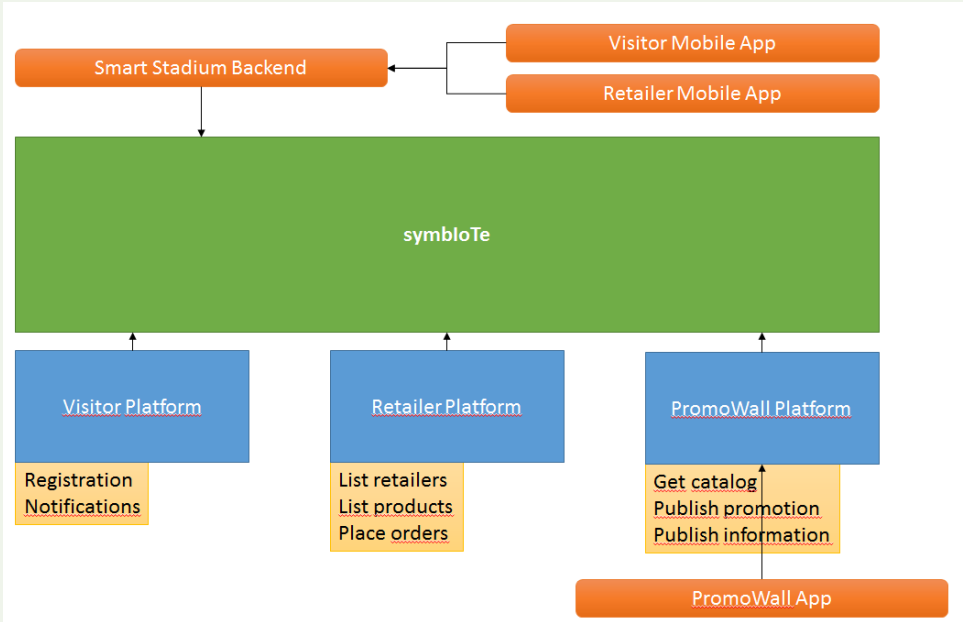
Step #	Step description and expected results	Status
1.	**Description**: Validate the quality of ordering service by checking the amount of orders sent by visitors and the amount of received ones by retailers **Expected Results:** • More than 80% **Measured Results:** • Number of orders in retailer's database: 27 • Number of errors in retailer's server logs regarding order registration process: 0 • Percentage of orders sent by users and received by retailer: 100% **Comments**: Any additional comment	passed
2	**Description**: Check number of orders successfully completed. **Expected Results:** • More than 80% **Measured Results:** • Number of orders in retailer's database: 27 • Number of finished orders in retailer's (completed or rejected by retailer): 27 • Percentage of successfully completed orders: 100% **Comments**: *Any additional comment*	passed
3	**Description**: *Check number of orders successfully completed.* **Expected Results:**	passed

	• More than 80% **Measured Results:** • Number of promotions in retailer's database: 36 • Number of promotions in visitor's database (inbox): 36 (all of them were addressed to visitors) • Number of promotions in Promowall's database: 30 (6 weren't published in Promowall, they were just push messages to visitors): 30 • Percentage of promotions successfully provided to users and Promowalls: 100% **Comments**: *Any additional comment*	

8.4.4 Promowall application evaluation

Test-card #	**SS-PW.01**	**Execution Status**	**PASSED**
Test Name	Use case specific KPIs for Promowall App		
Objectives	Validate all use case specific KPIs defined for Promowall App are covered and passed.		
KPIs to measure	DESCRIPTION	PRODEDURE	TARGET VALUE
	Percentage of received promotions from retailers	Check retailer's and Promowall's databases	More than 80%
Related Use Cases	*Smart Stadium*		
Designers	*Juan Belmonte Rodríguez*		
Related Test-cards	*SS-GN.01*		

Trial resources and topology	
Required resources	In order to run all tests, the following resources must be available: - Mobile application available in stores (Google Play and App Store) - Visitor Platform registered and accessible, with registered resources - Retailer Platform registered and accessible, with registered resources - Beacon platform registered and accessible, with registered resources - Promowall platform registered and accessible, with registered resources

Topology details	

Test description		
Step #	**Step description and expected results**	**Status**
1.	**Description**: Validate that retailers send promotions to be displayed in Promowalls across the stadium. **Expected Results:** • More than 80% **Measured Results:** • Number of promotions in all retailer's database: 36 • Number of promotions stored and published in Promowall: 30 • Percentage of received promotions from retailers: 83.33% **Comments**: *Any additional comment*	passed