Project Acronym:	**VICINITY**
Project Full Title:	**Open virtual neighbourhood network to connect intelligent buildings and smart objects**
Grant Agreement:	**688467**
Project Duration:	**48 months (01/01/2016 - 31/12/2019)**

Deliverable D5.2

VICINITY value-added services implementation framework

Work Package:	WP5 – Value-Added Services Implementation
Task(s):	T5.2 Implementation of value-added services
Lead Beneficiary:	GNOMON
Due Date:	30th September 2018 (M33)
Submission Date:	5th November 2018 (M35)
Deliverable Status:	Final
Deliverable Type:	D
Dissemination Level:	PU
File Name:	**VICINITY_D5_2_VICINITY_value-added_services_implementation_framework_1.0.pdf**

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°688467

VICINITY Consortium

No	Beneficiary		Country
1.	TU Kaiserslautern (Coordinator)	UNIKL	Germany
2.	ATOS SPAIN SA	ATOS	Spain
3.	Centre for Research and Technology Hellas	CERTH	Greece
4.	Aalborg University	AAU	Denmark
5.	GORENJE GOSPODINJSKI APARATI D.D.	GRN	Slovenia
6.	Hellenic Telecommunications Organization S.A.	OTE	Greece
7.	bAvenir s.r.o.	BVR	Slovakia
8.	Climate Associates Ltd	CAL	United Kingdom
9.	InterSoft A.S.	IS	Slovakia
10.	Universidad Politécnica de Madrid	UPM	Spain
11.	Gnomon Informatics S.A.	GNOMON	Greece
12.	Tiny Mesh AS	TINYM	Norway
13.	HAFENSTROM AS	HITS	Norway
14.	Enercoutim – Associação Empresarial de Energia Solar de Alcoutim	ENERC	Portugal
15.	Municipality of Pilea-Hortiatis	MPH	Greece

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

[1] ***Deliverable Type:***

R:	Document, report (excluding the periodic and final reports)
DEM:	Demonstrator, pilot, prototype, plan designs
DEC:	Websites, patents filing, press & media actions, videos, etc.
OTHER:	Software, technical diagram, etc.

[2] ***Dissemination level:***

PU:	Public, fully open, e.g. web
CO:	Confidential, restricted under conditions set out in Model Grant Agreement
CI:	Classified, information as referred to in Commission Decision 2001/844/EC.

Authors List

Leading Author (Editor)				
Surname		First Name	Beneficiary	Contact email
Zandes		Dimitrios	GNOMON	d.zandes@gnomon.com.gr
Co-authors (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Abreu	Vítor	ENERC	v.abreu@enercoutim.eu
2.	Belesioti	Maria	OTE	mbelesioti@oteresearch.gr
3.	Berler	Alexander	GNOMON	a.berler@gnomon.com.gr
4.	Filosofov	Dmitry	TINYM	dmitry@tiny-mesh.com
5.	Guan	Yajuan	AAU	ygu@et.aau.dk
6.	Heinz	Christopher	UNIKL	heinz@cs.uni-kl.de
7.	Hovstø	Asbjorn	HITS	hovsto@online.no
8.	Karkaletsis	Kostas	GNOMON	k.karkaletsis@gnomon.com.gr
9.	Koutli	Maria	CERTH	mkoutli@iti.gr
10.	Larsen	Ruben	TINYM	ruben@tiny-mesh.com
11.	Oliveira	João	ENERC	j.oliveira@enercoutim.eu
12.	Poljakov	German	TINYM	german@tiny-mesh.com
13.	Samovich	Natalie	ENERC	n.samovich@enercoutim.eu
14.	Stavropoulos	Liverios	GNOMON	l.stavropoulos@gnomon.com.gr
15.	Sundvor	Mariann	TINYM	mariann@tiny-mesh.com
16.	Sveen	Flemming	HITS	flsveen@online.no
17.	Theologou	Natalia	CERTH	nataliath@iti.gr
18.	Tryferidis	Athanasios	CERTH	thanasic@iti.gr

Reviewers List

List of Reviewers (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Heinz	Christopher	UNIKL	heinz@cs.uni-kl.de
2.	Perea Escribano	Carmen	ATOS	carmen.perea@atos.net
3.	Wall	Nigel	CAL	nw@nigel-wall.co.uk

Revision Control

Version	Date	Status	Modifications made by
0.1	7 July 2018	Initial TOC	Zandes (GNOMON)
0.2	30 August 2018	Updated Draft	Koutli, Theologou, Tryferidis (CERTH)
0.3	6 September 2018	Updated Draft	Zandes (GNOMON)
0.4	27 September 2018	Updated Draft	Koutli, Theologou, Tryferidis (CERTH), Zandes, Karkaletsis, Stavropoulos, Berler (GNOMON), Belesioti (OTE)
0.5	5 October 2018	Updated Draft – Oslo Pilot Site Use Cases consolidated	Zandes, Stavropoulos (GNOMON), Filosofov, Larsen, Poljakov, Sundvor (TINYM)
0.6	18 October 2018	Updated Draft – Martim Longo Pilot Site Use Cases partially consolidated, AAU Testing Use Case consolidated, ANNEX II on homomorphic encryption consolidated	Zandes (GNOMON), Abreu, Samovich (ENERC), Heinz (UNIKL), Guan (AAU)
0.7	19 October 2018	Updated Draft – Martim Longo Pilot Site Use Cases consolidated, Tromso Pilot Site Use Case partially consolidated, Conclusion consolidated	Zandes (GNOMON), Abreu, Samovich, Oliveira (ENERC), Sveen, Hovstø (HITS), Heinz (UNIKL)
0.8	22 October 2018	Updated Draft – All Pilot Site Use Cases fully consolidated, updated version for QAR	Zandes, Karkaletsis (GNOMON), Filosofov, Larsen (TINYM), Sveen, Hovstø (HITS), Abreu, Samovich, Oliveira (ENERC), Koutli, Theologou (CERTH)

0.9	1 November 2018	Final version for submission after QAR results	Zandes, Berler (GNOMON), Larsen, Sundvor (TINYM), Sveen, Hovstø (HITS), Abreu, Samovich, Oliveira (ENERC), Koutli, Theologou (CERTH)
1.0	5 November 2018	Submission to the EC	Carna Zivkovic

Executive Summary

The current deliverable provides an overview of the Value-Added Services developed under the concept of each Use Case within Task T5.2. In fact, it is a continuation of previously submitted work as both Use Cases and Value-Added Services were identified and defined in detail in deliverable D5.1. It contains information per Use Case (plus one testing case), in a meaningful tabular form, that focuses on a wide range of details such as functionalities delivered per Value-Added Service, technical/deployment details per Value-Added Service, respective User Interfaces used, ethical issues, etc.

This deliverable (D5.2 – Value-Added Services implementation framework) is the last and most important part to reach milestone MS6 which was aiming at making a first version of VICINITY Value-Added Services available.

To make the included information as meaningful as possible, the deliverable is organized in a straightforward manner, specifically, in sections/chapters that correspond to each Use Case and contain tables of summarized information for the respective Value-Added Services and User Interfaces plus a section of possible ethical issues per case and the way they were taken into consideration during the implementation process.

Since the implemented Value-Added Services are prototypes, the last chapter provides the future steps to follow, having also in mind Task T5.3 (i.e. Continuous operation and upgrades of value-added services), plus conclusions made on whether the deliverable's objectives were achieved.

Finally, two annexes close the document. The first contains information regarding the templates used on the deliverable's structure for the Value-Added Service/User Interface description, whereas the second one presents the concept of homomorphic encryption to be used as a possible extension for data anonymization/privacy.

Table of Contents

Executive Summary 7

1. Introduction 13

1.1. Context within VICINITY 13
1.2. Deliverable objectives and scope... 14
1.3. Structure of the Deliverable... 14

2. Oslo Pilot Site (NO) – Buildings (TINYM) 16

2.1. Use Case 1a.1 – Predictive operations.. 16
2.1.1. VICINITY Value-Added services... 16
2.1.2. User Interfaces 18
2.2. Use Case 1a.2 – Resource management 20
2.2.1. VICINITY Value-Added services... 22
2.2.2. User Interfaces 23
2.3. Ethics 25
2.4. VICINITY Value.. 25

3. Tromsø (NO) – Neighbourhood Smart Parking Assisted Living ecosystem (HITS) 27

3.1. Pilot Use Case 1b.1: Shared parking/priority parking 27
3.1.1. VICINITY Value-Added services... 28
3.1.2. User interfaces.. 30
3.2. Pilot Use Case 1b.2: eHealth Emergency parking 35
3.2.1. VICINITY Value-Added services... 36
3.2.2. User Interfaces 39
3.3. Ethics 41
3.4. VICINITY Value.. 41

4. Martim Longo (PO) – Neighbourhood GRID ecosystem (ENERC)..............................43

4.1. Use Cases 2.1, 2.2, 2.3, 2.4, 2.5, 2.10, 2.11 – Energy Efficiency and IEQ Management in Municipal Cluster of buildings – Municipal Services 43
4.1.1. VICINITY Value-Added services... 44
4.1.2. User interfaces.. 45
4.2. Use Case 2.9 – UV (Ultraviolet radiation) info services for Citizens and Tourists – Local to Local Services.. 46
4.2.1. VICINITY Value-Added Services 47
4.2.2. User interfaces.. 47
4.3. Use Cases 2.6, 2.7 – Distributed Energy assets management – Platform Services...................... 48
4.3.1. VICINITY Value-Added Services 48
4.3.2. User Interfaces 49
4.4. Ethics 50
4.5. VICINITY Value.. 50

5. Pilea-Hortiatis (GR) – eHealth & Assisted Living (CERTH – GNOMON – MPH) 52
5.1. Use Case 3.1 – eHealth and Assisted Living for elderly people at home 52
5.1.1. VICINITY Value-Added Services 52
5.1.2. User Interfaces 58
5.2. Use Case 3.2 – Health improvement for the middle-aged persons 61
5.2.1. VICINITY Value-Added Services 62
5.2.2. User Interfaces 67
5.3. Ethics 74
5.4. VICINITY Value 74
6. AAU Microgrid-IoT Testing Lab (DK) – Residential Microgrid & Smart Parking (AAU) 76
6.1. Testing Case – Optimal usage of parking slots by considering energy costs 76
6.1.1. VICINITY Value-Added Services 76
6.1.2. User Interfaces 77
6.2. Ethics 79
6.3. VICINITY Value 79
7. Conclusion 80
8. References 81
Annex I – Templates used 82
Value-Added Services Template 82
User Interfaces Template 82
ANNEX II – Homomorphic Encryption for data anonymization 83
Motivation and Impact 83
Anonymous Data Aggregation as a Micro Service for Value-Added Services 83
Homomorphic encryption 84
Applications in VICINITY pilots and intended prove-of-concept 84
Integration into VICINITY components 85

List of Tables

List of Figures

Figure 1-1 Context of D5.2 within VICINITY 13
Figure 2-1 Use Case 1a.1 - Overview 16
Figure 2-2 Use Case 1a.2 - Overview 21
Figure 3-1 Use Case 1b.1 associated Value-Added Services and User-Interfaces 27
Figure 3-2 Use Case 1b.2 associated Value-Added Services and User Interfaces.. 36
Figure 4-1 Pilot Site Overview.. 43
Figure 5-1 Use Case 3.1 - Overview 52
Figure 5-2 Use Case 3.2 - Overview 62
Figure ANNEX II-1: Integrating Homomorphic Encryption into the VICINITY Architecture 86

List of Definitions & Abbreviations

Abbreviation	Definition
CoAP	Constrained Application Protocol
CRUD	Create, Read, Update, Delete
DLT	Distributed Ledger Technology
DoW	Description of Work
DPA	Data Protection Authority
DTLS	Datagram Transport Layer Security
EC	European Commission
EU	European Union
FHIR	Fast Healthcare Interoperability Resources
GDPR	General Data Protection Regulation
HVAC	Heating, Ventilation and Air Conditioning
IDE	Integrated Development Environment
IEQ	Indoor Environmental Quality
IoT	Internet of Things
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
LoRaWAN	Long Range Wide Area Network
MAC	Media Access Control
MS	Milestone
O&M	Observations and Measurements
P2P	Peer to Peer
PC	Personal Computer
POI	Point Of Interest
POJO	Plain Old Java Object
SDK	Software Development Kit
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol

UC	Use Case
UDP	User Datagram Protocol
UI	User Interface
UV	Ultraviolet
VAS	Value-Added Service

1. Introduction

The deliverable describes the implementation of the Value-Added Services (VAS) developed by the relevant responsible partners under the scope of each pilot Use Case (UC).

The document focuses on delivering a technical overview of the offered solutions for each UC as a continuation of D5.1 where the VASs were first defined. However, it also tries to underline any possible ethical issues arisen in each pilot and the level they were taken into consideration.

1.1. Context within VICINITY

Figure 1-1 gives an overview of the context of D5.2 within VICINITY. As already mentioned, D5.2 is the most important step to reach Milestone 6 (MS6) which marks the implementation of the UCs and VASs defined in D5.1, at least in a first prototype version, with the use of the VICINITY server components/services and client infrastructures that were made available by MS5.

The work also has been extended to reflect the outcomes of discussions with the reviewers in the Brussels Technical Check meeting. It reflects a reaction of VICINITY on the reviewers' recommendation: *"It is expected that the project makes its intentions clear on DLT. What can be offered to the LSPs and IoT FA. It is recommended that the project (in particular the academic partners) show more willingness to experiment with DLT and IoT."*

UNIKL investigates that direction, but not based on DLT instead on homomorphic encryption technology. The approach is described in the Annex II of this document.

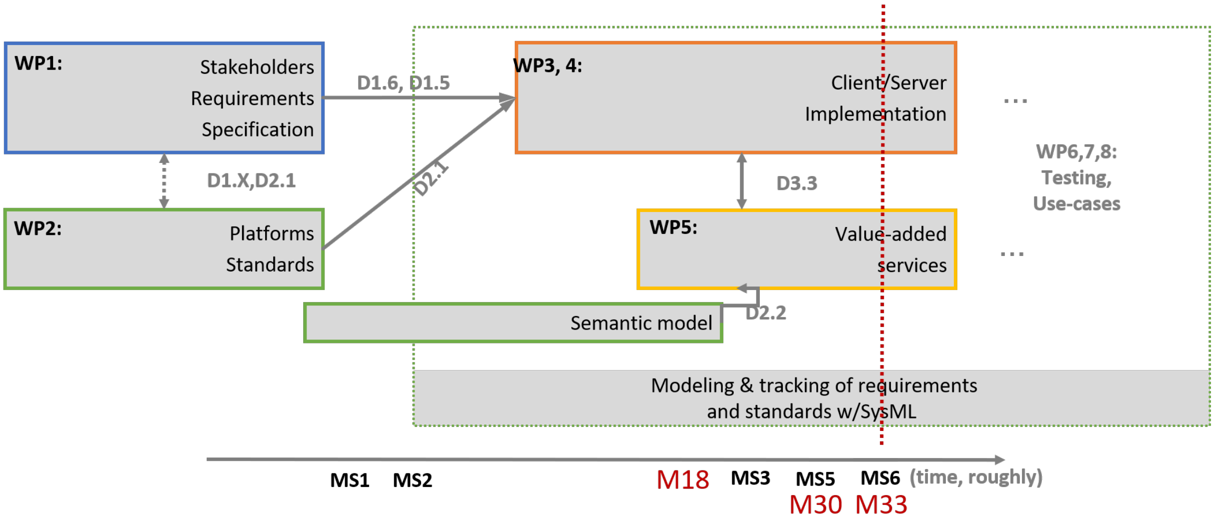


Figure 1-1 Context of D5.2 within VICINITY

Regarding the relation to other deliverables, the current document builds on the results of previous deliverables, specifically:

- **D5.1:** Value-added services definition requirements and architectural design

The outcome of this deliverable will form the basis of work for the following deliverables:

- **D5.3:** Value-added services deployment, validation, upgrade and evaluation
- **D7.2-D7.5:** Report on each Pilot's Use Cases installation
- **D8.2-8.5:** Pilot results for each Use Case
- **D8.6:** Evaluation of user experience and performance of VICINITY framework & value-added services
- **D9.4:** Finalizing the data specification and data management plan

1.2. Deliverable objectives and scope

The deliverable shapes an overview of the VASs implemented under the scope of each pilot Use Case, further summarizing functionalities delivered per VAS, examples of use, technical/deployment requirements per VAS, User Interfaces (UI) created for each case and possible pending issues for the future.

In previous deliverables, mainly D5.1, a VAS was defined as a piece of software that is fully integrated with the VICINITY infrastructure and implements an algorithm. The latter could be either a simple calculation or a more advanced operation, there are literally no boundaries on that. Input is provided by relevant IoT data from other IoT infrastructures that are part of the VICINITY network and could also be considered as VICINITY entities. As a result, an additional goal set for D5.2 is to offer a summary of all the algorithms running under each VAS plus the input they require to produce the expected results.

Furthermore, as the UCs deal with real-life scenarios, it is expected that various ethical issues might arise per case. Therefore, one of the main objectives of the current deliverable is to highlight those issues and describe the way they have been taken into consideration and have been overcome during implementation.

1.3. Structure of the Deliverable

The report contains 8 chapters and 2 annexes. In particular:

- **Chapter 1**: Introduction to the deliverable regarding its scope, objectives and defining the related documents.
- **Chapter 2:** Oslo Science Park (NO) – Buildings and Smart Transport Use Cases and VAS.
- **Chapter 3**: Tromsø (NO) – Neighbourhood Smart Parking Assisted Living eco-system Use Cases and VAS.
- **Chapter 4**: Martim Longo (PO) – Neighbourhood GRID ecosystem Use Cases and VAS.
- **Chapter 5**: Pilea-Hortiatis (GR) – eHealth & Assisted Living Use Cases and VAS.
- **Chapter 6**: AAU Microgrid-IoT Testing Lab (DK) – Residential Microgrid & Smart Parking (AAU)
- **Chapter 7**: Conclusion
- **Chapter 8**: References
- **ANNEX I**: Templates used
- **ANNEX II**: Homomorphic Encryption for data anonymization

2. Oslo Pilot Site (NO) – Buildings (TINYM)

2.1. Use Case 1a.1 – Predictive operations

The current use case deals with providing staff with data on room usage and notifications on mobile devices when a room needs cleaning. The data is collected by IoT sensors which detect when a door has been opened. The user may set thresholds to be triggered when a room needs cleaning. A given notification message, defined by the user as well, will be sent when the threshold is reached.

The Value-Added Service will provide cleaning and waste collection teams with information and alarms that can improve their efficiency and the quality of their service offerings.

The team can set and customise alarms with threshold values for the number of person movements. An alarm is then issued if a toilet or room has been used a certain amount of times. In summary, the service helps to reorganise cleaning and waste removal from a frequency based to an on-demand system.

Below a figure showing VICINITY as facilitator, services, and the use case of the Predictive Operations.

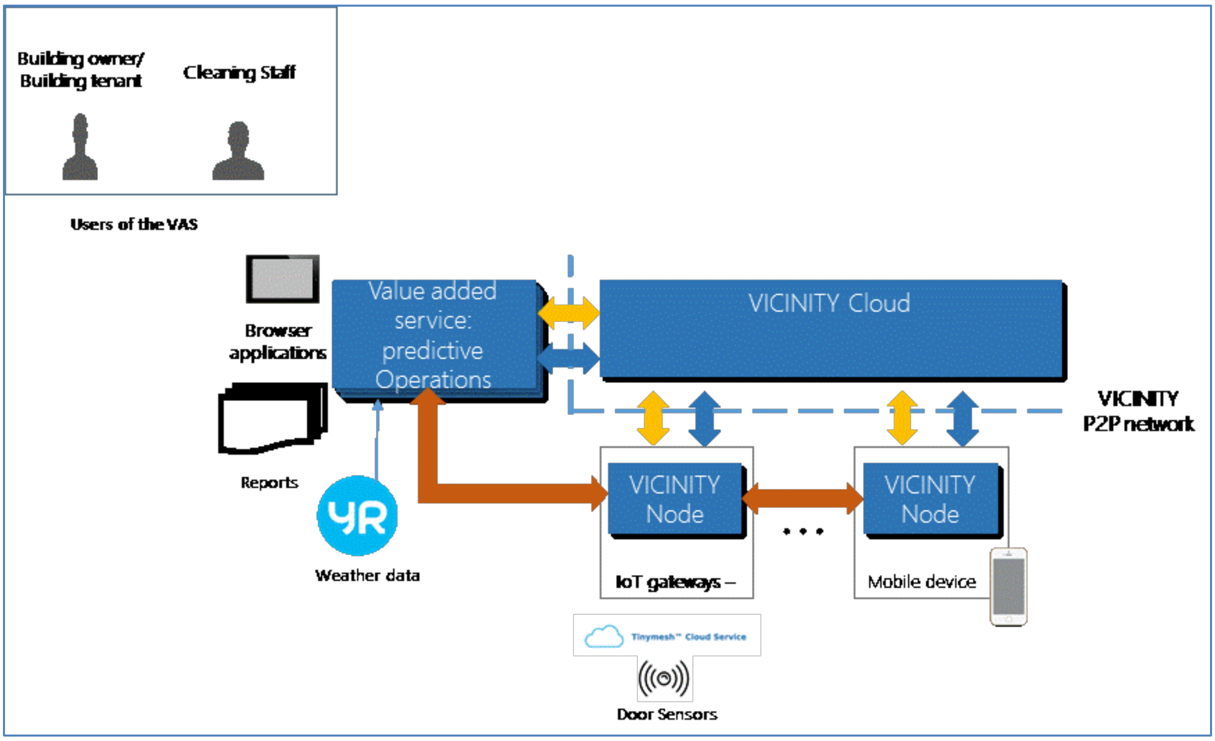


Figure 2-1 Use Case 1a.1 - Overview

2.1.1. VICINITY Value-Added services

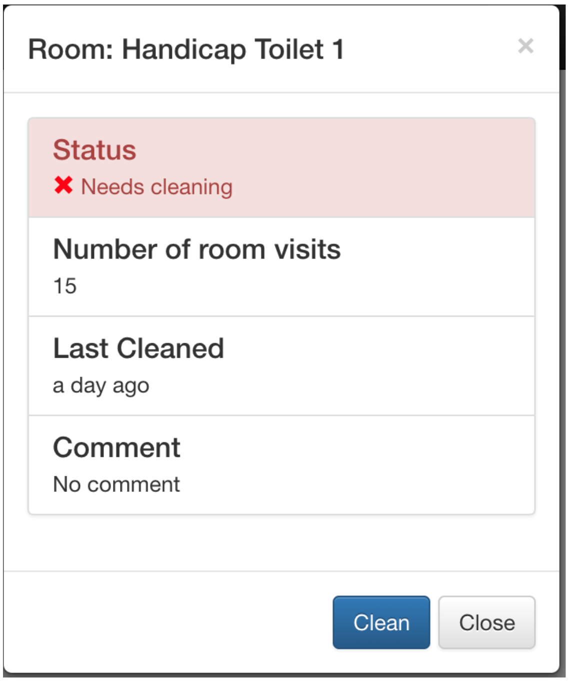
2.1.1.1. Cleaning and waste removal notification service and warning

VAS 1a.1.1	Cleaning and waste removal notification service and warning
Related Use Case	UC 1a.1 – Predictive operations
Functionalities delivered	**Get Admin**

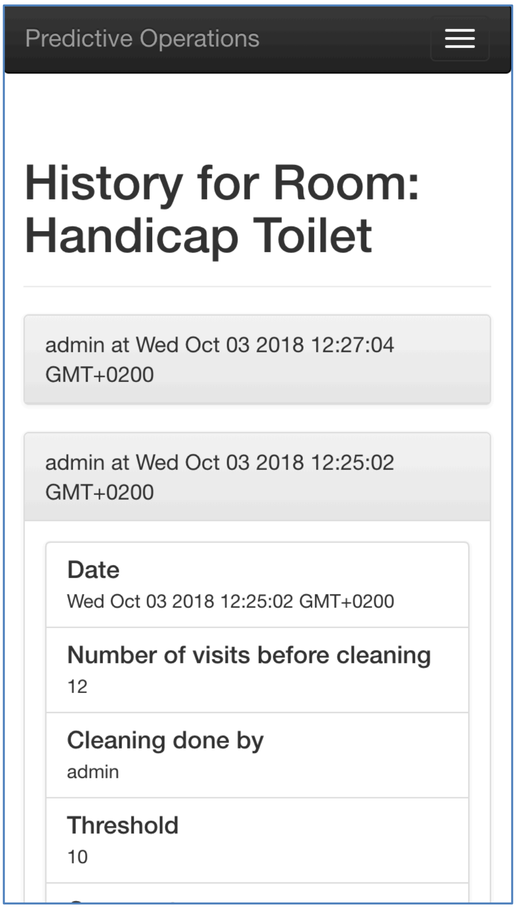
	○ Functionality: The service returns a list of rooms, allowing the user to access details and/or mark them as cleaned when desired. *notice: internal functionality for the VAS to work properly **Get Room** ○ Functionality: The service returns data on a room provided by a door sensor that registers the number of times a room has been visited. *notice: internal functionality for the VAS to work properly **Send Message** ○ Functionality: The service sends a customizable SMS message when the number of room visits reaches a predefined threshold. *notice: internal functionality for the VAS to work properly **Set Clean** ○ Functionality: Resets the counter of visits for a room, indicating that it has been cleaned. *notice: internal functionality for the VAS to work properly **Receive Door Event** ○ Functionality: The service returns data on a room provided by a door sensor that registers the number of times a room has been visited. The service will then offer other actions in response to events. ○ Endpoint: /objects/{oid}/events/{eid}
Selected implementation algorithms	The VAS compiles a list of rooms with requisite door sensors and keeps track of the number of times each individual door sensor has been triggered. When the sensor is triggered enough times to reach its threshold, an SMS message is sent to the assigned phone number. The rooms can have their visit counters reset by being marked as cleaned.
Deployment/Hosting of VAS	The VAS will run on an Azure server. Python with a Django framework is used alongside with PostgreSQL as the selected database system.
Pending issues	Potential increase in the amount of rooms and users. Furthermore, Open Call sensors are likely to be used to more accurately assess how many individuals visit a room, in combination with other room usages besides cleaning.
Deviations/changes	Potential change in the type of room sensor to improve the accuracy of measurements. The VICINITY agent version will be updated as well, and consequently some endpoints may change.

2.1.2. User Interfaces

2.1.2.1. Reports

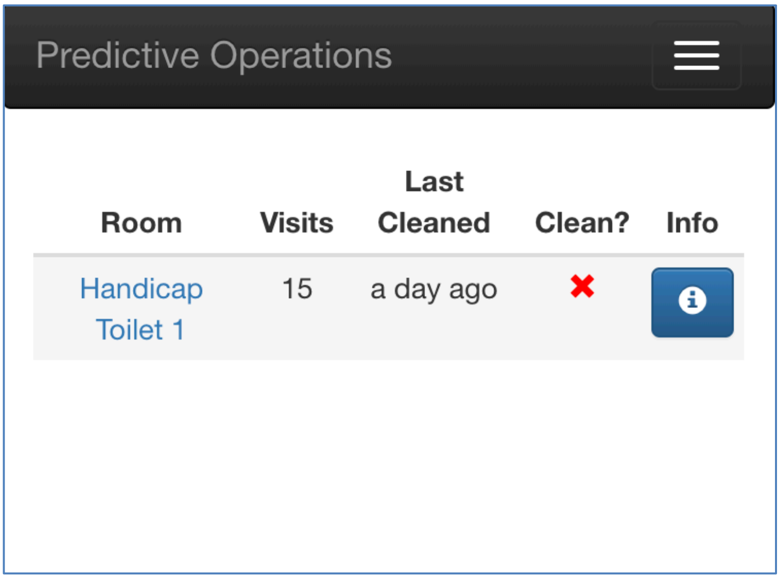
UI 1a.1.1	**Reports**
Related Use Case	UC 1.a.1 – Predictive operations
Functionalities delivered	The UI lets the user view the current cleaning status of a room and to reset the visit counter by marking the room as cleaned.
Example of use	A user receives a notification indicating that a room has reached its visit threshold. He may view the last time it was cleaned and the number of visits it has received since. After the room receives maintenance, the user may then mark the room as clean. The latter action resets the visit counter.
Screenshots	 Room: Handicap Toilet 1 Status ✖ Needs cleaning Number of room visits 15 Last Cleaned a day ago Comment No comment
Deviations/changes	The UI was developed on site as opposed to relying on Power Bi and other products.

2.1.2.2. Individual and statistical analysis of rooms

UI 1a.1.2	**Individual and statistical analysis of rooms**
Related Use Case	UC 1.a.1 – Predictive operations
Functionalities delivered	Each individual room has a history with data on: • when a room has been accessed or cleaned • what the visit threshold is • how often it has been visited
Example of use	The user views the history of a room via the UI. The latter provides past data on its use, allowing the user to prepare new thresholds or check when the room was last accessed by staff.
Screenshots	 Predictive Operations History for Room: Handicap Toilet admin at Wed Oct 03 2018 12:27:04 GMT+0200 admin at Wed Oct 03 2018 12:25:02 GMT+0200 Date Wed Oct 03 2018 12:25:02 GMT+0200 Number of visits before cleaning 12 Cleaning done by admin Threshold 10
Deviations/changes	The UI was developed on site as opposed to relying on Power Bi and other products.

Horizon 2020
European Union funding
for Research & Innovation

2.1.2.3. Room usage and warnings

UI 1a.1.3	**Room usage and warnings**
Related Use Case	UC 1.a.1 – Predictive operations
Functionalities delivered	Each individual room allows the user to: • view the amount of times the room has been visited • reset the visit counter (by marking it as cleaned) • view when the room was last cleaned • set a threshold when a notification message should be sent • access room history
Example of use	The user establishes a threshold for a room via the UI as well as a corresponding message triggered when this threshold is reached. Therefore, once the visit counter reaches the threshold the message will be sent. The counter shall be reset via the UI to mark the room as cleaned.
Screenshots	 Predictive Operations Room | Visits | Last Cleaned | Clean? | Info Handicap Toilet 1 | 15 | a day ago | ✖ | ℹ
Deviations/changes (if applicable)	None

2.2. Use Case 1a.2 – Resource management

The current use case collects utility data, compiling it and graphing it according to user specification. The user may use the data for setting alarms to be triggered when utility consumption exceeds desired thresholds and potentially manage the energy consumption of appliances and processes automatically. The data is collected via IoT devices that provide information on water and electricity consumption, while access to IoT devices in the pilot site is handled via local partners.

The service will support daily operations and decision making on energy management, load optimisation and Heating, Ventilation and Air Conditioning (HVAC) control. Combined with weather forecast data from YR[1] the service can predict upcoming electricity peak loads based on estimated needs for additional heating or cooling of the premises. The service generates value through electricity load harmonization to save on tariffs, as well as to discover consumption trends for improved operations planning.

Below a figure showing VICINITY as facilitator, services, and the use case of the Resource management.

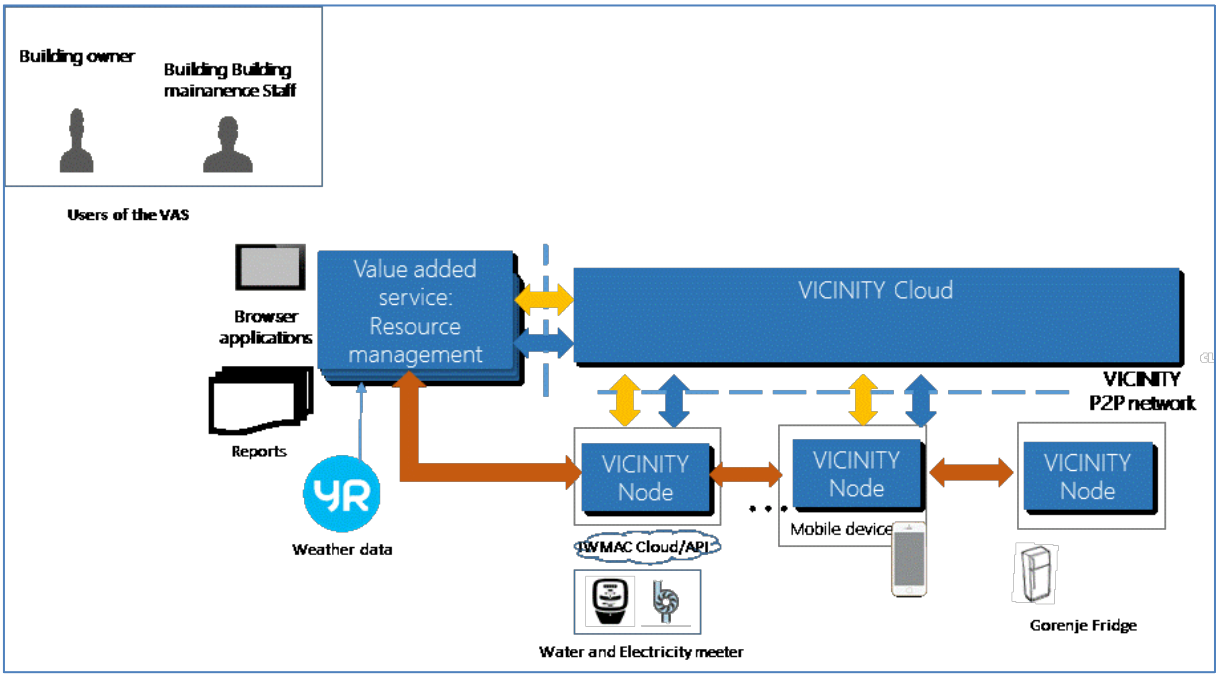


Figure 2-2 Use Case 1a.2 - Overview

[1] https://yrkundesenter.zendesk.com/hc/en-us

2.2.1. VICINITY Value-Added services

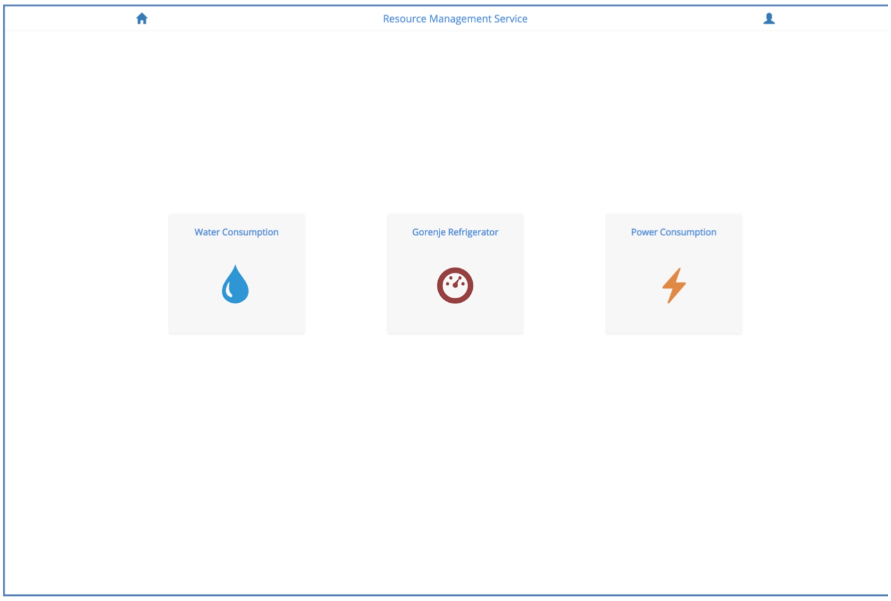
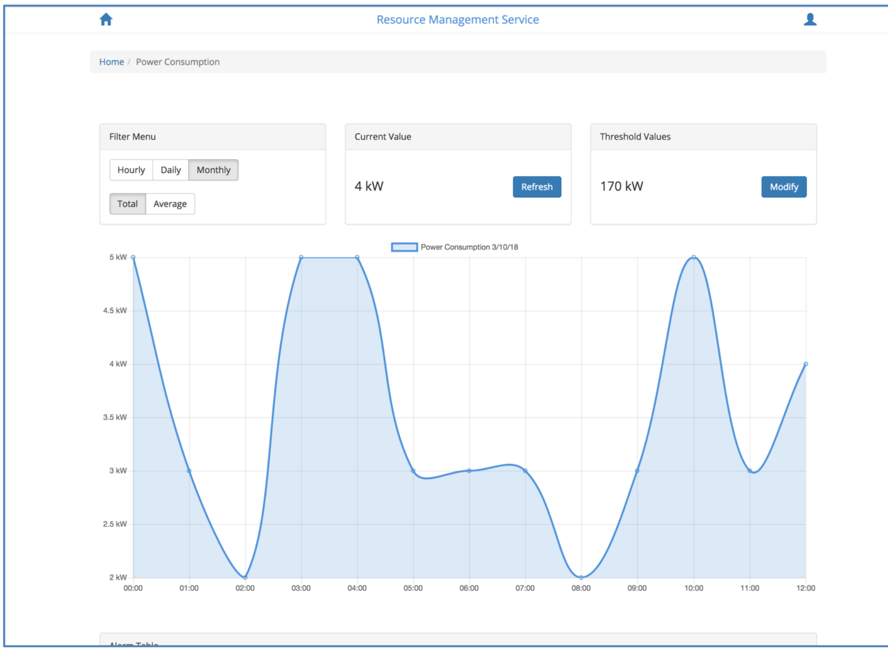
2.2.1.1. Resource consumption and alarm service

VAS 1a.2.1	**Resource consumption and alarm service**
Related Use Case	UC 1a.2 – Resource management
Functionalities delivered	**Get Admin** ○ Functionality: Returns a list of available utility data sets. **notice*: internal functionality for the VAS to work properly **Receive Water Consumption** ○ Functionality: Returns a graph of water consumption on a time basis (daily, weekly, monthly), a list of alarms (when thresholds are exceeded) and a list of histories. ○ Endpoint: /objects/{oid}/events/{eid} **Receive Electricity Consumption** ○ Functionality: Returns a graph of electricity consumption on a time basis (daily, weekly, monthly), a list of alarms (when thresholds are exceeded) and a list of histories. ○ Endpoint: /objects/{oid}/events/{eid} **Set Gorenje refrigerator settings** ○ Functionality: Reads and manipulates temperature settings from the Gorenje refrigerator. ○ Endpoint: /objects/{oid}/properties/{pid} **Get Gorenje refrigerator temperature** ○ Functionality: Reads temperature from the Gorenje refrigerator. ○ Endpoint: /objects/{oid}/properties/refrigerator_temperature
Selected implementation algorithms	IWMAC, a utility data company, provides utility data, via a device on their network, which are then compiled, visualized and used in predictive cases. The latter will then be used to automatically manage appliance's energy usage in relation to energy tariffs.
Deployment/Hosting of VAS	While the calculations and handling of data are done locally, utility data is provided via an off-site device in the hands of IWMAC.
Pending issues	None
Deviations/changes	The {eid} is a unique property of the device used in each case and, therefore, not suitable for specific references on the endpoints above.

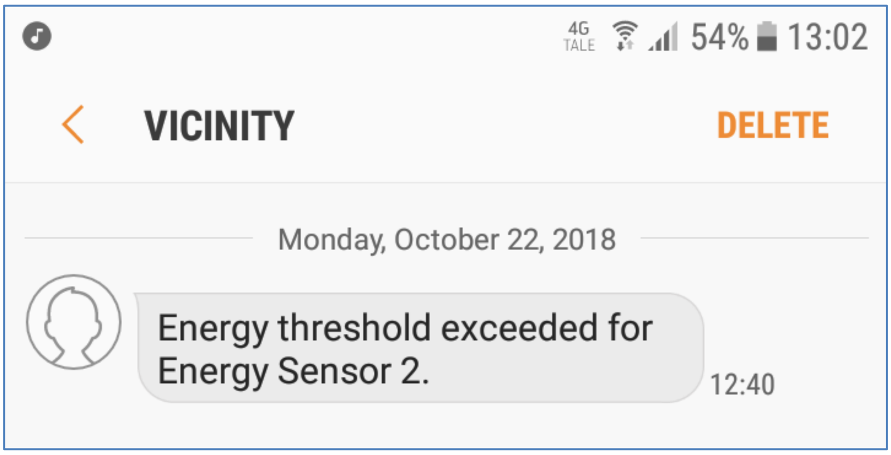
2.2.2. User Interfaces

2.2.2.1. Individual data for each meter

UI 1.a.2.1	**Individual data for each meter**
Related Use Case	UC 1a.2 – Resource management
Functionalities delivered	The menu allows the user to: • return to the home screen • access profile • access the different consumption pages Each consumption page allows the user to: • change the timescale of the graph (hourly, daily, monthly) • change the type of data (total, average) • refresh the current consumption value • modify the alarm threshold
Example of use	The user clicks the energy icon (“Power Consumption”) on the menu to view energy consumption data. On the energy consumption page, by clicking the “monthly” tab, the graph changes to a 30-day time line. The latter allows the user to view a history of data and to, thereby, compare it to the current value. The user can also set an alarm to record the times when the consumption levels surpass the desired threshold.

Screenshots	 Resource Management Service Water Consumption Gorenje Refrigerator Power Consumption  Resource Management Service Home / Power Consumption Filter Menu Hourly Daily Monthly Total Average Current Value 4 kW Refresh Threshold Values 170 kW Modify Power Consumption 3/10/18 5 kW 4.5 kW 4 kW 3.5 kW 3 kW 2.5 kW 2 kW 00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 Alarm Table
Deviations/changes	None

2.2.2.2. Triggering alarm

UI 1.a.2.2	Alarm on Mobile Device
Related Use Case	UC 1a.2 – Resource management
Functionalities delivered	The UI will notify the user via a mobile notification when the current value reaches the threshold.
Example of use	Assuming a water leak impacts water usage, the UI will notify the user once the current consumption surpasses the threshold.
Screenshots	 Monday, October 22, 2018 Energy threshold exceeded for Energy Sensor 2. 12:40
Deviations/changes	None

2.3. Ethics

The data the VASs collect in both UCs is anonymous. Beyond that, there is also an agreement, in accordance with relevant national law and regulations, with local partners who also regulate this data.

2.4. VICINITY Value

Based on the initial identification of how VICINITY acts as an enabler for the specific use cases (based on the information provided in D5.1 under sections 3.2.6, 3.3.6), the assessed added value provided is summarized in the following tables,

Value of VICINITY to use case 1a.1	
Problem	Solution
Lack of statistics and demonstrations of value for cleaning personnel	The UI allows cleaning staff to demonstrate their value to current or potential employers. The usage statistics can prove whether a room has been serviced sufficiently and will allow the user to apply the right amount of attention to individual rooms. VICINITY provides long term aggregate data, allowing staff to compile data over clients and locations.
Contextual data that may affect user tasks	VICINITY allows importing weather data to allow for more optimized sensors thresholds and thereby a better user experience.

Value of VICINITY to use case 1a.2	
Problem	Solution
Availability of utility consumption data	IWMAC provides data from their server via an adapter connected to their network. This data is then accessed remotely through Vicinity and used on site.
Controlling appliances remotely	VICINITY allows the user to automatically manage the electricity consumption of other devices connected to Vicinity in response to the overall consumption. VICINITY facilitates the automation of these tasks, while providing statistics.

3. Tromsø (NO) – Neighbourhood Smart Parking Assisted Living ecosystem (HITS)

3.1. Pilot Use Case 1b.1: Shared parking/priority parking

The current Use Case deals with will sharing private and public areas known as parking space. Currently, 25% to 42% of city traffic consists of vehicles looking for free parking (ref. Grant Agreement). Drivers routinely search for nearby parking to their destination. The scenario involves authorized healthcare personnel which routinely plan their journey by car to visit their care persons in a city centre. Additionally, private and public owners of a parking facility are involved by offering their parking facility when not in use.

A simple ticketing service is offered to the benefit of parking space owner and vehicle user. A schematic representation of the Use Case in relation to VICINITY architecture (i.e. Organisations, Users, Things, Friendship, etc.) and the way information is combined and distributed to the different actors is shown in **Fehler! Verweisquelle konnte nicht gefunden werden.**, but all functionality will not be demonstrated.

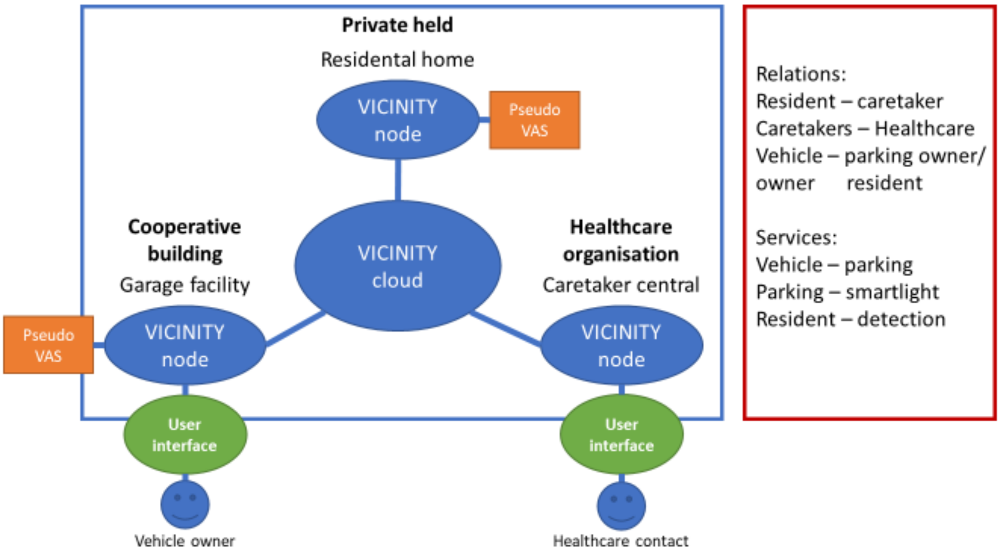


Figure 3-1 Use Case 1b.1 associated Value-Added Services and User-Interfaces

On the top level, this use case offers real-time operation of underground parking facilities. Ownership and administration of the parking space will be handled by the building manager and the owner of the physical parking space. “Teaterkvarteret” and “Himmel&Hav” consist of community organised parking space owned by the tenants. The ownership of parking facility is handled by the cooperative but are contractually allocated tenants of the building cluster.

Currently, the main target group that will be assigned priority parking is healthcare personnel, care centre and blue light agencies (medical, fire department and police). Further developments might also include drivers/passengers with physical disabilities, other residents, etc.

The VASs set out to offer solutions addressing,

- priority parking allocation based on booking ID
- eHealth priority parking, blue light agencies
- support for adaptable car sizes and position in relevance to needs
- triggers from building (e.g. panic button) and assisted living (e.g. smart appliances)

The priority parking will be contract based. Three kinds of contracts are envisioned,

- short time contracts (ad-hoc requests)
- long term contracts (daily between certain hours)
- transferral of ownership (not to be implemented)

Additionally, different sets of subscription services might be implemented.

3.1.1. VICINITY Value-Added services

3.1.1.1. Offline Management

VAS 1b.1.1	**Offline management**
Related Use Case	UC 1b.1 – Shared parking/Priority parking
Functionalities delivered	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Selected implementation algorithms	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Deployment/Hosting of VAS	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Pending issues	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Deviations/changes	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)

3.1.1.2. Real-time operation

VAS 1b.1.2	**Real-time operation**
Related Use Case	UC 1b.1 – Shared parking/Priority parking
Functionalities delivered	**Get booking status** ○ Functionality: Receives status of ongoing events (booking) ○ Endpoint: /objects/{vas_vicinity_oid}/properties/booking **Get parking sensor status**

	○ Functionality: Receives occupancy and operational status from parking sensor ○ Endpoint: /objects/{vas_vicinity_oid}/properties/status **Set smart light** ○ Functionality: Sets colour of smart light in accordance to status of a given parking sensor or ongoing events. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/color
Selected implementation algorithms	The commands sent to the gateway are DTLS over CoAP. DTLS is essentially an implementation of SSL over UDP. The gateway for the smart light needs to be defined. The current installation is based on IKEA smart light and appears with a hostname of the form gw:xx-xx-xx-xx-xx-xx, where the last bit is the MAC address of the gateway. It will also be necessary with the key (security code) that is located physically on the bottom of the gateway. It will be an alphanumeric code of length 16, listed below the serial number (which is the MAC address).
Deployment/Hosting of VAS	The VAS is composed by two components. A Python application and a Java web application. It uses Python 3.6, Java 8 and SQL Database as requirements and can run on any PC.
Pending issues	Updates will most likely occur on irregular intervals due to testing, feedback and evaluation in a production environment. Beta software is expected to produce errors or/and improvements will be identified as end users get actively involved in daily use.
Deviations/changes	The predefined VASs Offline management, Active maintenance and Smart parking event analysis will not be implemented as described in D5.1.

3.1.1.3. Active maintenance

VAS 1b.1.3	**Active maintenance**
Related Use Case	UC 1b.1 – Shared parking/Priority parking
Functionalities delivered	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Selected implementation algorithms	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Deployment/Hosting of VAS	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)

Pending issues	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Deviations/changes	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)

3.1.2. User interfaces

3.1.2.1. Administration of parking space

UI 1b.1.1	**Administration of parking space**
Related Use Case	UC 1b.1 – Shared parking/Priority parking
Functionalities delivered	**Get list of parking space** ○ Functionality: Returns a list of parking space with properties and ownership. The list will be displayed in either a table or at a map. **Get list of parking sensors** ○ Functionality: Returns a list of registered parking sensors and assigned parking space. The list will be presented alongside the overview of parking space. **Get list of bookings** ○ Functionality: Returns a list of ongoing bookings with information about vehicle, ownership, timeslots and more. The list will be associated with the list of parking space and be presented in a status field. **Set parking space** ○ Functionality: Register, change or remove parking space. This includes the assigned (virtual) parking lot. Properties relevant to the parking space is ownership of the space, dimensions and other properties describing access, POI, etc. **Set booking** ○ Functionality: Assigns or changes the status or assign of a booking. It will allow to change assigned parking space, assigned priority or assigned vehicle. This may also include to cancel an ongoing booking or to create a new booking request.
Example of use	Parking sensors are placed on assigned parking space. The objects returned from the endpoints describe the status as either vacant or occupied. The request is sent to an individual parking slot <int:parking_slot_id>. Each parking slot is assigned a unique ID that refers to the parking sensor that is placed at that location.

Whenever a specific parking space is requested, <res_id> is returned. This object contains information related to that specific booking. <res_id> is also used for cancelling the reservation.

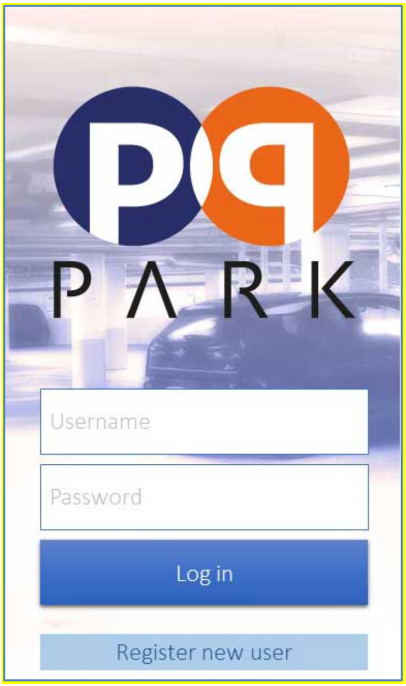
A booking in progress also supports events that are sent when certain situations occur. So far, only timeouts are supported, which again triggers business logic that changes the colour settings of the smart light (sequence of on/off when booking is in effect/about to timeout),

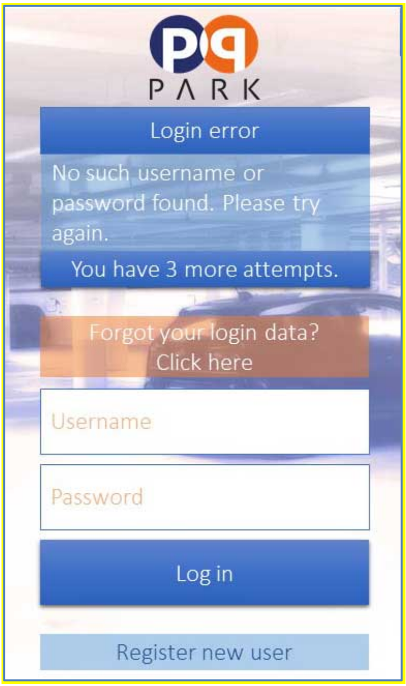
- objects/<subscriber_id>/events/<eid>
  , source: vas-adapter-hits/api/urls.py

, there is additional functionality available for administrative purposes through the API,

- PNI client java supports CRUD functionality (update, insert, remove)
- List of all registered parking spaces: public List<ParkingLot> getParkingLots()
- List of all registered parking sensors: List<ParkingSensor> getSensors(ObjectNode filters)
  , source: https://github.com/vicinityh2020/vicinity-adapter-hits/blob/master/src/main/java/com/hits/vicinity/adapter/api/pni/PniClient.java

Screenshots

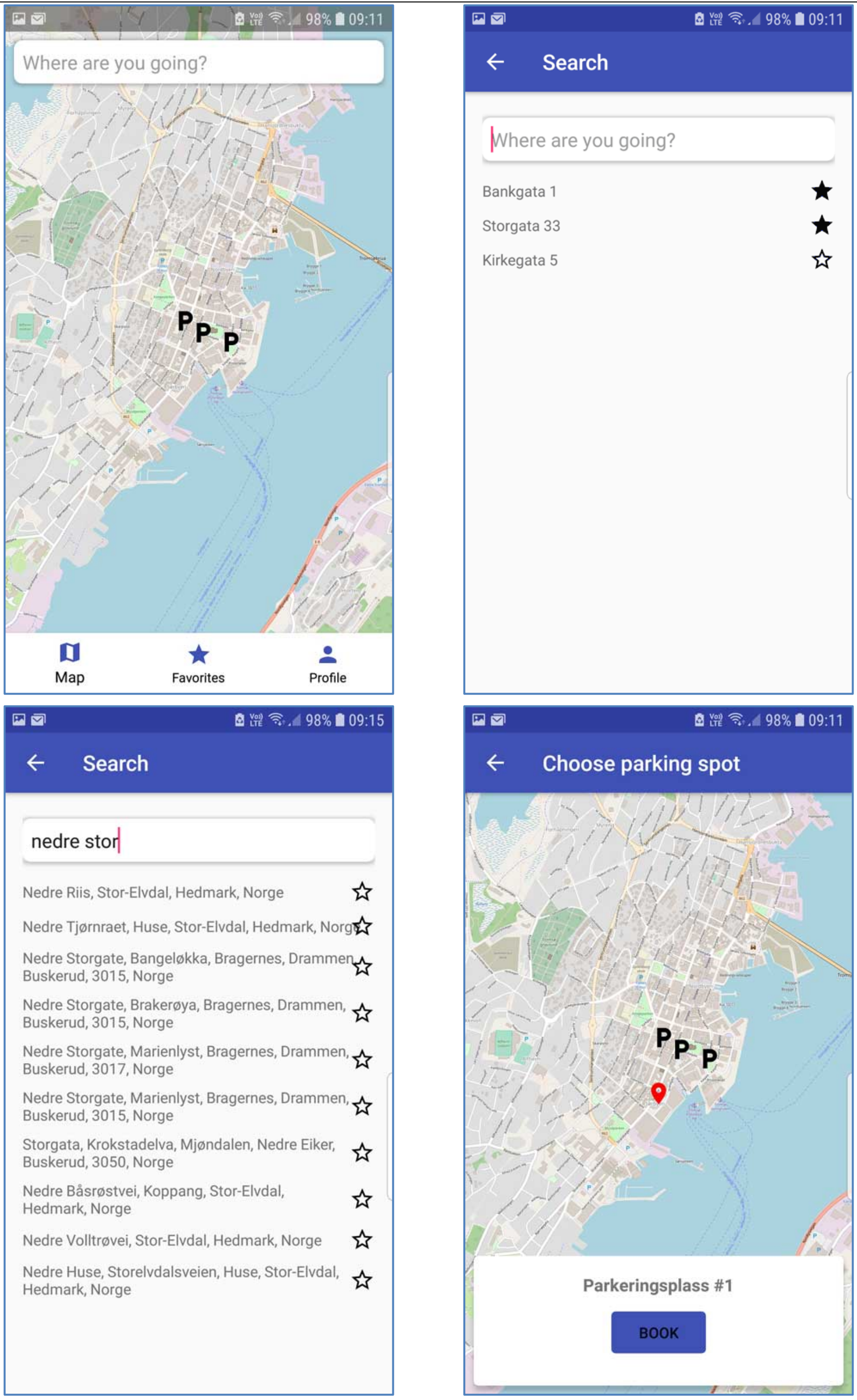




Login error

No such username or password found. Please try again.

You have 3 more attempts.



Search

Where are you going?

- Bankgata 1 ★
- Storgata 33 ★
- Kirkegata 5 ☆

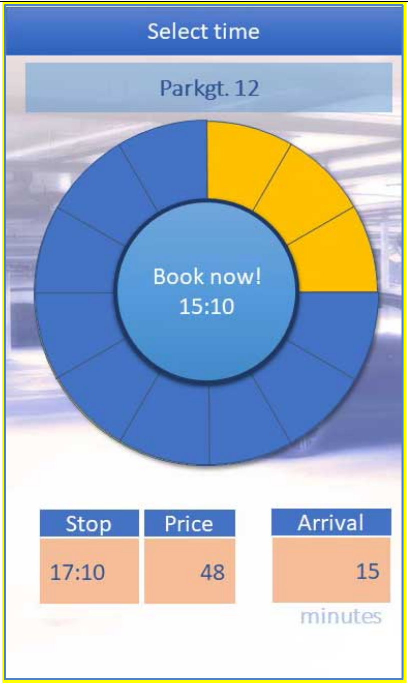
Search

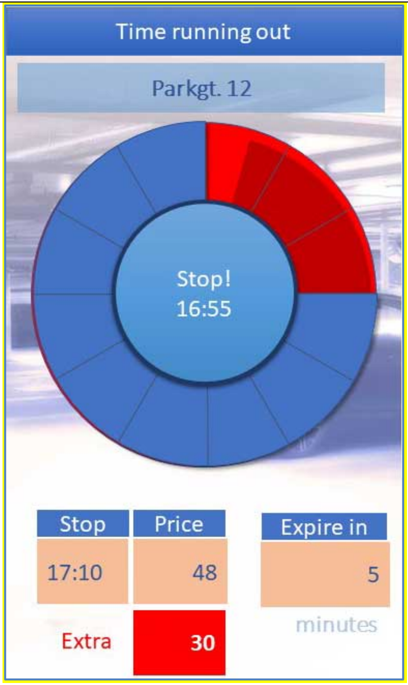
nedre stor

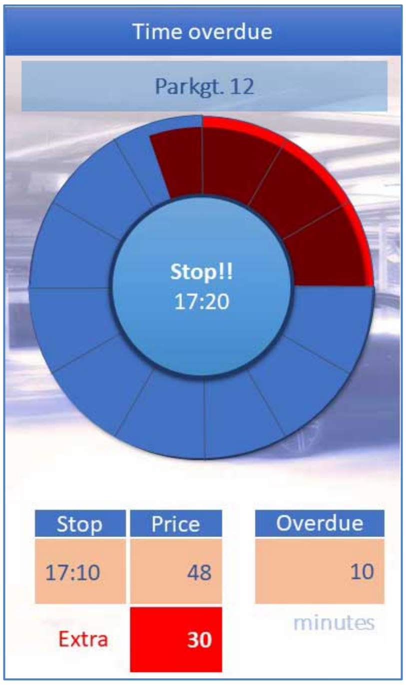
- Nedre Riis, Stor-Elvdal, Hedmark, Norge ☆
- Nedre Tjørnraet, Huse, Stor-Elvdal, Hedmark, Norge ☆
- Nedre Storgate, Bangeløkka, Bragernes, Drammen, Buskerud, 3015, Norge ☆
- Nedre Storgate, Brakerøya, Bragernes, Drammen, Buskerud, 3015, Norge ☆
- Nedre Storgate, Marienlyst, Bragernes, Drammen, Buskerud, 3017, Norge ☆
- Nedre Storgate, Marienlyst, Bragernes, Drammen, Buskerud, 3015, Norge ☆
- Storgata, Krokstadelva, Mjøndalen, Nedre Eiker, Buskerud, 3050, Norge ☆
- Nedre Båsrøstvei, Koppang, Stor-Elvdal, Hedmark, Norge ☆
- Nedre Volltrøvei, Stor-Elvdal, Hedmark, Norge ☆
- Nedre Huse, Storelvdalsveien, Huse, Stor-Elvdal, Hedmark, Norge ☆

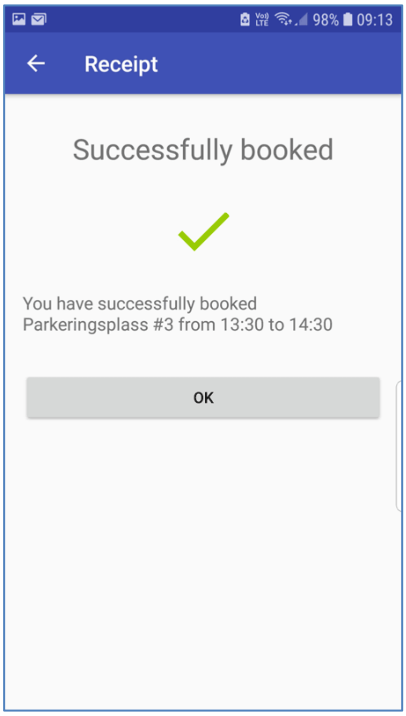
Choose parking spot

Parkeringsplass #1









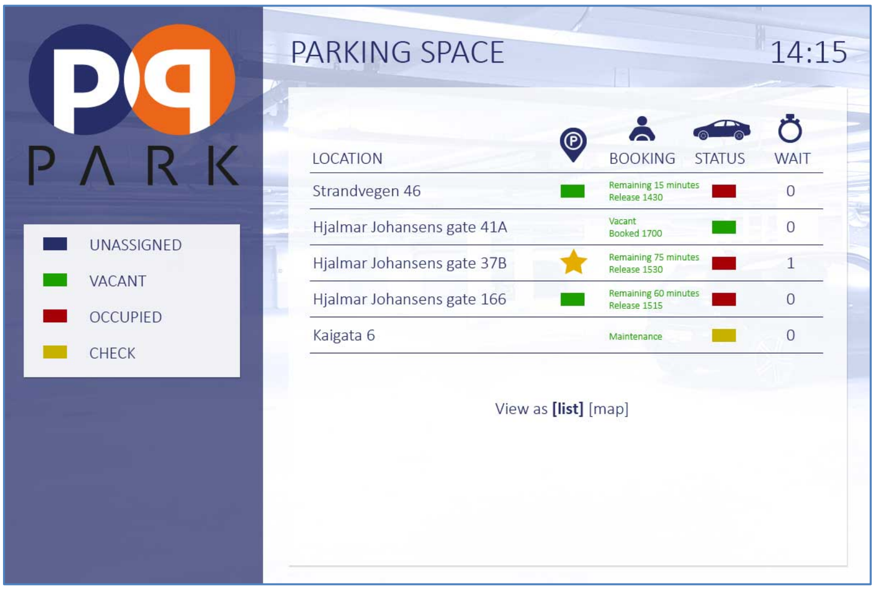
Successfully booked

You have successfully booked Parkeringsplass #3 from 13:30 to 14:30

Deviations/changes	See VAS 1b.1.2

3.1.2.2. Real time information about parking space

UI 1b.1.2	**Real time Information about parking space**
Related Use Case	UC 1b.1 – Shared parking/Priority parking

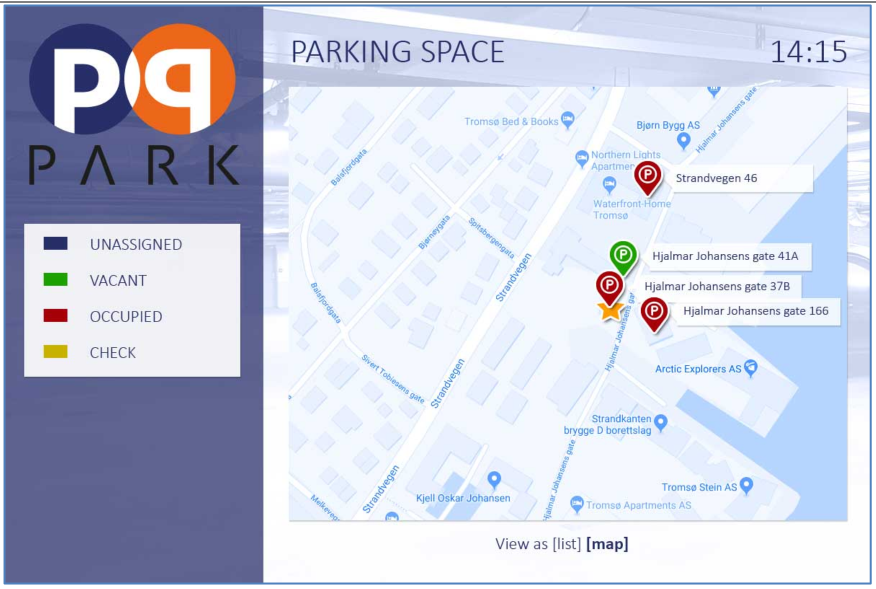
Functionalities delivered	**Get list of parking space** ○ Functionality: Returns a list of parking space with properties and ownership. The list will be displayed in either a table or at a map. **Get list of parking sensors** ○ Functionality: returns a list of registered parking sensors and assigned parking space. The list will be presented alongside the overview of parking space. **Get list of bookings** ○ Functionality: returns a list of ongoing bookings with information about vehicle, ownership, timeslots and more. The list will be associated with the list of parking space and be presented in a status field.
Example of use	Providing overview of availability and status of parking space and installed parking sensors. Colour codes indicates status; green: vacant, red: occupied, yellow: error. Overview is presented as either a list or as positions on a map.
Screenshots	

🟦	UNASSIGNED
🟩	VACANT
🟥	OCCUPIED
🟨	CHECK

PARKING SPACE 14:15

LOCATION		BOOKING	STATUS	WAIT
Strandvegen 46	🟩	Remaining 15 minutes Release 1430	🟥	0
Hjalmar Johansens gate 41A		Vacant Booked 1700	🟩	0
Hjalmar Johansens gate 37B	★	Remaining 75 minutes Release 1530	🟥	1
Hjalmar Johansens gate 166	🟩	Remaining 60 minutes Release 1515	🟥	0
Kaigata 6		Maintenance	🟨	0

View as **[list]** [map]

	 PQ PARK UNASSIGNED VACANT OCCUPIED CHECK PARKING SPACE 14:15 Strandvegen 46 Hjalmar Johansens gate 41A Hjalmar Johansens gate 37B Hjalmar Johansens gate 166 Tromsø Bed & Books Bjørn Bygg AS Northern Lights Apartment Waterfront-Home Tromsø Arctic Explorers AS Strandkanten brygge D borettslag Tromsø Stein AS Tromsø Apartments AS Kjell Oskar Johansen Strandvegen View as [list] [map]
Deviations/changes	See VAS 1b.1.2

3.1.2.3. Maintenance of parking sensors

UI 1b.1.3	**Maintenance of parking sensors**
Related Use Case	UC 1b.1 – Shared parking/Priority parking
Functionalities delivered	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Example of use	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Screenshots	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)
Deviations/changes	Specified in D5.1, removed from implementation (see 3.1.1.2, VAS 1b.1.2, Deviations/changes)

3.2. Pilot Use Case 1b.2: eHealth Emergency parking

The current Use Case deals with will prioritizing shared private and public parking facility. Currently, nearly 10% of accidents in managed homes happen when the caring client is not able to report on the accident (ref. Grant Agreement). The scenario involves smart appliances (i.e. fridge and oven) can act as "panic buttons" to do a "silent" report to care centre. The authorized healthcare personnel will be reported about the accident by the smart appliances, specifically,

- For a refrigerator that its door has been left open more than normal time (i.e. more than 30 seconds)

- For an oven that its power, cooking and/or heat have been left on for an abnormal time (i.e. more than 15 minutes)

The healthcare personnel will automatically be assigned a vehicle and a priority parking at the closest facility to their client's assisted living building.

A schematic representation of the Use Case in relation to VICINITY architecture (i.e. Organisations, Users, Things, Friendship, etc.). A priority ticketing service is offered to the benefit of parking space owner and vehicle user. How all the information is combined and distributed to the different actors is shown in Figure 3-2, but not all functionality will be demonstrated.

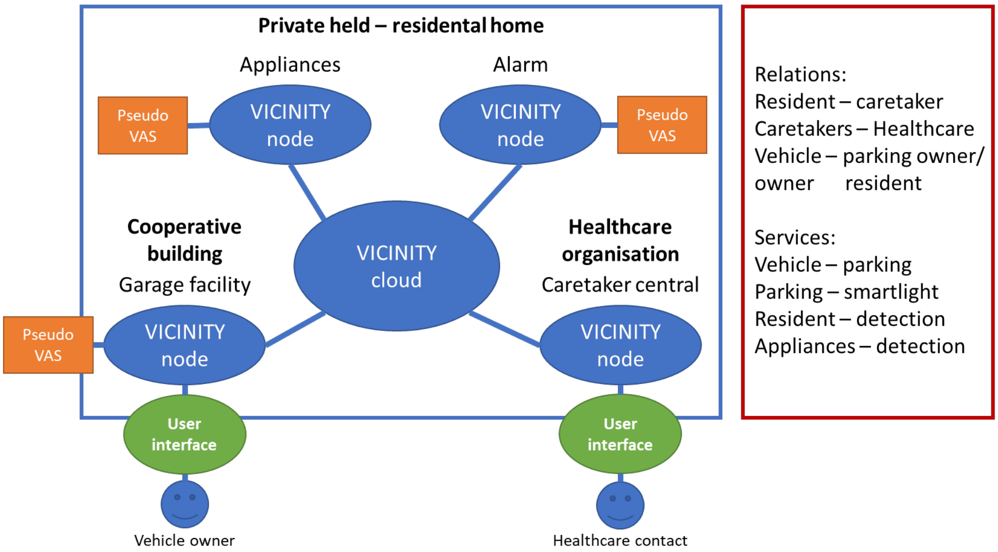


Figure 3-2 Use Case 1b.2 associated Value-Added Services and User Interfaces

Ownership and administration of the parking space will be handled by the building manager and the owner of the physical parking facility.

Currently, the main target group that will be assigned priority parking is healthcare personnel, care centre and blue light agencies (medical, fire department and police).

The value-added services set out to offer solutions addressing,

- priority parking allocation based on booking ID
- eHealth priority parking, blue light agencies
- triggers from building (e.g. panic button) and assisted living (e.g. smart appliances)

3.2.1. VICINITY Value-Added services

3.2.1.1. Neighbourhood data processing

VAS 1b.2.1	**Residence data processing**
Related Use Case	UC 1b.2 – eHealth Emergency parking
Functionalities delivered	**Get booking status** ○ Functionality: Receives status of ongoing events (booking) ○ Endpoint: /objects/{vas_vicinity_oid}/properties/booking

	Get parking sensor status ○ Functionality: Receives occupancy and operational status from parking sensor ○ Endpoint: /objects/{vas_vicinity_oid}/properties/status **Set smart light** ○ Functionality: Sets colour of smart light in accordance to status of a given parking sensor or ongoing events. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/color **(Get) Smart appliance/refrigerator** ○ Functionality: Returns the status of door. Status can be either open or closed. ○ Endpoint: ○ /objects/{vas_vicinity_oid}/properties/door **(Get) alarm** ○ Functionality: Returns the status of the alarm VAS. Status can be either true or false. ○ Endpoint: ○ /objects/{vas_vicinity_oid}/properties/activated ** The endpoints have not been made available yet, but they are expected to be in the format described above*
Selected implementation algorithms	The VAS will be event based, catching changes status (close, open, warning on, warning off). Business logic for smart appliances will be implemented in Java and will examine status changes for certain conditions. The smart refrigerator VAS will trigger a warning similar to an alarm button if the door is left open for a specified period of time (currently set to 5 minutes).
Deployment/Hosting of VAS	The VAS will be based on a central server with Linux (Ubuntu). The server will support PostgreSQL, Python3 scripts and Java. The server will be running Apache Tomcat. To be able to communicate with the devices, some ports need to be opened in the firewall/gateway and be set to both send and receive (in/out): • UDP: 11760 • TCP: 3000, 5684, 8000, 8050, 8080, 8100, 8181, 8282, 9997, 9998 Disclaimer: some ports may be closed while others may be opened instead. This will be system specific on deployment as changes in smart light. Alarm devices and smart appliances may affect which ports needs to be accessible.

Pending issues	Updates for smart appliances, smart light and smart alarms/warnings are expected as partners update their adapters and the project receives feedback from test users. Three release candidates are planned ultimo October, November and medio December.
Deviations/changes	Event and data-handling of equipment in residential homes has been consolidated. Adapters supporting through necessary functionality has or will be delivered by other VICINITY partners. This affects mainly the Gorenje smart appliances: refrigerator and oven. Additionally, VAS supporting alarms is being prepared by CERTH/GNOMON/MPH.

3.2.1.1. Smart parking event analysis

VAS 1b.2.2	**Smart parking event analysis**
Related Use Case	UC 1b.2 – eHealth Emergency parking
Functionalities delivered	**Get list of priority messages** ○ Functionality: Returns a list of warnings that have triggered a priority message alongside the device that triggered the warning. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/warnings **Get booking status** ○ Functionality: Receives status of ongoing events (booking) ○ Endpoint: /objects/{vas_vicinity_oid}/properties/booking **Get parking sensor status** ○ Functionality: Receives occupancy and operational status from parking sensor ○ Endpoint: /objects/{vas_vicinity_oid}/properties/status
Selected implementation algorithms	See descriptions in UC1b.1 When emergency arises (i.e. an alarm signal was triggered as described in 1b.2.1), authorised users are permitted to use priority booking. • Priority booking is a functionality that puts the booking request first in line. • If other vehicles are waiting for a parking space to become vacant, the priority parking request is put first in line. • If other vehicles already have booked but not arrived at the parking space, the booking will be rescheduled. • Messages describing the nature of rescheduled parking priority will be made available to affected parties.

Deployment/Hosting of VAS	The VAS will be based on a central server with Linux (Ubuntu). The server will support PostgreSQL, Python3 scripts and Java. The server will be running Apache Tomcat.
Pending issues	Service has not been integrated with Gorenje adapter for smart appliances (refrigerator, oven) yet. An update is expected ultimo November. Furthermore, see VAS 1b.1.2
Deviations/changes	Additional functionality related to optimized parking has been temporarily removed. This is due to this feature is considered irrelevant on the pilot site. All parking space are located close to each other, they feature similar properties when it comes to access/authorisation, physical dimensions and locations.

3.2.2. User Interfaces

3.2.2.1. Administration of priority parking

UI 1b.2.1	**Administration of priority parking**
Related Use Case	UC 1b.2, UC 1b.1
Functionalities delivered	**Get list of priority messages** ○ Functionality: Returns a list of warnings that have triggered a priority message. A priority message may open for priority booking level 2, otherwise known as emergency booking. This list will be displayed in either a table or at map. **Get list of parking space** ○ Functionality: Returns a list of parking space with properties and ownership. The list will be displayed in either a table or at a map. **Get list of parking sensors** ○ Functionality: Returns a list of registered parking sensors and assigned parking space. The list will be presented alongside the overview of parking space. **Get list of bookings** ○ Functionality: returns a list of ongoing bookings with information about vehicle, ownership, timeslots and more. The list will be associated with the list of parking space and be presented in a status field. **Set booking** ○ Functionality: Assigns or changes the status of an assignment of a booking. It will allow to change assigned parking space, assigned

	priority or assigned vehicle. This may also include to cancel an ongoing booking or to create a new booking request.
Example of use	The service is only made available to authorized personnel. Depending on the authorisation level, the user will be able to see just one entry (their own), or a list related to larger set of registered parking spaces. The interface allows changes to make on ongoing activities (normal parking requests, priority parking requests). It also presents the status of the parking sensor placed at the allocated/booked parking space.
Screenshots	See mobile interfaces described in UI 1b.1.1. These will be extended with functionality relevant to the user role and situation (warning level 1 and level 2).
Deviations/changes	See VAS 1b.1.2 and VAS 1b.2.2

3.2.2.2. Real time information about priority parking in progress

UI 1b.2.2	**Real time Information about priority parking in progress**
Related Use Case	UC 1b.2, UC 1b.1
Functionalities delivered	Similar to UC 1b.2. Additional functionality includes a note in the list displaying parking space that informs that priority parking is in progress.
Example of use	When an emergency arises (i.e. an alarm signal was triggered as described in 1b.2.1), authorised users are permitted to use priority booking. • Priority booking is a functionality that puts the booking request first in line. • If other vehicles are waiting for a parking space to become vacant, the priority parking request is put first in line. • If other vehicles already have booked but not arrived at the parking space, their booking will be rescheduled. • Messages describing the nature of rescheduled parking priority will be made available to affected parties.
Screenshots	See UI 1b.1.2, The real time information is combined with other relevant data. Information will be added in an iterative process.
Deviations/changes	Original descriptions referred to already parked vehicles receiving requests to reallocate their cars. Later discussions with stakeholders signalled such functionality provided little to no benefit and thus was dropped. Furthermore, see VAS 1b.1.2

3.3. Ethics

Transport and parking normally do not cover personal and sensitive data. But when combined with eHealth data, then personal data may be sensitive. Data protection and privacy concerns have been a showstopper for eHealth ambitions in the cloud sector so far. The "Shared/priority parking" pilot will not involve data of care receivers. Care centre and healthcare personnel are managed by national regulation on privacy about their clients. The "emergency parking" might be applied for acquiring data on person's daily life, including data when fridge and oven are used. The data from smart appliances (i.e. fridge and oven) will only be used as an alarm signalling a case of emergency or the case when the care receiver needs assistance. Any participation within VICINITY as a user or patient is entirely voluntary. Participants will have the right to withdraw from the research at any time and to withdraw retrospectively any consent given.

Implementation of the VASs for this pilot had to take into consideration the above restrictions concerning sensitive medical data. During design phase of the VASs of another VICINITY pilot, the implementation of a service, which will be responsible for storing and handling the sensors data. This is VAS 3.1.1 for Use Case 3.1 and VAS 3.2.1 for Use Case 3.2. Respecting the GDPR regulation, the VAS is keeping a record of all the data transactions and can inform each user about who has requested access to their data. Moreover, it is responsible for granting access to specific user data, only to certain VICINITY users and it also gives the ability to users to delete their personal records at any time (right to be forgotten). The rest of the VASs of the pilot, acquire the data they need for processing only through this VAS and do not have direct access (no VICINITY Contract) to the sensor data.

3.4. VICINITY Value

Based on the initial identification of how VICINITY acts as an enabler for the specific use cases (based on the information provided in D5.1 under sections 4.2.6, 4.3.6), the assessed added value provided is summarized in the following tables,

Value of VICINITY to use case 1b.1	
Problem	Solution
Tenants in buildings have available parking when not using their parking themselves	The smart parking admin app will assist in specifying availability for rental of their parking facility. Payments for rentals will be assigned their private bank account. Healthcare personnel can plan their tour visiting care clients and apply parking as close as possible to their clients in order to work more efficiently with their clients.
Healthcare person is stuck in traffic or cannot find parking close to their clients. Parking fines will not be covered by the care centre	The smart parking app is available for care centre and can plan the tour visiting their clients and apply Priority Parking as close as possible to their clients in order to work more efficiently with their clients.

Value of VICINITY to use case 1b.2	
Problem	Solution
Lack of Semantic interoperability	Utilized two different IoT Platforms integrated with both parking and smart home domain sensors, namely: PNI cloud and Gorenje Cloud service.
Emergency happens at care receiver side by not having access to her Panic button	The care receiver keeps her door to fridge open. An alarm will be triggered to care centre putting the case as an emergency and acquire authorised healthcare personnel, vehicle and priority parking as close as possible to the care receiver.
Emergency when cooking or baking in oven, food over-cooked/boiled	The care receiver cooks or bakes her food too long time in oven. Danger of fire. An alarm will be triggered to care centre putting the case as an emergency and acquire authorised healthcare personnel, vehicle and priority parking as close as possible to the care receiver.

4. Martim Longo (PO) – Neighbourhood GRID ecosystem (ENERC)

4.1. Use Cases 2.1, 2.2, 2.3, 2.4, 2.5, 2.10, 2.11 – Energy Efficiency and IEQ Management in Municipal Cluster of buildings – Municipal Services

In this set of Use Cases (Solarlab at the Solar Demo platform, Municipal School, Sports Centre - Swimming Pool and Retirement home) multiple types of sensors and equipment are used to collect real-time information from the buildings. Specifically, sensor readings and alert levels of temperature, CO2, humidity, noise, motion and energy consumption are displayed. Historical data of sensor readings may be retrieved for further analysis.

An overview of the pilot site and the VICINITY integration under the concept of the specific Use Cases can be seen in the following figure.

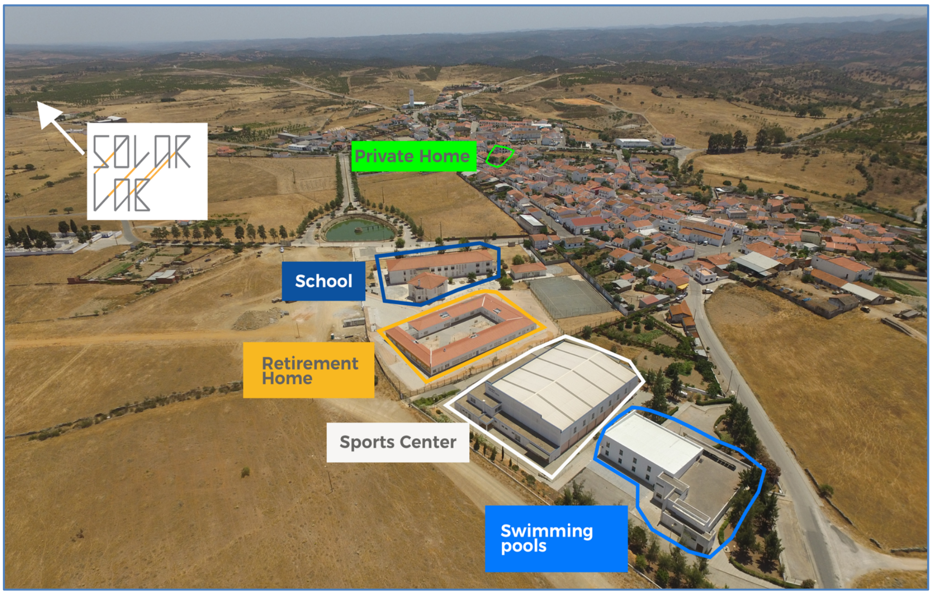


Figure 4-1 Pilot Site Overview

The primary goal is to facilitate dynamic data collection from various sensors enabling a **Dynamic Building Audit** (see UI 2.1-2.5).

All the use cases of the Municipal Services group require the following, real time display of sensor data; real time display of alert levels; display of sensor data over multiple date and time ranges; and geospatial location of measured values.

Since this is a common pattern in Indoor Environment Monitoring, the solution being implemented is based on a configurable **Environment Dashboard** that presents real time sensor data over building and sensor installation plans. Geospatial location and relationship between sensor data are relevant for most users: for an occasional user it allows a simplified location of alert levels and for advanced users it provides context to the observed values (e.g. sun exposure, UV radiation, shade and temperature and humidity).

Recognition of this pattern and the absence of a relevant end-to-end prototype based on the parametrization of readily available IoT tools, led the project team in late September to try a full restart in software implementation using a full-fledged IDE (NetBeans) and SDK (Java SE Development Kit).

Using documentation and Java source code publicly available at https://github.com/vicinityh2020, we were able to implement in about two weeks a hard-coded end-to-end prototype fully integrated with VICINITY, that show cases the technology required to implement an Environment Dashboard, including: a VICINITY device adapter to gather sensor data; a VICINITY service adapter to assemble the dataset of sensor data required by the user interface service; and a user interface service producing a webpage that presents real time sensor data over a jpeg image of the building and sensor installation plans.

Some features relevant to the use cases presented require historical sensor data analysis with specific algorithms and are being implemented as specific add-on modules extending the main Environment Dashboard.

VAS 2.1 approaches allow the creation of solutions for a better indoor comfort and energy rationalization, reducing costs, consumptions and CO_2 emissions. Maintaining or improving the quality of indoor environment and the productivity of teams (e.g. students) using the spaces is an important step towards sustainability in the medium to long term. It will be the first step towards the important digitization of Municipal Buildings.

Based on stakeholders' feedback additional functionality is discussed,

- noise level incentives in the classrooms
- boundaries for Building IEQ will be incorporated
- links to resources use will be leveraged

4.1.1. VICINITY Value-Added services

Municipal Services Category of VAS would demonstrate test an offering of - ***Dynamic Building Audit and the IEQ services***

4.1.1.1. Municipal Services, IEQ Smart School, Dynamic Audit

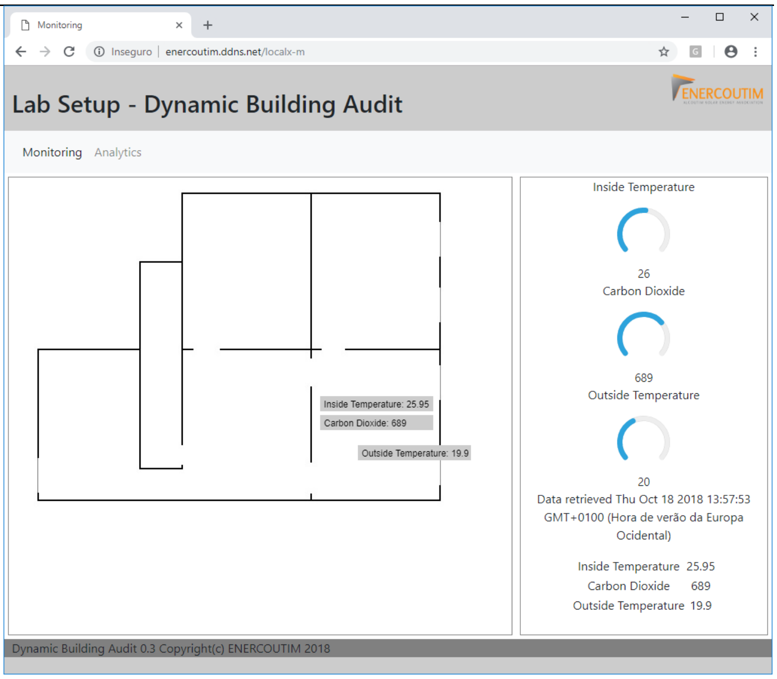
VAS 2.1	**Municipal Services, IEQ Smart School, Dynamic Audit**
Related Use Cases	UC 2.1, 2.2, 2.3, 2.4, 2.5, 2.10,2.11
Functionalities delivered	**Get Measurements** ○ Functionality: Returns sensor real-time data (currently: temperature and carbon dioxide) ○ Endpoint: /{vas_url}/service/{oid}/property/{pid}
Selected implementation algorithms	Dynamic Building Audit VAS is responsible for combining data from different sensor adapters and evaluate alert levels, preparing information for the user interface.
Deployment/Hosting of VAS	Dynamic Building Audit VAS is currently being deployed with an embedded Grizzly Java webserver to Raspberry Pi with Raspbian and distributed on site.
Pending issues	Performance limitations: • Deployment to a central location using a standard x64 server running a Jetty or Tomcat webserver is under analysis due to performance limitations of Raspberry Pi. Implementation issues:

	- Development of VICINITY device adapter for K&Z sensors with dataTaker dataloggers is ongoing. - Research and assessment criteria of alert levels is ongoing. Productization issues: - Development of a more configurable adapter for Energomonitor sensors is ongoing. - Development of a more configurable Environment Dashboard is ongoing.
Deviations/changes	VAS implementation is not complete but there are no planned changes to the features outlined in the VAS uses cases.

4.1.2. User interfaces

4.1.2.1. Municipal Services, IEQ Smart School, Dynamic Audit

UI 2.1-2.5	**Municipal Services, IEQ Smart School, Dynamic Audit**
Related Use Case	UC 2.1, 2.2, 2.3, 2.4, 2.5, 2.10, 2.11
Functionalities delivered	Web based Environment Dashboard and Analytics with: - real time display of sensor data; - real time display of alert levels; - display of historical sensor data; - display of location of measured values on building plan.
Example of use	- Monitoring: User may access the environment dashboard using any common browser for computers and mobile devices and check if there is any alert level. - Analytics: User may access the analytics console using any common browser for computers and mobile devices and assess common comfort levels using historical data.
Screenshots	

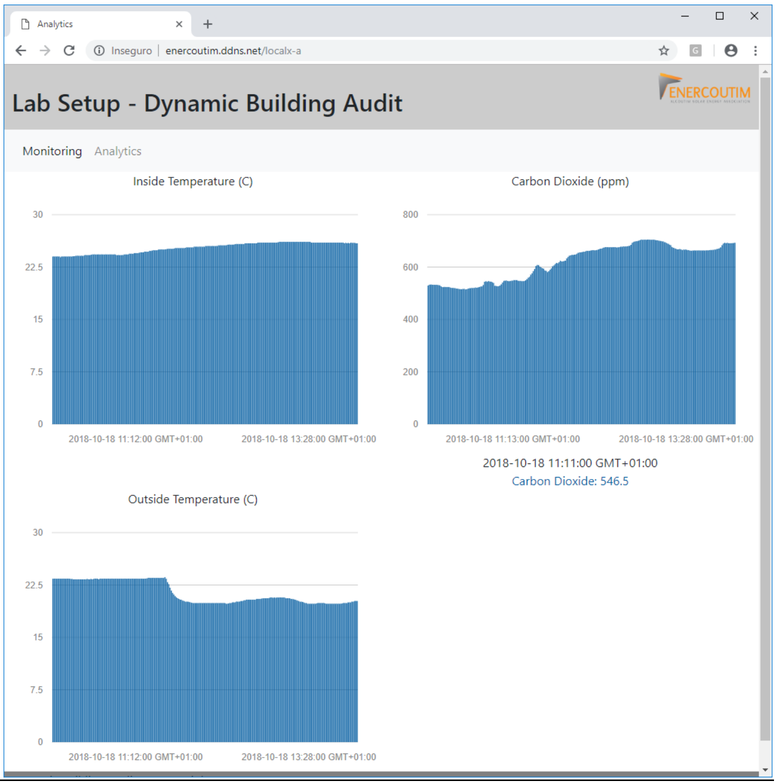


Lab Setup - Dynamic Building Audit

Data retrieved Thu Oct 18 2018 13:57:53 GMT+0100 (Hora de verão da Europa Ocidental)

Inside Temperature 25.95
Carbon Dioxide 689
Outside Temperature 19.9

Dynamic Building Audit 0.3 Copyright(c) ENERCOUTIM 2018


Analytics
Inseguro | enercoutim.ddns.net/localx-a
ENERCOUTIM
Monitoring Analytics
Inside Temperature (C)
30
22.5
15
7.5
0
2018-10-18 11:12:00 GMT+01:00
2018-10-18 13:28:00 GMT+01:00
Carbon Dioxide (ppm)
800
600
400
200
0
2018-10-18 11:13:00 GMT+01:00
2018-10-18 13:28:00 GMT+01:00
2018-10-18 11:11:00 GMT+01:00
Carbon Dioxide: 546.5
Outside Temperature (C)
30
22.5
15
7.5
0
2018-10-18 11:12:00 GMT+01:00
2018-10-18 13:28:00 GMT+01:00

Lab Setup - Dynamic Building Audit

Deviations/changes	VAS implementation is not complete but there are no planned changes to the features outlined in the VAS uses cases.

4.2. Use Case 2.9 – UV (Ultraviolet radiation) info services for Citizens and Tourists – Local to Local Services

Information services for citizens and tourists require an additional dissemination effort. We cannot expect occasional users to search for specific information on ultraviolet radiation since they are not aware of its existence.

Most citizens and tourists are likely to visit websites related to local businesses or the municipality itself. The solution being implemented is based on a simple widget that can be embedded as an iframe in external websites displaying information about UV radiation and linked to a webpage with additional information (UV index, ambient temperature, humidity and practical recommendations).

Monitoring the ultraviolet radiation index will provide real-time prediction and reporting of levels and prevent any harmful effects on the health of students and the elderly. It is also an increasingly important tool in places where radiation levels are normally quite high, bringing medium-term savings in treatments against for example skin cancer.

4.2.1. VICINITY Value-Added Services

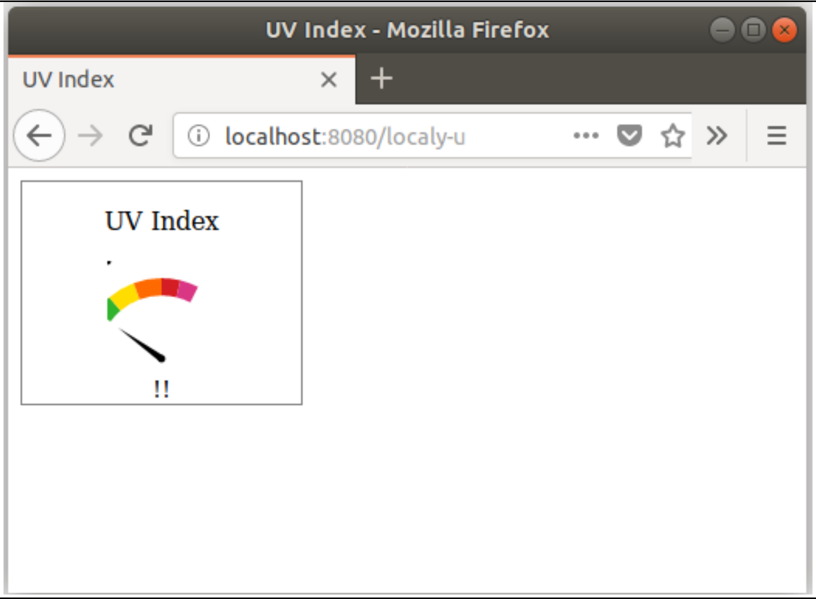
4.2.1.1. Local to Local Services, UV for Citizens

VAS 2.2	**Local to Local Services, UV for Citizens**
Related Use Case	UC 2.9
Functionalities delivered	**Get Measurements** ○ Functionality: Returns UV sensor real-time data ○ Endpoint: /{vas_url}/service/{oid}/property/{pid}
Selected implementation algorithms	UV for Citizens VAS is responsible for retrieving data from the K&Z UV sensor adapter and evaluate alert levels, preparing information for the user interface.
Deployment/Hosting of VAS	UV for Citizens VAS is currently being deployed with an embedded Grizzly Java webserver to Raspberry Pi with Raspbian and distributed on site.
Pending issues	Performance limitations: • Deployment to a central location using a standard x64 server running a Jetty or Tomcat webserver is under analysis due to performance limitations of Raspberry Pi. Implementation issues: • Development of VICINITY device adapter for K&Z sensors (including the UV Sensor) with dataTaker dataloggers is ongoing.
Deviations/changes	VAS implementation is not complete but there are no planned changes to the features outlined in the VAS uses cases.

4.2.2. User interfaces

4.2.2.1. Local to Local Services, UV for Citizens

UI 2.9	**Local to Local Services, UV for Citizens**
Related Use Case	UC 2.9

Functionalities delivered	Iframe embeddable widget showing UV radiation and alert level.
Example of use	User accessing websites related to local businesses or public services sees a widget showing the UV radiation and alert levels.
Screenshots	 UV Index - Mozilla Firefox UV Index localhost:8080/localy-u UV Index !!
Deviations/changes	VAS implementation is not complete and is currently targeting an embeddable widget showing UV radiation and a complimentary website with additional information (UV index, ambient temperature, humidity and practical recommendations).

4.3. Use Cases 2.6, 2.7 – Distributed Energy assets management – Platform Services

While a solution for this set of Use Cases requires implementation of specific features related to O&M data, it could also benefit from an Environment Dashboard presenting the infrastructure layout.

Specific algorithms are being implemented as add-on modules extending the main Environment Dashboard.

By monitoring the meteorological conditions of the area, namely solar radiation and solar module production performance, the VASs under Use Cases 2.6, 2.7 will allow more accurate predictions and scheduling for actions like washing of the solar systems plus usage optimization of the resources and the equipment.

4.3.1. VICINITY Value-Added Services

4.3.1.1. Platform Services. Smart Clean. O&M for distributed renewable production resources

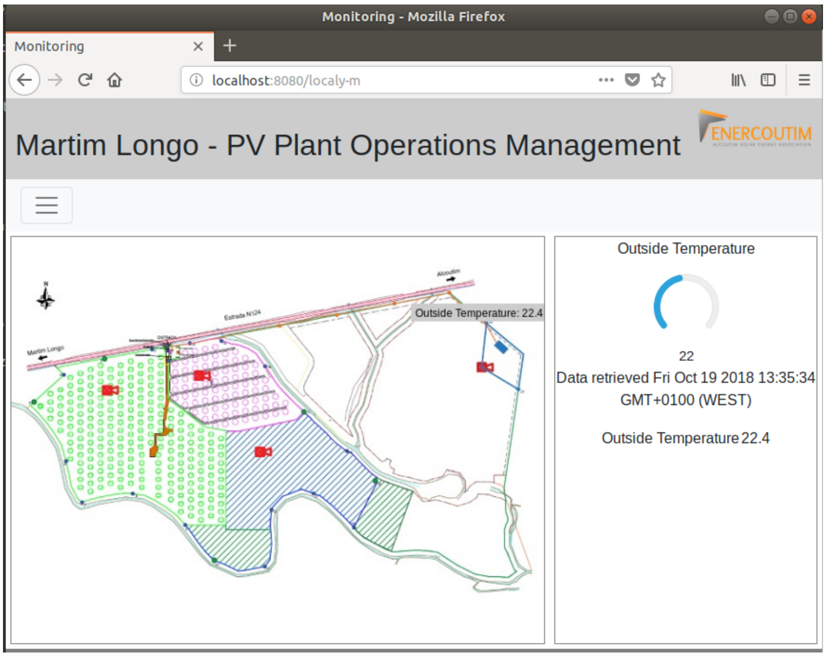
VAS 2.3	**Platform Services. Smart Clean. O&M for distributed renewable production resources.**
Related Use Case	UC 2.6, 2.7
Functionalities delivered	**Get Measurements** ○ Functionality: Returns UV sensor real-time data

	Endpoint: /{vas_url}/service/{oid}/property/{pid}
Selected implementation algorithms	PV Plant Operations Management VAS is responsible for combining data from different sensor adapters and evaluate opportunity for current maintenance operations, preparing information for the user interface.
Deployment/Hosting of VAS	PV Plant Operations Management VAS is currently being deployed with an embedded Grizzly Java webserver to Raspberry Pi with Raspbian and distributed on site.
Pending issues	Performance limitations: • Deployment to a central location using a standard x64 server running a Jetty or Tomcat webserver is under analysis due to performance limitations of Raspberry Pi. Implementation issues: • Development of VICINITY device adapter for K&Z sensors with dataTaker dataloggers is ongoing; • Data of the energy production from PV system/energy available from pyrheliometer and pyranometer (if production is below x% of what should be estimated production, it means it needs to be cleaned) • Access to external weather forecasting service for the next 8 days; • Availability of cleaning equipment • Availability of specialized human resources for cleaning operations.
Deviations/changes	VAS implementation is still in progress and the features outlined in the uses cases are being reanalysed for identification of overlapping features and integration with current operations management system.

4.3.2. User Interfaces

4.3.2.1. Platform Services. Smart Clean. O&M for distributed renewable production resources

UI 2.6, 2.7	**Platform Services. Smart Clean. O&M for distributed renewable production resources.**
Related Use Case	UC 2.6, 2.7
Functionalities delivered	Web based Environment Dashboard and Analytics with: • real time display of sensor data (currently temperature); • real time display of alert levels; • display of historical sensor data; • display of platform plan.
Example of use	Monitoring (currently not viable due to unimplemented service features): User may access the environment dashboard using any

	common browser for computers and mobile devices and check if there are an alert to request the need to clean.
Screenshots	 Monitoring - Mozilla Firefox Monitoring localhost:8080/localy-m Martim Longo - PV Plant Operations Management Outside Temperature 22 Data retrieved Fri Oct 19 2018 13:35:34 GMT+0100 (WEST) Outside Temperature 22.4
Deviations/changes	VAS implementation is not complete yet and the features outlined in the uses cases are being reanalysed for identification of overlapping features and integration with current operations management system.

4.4. Ethics

The collection of data in each of the Municipal buildings has the consent of the Municipality and the information extracted from it will be properly stored and in compliance with the GDPR regulation.

No personal information will be collected in any of our VASs. Instead, VAS 1 will collect information referring to the behavior of buildings, VAS 2 will collect and provide useful information to the citizens without any intervention to their privacy and in VAS 3 information will be restricted to each of the companies that will contract the smart cleaning service. The latter contract will essentially be a pre-agreement of acceptance and confidentiality for the collected data.

In addition to the above, any participation in the VICINITY platform as a user is entirely voluntary and lacks a record that will require compliance with all data protection requirements. There is always the right for the user to withdraw at any time and to withdraw any consent that has been given.

Enercoutim will also provide all necessary actions to be in full compliance with Portuguese and European laws regarding the storage and processing of data for each of its VASs.

4.5. VICINITY Value

Based on the initial identification of how VICINITY acts as an enabler for the specific use cases (based on the information provided in D5.1 under sections 5.2.6, 5.3.6, 5.4.6), the assessed added value provided is summarized in the following tables,

Value of VICINITY to use case 2.1, 2.2, 2.3, 2.4, 2.5, 2.10, 2.11	
Problem	Solution
Lack of **IoT Semantic interoperability**	Integrated two different IoT Platforms optimized for specific usages that provide complimentary data: weather stations (K&Z sensors with dataTaker datalogger) and building domain sensors (Energomonitor).

Value of VICINITY to use case 2.9	
Problem	Solution
Reuse of existing IoT and IT infrastructure.	Considering the IoT and IT infrastructure required to operate the pilot installations and the information available, this use case reuses sensor data from the weather station to deliver a local public service (UV Index), demonstrating VICINITY as an enabler of infrastructure usage efficiency.

Value of VICINITY to use case 2.6, 2.7	
Problem	Solution
Lack of IoT Semantic interoperability.	Integrated two different IoT Platforms optimized for specific usages that provide complimentary data: weather stations (K&Z sensors with dataTaker datalogger) and PV energy production equipment sensors (SMA).

5. Pilea-Hortiatis (GR) – eHealth & Assisted Living (CERTH – GNOMON – MPH)

5.1. Use Case 3.1 – eHealth and Assisted Living for elderly people at home

The current Use Case deals with the remote monitoring of elder citizens living alone under the concept of assisted living that will be offered as a service by the Municipality. Everyday routine tasks and medical data collected by relevant IoT devices consist the main scope of monitoring. The scenario involves authorized healthcare personnel to interpret respective information gathered by the Value-Added Services plus officially registered guardians (e.g. relative) per citizen, mainly for notifications in case of abnormal detections.

The following figure provides a schematic representation of the Use Case in relation to the VICINITY architecture (Organisations, Users, Things, Friendships, etc).

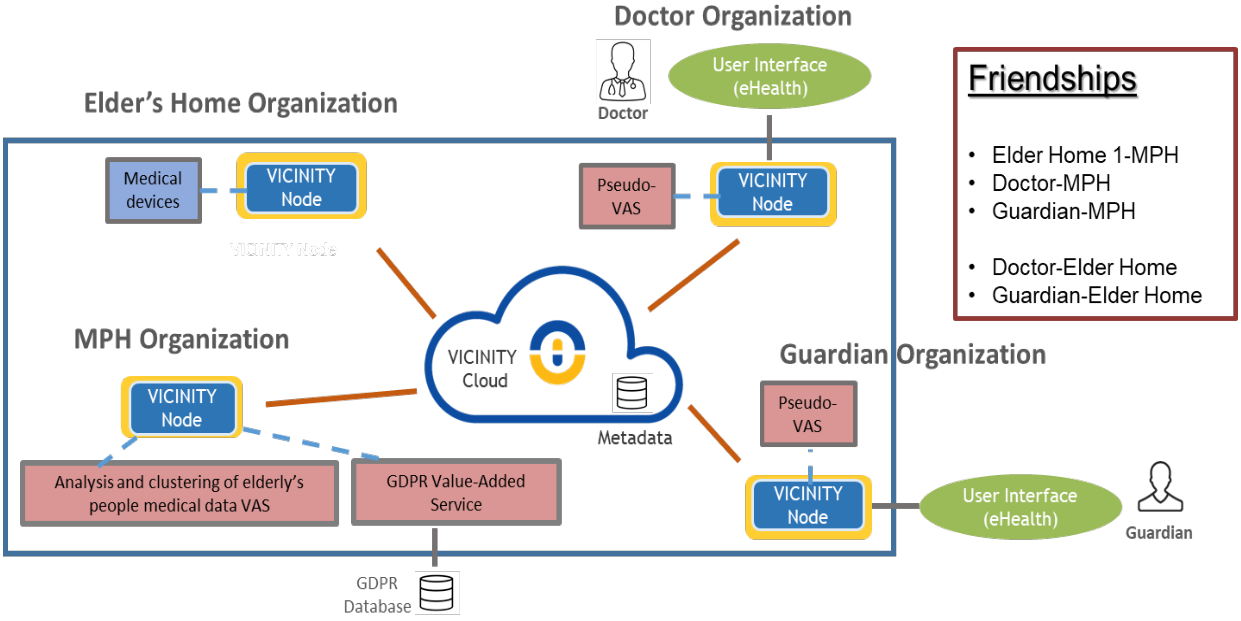


Figure 5-1 Use Case 3.1 - Overview

5.1.1. VICINITY Value-Added Services

5.1.1.1. Privacy-preserving Data Gathering and Storage ft. GDPR data auditing

VAS 3.1.1	**Privacy-preserving Data Gathering and Storage ft. GDPR data auditing**
Related Use Case	UC 3.1 – eHealth and Assisted Living for elderly people at home
Functionalities delivered	**Push Measurement** ○ Functionality: Stores an elder's citizen measurement, sent by the relevant Raspberry Pi deployed in-house, to MPH's existing infrastructure that was used for the case so far. After the latter successful storage, the measurement is also sent to an already

	existing solution (i.e. eHealthPass) developed by GNOMON under the FHIR Specification, a standard used for exchanging health information electronically, that was extended and integrated for the VICINITY case. Finally, after the measurement is successfully stored in both systems above, it is forwarded to VAS 3.1.2. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/prop **Get Measurements** ○ Functionality: Returns a set of user's measurements as fetched by MPH's and eHealthPass databases described above. The latter functionality will be used either by healthcare personnel authorized by MPH especially for the VICINITY case or by the respective user's official guardian. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/measurements **Get Logs** ○ Functionality: Returns a set of audit logs containing information related to user's account access. Such logs are generated after actions like the previous measurement push. This functionality will be used by the user's authorized guardian. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/logs **Get Statistics** ○ Functionality: Returns a set of measurements per user and per date for a specific period and measurement type (e.g. blood pressure measurements per user for the last month). Such statistics are then processed by other VASs to produce useful conclusions for the Municipality's healthcare personnel. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/statistics **Delete Account** ○ Functionality: Deletes user's account and all measurements that were stored and related to him/her. This way, the VAS guarantees patient's right to be forgotten which is one of the main GDPR principles. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/delete-account
Selected implementation algorithms	GDPR VAS is responsible for the proper storage of the measurements, plus the necessary generation of relevant audit logs, before forwarding them to VAS 3.1.2 as well as functionalities related to fetching sets of measurements, statistics or logs. Its functionality can be summarized in keeping the control of the state (by logs, etc) before forwarding the

	VICINITY requests and does not involve any further processing than that. Moreover, it gives the user the right to be forgotten by deleting his account and all earlier stored data connected with it.
Deployment/Hosting of VAS	GDPR VAS is deployed on MPH server as a Spring Boot application built with Java 1.8. The latter server is also the one running MPH's existing infrastructure where relevant MySql is deployed as well. eHealthPass application that is used alongside GDPR VAS, mainly for the storage as described above, is deployed on an Apache Tomcat 8.0.5 and built with Java 1.8.
Pending issues	Potential updates might occur due to the continuous actual operation and evaluation of the VAS which might produce errors or/and improvements that escaped our attention during development.
Deviations/changes	None

5.1.1.2. Analysis and clustering of elderly’s people medical data to detect unusual behavioural events

VAS 3.1.2	**Analysis and clustering of elderly’s people medical data to detect unusual behavioural events**
Related Use Case	UC 3.1 – eHealth and Assisted Living for elderly people at home
Functionalities delivered	**Push Measurement** ○ Functionality: The service receives new measurements of blood pressure monitors, panic buttons and weight scales. It calculates statistics for the elder, as the frequency of taking measurements for this week and the usual time, creates or updates the respective table entries in the DB and checks for abnormal values e.g. a measurement taken after midnight. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/push_measurement **Get User Profile** ○ Functionality: The service returns some metrics for the elder as the mean frequency, mean blood pressure, usual time of day the measurement is taken, user type etc. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/user_profile **Get Latest Frequencies** ○ Functionality: The service provides the frequency of measurements per week, for the past x weeks. (frequency diagram) ○ Endpoint: /objects/{vas_vicinity_oid}/properties/latest_frequencies **Get Latest Notifications** ○ Functionality: The service returns the x latest notifications for a specific elder. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/notifications

Selected implementation algorithms	A processing of blood pressure, panic button and weight measurements of elders is performed by this VAS. The elders are classified according to the frequency of taking measurements (frequent, normal, sparse), the usual time they take measurements, their mean blood pressure etc. Notifications are also produced in cases that the frequency of measurements or the time/value of a measurement is out of the specified limits.
Deployment/Hosting of VAS	The VAS is a Java 8 web application, which needs a connection to an SQL database. It can be run on any PC, by executing the compiled .jar file.
Pending issues	We expect to have updates on the service after the actual deployment based on the user experience and as part of T5.3.
Deviations/changes	We were not able to use real medical data during the analysis of requirements of the VAS, but we were able to identify them with the help of the municipal doctor of Pilea-Hortiatis.

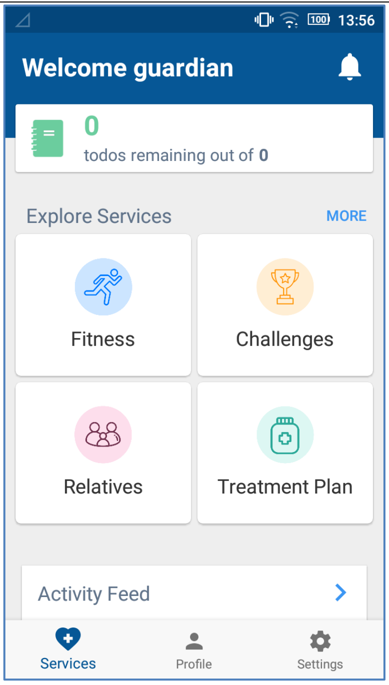
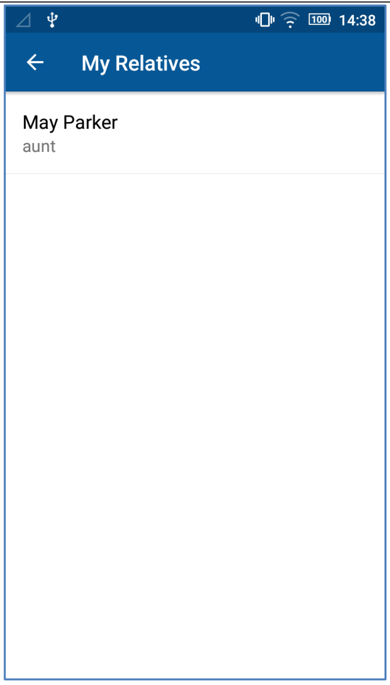
5.1.1.3. Triggering abnormal detection in homes

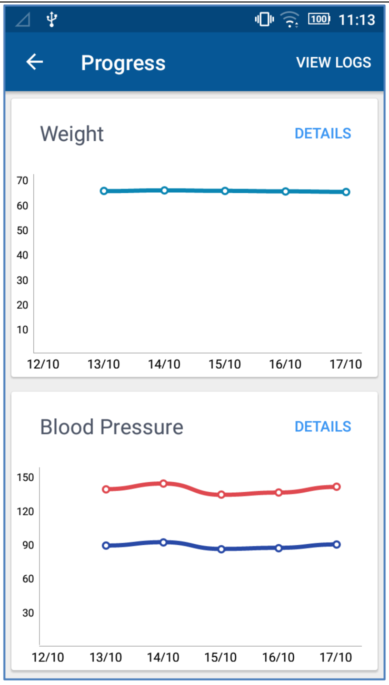
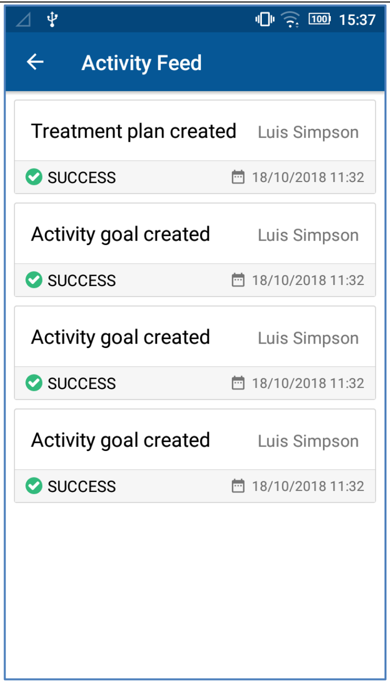
VAS 3.1.3	**Triggering abnormal detection in homes**
Related Use Case	UC 3.1 - eHealth and Assisted Living for elderly people at home
Functionalities delivered	**Calculate metrics (activity level, mobility level etc)** ○ Functionality: For each user a pre-processing of the building sensor measurements of the past day is performed. Metrics that were chosen from respective literature on the subject include activity level, mobility level, non-response interval, rooms usage, rooms duration. **notice*: internal functionality for the VAS to work properly **Calculate abnormal points** ○ Functionality: For each user perform an abnormal detection algorithm for the past day's calculated metrics. The models for each user, metric have been calculated offline by using 30 days training data. **notice*: internal functionality for the VAS to work properly **Get previous day behaviour** ○ Functionality: Provides the behaviour of the elder during the last day. The rooms he/she visited and at which times. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/behavior **Get Rooms Visits and duration**

	○ Functionality: Returns the number of visits and the duration per room for a specific day. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/rooms_visits_duration **Get metrics** ○ Functionality: Returns the metric values for a specific elder and a specific day. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/{metric_name} **Get abnormal points** ○ Functionality: Returns the abnormal events for a specific elder and a specific day. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/abnormal **Get notifications** ○ Functionality: The service returns the latest notifications for a specific elder. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/notifications
Selected implementation algorithms	The VAS is performing a pre-processing of the building sensor data to produce valuable metrics, regarding the behaviour of the elder in the house. A 30-day training period should be performed before the algorithm can detect any abnormal behaviour of a specific elder's behaviour.
Deployment/Hosting of VAS	The VAS is composed by two components. A Python application and a Java web application. It uses Python 3.6, Java 8 and SQL Database as requirements and can run on any PC.
Pending issues	We expect to have updates on the service after the actual deployment based on the user experience and as part of T5.3.
Deviations/changes	Since abnormal events are not that frequent, we have identified the need to also present the behavioural patterns of the elders as an additional help to their guardians to better monitor them. In case of abnormality a notification to the guardians' UI will appear, instead of signal to the call center.

5.1.2. User Interfaces

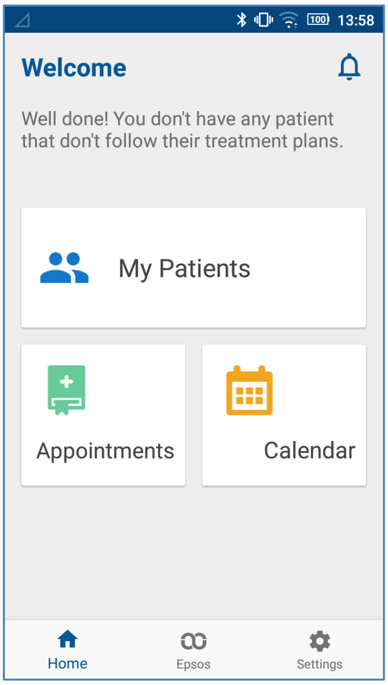
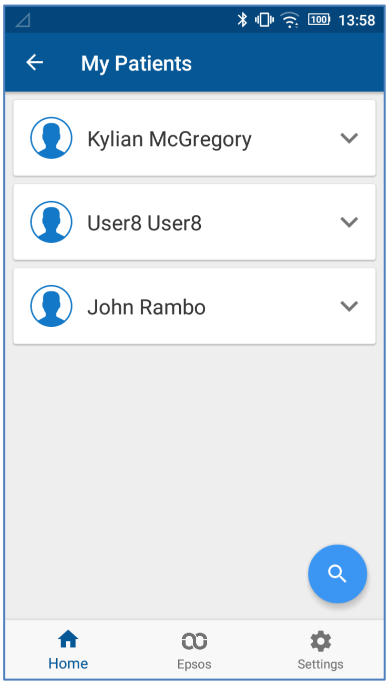
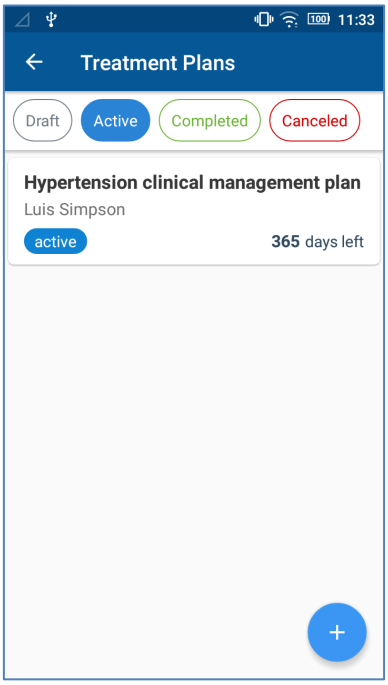
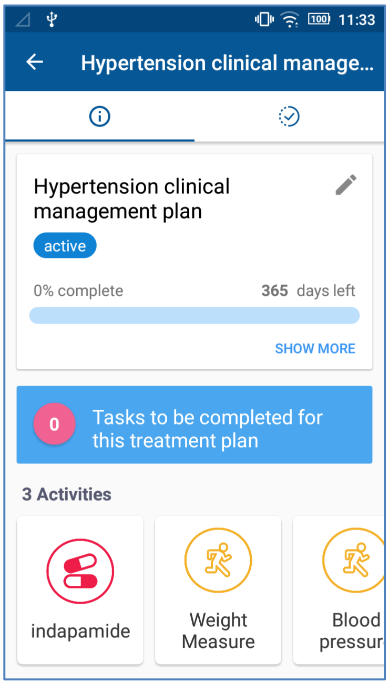
5.1.2.1. Individual medical history tracking (for Relatives)

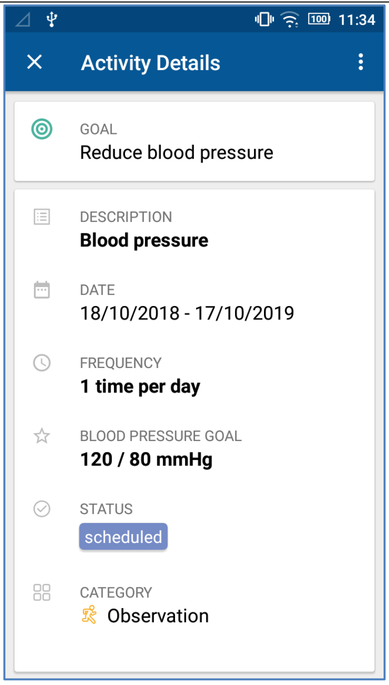
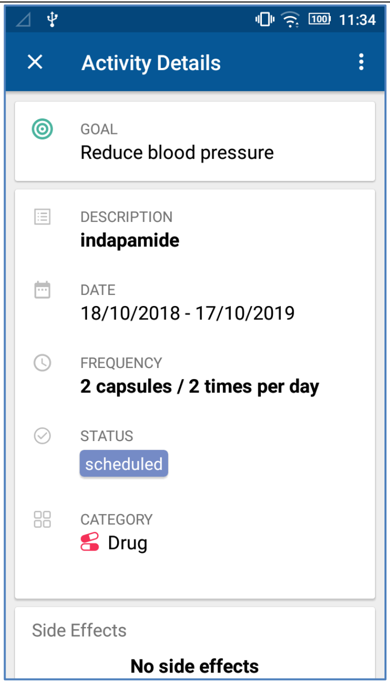
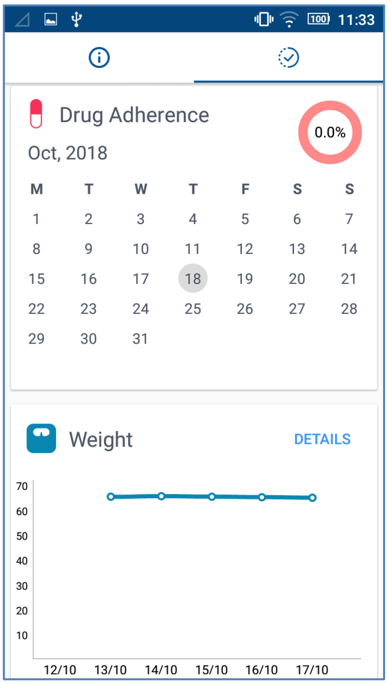
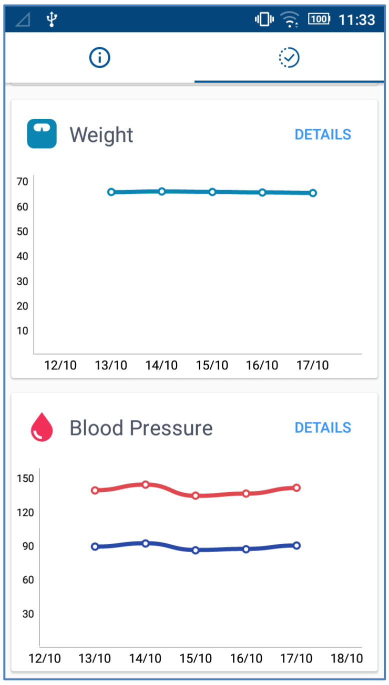
UI 3.1.1	**Individual medical history tracking (for Relatives)**
Related Use Case	UC 3.1 – eHealth and Assisted Living for elderly people at home
Functionalities delivered	**Get list of relatives (protected members)** ○ Functionality: Returns a list of elder citizens for whom the logged in user is officially registered as guardian. **Get relative's measurements** ○ Functionality: Returns a list of measurements for a specific type (e.g. blood pressure) and elder for whom the logged in user is officially registered as guardian. **Get relative's account logs** ○ Functionality: Returns a list of audit logs containing account access information for an elder for whom the logged in user is officially registered as guardian.
Example of use	A user, who is also officially registered as guardian of an elder's citizen, logs in to eEhealthPass. By clicking the "Relatives" section of the home dashboard, a list of relatives/protected members is shown up. He can then select the relative of his choice and check related measurements (e.g. blood pressure) and account access audit logs on the respective screens.
Screenshots	 13:56 Welcome guardian 0 todos remaining out of 0 Explore Services MORE Fitness Challenges Relatives Treatment Plan Activity Feed Services Profile Settings 

	 11:13 Progress VIEW LOGS Weight DETAILS 70 60 50 40 30 20 10 12/10 13/10 14/10 15/10 16/10 17/10 Blood Pressure DETAILS 150 120 90 60 30 12/10 13/10 14/10 15/10 16/10 17/10  Treatment plan created Luis Simpson SUCCESS 18/10/2018 11:32 Activity goal created Luis Simpson SUCCESS 18/10/2018 11:32 Activity goal created Luis Simpson SUCCESS 18/10/2018 11:32 Activity goal created Luis Simpson SUCCESS 18/10/2018 11:32
Deviations/changes	None

5.1.2.2. Individual and clustering analysis of patients' medical history (for Health Care Professionals)

UI 3.1.2	**Individual and clustering analysis of patients' medical history (for Health Care Professionals)**
Related Use Case	UC 3.1 - eHealth and Assisted Living for elderly people at home
Functionalities delivered	**Get list of patients** ○ Functionality: Returns a list of patients who gave consent to the logged in health care professional. **Generate treatment plan for patient** ○ Functionality: The logged in health care professional can generate a treatment plan for any of the patients who have given consent to him. **Get patient's measurements (treatment plan progress)** ○ Functionality: Returns a list of patient's measurements (e.g. blood pressure, weight) who has given consent to the logged in health care professional. The measurements are presented in diagram form as progress of a respective treatment plan. **Abnormal detection notification** ○ Functionality: The logged in health care professional receives a notification related to an abnormal detection of a patient's progress (e.g. high blood pressure).

Example of use	The doctor logs in eHealthPass mobile application. By clicking “My Patients” button of the main dashboard, he is transferred to the list of patients who have given consent to him. After picking any of them, he generates a treatment plan and tracks its progress by listing patient’s measurements (e.g. blood pressure) in a daily diagram form.
Screenshots	 13:58 Welcome Well done! You don't have any patient that don't follow their treatment plans. My Patients Appointments Calendar Home Epsos Settings  13:58 My Patients Kylian McGregory User8 User8 John Rambo Home Epsos Settings  11:33 Treatment Plans Draft Active Completed Canceled Hypertension clinical management plan Luis Simpson active 365 days left  11:33 Hypertension clinical manage... Hypertension clinical management plan active 0% complete 365 days left SHOW MORE 0 Tasks to be completed for this treatment plan 3 Activities indapamide Weight Measure Blood pressur

	 11:34 Activity Details GOAL Reduce blood pressure DESCRIPTION **Blood pressure** DATE 18/10/2018 - 17/10/2019 FREQUENCY **1 time per day** BLOOD PRESSURE GOAL **120 / 80 mmHg** STATUS scheduled CATEGORY Observation  11:34 Activity Details GOAL Reduce blood pressure DESCRIPTION **indapamide** DATE 18/10/2018 - 17/10/2019 FREQUENCY **2 capsules / 2 times per day** STATUS scheduled CATEGORY Drug Side Effects **No side effects**  11:33 Drug Adherence 0.0% Oct, 2018 M T W T F S S 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 Weight DETAILS 70 60 50 40 30 20 10 12/10 13/10 14/10 15/10 16/10 17/10  11:33 Weight DETAILS 70 60 50 40 30 20 10 12/10 13/10 14/10 15/10 16/10 17/10 Blood Pressure DETAILS 150 120 90 60 30 12/10 13/10 14/10 15/10 16/10 17/10 18/10
Deviations/changes	None

5.2. Use Case 3.2 – Health improvement for the middle-aged persons

The current Use Case deals with the promotion of a healthy lifestyle on behalf of the Municipality that will result on the health improvement of its middle-aged citizens. Everyday routine tasks and medical data collected by relevant IoT devices consist the main scope of monitoring that will end up in a municipal-scale competition (“urban marathon”) to reward the health improvement achievements. The scenario involves authorized healthcare personnel (dietician) to interpret respective information.

The following figure provides a schematic representation of the Use Case in relation to the VICINITY architecture (Organisations, Users, Things, Friendships, etc).

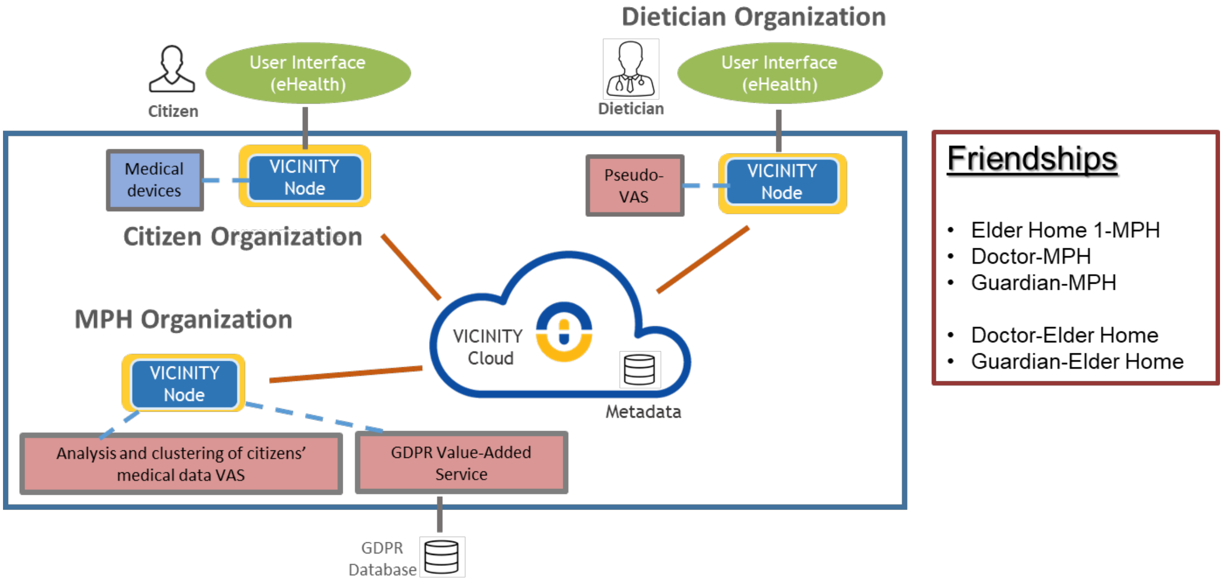


Figure 5-2 Use Case 3.2 - Overview

5.2.1. VICINITY Value-Added Services

5.2.1.1. Privacy-preserving Data Gathering and Storage ft. GDPR data auditing

VAS 3.2.1	**Privacy-preserving Data Gathering and Storage ft. GDPR data auditing**
Related Use Case	UC 3.2 – Health improvement for the middle-aged persons
Functionalities delivered	**Push Measurement** ○ Functionality: Similar to VAS 3.1.1 (see Section 5.1.1.1) with the main difference of the user being a middle-aged citizen using a mobile application as a UI instead of an elder sending measurements via Raspberry Pi. The measurement is then sent to eHealthPass and forwarded to VAS 3.2.2. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/prop **Get Measurements** ○ Functionality: Returns a set of user's measurements as fetched by eHealthPass database described above. A user shall be able to access all his/her measurements pushed by the previous functionality. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/measurements **Get Logs** ○ Functionality: Returns a set of audit logs containing information related to user's account access. Such logs are generated after

	actions like the previous measurement push and are stored in eHealthPass database in a similar manner as the latter. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/logs **Get Statistics** ○ Functionality: Returns a set of measurements per user and per date for a specific period and measurement type (e.g. number of steps per user for the last month). Such statistics are then processed by other VASs to produce useful conclusions for the Municipality's medical personnel. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/statistics **Register to Urban Marathon** ○ Functionality: Forwards the request to VAS 3.2.2 User Registration functionality. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/register **Delete Account** ○ Functionality: Deletes user's account and all measurements that were stored and related to him/her. This way, the VAS guarantees patient's right to be forgotten which is one of the main GDPR principles. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/delete-account
Selected implementation algorithms	See section 5.1.1.1
Deployment/Hosting of VAS	See section 5.1.1.1
Pending issues	See section 5.1.1.1
Deviations/changes	See section 5.1.1.1

5.2.1.2. Individual Statistical Analysis of data from wearables, medical devices, beacons

VAS 3.2.2	**Individual Statistical Analysis of data from wearables, medical devices, beacons**
Related Use Case	UC 3.2 – Health improvement for the middle-aged persons
Functionalities delivered	**User Registration** ○ Functionality: Stores to MySQL Database information (UserId, height, gender) about the new user who will register at the Urban Marathon. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/Register-user **Push Measurement** ○ Functionality: Stores to MySQL Database information (weight, steps, times to gym) when an event is triggered and calculates the gained and final points of the user depending on user's gender and BMI (Body Mass Index). Each user depending on these characteristics has 2 weeks targets / rules regarding weight loss, steps and visits to Municipality's gym. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/push_measurement **Get Rules** ○ Functionality: This service receives weight, height and gender, calculates BMI and returns the rules regarding weight loss, steps and visits to Municipality's gym for these traits. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/Get-rules **Get Ranking** ○ Functionality: This service returns ranking of the user and other information about his progress and the Urban Marathon. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/user_ranking **Get Points for User** ○ Functionality: This service returns last gained points for event for the user. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/get_points_for_user
Selected implementation algorithms	Algorithms used for the Urban Marathon were implemented and detailed based on the guidance provided by experienced medical staff (dietician) to produce the final urban marathon ranking system. Aspects considered were: (i) the gender and age of the citizens, (ii) target weight loss, (iii)

	daily steps and (iv) target visits to gym per week for each citizen. Points are awarded to citizens if they accomplish their goals or in proportion to their targets. Anonymized data are saved to calculate points for each citizen in database (MySQL) deployed on the MPH secure server, whereas specific personal data of the citizens are only stored and processed by the GDPR VAS 3.2.1.
Deployment/Hosting of VAS	Urban Marathon VAS is deployed on MPH Server on Liferay (Liferay Portal Community Edition 6.2 CE GA6) running a Tomcat server 7.0.56 with Java (JRE) 1.7. The database used is MySQL.
Pending issues	Potential updates might occur due to the continuous actual operation and evaluation of the VAS which might produce errors or/and improvements that escaped our attention during implementation.
Deviations/changes	After discussions with the dietician of the municipality, we have decided to exclude from this VAS calculations any measurements regarding blood pressure, which deviates from what we initially described in D5.1 “VICINITY Value-Added Services definition, requirements and architectural design”. This use case’s IoT infrastructure includes activity trackers, weight scales and beacons.

5.2.1.3. Aggregated Statistical Analysis of data from wearables, medical devices, beacons

VAS 3.2.3	**Aggregated Statistical Analysis of data from wearables, medical devices, beacons**
Related Use Case	UC 3.2 – Health improvement for the middle-aged persons
Functionalities delivered	**Get Citizens** ○ Functionality: It returns the number of citizens participating in the Urban Marathon as fetched by the database. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/GetCitizens **Get Weight Steps Statistics** ○ Functionality: Calculates and returns, • average weight for men and women, • percentage of women and men that are overweight, • average steps per day and per month for women and men • BMI categories for men and women per month ○ Endpoint: /objects/{vas_vicinity_oid}/properties/WeightStepsStatistics **Get Gym Statistics** ○ Functionality: Calculates and returns, • visits to each gym of the 10 cooperating with the MPH for the Urban Marathon from men and women, • average visits to the gym per month for women and men ○ Endpoint: /objects/{vas_vicinity_oid}/properties/GymStatistics **Get Points** ○ Functionality: Calculates and returns, • average points per activity (weight loss, times to gym, steps) for women and men • ranking of the users – anonymised ○ Endpoint: /objects/{vas_vicinity_oid}/properties/PointsRanking
Selected implementation algorithms	This VAS is making anonymised statistical analysis of the middle-aged citizens of MPH regarding their fitness and wellbeing. The VAS is calculating the average values regarding citizens' weight and steps for each gender and shows statistics regarding the usage of local gyms.

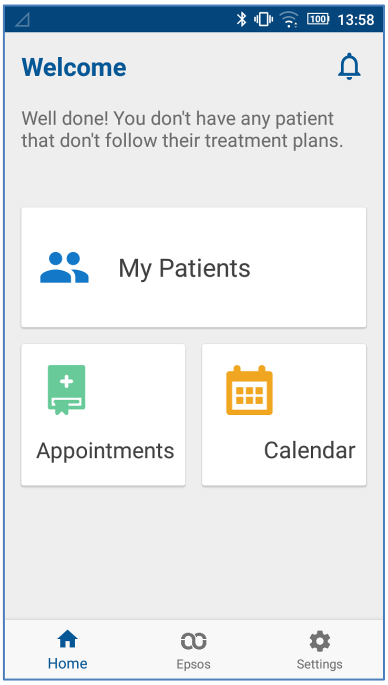
Deployment/Hosting of VAS	Same as VAS 3.2.2
Pending issues	Potential updates might occur due to the continuous actual operation and evaluation of the VAS which might produce errors or/and improvements that escaped our attention during implementation.
Deviations/changes	None

5.2.2. User Interfaces

5.2.2.1. Individual and statistical analysis of patients' history (for Dieticians)

UI 3.2.1	**Individual and statistical analysis of patients' history (for Health Care Professionals)**
Related Use Case	UC 3.2 - Health improvement for the middle-aged persons
Functionalities delivered	**Get list of patients** ○ Functionality: Returns a list of patients who gave consent to the logged in dietician. **Register patient to Urban Marathon** ○ Functionality: The logged in dietician can register any of the patients who have given consent to him to the municipality's urban marathon by providing weight measurement. A diet management treatment plan is automatically generated on the background. **Get patient's measurements (treatment plan progress)** ○ Functionality: Returns a list of patient's measurements (e.g. steps, weight) who has given consent to the logged in dietician. The measurements are presented in diagram form as progress of a respective treatment plan.
Example of use	The dietician logs in eHealthPass mobile application. By clicking "My Patients" button of the main dashboard, he is transferred to the list of patients who have given consent to him. After picking any of them, he registers any of them to the municipality's urban marathon by providing a weight measurement to the respective input. A treatment plan is automatically generated and the dietician can track its progress by listing patient's measurements (e.g. steps, weight, etc) in a daily diagram form.

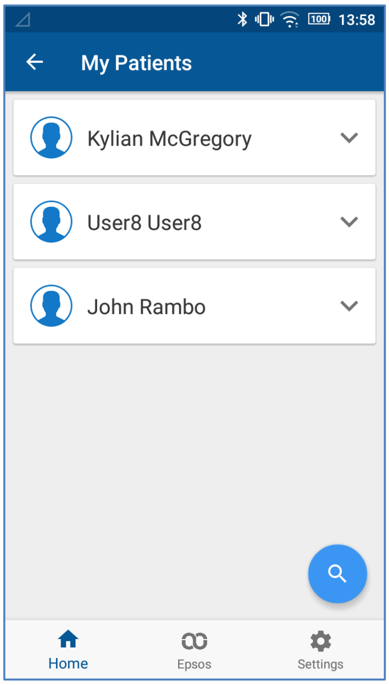
Screenshots



13:58

Welcome

Well done! You don't have any patient that don't follow their treatment plans.



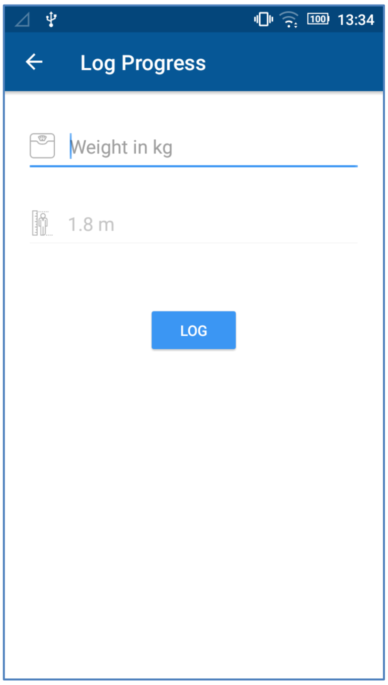
13:58

My Patients

Kylian McGregory

User8 User8

John Rambo

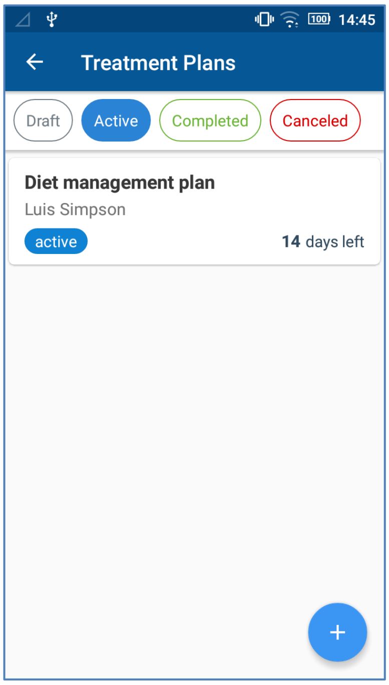


13:34

Log Progress

Weight in kg

1.8 m



14:45

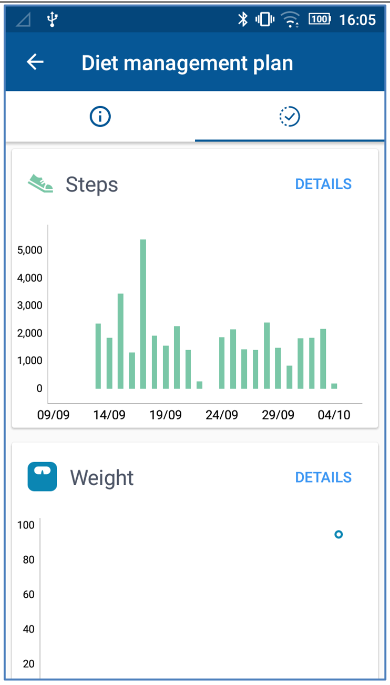
Treatment Plans

Diet management plan

Luis Simpson

active

14 days left

	
Deviations/changes	None

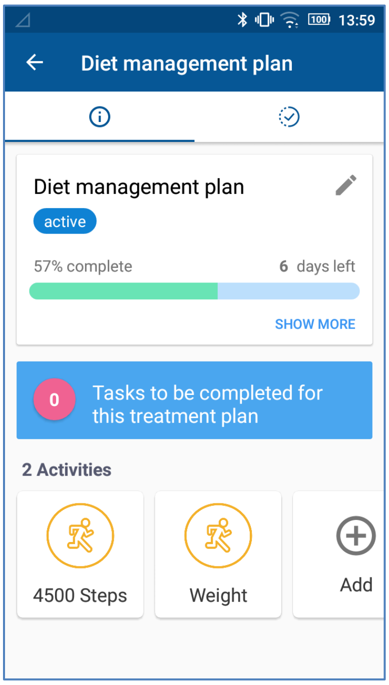
5.2.2.2. Tracking progress and bonus achieved (for citizens)

UI 3.2.2	**Tracking progress and bonus achieved (for citizens)**
Related Use Case	UC 3.2 - Health improvement for the middle-aged persons
Functionalities delivered	**View treatment plan** ○ Functionality: The logged in user can view the, generated by the dietician, diet management treatment plan to follow under the concept of the Urban Marathon. **Synchronize IoT devices** ○ Functionality: The logged in user can synchronize his IoT devices (i.e. activity tracker, weight scale monitor) with his mobile via the respective functionality offered by eHealthPass. The synchronized data will be pushed so that the user gains the relevant points for the Urban Marathon. **Get Urban Marathon ranking** ○ Functionality: The logged in user can check his total points and general ranking among the rest of Urban Marathon participants. **Get measurements** ○ Functionality: Returns a list of user's measurements (e.g. steps, weight). The measurements are presented in diagram form as progress of a respective diet management plan.

Example of use

The user logs in eHealthPass and checks his diet plan running under the Urban Marathon concept. As a main goal of the plan is to exceed the daily limit of 4,500 steps, he synchronizes his activity tracker with the application and pushes the relevant data fetched. He can then check the result the latter action had on his overall ranking.

Screenshots



Diet management plan

Diet management plan

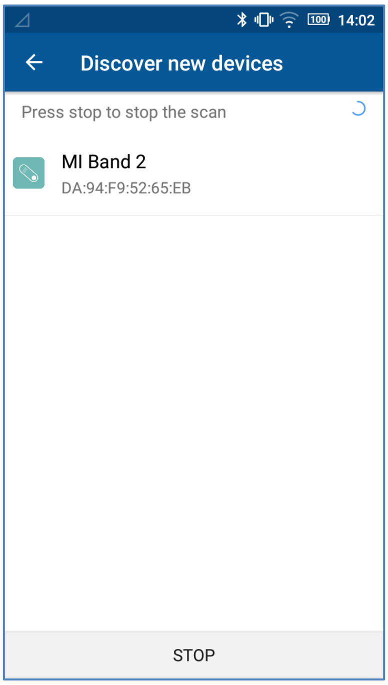
active

57% complete

6 days left

0 Tasks to be completed for this treatment plan

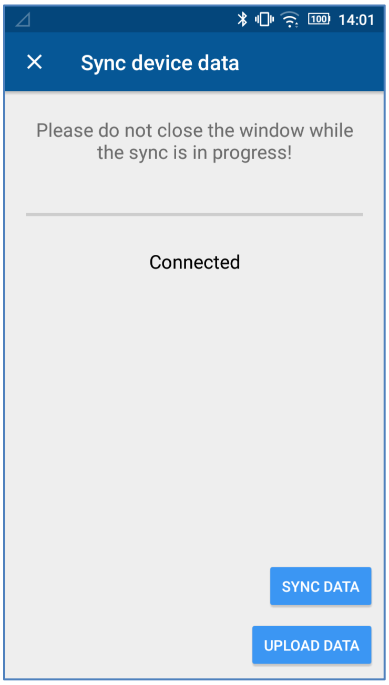
2 Activities



Discover new devices

Press stop to stop the scan

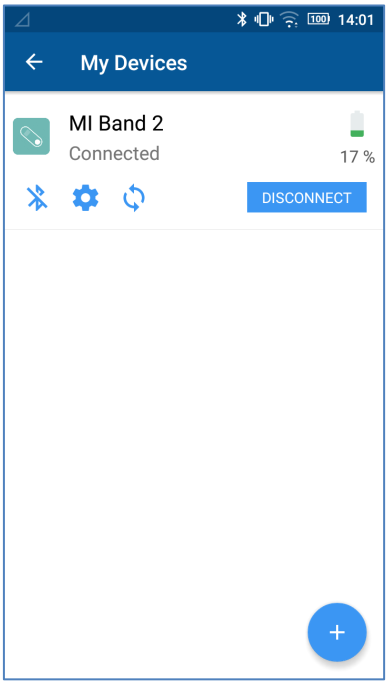
MI Band 2
DA:94:F9:52:65:EB



Sync device data

Please do not close the window while the sync is in progress!

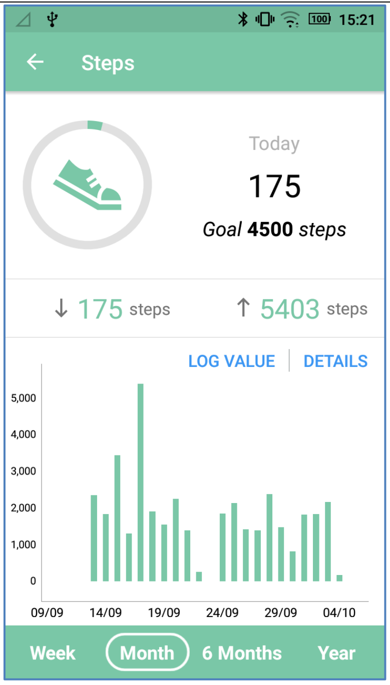
Connected



My Devices

MI Band 2
Connected

17 %



15:21

Steps

Today

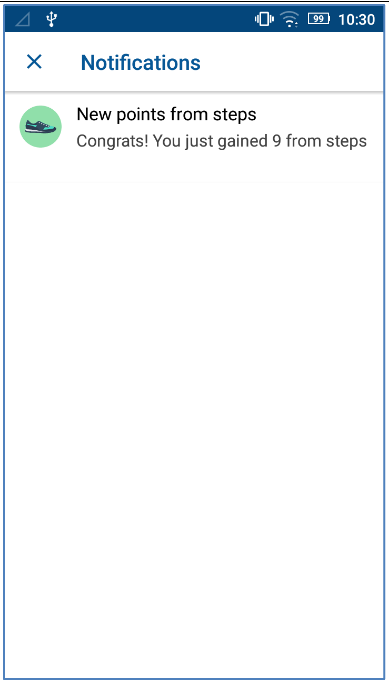
175

Goal **4500** *steps*

↓ 175 steps ↑ 5403 steps

LOG VALUE | DETAILS

Week | Month | 6 Months | Year

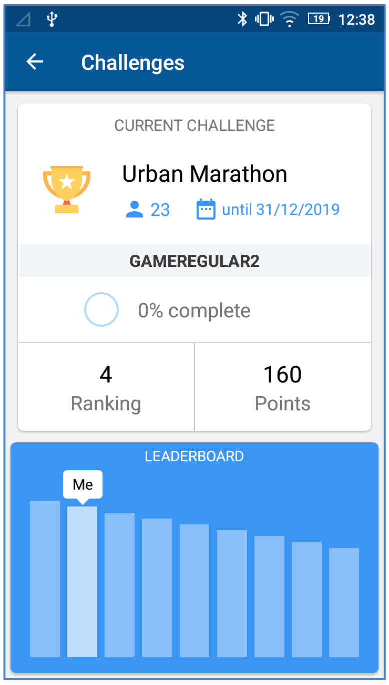


10:30

Notifications

New points from steps

Congrats! You just gained 9 from steps



12:38

Challenges

CURRENT CHALLENGE

Urban Marathon

23 until 31/12/2019

GAMEREGULAR2

0% complete

4 Ranking

160 Points

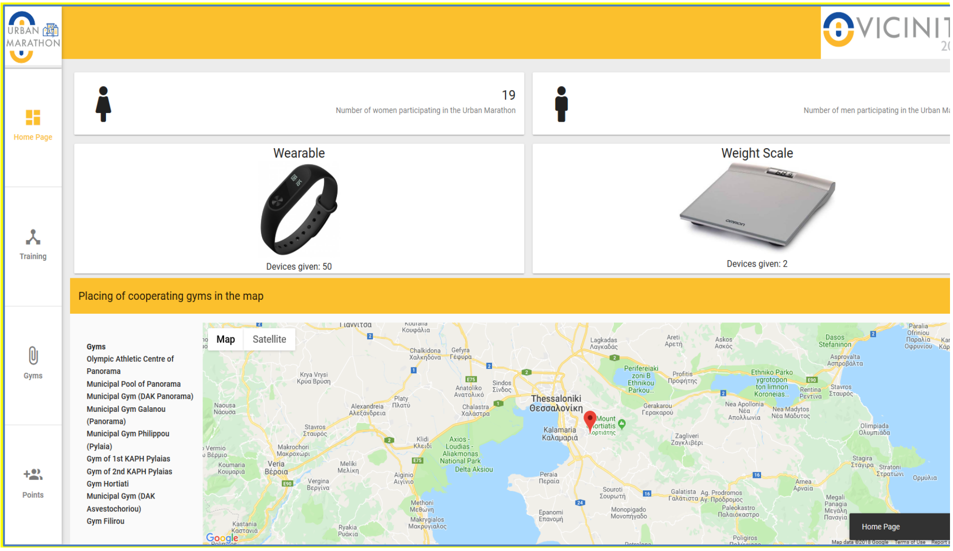
LEADERBOARD

Deviations/changes	None

5.2.2.3. Aggregated statistical analysis of municipality citizens progress (for Municipalities)

UI 3.2.3	**Aggregated statistical analysis of municipality citizens progress (for Municipalities)**
Related Use Case	UC 3.2 - Health improvement for the middle-aged persons
Functionalities delivered	Functionalities: • View number of women and men participating in the Urban Marathon, cooperating gyms and IoT devices given to the citizens • View weight statistics for women and men, the percentage of women and men that are overweight, average steps per day and per month for women and men, BMI categories for men and women per month • View visits to each gym of the 10 cooperating with the MPH for the Urban Marathon from men and women, average visits to the gym per month for women and men • View average points per activity (weight loss, times to gym, steps) for women and men and anonymised ranking of the users
Example of use	An admin from the MPH logs in with his credentials to the Municipality Web UI and views the anonymised statistical analysis of the municipality citizens' data, keeping track of their health condition. The UI presents how this use case facilitates health improvement of middle-aged people.

Screenshots



19
Number of women participating in the Urban Marathon

Number of men participating in the Urban Ma

Wearable

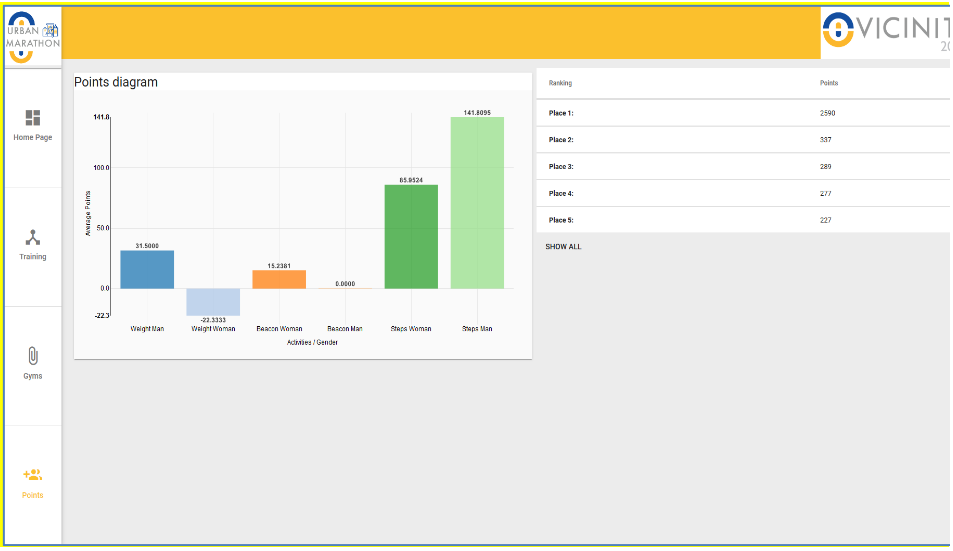
Devices given: 50

Weight Scale

Devices given: 2

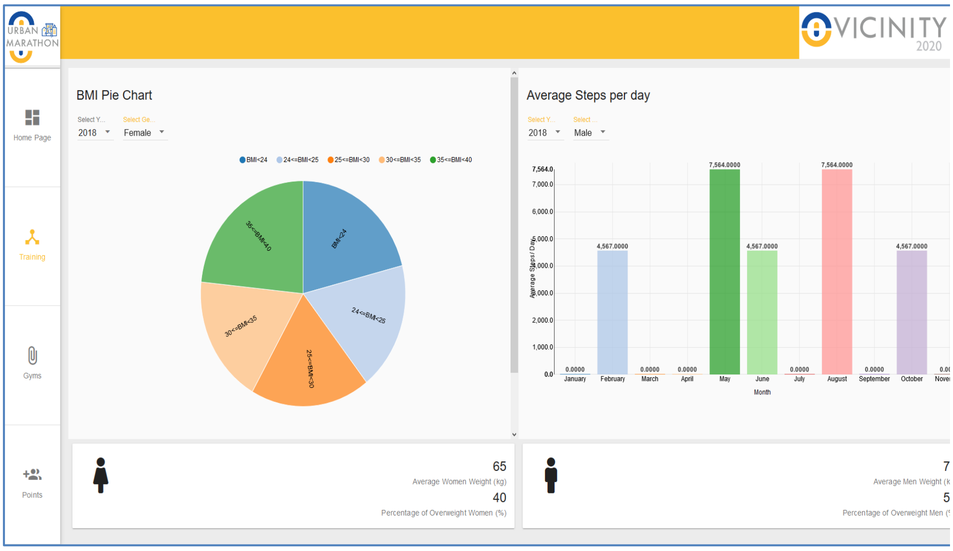
Placing of cooperating gyms in the map

Gyms
Olympic Athletic Centre of Panorama
Municipal Pool of Panorama
Municipal Gym (DAK Panorama)
Municipal Gym Galanou (Panorama)
Municipal Gym Philippou (Pylaia)
Gym of 1st KAPH Pylaias
Gym of 2nd KAPH Pylaias
Gym Hortiati
Municipal Gym (DAK Asvestochoriou)
Gym Filirou



Ranking	Points
Place 1:	2590
Place 2:	337
Place 3:	289
Place 4:	277
Place 5:	227

SHOW ALL



Select Y... 2018
Select Ge... Female

Select Y... 2018
Select ... Male

65
Average Women Weight (kg)
40
Percentage of Overweight Women (%)

7
Average Men Weight (k
5
Percentage of Overweight Men (

Deviations/changes

None

5.3. Ethics

EHealth is an area where highly personal and sensitive data is processed. Data protection and privacy concerns have been a showstopper for eHealth ambitions in the cloud sector so far. The Healthcare pilot scenario will address two use cases, the so-called "ehealth and assisted living" and the "fitness and preventive medicine". In the latter two use cases, technology will be applied for acquiring data on person's daily life, including health data. This will be achieved with the help of wearable sensors and wearable computing and any participation within VICINITY as a user or patient is entirely voluntary. Participants will have the right to withdraw from the research at any time, and to withdraw retrospectively any consent given.

Implementation of the VASs of this pilot had to take into consideration the above restrictions concerning sensitive medical data. During the design phase of the VASs of this pilot, we had already defined the implementation of a service, which will be responsible for storing and handling the sensors data. This is VAS 3.1.1 for Use Case 3.1 and VAS 3.2.1 for Use Case 3.2. Respecting the GDPR regulation, the VAS is keeping a record of all the data transactions and can inform each user about who has requested access to their data. Moreover, it is responsible for granting access to specific user data, only to certain VICINITY users and it also gives the ability to users to delete their personal records at any time (right to be forgotten). The rest of the VASs of the pilot, acquire the data they need for processing only through this VAS and do not have direct access (no VICINITY Contract) to the sensor data.

Certain actions, complementary to the actual VAS development, were also needed for this pilot. More specifically, the municipality (MPH) has prepared consent forms for all the participants in the project, regarding the usage of their medical data. These forms need to be signed by the elder, or the middle aged respectively, person before he/she connects to VICINITY platform.

Furthermore, the above two use cases need to conform to Greek regulations regarding data protection. For this reason, it was requested and acquired, a certificate from the Hellenic Data Protection Authority (DPA) regarding the MPH server which will host the aforementioned VASs.

5.4. VICINITY Value

Based on the initial identification of how VICINITY acts as an enabler for the specific use cases (based on the information provided in D5.1 under sections 6.2.6, 6.3.6), the assessed added value provided is summarized in the following tables,

Value of VICINITY to use case 3.1	
Problem	Solution
Lack of **IoT Semantic interoperability**	Deployed and utilized 3 different IoT Platforms integrated with both health and building domain sensors, namely: **IoTivity**, **NodeRED** and **Gorenje cloud** service
Handling user medical data respecting user **privacy** and user **rights**	- VICINITY enables the transmission of collected medical devices data (from 40 different elderly houses) ONLY to the contracted GDPR service, which is deployed on the secure server hosted by the municipality of MPH. - The collected data are NOT stored by the VICINITY platform (VICINITY only handles transition permission), but on the secure MPH server.

Value of VICINITY to use case 3.2	
Problem	Solution
Device Semantic interoperability	The **Xiaomi Mi Band 2** Activity Tracker has been fully integrated to VICINITY with a corresponding VICINITY adapter. Similar devices from other vendors (e.g. **Garmin, Fitbit**, etc) are on the process of integration, allowing vendor lock prevention.
Handling user medical data respecting user **privacy** and **rights**	- VICINITY enables the transmission of collected medical devices data (from 50 different middle-aged citizens) ONLY to the contracted GDPR service, which is deployed on the secure server hosted by the municipality of MPH. - The collected data are NOT stored by the VICINITY platform (VICINITY only handles transition permission), but on the secure MPH server.

6. AAU Microgrid-IoT Testing Lab (DK) – Residential Microgrid & Smart Parking (AAU)

Chapter 6 is introduced in this deliverable to present the Value-Added Services implemented by AAU. These Value-Added services were not defined in D5.1 and will not be used in any of the above pilot sites. Instead, they will be used for VICINITY testing by AAU Lab in Tasks 6.1 and 6.2 which is the main reason they are presented in this deliverable. An additional future use could be related to the 2nd VICINITY Open Call where it could be used by the Open Call partners as a testing environment.

6.1. Testing Case – Optimal usage of parking slots by considering energy costs

The current testing case deals with providing users with data about the numbers of free parking slot and the real-time charging price for EV. Parking slot usage data is collected through VICINITY by using three parking sensors to achieve monitoring function. A residential microgrid, which consists of PV, wind turbine and battery, is emulated to supply power to three EV chargers. The real-time charging price is calculated by considering the simulated real-time utility electricity price, state of charge of batteries, and forecasts of the PV and wind turbine power generation. The parking slot usage and the real-time charging price will be sent automatically to users after subscribing Energy Management System-Parking VAS.

6.1.1. VICINITY Value-Added Services

6.1.1.1. Vacant parking slot and charging price notifications service

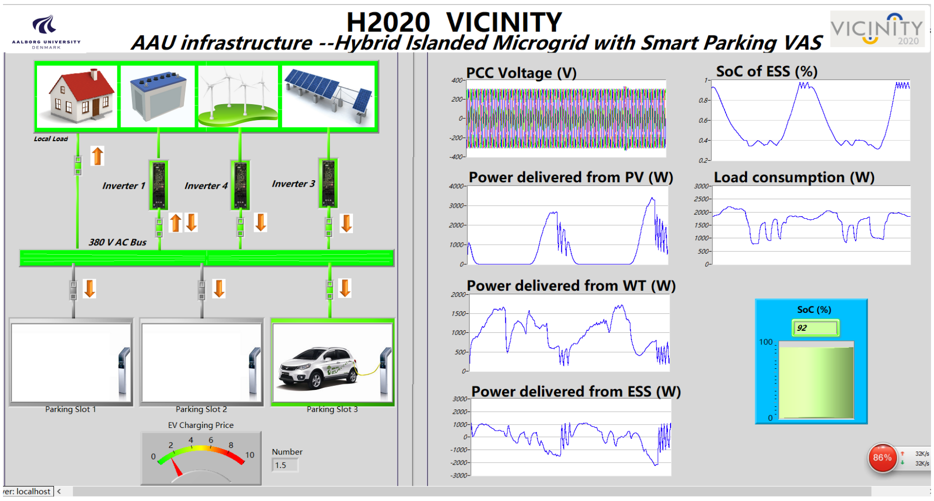
VAS 6.1.1.1	**Vacant parking slot and charging price notifications service**
Related Use Case	Testing Case
Functionalities delivered	**Get active power consumption** ○ Functionality: Returns active power consumption in the residential microgrid. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/Load_ActivePower **Get active power generation of PV** ○ Functionality: Returns active power generation of the PV system in the residential microgrid. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/PV_ActivePower **Get active power generation of WT** ○ Functionality: Returns active power generation of the wind turbine in the residential microgrid. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/WT_ActivePower **Get the state of charge of energy storage systems**

	○ Functionality: Returns state of charge of batteries in the residential microgrid. ○ Endpoint: /objects/{vas_vicinity_oid}/properties/BMS_SoC **Send Message** ○ Functionality: The service sends the number of free parking slots and the charging price for EVs to the subscribers. ○ Endpoint: /events/ParkingAndChargingStatus
Selected implementation algorithms	The VAS monitors the parking lot usage and calculates the real-time charging price. The parking slot usage and the real-time charging price will be sent automatically to users to let them choose the preferred parking time.
Deployment/Hosting of VAS	The VAS is established based on Python 3, which is connected to LabVIEW-based Energy Management System through TCP/IP. It can be run on any PC by executing the .py file.
Pending issues	Potential improvements by integration with other devices and sensors
Deviations/changes	Potential changes in sending more information to increase the VAS function.

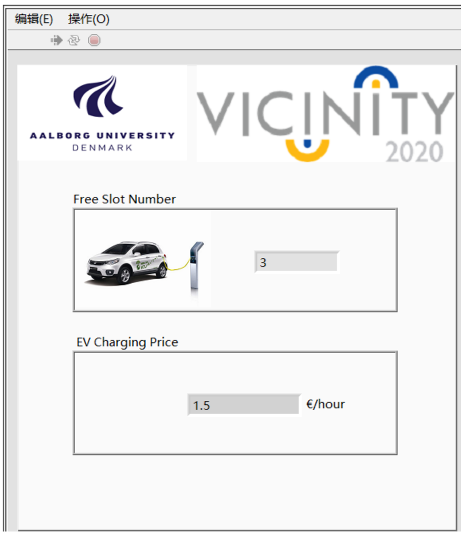
6.1.2. User Interfaces

6.1.2.1. Residential Microgrid Energy Management System with Smart Parking

UI 6.1.2.1	Residential Microgrid Energy Management System with Smart Parking
Related Use Case	Testing Case
Functionalities delivered	Detailed operation performance of the residential microgrid and the parking lot usage are shown in the UI: • Active power outputs of PV/WT/Batteries • Parking lot usage • Real-time charging price
Example of use	The user can choose the preferred parking time based on the vacant parking slot number and real-time charging price. The residential microgrid manager monitors the operation performance of devices, energy resources, and the usage status of the parking lot.

Screenshots	 H2020 VICINITY AAU infrastructure --Hybrid Islanded Microgrid with Smart Parking VAS Local Load Inverter 1 Inverter 4 Inverter 3 380 V AC Bus Parking Slot 1 Parking Slot 2 Parking Slot 3 EV Charging Price Number PCC Voltage (V) SoC of ESS (%) Power delivered from PV (W) Load consumption (W) Power delivered from WT (W) Power delivered from ESS (W) SoC (%) Number 1.5 SoC (%) 92
Deviations/changes	None

6.1.2.2. AAU Smart Parking UI

UI 6.1.2.2	**AAU Smart Parking UI**
Related Use Case	Testing Case
Functionalities delivered	The user views the real-time charging price and the number of vacant parking slots from the UI.
Example of use	The user can subscribe to the vacant parking slot and charging price notifications service to receive the real-time charging price for EVs and the number of vacant parking slots automatically.
Screenshots	AAU Smart Parking UI  编辑(E) 操作(O) AALBORG UNIVERSITY DENMARK VICINITY 2020 Free Slot Number 3 EV Charging Price 1.5 €/hour
Deviations/changes	None

6.2. Ethics

The parking sensor data collected is anonymous. The power output data is collected at a neighbourhood network level that does not have direct access to any specific user's consumption and generation data. In addition, there is a contract (friendship) between service user and service provider.

6.3. VICINITY Value

Based on the initial identification of how VICINITY acts as an enabler for the specific testing cases, the assessed added value provided is summarized in the following tables,

VICINITY to testing case 6.1	
Problem	Solution
Device Semantic interoperability	LabVIEW-based energy management system and LoRaWAN-based parking sensors, that cover energy and transport domains, have been integrated to VICINITY with corresponding VICINITY adapters.
Sharing vacant EV parking slots respecting charging price	The vacant parking slot and charging price notifications service can automatically publish real-time charging price and vacant EV parking slots for subscribed users through VICINITY.

7. Conclusion

This deliverable has presented a detailed definition of the value-added services, including requirements and architectural design specification, to be implemented and demonstrated per each pilot Use Case. The information has been presented in a structured way such that comparisons may be made between the different pilots.

Deliverable D5.2 has been the last and most important step to reach milestone MS6, which was aiming to make a first, prototype version of the VASs available. By presenting in detail what has been implemented so far for all the Use Cases running under each Pilot Site, it can be said that the objectives of the deliverable were achieved.

D5.2 is a continuation of the previously submitted D5.1, the current document provides information for all the VASs that were first introduced and presented in the latter deliverable. Some of the services have been withdrawn from the implementation process, as the result of various issues (technical, deployment, ethical, etc), that were not taken into consideration when D5.1 was developed.

For each Pilot Site, the deliverable provides an overview of the VASs implemented by presenting, in tabular form, functionalities delivered, examples of use, technical/deployment details, possible updates and deviations and ethical issues arisen for each of the services. A reader of the document can thus form an image of the status of the project as the information is presented in a quite easy, straightforward manner.

Finally, relevant Annexes are included with one of them offering an interesting perspective on the project. Specifically, Annex II suggests a possible future extension of homomorphic encryption that will guarantee additional data privacy and increase the credibility of the VICINITY project.

8. References

[1] http://www.vicinity-h2020.eu

[2] ICT 30 – 2015: Internet of Things and Platforms for Connected Smart Objects - http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/914-ict-30-2015.html

Annex I – Templates used

Value-Added Services Template

VAS x.y	**Name of the VAS**
Related Use Case	Use case related to the VAS
Functionalities delivered	List of the implemented functionalities along with respected endpoints
Selected implementation algorithms	Add details when applicable on technical aspects, e.g. modelling approach, clustering/statistical analysis process as finally selected and implemented for delivering the abovementioned functionalities
Deployment/Hosting of VAS	Add VAS hosting information e.g. deployment on Linux PC running a Tomcat server, positioning (distributed on site or on a central location) etc.
Pending issues	Potential updates planned in the course of the realisation - e.g. to be delivered in the next 1-2 months
Deviations/changes	Deviations comparing to what was described in D5.1 for this VAS.

User Interfaces Template

UI x.y.z	**Name of the UI**
Related Use Case	Related Use Case for the UI
Functionalities delivered	UI offered functionality
Example of use	Short story-telling of use of UI per user (e.g. showing in a step-by-step manner)
Screenshots	Screenshots of final version of UIs (Not mock-ups), add as many required to cover abovementioned functionalities
Deviations/changes	Deviations comparing to what was described in D5.1 for this VAS.

ANNEX II – Homomorphic Encryption for data anonymization

This annex describes some features that were added as reaction to the discussion with the reviewers. As a critical issue in the architecture of VICINITY, the loss of control over the data towards the operators of value-added services as described in the DoW was recognized. In the work of research partners, these limitations are tackled as follows:

- UNIKL has investigated and implemented a smart contract use case within a UNIKL test lab based smart energy scenario using DLT technology; however, DLT and smart contracts do not offer any anonymization, just pseudonymization.
- CERTH has investigated the use of DLT in both a broad and a VICINITY-wise context, and how it could help extending the VICINITY concept and architecture to a fully decentralized system. This work will be reported as part of the upcoming deliverables of work packages 3 or 4 (VICINITY client/server operation and continuous upgrades).
- UNIKL has investigated and is implementing a VICINITY micro-service for anonymous data-aggregation; it will be used within the UNIKL (emulation-based) test lab, and likely also in one of the VICINTY use cases.

In the following we give a brief overview of the VICINITY micro-service for anonymous data-aggregation.

Motivation and Impact

The VICINITY project gives privacy considerations much thought. Indeed, privacy is one of the unique selling points compared with other platforms. Considering that GDPR implementation requires a high effort for users of IoT platforms, and management of privacy is supported by a very few platforms (e.g. data.me), giving users the ultimate control over their data is a promise we strive to fulfil. The VICINITY P2P networks and its involved components are designed and implemented following current best practices in terms of data security. We can assume personal data to be secure while being transmitted from one peer to another. By default, value-added services (VAS) have no access to any personal information, nor private sensor readings, etc. The user shall explicitly agree to his data being read and process by VAS.

However, VAS can be offered from 3rd parties. This has two implications:

- The operator of a VAS can be malicious. Even though this VAS can immediately be removed from VICINITY, private data might have leaked out without the user's consent.
- Even a non-malicious operator of a VAS has always to deal with personal (maybe even health) data. Handling such data involves a high legal complexity even beyond GDPR.

To overcome these two issues, UNIKL is adding a VICINITY Micro Service for "anonymous data-aggregation" to the VICINITY IoT platform.

Anonymous Data Aggregation as a Micro Service for Value-Added Services

In some scenarios and use cases, a (value-added) service or its business model requires aggregated data, but not personal data. As an example, consider the predicted overall energy consumption of a building, district or municipality in a smart grid. Currently, a service operator will collect the individual, personal data and calculate the sum on their own. However, for operating his service, e.g. demand side management, and business model, he does not need the individual consumption of all appliances that is personal data with all legal implications: he only needs the aggregation in form of the sum. Hence, the access to personal data is a useless overhead, creating loss of trust and administrative overhead with no need.

The VICINITY anonymous data aggregation micro-service solves these issues:

- No personal data leaves the site and the control of the user.
- The VICINITY IoT platform resp. its micro-service aggregates anonymous data.
- The VAS operator can avoid dealing with aggregated, non-personal data.

For the implementation, we use specific encryption techniques: homomorphic encryption.

Note that:

- The question if GDPR is applicable is legally under discussion; If data have been aggregated such that no data about an identifiable person can ever be extracted, then GDPR does not apply. However, if personal data has been encrypted and a key exists or could be created to recover these data, then GDPR would apply.
- The solution can even offer algorithm accountability which is gaining impact as the user has control over which kind of analysis and decisions are made with on his personal data.

Homomorphic encryption

(Partially) Homomorphic encryption is a form of encryption that enables (limited) mathematical operations to be performed on the encrypted cyphertext, without the need to decrypt the information first. The generated results match the result of the operations, as if they had been performed on the plaintext.

Formally speaking, if $E(x)$ denotes the encryption of the data x, this means (w.l.o.g.) e.g. for multiplications of data x, y,

$$E(x) * E(y) = E(x * y)$$

Ultimately, one does not even need access the plaintext to calculate e.g. the sum. This feature can be utilized to first encrypt personal data, like the energy consumption mentioned in the above example, using a homomorphic encryption scheme. Only the encrypted data is then transmitted to the receiving end, like the VICINITY node running the micro-service for anonymous data-aggregation. The micro-service can then sum up multiple encrypted data, even though it cannot read the individual plaintexts. After the calculation, the individual parties need to decrypt their part of the encrypted sum and now only the overall sum will be available as plaintext to the VAS.

UNIKL is currently looking into ways to further exploit the P2P nature of VICINITY to further optimize this process and enhance its performance.

Applications in VICINITY pilots and intended prove-of-concept

Discussions in the VICINITY consortium have shown that the micro-service described above is useful for some of the VICINITY pilots as of now. The team at ENERC has already given some scenarios where this very technique can be applied. Further testing and a first prototype are however still required.

To this end, UNIKL will first implement a use case for aggregation of energy consumption in a virtual, simulated/emulated environment with Hardware-in-the-loop. This will enable us to scale up the scenario by adding more and more virtual devices to test scalability. We will conduct performance analysis to evaluate if and to what extent, the proposed methodology can be applied to the Portuguese pilot site or any other potential, future application domain.

One use-case is depicted in Figure ANNEX II-1: Multiple household appliances are equipped with smart energy meters. Their data is collected for internal evaluation (e.g. how many hours and energy, one has wasted watching TV). However, for external use, only the overall energy consumption is required, yet still making sure that each device is indeed reporting its consumption. To this end, data can be encrypted using a homomorphic encryption scheme and sent to the aggregation micro-service. While the receiving side will still be able to validate, that all appliances did send their data, none of the individual readings is exposed, as only the ciphertext is transmitted. The aggregation micro-service can calculate the sum, by adding up all ciphertexts. The resulting sum can then be decrypted, of course giving back the same result, as if it were calculated on the individual plaintexts.

Integration into VICINITY components

Encryption/Decryption, in general, and homomorphic encryption schemes, in particular, are costly in terms of computational effort. However, the same applies to the encryption inside the P2P network already in place. This needs to be implemented on some gateway device on the edge, also running other VICINITY components like the VICINITY Gateway API or the VICINITY agent. The homomorphic encryption can be available as a kind of micro-service that is taking place on the VICINITY nodes and before data is sent out to the P2P network at all.

Figure ANNEX II-1 illustrates the potential Integration of such a microservice into the current VICINITY architecture: VICINITY Nodes interested in using Homomorphic encryption can add the respective microservice in addition to the other components such as the Agent and the Gateway-API. Instead of directly transmitting new e.g. sensor readings (marked in red), data is first encrypted with the homomorphic encryption scheme. The encrypted value is then sent, similar to regular payload, through the rest of the chain (agent, gtw-api) and through the VICINITY network to the receiving end. The same holds for all other measurements, which will encrypt their data the same way.

On the receiving end, instead of forwarding individual readings to the Value-added Service directly, the encrypted data is again sent to the homomorphic encryption microservice, which will aggregate the encrypted values (e.g. calculate the sum over all received data) and will only then decrypt and forward the aggregated data to the VAS. In both (encrypted and normal) cases, the VAS will end up with an aggregation of data, which it requires to operate. In contrast to the normal procedure, where the aggregation is calculated by the VAS itself, with homomorphic encryption the VAS will never know any individual, private data. This further enhances the “privacy by design” philosophy, which is fundamental to VICINITY from the very beginning.

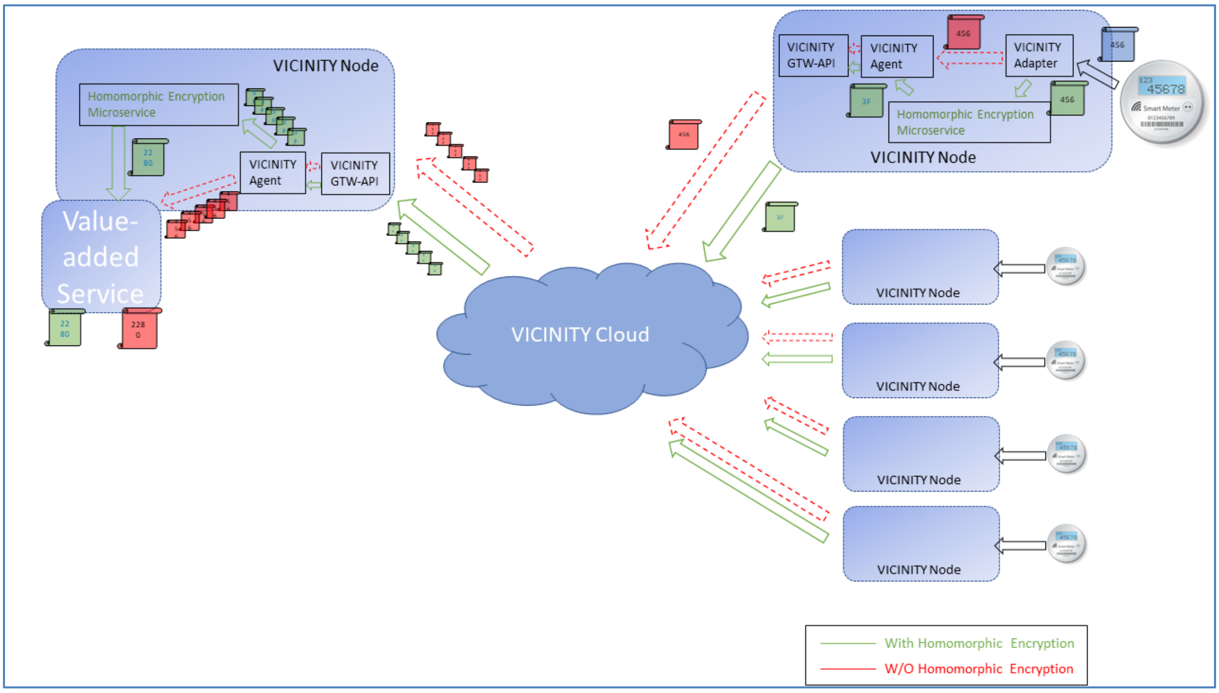


Figure ANNEX II-1: Integrating Homomorphic Encryption into the VICINITY Architecture