Ref. Ares(2018)6294307 - 07/12/2018

MOBISTYLE
MOtivating end-users Behavioral change by combined ICT based modular Information on energy use, indoor environment, health and lifeSTYLE

Contract No.: 723032

Report: Applicable hardware and software solutions for sensing technologies

Work Package: Work package 4, Task 4.1
Deliverable: D 4.1

Status: Public

Prepared for:
European Commission
EASME
Project Advisor: Mr Pau Rey-García

Prepared by:
DEMO Consultants BV
November 30, 2018

This project has received funding from the European Union's H2020 framework programme for research and innovation under grant agreement no 723032. The sole responsibility for the content lies with the authors. It does not necessarily reflect the opinion of the European Communities. The European Commission is not responsible to any use that may be made of the information contained therein.

H2020-EE07-2016-IA

Contents

Executive summary

Deliverable 4.1 describes the hardware and software solutions for sensing technologies used within MOBISTYLE.
Information provided in this deliverable is the follow up of the previous deliverable D4.2 (M12) where the overall MOBISTYLE Platform and workflow were described, presenting the different possibilities for the development of the MOBISTYLE Platform.

Deliverable 4.1 describes the decision taken for the development of the MOBISTYLE platform in reference to the connection between the demonstration cases and the MOBISTYLE database for data collection, to the setup of the database itself and to how sensors data are then shared with the end user applications.

In each demonstration case (Italy, Slovenia, Poland, Denmark, and Netherlands), it has been necessary to add new sensing network to the existing one to achieve the purposes of MOBISTYLE. More information on the specifications of each sensor chosen in the demonstration cases and of the existing ones is available in D2.5.

Sensor data are locally stored and a pull system is defined on the side of the demonstration cases to guarantee data transfer to the MOBISTYLE platform. The type of connection varies from case to case. All data are collected, pre-processed, harmonized and stored in the cloud based MongoDB database.

Three different kind of information are made available for the development of the end user applications:

a) metadata regarding the location of each sensor in the room / apartment / demonstration case, the sensor type and what they measure;
b) sensor data available both as last measurement available and measurement within a range of time;
c) KPI values as defined specifically for each demonstration case.

The data flow from the servers of the demonstration cases to the user applications developed by the other ICT partners is currently operational and described in the current deliverable D4.1. However, the final platform and all the final interfaces to satisfy the interoperability requirements and the integration between different software tools and mobile applications will be provided in M42 in deliverable D4.3. An intermediate D4.3 will be available in M30.

1. Introduction

The building sector is one of the greatest energy consumers, accounting around 40% of the total energy use [1]. For this reason, the design of energy efficient buildings and improvements in the energy performance of the existing building stock is of vital importance [2]. However, working on the building itself and its components and installations are not enough to reduce the energy consumption. Users have a great impact on the behaviour of the building and of energy consumption.

In this context, the goal of MOBISTYLE is to raise consumer awareness and motivate behavioural change by providing attractive personalized combined knowledge services on energy use, indoor environment, health, and lifestyle, by Information and Communication Technologies (ICT) tools. Providing more understandable information on energy, health and lifestyle will motivate end-users to change their behaviour towards optimized energy use and provide confidence in choosing the right thing. It will offer consumers more and lasting incentives that only information on energy use.

In WP4 Practical ICT tools are developed, including:

- robust and cost-effective sensing technologies that can be deployed with a minimal setup in small and large-scale installation spaces;
- an integration platform with modular configuration for data and software interoperability, inter-connecting sensor networks that aim to improve the range and type of energy-efficient behaviours;
- a set of software applications for mobile devices and wearable, to enable energy-efficient behaviours of the end-users.

The main aspects of ICT tool development that are covered by MOBISTYLE are:

- embedding strategies of emotional, corporeal seduction for stimulating of end-users, without significantly increasing cognitive load in the tools;
- including mobile technologies, grounded in conceptual narrative frameworks, to encourage end-users to feel an emerging need to delve further and become curious about energy, health and the resulting lifestyle;
- integrating narrative techniques from choreography, music, and serious gaming.

The first deliverable of WP4, D4.2, aimed to describe the overall MOBISTYLE architecture for providing feedback and raising awareness to the end user, starting from the data collection – coming from sensors – about energy use, indoor environmental quality and health.

The second deliverable of WP4 is the current D4.1 regarding the applicable hardware and software solutions for sensing technologies used in MOBISTYLE. This deliverable presents also the decision taken for the development of the MOBISTYLE platform in reference to the connection between the demonstration cases and the MOBISTYLE database for data collection, to the setup of the database itself and to how sensors data are then shared with the end user applications.

Chapter 2 wants to be a bridge between the previous deliverable 4.2 – in which we presented different possible directions for the development of the MOBISTYLE platform – and the current one 4.1 in which all the decision taken for the development of the MOBISTYLE platform are described.

Chapter 3 provides a description of the existing and new sensing technologies implemented in demonstration case, including details related to data structure, data transfer to MOBISTYLE, and challenges and limitations encountered. Moreover, information about the costs for setting up, managing and maintaining the infrastructure is given. More information on the existing and new sensors is available in D2.5 and D6.1.

In chapter 4, a detailed description of the MOBISTYLE central database and its connection to the demonstration cases and the user application is given. Measures on how to increase the performance of the database are given.

Chapter 5 describes which are the further developments that will be done, indicating the measures that will be taken to ensure high performance and data management during the upscale phase of the Polish case. Moreover a brief presentation of the ICT solutions (Dashboard and Game) that will be tested in the demonstration cases in the upcoming months is given. These ICT solutions will be further developed and improved thanks to the feedback provided by the focus groups. A more detailed description of these tools will be available in M30 with a first version of D4.3.

Chapter 6 describes conclusions for this part of the research.

2. The MOBISTYLE Platform in M12 and M24

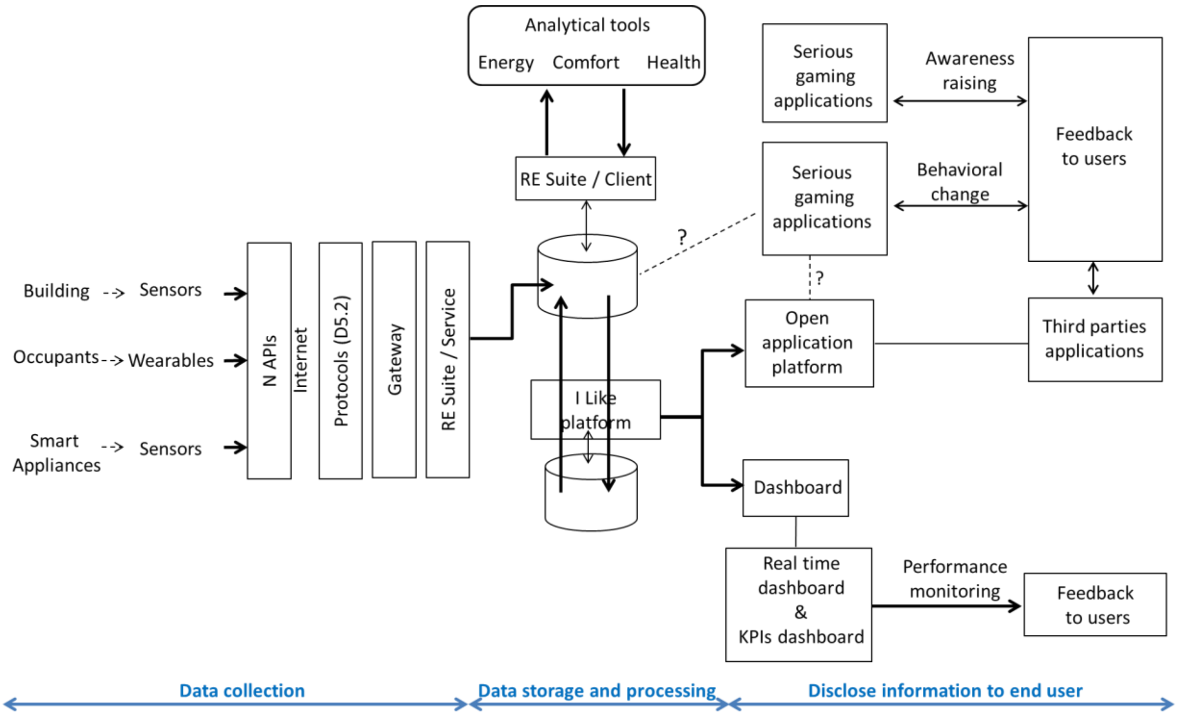


Fig. 1 - General scheme for MOBISTYLE architecture in M12 – D4.2[1]

Since the main aim of MOBISTYLE is to raise consumer awareness regarding energy use, indoor environment, health and lifestyle, information about energy use and behaviour need to be first collected and evaluated and then made available to the end user in an easy and understandable way.

D4.2 (M12) aimed at clarifying and supporting the MOBISTYLE platform architecture idea. The outcomes of this report were that the MOBISTYLE platform is the combination of two existing software application platforms developed by two partners of the consortium: RE Suite by DEMO and i-Like by Holonix. Raw data coming from the demonstration cases are collected, pre-processed and stored in the databases of RE Suite, while the user profile data are stored in the i-Like databases.

The functional, technical and interoperability requirements defined in D4.2 were satisfied during the implementation phase, being able to connect to tools for measuring energy and IEQ and for disclosing information to the end users.

[1] The scheme represents the general architecture in M12. Between M12 and M24, the research brought to a modification of the architecture and features: D5.2, that in M12 was supposed to be related to Protocols, is GDPR centred now; the Dashboard also includes motivational and awareness messages; health related parameters from warbles are stored in a separate database; open platform will probably be a new module with aggregated data. Fig. 6 shows the final architecture in M24.

Following the same structure of D4.2, the decision taken for the implementation of the MOBISTYLE platform will be presented in this chapter according to the different phases of the dataflow. More technical specifications will be given in chapter 4 of the present report.

Three main phases compose the overall dataflow:

- Phase 1: data collection
- Phase 2: data storage and processing
- Phase 3: disclosing information to the end user

Phase 1: data collection

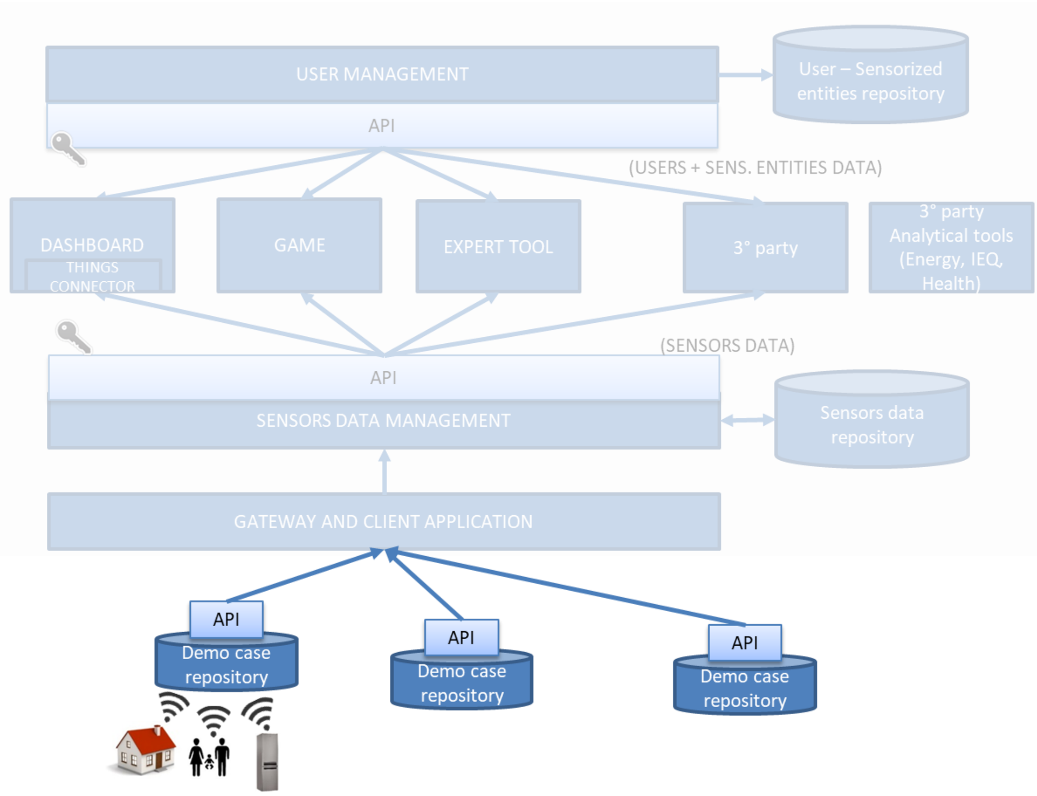


Fig. 2 – Data collection

In the MOBISTYLE demonstration cases, energy sensors and IEQ sensors are installed in the building. Wearable are tested in the Dutch demonstration case. The table below shows, for each demonstration case all the data type collected. More information about the sensors used to collect these data is available in D2.5. The Dutch case is not considered here since it is not connected to the MOBISTYLE platform.

Demonstration case	Data type
Poland	Total Electricity consumption (Apartment level)
	Electricity consumption (Smart plugs and appliances)
	Temperature
	Relative Humidity
	Window / Door opening
WHP appliances	Spin speed
	Water hardness level set

	Cycle duration
	Cycle type
	Water temperature
	Water consumption
	Energy consumption
Slovenia	Temperature
	Set point temperature
	Relative humidity
	CO2 concentration
	Window / Door opening
	Cooling ceiling convector ventilator switch status
	Room access / Occupancy
	Light Switching
	VOC (Volatile Organic Compounds)
	Solar shading
	Thermostat Adjustment
	Valve position radiator heating
Weather station	Wind speed
	Illumination S/E/W
	Precipitation 0'=rain, 10'=dry
	External temperature
	External air humidity
Italy	Total Electricity consumption (Apartment level)
	Electricity consumption (Smart plugs and appliances)
	Temperature
	Relative humidity
	CO2 concentration
	Window / Door opening
	Thermostat adjustment
	Occupancy (Actuator of energy)
	Status fan coil (on / off)
Denmark	Temperature
	CO2 concentration
	Relative Humidity
	Window / Door opening
	Domestic Cold Water consumption
	Domestic Hot Water consumption
	Heat
	Occupancy

Table. 1 – Data type list in the different demonstration cases

All the sensors installed in the demonstration cases communicate directly with the local databases and systems installed in the buildings themselves or owned by the supplier's system responsible for the setup and management of the sensing network. Therefore, how the sensors communicate with the database is responsibility of the different demonstration cases themselves.

On the other hand, it is very important to establish how the supplier's systems and databases communicate with the MOBISTYLE platform. All demonstration cases have developed a pull system which allows the MOBISTYLE database to retrieve data at a certain pre-established frequency: the MOBISTYLE platform interrogates the databases of the demonstration cases every 15 minutes. However, in the Danish case records are collected every 30 minutes due to performance limitations of the demonstration case side. The frequency in which the records are taken and stored in the demonstration cases' databases vary from case to case:

- Polish case: records are taken at a frequency of few seconds;
- Slovenian case: records are taken at a frequency that varies from 5 to 30 seconds, depending on the type of sensor;
- Italian case: records are taken at a frequency of one minute;
- Danish case: records are taken every 15 minutes.

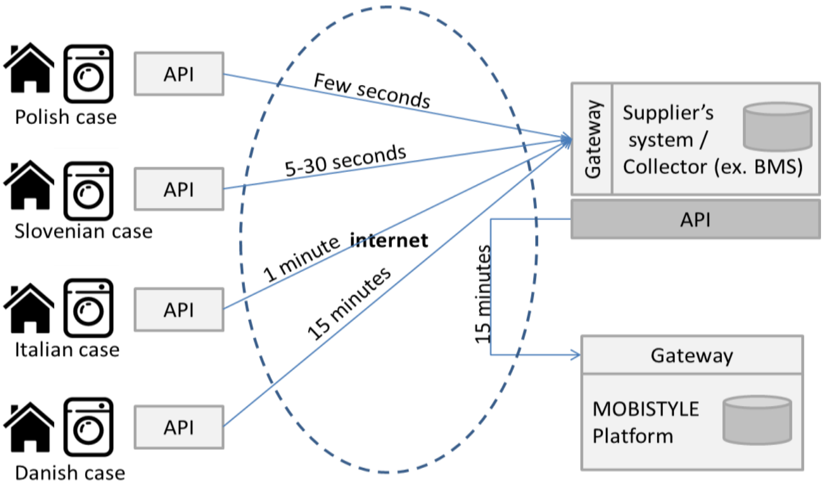


Fig. 3 – Data collection from sensors to MOBISTYLE Platform

Phase 1: Gateway

In D4.2 it has been described that 2 options were available for the location of the gateway: (1) gateway and its functionalities embedded into the platform; (2) gateway and its functionalities not integrated into the platform. The choice during the implementation fell into option 1.

As shown in Figure 4, a gateway is implemented into the MOBISTYLE platform which is the intermediary between the supplier of data (the external gateway) and the database for storage of data. The gateway subsequent retrieves data from external gateways, validates this data, processes this data and converts it to a standard format, and writes the data to the MOBISTYLE platform database. The choice to integrate the gateway into the platform was derived from the final decision of having the MOBISTYLE database developed by DMO as central data collector from the demonstration cases, before sharing the sensors' data with the other ICT tools developed within the consortium to disclose the information with the end user. In this way, by integrating the gateway into the platform, it is guaranteed full control over data flow, high performance and speed of the system because of less "steps" in between for the data flow from the demonstration cases and the user applications.

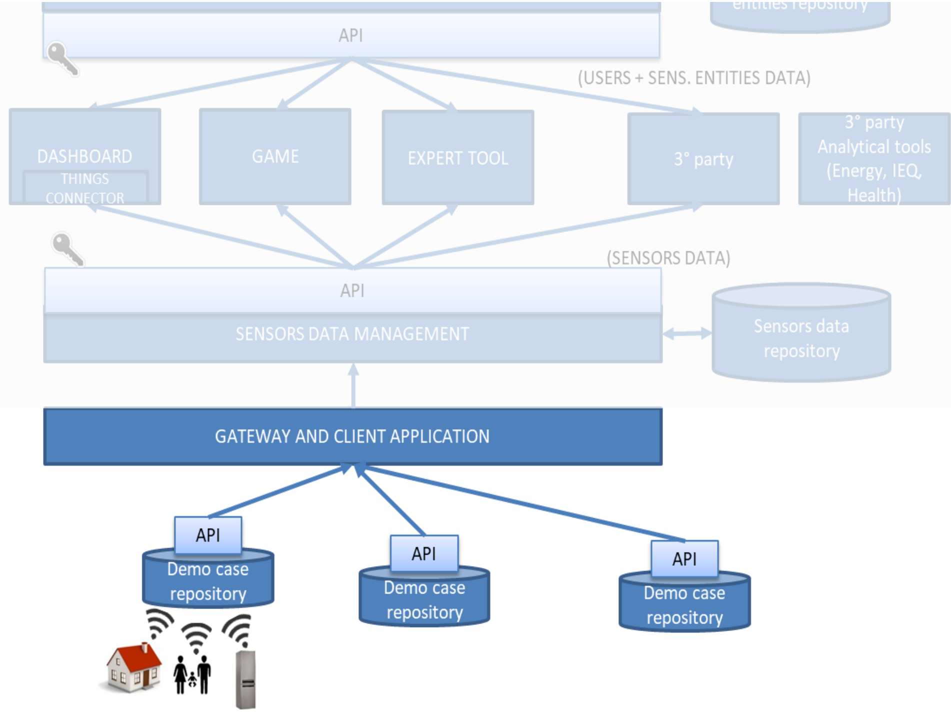


Fig. 4 – Gateway and Client Application

Phase 2: data storage and processing

The gateway and client application directly send the information to the sensors data management responsible for pre-processing the data to be stored in the sensors data repository.

In the same way, the sensor data management process the requests received from the ICT tools and will post-process all the sensors data and it will calculate the KPIs requested. More technical specifications are available in chapter 4.

For sensors and users data, there is a choice between SQL and NoSQL databases when it comes to data storage. SQL databases are used when data is small, consistent, and reliable and has a priori defined format. NoSQL on the other hand is mostly used in an opposite situations where amounts of data can be huge, data is inconsistent and format can differ between the records.

For sensors data, NoSQL database was chosen because there is a potential for huge amounts of data from demo cases, data format differs from demo case to demo case and sometimes even from sensor to sensor within the same demo case (e.g. Slovenian, Polish and Danish case). A well-known, open-source and document-oriented NoSQL database is MongoDB. This is implemented by the consortium partner DMO.

For the user data instead, an SQL repository is used by the i-Like platform developed by Holonix. Personal information is stored with modern cryptography algorithms. No one other than the i-Like platform can access directly to the database, a RESTFul API is the only way for third parties applications to retrieve the data. User data and profile are linked to sensors data by using the concept of the "sensorized entity". This is, as also explained in D4.2, a virtual structure that aggregate the sensors data to define a certain ecosystem to which they belong and to connect with the user data.

In the same way as for sensors data, user management system is responsible for pre/post-processing of the user information to disclose them to the ICT tools.

More technical information about the two repositories is shared in chapter 4 of the present deliverable.

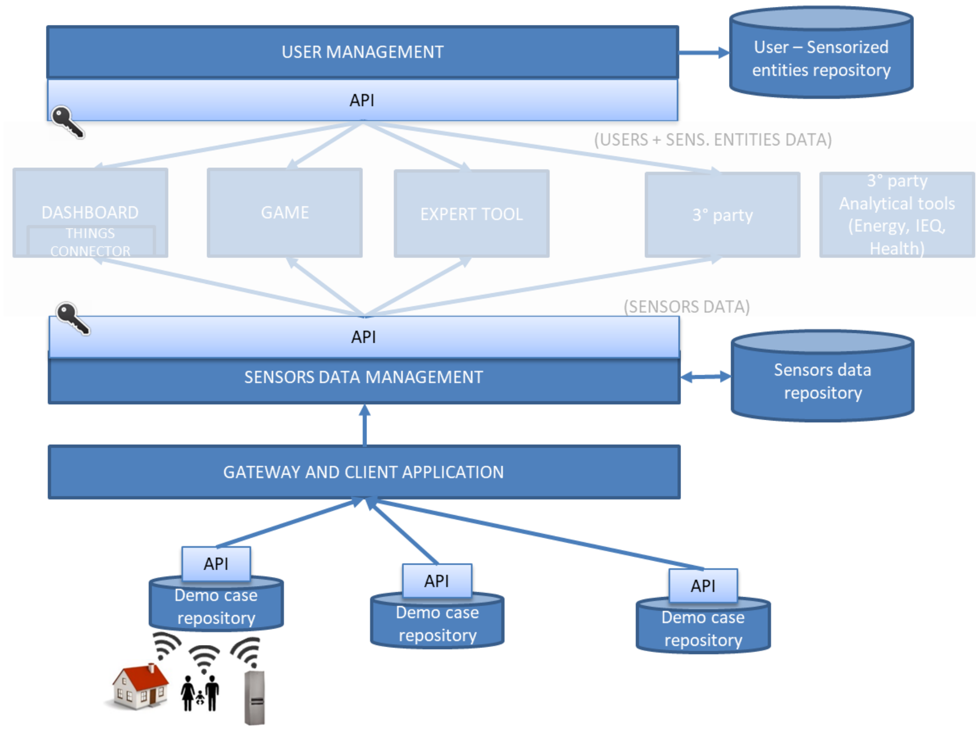


Fig. 5 – Sensors data and user management – Repositories

Phase 3: disclosing information to end user – Game, Dashboard, Expert tool

The disclosing of information to the end-user is done in three[2] distinct ways:

- ***Dashboard***
  The Dashboard developed by the consortium partner Holonix is designed for the non-residential demonstration cases (Slovenia, Italy). It visualizes the most important IEQ and Energy KPIs as well as trends of the most important IEQ and Energy data, chosen with the demonstration case holders and with the support of the energy and anthropological experts (IRI-UL, AAU, and POLITO). Moreover, the dashboard provide alerts and suggestions to the users based on specific thresholds.

- ***Game***
  The MOBISTYLE Game consists of a gamified app developed by HighSkillz. The Game aims at the behavioural change of its users regarding their energy use and also creates awareness on the associated health benefits. The acquisition of the desired MOBISTYLE behaviours is achieved through a series of gamified challenges, which were designed following the project's people centric approach. The Game is designed for the residential demonstration cases

[2] Detailed information on each approach is provided in chapter 5 of this report.

(Poland and Denmark) where such an approach would be most beneficial in collaboration with the demonstration case holders and the support of the energy experts (AAU, POLITO).

- ***Expert tool***

  The expert tool is a new support tool to be used by the experts in the evaluation phase defined in the methodology in WP3. The need of this tool was clarified thanks to the multiple web meetings between DMO, ICT partner responsible for the implementation of the MOBISTYLE database, and the energy and anthropological experts within the consortium (IRI-UL, AAU, and POLITO). As big amount of data are available and collected from the demonstration cases, in different structure and format, it became clear that a tool was necessary to make these data available for the expert in order to allow validation, check, filtering and export the data needed for the analysis. The expert tool is designed and developed by the consortium partner DEMO.

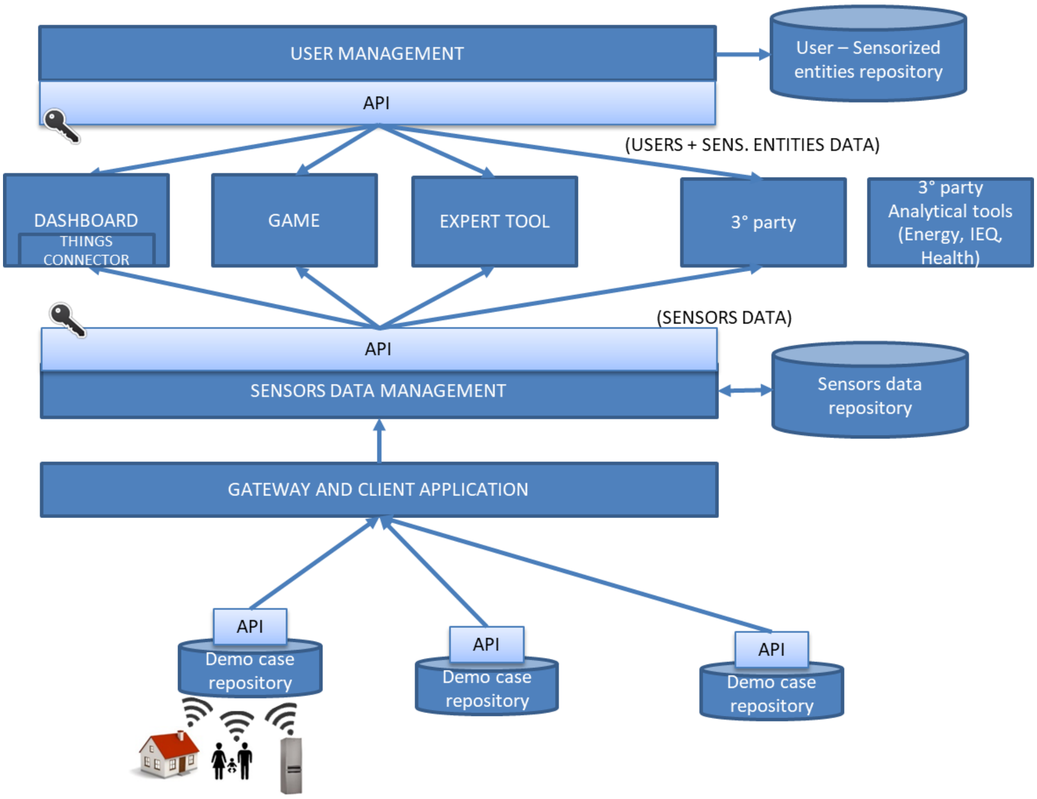


Fig. 6 – MOBISTYLE ICT tools and overall MOBISTYLE Platform

3. Hardware and software for sensing technologies

This chapter describes the technical characteristics of the existing and new sensing network installed in each demonstration case. It explains also limitations and challenges encountered during the setup, which have consequences then on the data transfer from the demonstration cases to the MOBISTYLE platform. Consequently, data structure and transfer is described, including an indication of the costs to setup the network, to manage and to maintain it.

Each demonstration case has an existing sensing network, mainly related in collecting energy data. BMS systems, SCADA systems or Smart Meters are already present in the buildings. But, in order to achieve the purposes of MOBISTYLE and according to the input coming from WP2 and WP3, additional sensors, with the aim of monitoring IEQ and occupancy, are installed. More information of the type of sensors is available in D2.5.

The first challenge encountered was to integrate the new sensors in the existing network. In some cases this was possible, in other not. The second challenge was related to making the sensors data available and accessible from the MOBISTYLE platform, meeting also the performance requirements defined by the users' applications, WP2 and WP3. The most important requirement, and biggest challenge, is to guarantee the data collection and transfer at a certain frequency. Some limitations derived from the structure of the sensing network in the demonstration cases which, in some cases, do not allow data collection so often. So the challenge was to understand how much 'delay' we can have in showing the data in the ICT tools in order to still give valuable information to the users. In agreement with WP2 and WP3, in this first stage it has been decided that data are collected from the MOBISTYLE platform and transferred to the ICT tools every 15 minutes. Therefore, in the ICT tools, the user will visualize data at the latest of 15 minutes before. After the testing phase, it will be evaluated the feasibility of increasing the frequency of data transfer from the demonstration cases to the MOBISTYLE platform, without compromising the performance of the different networks.

In the Polish case, a set of AMIPlus meters – for energy data – and Lerta Smart Home set – for IEQ data are used in combination. The challenge here is related to the fact that the collector is far from the sensors and the wireless connection is not strong, the data transfer from the AMI meters to MOBISTYLE is quite slow. Moreover, Whirlpool appliances are installed in some reference apartments. However, it was not possible to connect them neither to the sensing network already available, neither it was possible to have access to the WHP systems. For this reason DMO has implemented a SharePoint CMIS where data coming from the WHP appliances are collected. This is treated as additional data storage of the Polish case.

In the Slovenian case, the challenge was to make accessible data coming from the SCADA system already present in the building. For this reason, it was necessary to implement a new SFTP connected to SCADA and that the MOBISTYLE platform interrogates to fetch energy data. Moreover, IEQ data are available from the INAP systems (owner of the sensors). Here the challenge is related to the fact that sometime the connection between the sensors and the local storages is lost and there are authentication issues, thus data are sometimes lost. Thanks to weekly monitoring and validation of the data, the consortium is capable of minimizing the risk of loss of big amount of data.

In the Italian case an SQL database is accessible from MOBISTYLE and contains both IEQ and energy data. New and existing sensing systems are combined here. The challenge encountered here is the same as the Slovenian case, with loss of connection, authentication issues. Moreover, sometimes data are missing due to the change of position or removal of some smart plugs. However, thanks to weekly monitoring and validation of the data, the consortium is capable of minimizing the risk of loss of big amount of data.

In the Danish case the biggest challenge was to guarantee high frequency of data collection form the sensors without the system to crash so that the records are not stored in the SFTP. During setup the aim was to have records every 5 minutes. However the encountered limitations forced to have records every 15 minutes. Then, data are available on the SFTP to be fetched by MOBISTYLE every 30 minutes.

Additional information for each demonstration case is available in D6.1.

3.1. Polish case

3.1.1. Technical characteristics of the existing and new infrastructure

Technical description of the existing sensing network

The existing infrastructure is based on AMI (AMI – Advanced Metering Infrastructure) energy meters, which were made available free of charge by the TAURON Group within AMIPlus Smart City Wroclaw program. According to local standards, home electrical systems are part of TN-S or TN-C-S system. Each house and apartment qualified for the pilot program is equipped with a smart AMI-plus meter used to measure electricity consumption. The meter allows reading the actual value of the electricity consumption in a real time. End users have an access to the data through an application offered by the TAURON Group.

AMI meters communicate with the TAURON system with PLC messages, using a low demand power grid. In addition, AMI meters enable radio communication with other media meters and the Home Area Network (HAN), using wireless M-BUS technology. Wireless M-BUS communication in AMI meters is carried out by an integrated communication module, using the MEP (Multipurpose Extension Port) or by communication module mounted on the meter's perimeter board. M-bus wireless module is working with OMS and IEC protocols.

New systems/infrastructure implemented

Using combined set of AMIPlus meters and Lerta Smart Home set may let for better understanding consumers behaviour towards energy use. Each Lerta Smart Home set contains:

- humidity/temperature sensor,
- window/door opening sensors (4 pcs.),
- smart plugs (4 pcs.)
- gateway.

All sensors are wireless and communicate via gateway.

Lerta Smart Home set is able to measure IEQ (Indoor Environmental Quality) by humidity/temperature sensor and window/door opening sensors, as well as energy consumption using smart plugs connected to particular appliances.

Lerta System, which is collecting data from particular sensors is working in Kubernetes environment and is made available to MOBISTYLE users as a service in a cloud managed by Husar Labs. The Lerta platform uses the Google Cloud Platform infrastructure in Belgium, so the network architecture is managed by Google.

3.1.2. Limitations of the existing and new infrastructure

Access to system for users is carried out via the web application. System users must use a web browser that supports JavaScript in the ECMAScript 2015 and CSS3 standards.

Lerta Custom Platform (customer application) allows access to data, which are sent via HTTPS protocol. Requests are authenticated by the API based on a JWT token or login credentials. Additionally, in order to implement real-time data exchange (e.g. push notifications), applications use the Websockets connection, which is used for communication via the MQTT protocol.

3.1.3. Challenges during the set-up of the infrastructure

All sensors can be installed in the home without special knowledge and specialist help, so users have an influence on placing devices, which may result in change of sensor location and placing sensors in places not intended in MOBISTYLE requirements. Moreover, in some cases communication difficulties between smart meter and gateway has been noticed because of the distance between meter and gateway, what caused lack of data.

3.1.4. Data transfer to the MOBISTYLE database

In Polish demo case data are transferred in the following way:

a) Smart meter – Lerta Home Gateway
The telecommunication device Lerta Home Gateway uses digital wireless communication WirelessMBus to obtain data from the user's electricity meter. The meter is the communication initiator; it sends encrypted messages to surrounding potential receivers. The interval of sent data depends on the meter configuration.

b) Lerta Home Gateway – Lerta IoT
The telecommunication device Lerta Home Gateway uses LAN or WiFi network (in user's home) for data transmission. The communication protocol is MQTT version 3.1.1. Communication is encrypted using TLS keys.

c) Lerta Iot – Lerta MDM – Lerta Customer Platform
Data between individual modules are sent via HTTP or AMQP protocol. Requests are authenticated by API modules based on an internal authentication key system, e.g. for Kubernetes tools.

3.1.5. Data collection and storage

Type of information	Question	Answer
Data characteristics	Data format	JSON and CSV
	Type of data	Real time data collection. Historic data are transferred to the MOBISTYLE database every 15 minutes.

	Amount of data	Data showing energy consumption collected from AMI will weight approximately 30 MB/year for one energy meter. In the first testing phase, where we have 23 apartments estimated amount of data will be about 690MB/year. In the second phase, where we're planning to reach 1.000 apartments, estimated amount of 30 GB data will be shared yearly. Data acquisition from AMI assumes two sources: 1) Profiles – 6 values in every 15 minutes, means 576 records daily, 17.280 monthly, 2) Registers – 6 values daily, 180 monthly.
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	All data are stored locally on the server owned by Tauron measurement entity.
Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	Data are requested via API web service accessible via HTTPS requests. Every smart meter has its unique deviceID which is used to access device's details and the data transmitted by it.

Table. 2 – Data characteristics, data storage and interfaces Polish case

3.1.6. Costs for setting up the infrastructure

Urządzenie/MOQ	10	100	1 000	10 000	100 000	1 000 000
Motion Sensor	47,92	41,08	27,38	21,91	19,85	18,43
Window Sensor	36,65	31,40	20,94	16,75	15,18	14,10
Smoke Alarm	56,38	48,32	32,22	25,78	23,35	21,68
Heat Alarm	56,38	48,32	32,22	25,78	23,35	21,68
Flood Alarm	47,92	38,68	27,38	21,91	19,85	18,43
Humidity Sensor	40,87	35,03	23,35	18,68	16,93	15,72
SmartPlug	50,74	43,49	28,99	23,20	21,02	19,52
Gateway	206,51	179,58	125,72	104,17	93,41	82,63
Gateway + GSM + bateria	334,55	289,33	198,89	162,71	144,62	126,53

Table. 3 – Costs setup of sensing network Polish case

Prices above include devices delivered by the Lerta Company, who was the cheapest in our tender request. Prices may vary in final stage when we issue new request.

All the devices are plug&play so they will be installed by clients themselves. Additional cost is the platform – in pilot it is Lerta IoT which cost 500 EUR in total. This amount was calculated for 25 devices, it will rise having 1.000 participants. At this moment it's not possible to define exact cost for set up, maintenance and monitoring of the infrastructure, because it depends on a specific agreement conditions for 1000 households. We can assume that an average price for the services described above will be about 40 EURO per hour (based on average polish cost of similar services).

3.1.7. Contact person responsible for management of the infrastructure

Krzysztof Drożyński, e-mail: krzysztof.drozynski@lerta.energy

3.1.8. Data transfer from Whirlpool appliances to MOBISTYLE database

MOBISTYLE intends to provide feedback to users about energy, IEQ and health. Energy, IEQ and health sensors are installed and used. To get more information on energy consumption, also appliances are used. In most cases these are connected to smart plugs in order to collect the needed information. In the Polish case however, 15 Whirlpool smart appliances, model Whirlpool FSCR 12440, are made available and installed by the consortium partner (WHP) in as many apartment in the demonstration case.

Whirlpool FSCR 12440 is equipped with a smart 6th sense that adapts to the washing machine's load, reducing water, energy and time consumption by as much as 70%. The washing machine can be remotely controlled by the 6th Sense LIVE application. It is possible to plan laundry, get tips for the most efficient use of the appliance, and check the state of the washing machine on your phone or tablet. The washer is equipped with a reliable ZEN direct drive / DirectDrive engine that is electronically controlled with variable speed, making it energy efficient and boasting a long service life. ZEN technology delivers the lowest level of noise at spin between washing machines up to 10 kg and up to 1,400 revolutions per minute.

To achieve the purposes of MOBISTYLE, records will be collected in relation to spin speed, water hardness level set, cycle type and duration, water temperature and consumption, energy consumption.

The table below shows the main information regarding data type, storage and interface.

Type of information	Question	Answer
Data characteristics	Data format	JSON
	Type of data	Historic records collected once a week
	Amount of data	Not yet defined
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	Records from the appliances are stored in the Whirlpool databases. Every week, only the records needed for MOBISTYLE are extrapolated in JSON format and uploaded on SharePoint CMIS

Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	Every week, only the records needed for MOBISTYLE are extrapolated in JSON format and uploaded on SharePoint CMIS. Access to SharePoint through credentials.

Table. 4 – Data characteristics, data storage and interfaces Polish case – WHP appliances

Data transfer to the MOBISTYLE database

As it was no possible for WHP to provide access to the cloud to retrieve data, DMO has created a SharePoint CMIS where WHP is able to upload the collected data every week. This SharePoint is considered as one of the local storages on the side of the demonstration cases and a connection between the SharePoint CMIS and the MOBISTYLE database is guaranteed as in all the other demonstration cases.

The SharePoint CMIS environment is based on Microsoft SharePoint 2013 technology which has been extended with a 'wrapper' layer to provide CMIS-interoperability. CMIS (Content Management Interoperability Services) is an open standard for access and operations on Content Management Systems (CMS), and provides easy interoperability through an API. In addition to this the native SharePoint 2013 interfaces are available.

One key strength of SharePoint-environments is that they are both human, through the web interface, as well as machine readable, through the API's. Because of this fact, WHP can upload the collected data from their appliances easily through the web interface.

Then, the data collection service on the platform's side will (programmatically) connect to the SharePoint CMIS through the API and retrieve the collected data. Therefore, the SharePoint CMIS can be characterized as an external data source and is integrated into the data collection methodology similarly as for example the Danish Case (file-based data collection).

3.2. Slovenian case

3.2.1. Technical characteristics of the existing and new infrastructure

Technical description of the existing sensing network

The existing infrastructure is a GE iFix SCADA with distributed architecture PLCs all behind firewall. External shading position, occupancy from access, set point temperature, convector ventilator switch status, window switch and valve status for heating (radiator) and cooling (celling convector) and lights use for each circuit are monitored by signals from existing SCADA based central controls system. Software provider of SCADA solution is General Electrics (iFix). Room automation is based on Honeywell Lynx PLCs (for local heating/cooling and shading) and lighting on Robotina Cybro PLCs. SCADA integrator is Metronik, who also maintains the system. PLCs communicate with control system via Bacnet or CanBus communication network.

Energy and water meters are installed at the premises, for building level measurements. Networks associated with central control systems are separated from other networks and behind heavily secured firewalls. SCADA controls buildings systems, and monitors several sub systems, that are integrated e.g. active fire protection, access control system, schedules etc. High performance servers with optic communication and several different levels wireless networks are available on site.

New systems/infrastructure implemented

Energy Information System (EIS) is implemented on the servers in the demo buildings collecting data from installed meters and monthly data from bills. All energy related measurements on the building level are saved here and available for analysis.

IEQ (Indoor Environmental Quality) sensors are installed in demo rooms (CO_2, RH, and T). Additional IEQ sensors were installed next to existing thermostats. Sensors provider is INAP.
The image on the left shows an installed IEQ sensor.

Fig. 7 – IEQ sensors

Wireless multi sensors communicate via gateway to a cloud. These sensors use existing hidden wireless networks to connect to the local network. Data is transferred to a gateway installed on servers that is allowed to transfer measurements to a company's cloud.
The solution to gather data from SCADA takes the data from integrated process database and transferred it to a database on a separated computer (virtualized on local servers) behind firewalls. Here files (.csv) of pre-agreed formats are generated periodically. Special service was built to firstly generate and secondly transfer files by one way communication via exemption on firewall to a SFTP server. Here data is extracted by DEMO periodically.

3.2.2. Limitations of the existing and new infrastructure

Safety concerns connected with external parties accessing SCADA. To exclude possible security breached that could result is equipment failures, safety was nr. one concern when planning and executing data flow from the SCADA system. Top IT experts from Faculty of Computer and Information Science, company responsible for firewalls and SCADA provider were consulted. Several difficulties occurred when trying to obtain data that already existed on control system. There were issue with Software versions, compatibility, licenses, integration possibilities, automation of the processes (it is easy to do it manually, but to do it periodically, taking in to account possible limitations, it becomes the challenge).

3.2.3. Challenges during the set-up of the infrastructure

The main problem during the implementation was the complexity of existing infrastructure. There was no person that would know all the structures on site: from SCADA to IT on the building. There were various people involved (head of project from system integrator, head of IT at the premises…). One circumstance that was encountered and made the project implementation more difficult was that the head of IT left to another job during the set up.

It was neither possible nor cost effective to integrate new sensors for IEQ, in to existing SCADA, because modifications would need to be done on PLC, database and interface. Therefore we decided to make separate sensing network utilizing available hidden Wi-Fi network.

3.2.4. Data transfer to the MOBISTYLE database

INAP IEQ sensors data is accessible via API from companies cloud. Data form SCADA is periodically dumped to a SFTP server and there accessible by DEMO.

3.2.5. Data collection and storage

Type of information	Question	Answer
Data characteristics	Data format	JSON and CSV
	Type of data	INAP sensors record with a frequency of 30 seconds Sensors connected to SCADA record with a frequency of 5 seconds. Historic data are transferred to the MOBISTYLE database every 15 minutes.
	Amount of data	8 IEQ sensors x 4 transferred: 32 Data for 12 room x 31: 372 Weather station: 9 Total nr. of parameters: 413 This is done 4 times per hour or 1.189.440 points per month. Not all are used further on.
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	All data are stored locally on SFTP and INAP database.
Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	IEQ records from INAP sensors are requested via API web service accessible via HTTPS requests. Records from SCADA are available by accessing the SFTP server.

Table. 5 – Data characteristics, data storage and interfaces Slovenian case

3.2.6. Costs for setting up the infrastructure

Not possible to get the data for parts allocated to MOBISTYLE only; whole SCADA on the demo building with periphery equipment was ~1M€ (new building). Costs for getting the data from the system were one time. Work load was associated with technical difficulties that are case specific and, consequently, the hourly rate depended on the different tasks.

3.2.7. Contact person responsible for management of the infrastructure

Jure Vetršek, email: Jure.Vetrsek@iri.uni-lj.si

3.3. Italian case

3.3.1. Technical characteristics of the existing and new infrastructure

Technical description of the existing sensing network

The existing system of the case study is equipped with a fully operative BEMS (Building Energy Management Systems) and SMS (Security Management Systems) based on KNX standard as communication protocol. Currently, the system monitors indoor air temperature (and thermostat adjustment), window opening and door openings in each room of the hotel.

The network architecture is based on KNX standard with IP/USB interface. The transmission BUS is connected to a server responsible for logging, storing, and sharing the collected data. The server permits to share data, through a software platform, to every type of computer device, tablet or smartphone. The existing communication protocol of the hotel will be exploited to define the final sensing structure and to acquire data useful for the scope of MOBISTYLE project. The communication protocol is based on the KNX standard (see EN 50090, ISO/IEC 14543 – https://www.knx.org), which is an open standard for building automation in commercial and residential buildings. The case study makes use of the ABB i-bus KNX in Hotel Guest Rooms. Further information on the protocol can be found here:

(https://www.ivoryegg.co.uk/site_files/b5f761d5/7e3b/49a7/998b/8214e814008f_2CDC500118N0201_Presentation_Hotel_Guest_Rooms.pdf?1498544956).

New systems/infrastructure implemented

The new infrastructure is integrated in the existing system and consists of the installation and configuration of compatible smart meters, smart plugs, and IEQ sensors.

The Building Management Systems allows for storing and accessing the monitored data of the new infrastructure in the SQL database (https://www.microsoft.com/en-gb/sql-server).

3.3.2. Limitations of the existing and new infrastructure

The main limitation was the choice of new sensors that ought to be compatible with the existing sensor system and communication protocol.

This problem was overcome by a careful selection of additional KNX sensors that are aligned with the KNX communication protocol, such as KNX smart plugs. Also the Schneider smart meters and IEQ sensors are supported by KNX bus connectivity. No other important limitations to the feasibility of a new infrastructure were observed.

3.3.3. Challenges during the set-up of the infrastructure

The main challenge during the set-up of the infrastructure was related to connectivity problems implying loss of data during the initial monitoring period. Another challenge is the management of the smart plugs. The smart plugs are just attached to the outlets of the electric devices/electric sockets and can therefore be easily removed by the hotel guests/staff. As an example, one smart plug did cause electric power cuts and was removed by the hotel staff. The uncontrolled removal of smart plugs led to unexpected data loss.

3.3.4. Data transfer to the MOBISTYLE database

Data can be visualized and downloaded with the **SQL server management studio** (https://www.microsoft.com/en-gb/sql-server) that permits to view data from both existing and added sensors. It is also used by the hotel technicians for configuring, managing, and administering all components. The monitoring data is extracted from here and transferred to the MOBISTYLE database.

3.3.5. Data collection and storage

Type of information	Question	Answer
Data characteristics	Data format	MS SQL Table
	Type of data	Records taken every minute and transferred to the MOBISTYLE database every 15 minutes.
	Amount of data	Minimum amount of data to be transferred: 5 IEQ sensors x 3 variables (temperature, relative humidity, CO_2 concentration) 4 smart meters x 1 variable (electricity consumption of apartment) 15 smart plugs x 2 variables (electricity consumption and instant power of electric devices) Data is extracted in 15-minutes intervals (4 times per hour); this corresponds to a minimum of 145.824 data points per month, on average 163.422.
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	BMS (Building Management System) collect all the data and store them in an SQL database.
Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	Access to SQL server via username and password.

Table. 6 – Data characteristics, data storage and interfaces Italian case

3.3.6. Costs for setting up the infrastructure

The table below presents the costs of measuring devices installed in February 2018. The table also includes the costs for the installation and configuration of the new sensors.

Description	Code	Cost per unit	Number of devices	Costs
IEQ sensor	MTN6005-0001	265.96 €	5	1329.8 €
Smart meter	ZN1IO-KES+ ZN1AC-CST60	256.2 €	5	1281 €
Smart plug	RF-AZK1ST.01	181.78 €	15	3788.1 €

Men work (installation and configuration of sensors)				1000.0 €
	Total			9398.9 €

Table. 7 – Costs setup sensing network Italian case

3.3.7. Contact person responsible for management of the infrastructure

The company that installed the current monitoring system, "Big Studio S.r.l.", plays an active role in the project during the installation process, component configuration and system upgrade.

The main contact person is Dario Defilippi (dario.defilippi@bigsrl.it).

3.4. Danish case

3.4.1. Technical characteristics of the existing and new infrastructure

Technical description of the existing sensing network

All apartments in residential housing area in Kildeparken have smart Heat and Water meters installed using Wired M-Bus data network and the utility metering company Varmekontrol manages them. The sensor network can be extended using for example M-Bus standard protocol.

In some buildings another utility company are delivering the consumption data to the housing association Himmerland. This in future would require an extended communication and new agreements with another company.

New systems/infrastructure implemented

In order to extend the number of households and apply MOBISTYLE solution, new indoor environmental quality (IEQ) monitoring devices will be needed to install. Mostly AMR-Wireless M-Bus devices from LANSEN systems (temperature/humidity, door and windows, CO2) are installed to extend the existing sensor network. Therefore, new data collectors (Wireless M-Bus collector) are also needed to manage the increase of data flow. Collector for MOBISTYLE project is CMe2100 (gen.3), an M-Bus metering gateway for mobile network. Furthermore, for MOBISTYLE purposes the data transfer from the existing wired M-Bus Heat and Water meters is transformed into wireless data flow to the new data collectors.

To provide a secure file transfer and data storage ista Denmark A/S (Varmekontrol) is using an SFTP network protocol for MOBISTYLE project, and a SQL local server for sorting and analysing the data before uploading it to the MOBISTYLE SFTP.

3.4.2. Limitations of the existing and new infrastructure

One of the limitations of MOBISTYLE infrastructure is the use of data collectors to the limit without losing the data. This is explained in the following.

Existing consumption meters (Heat, Hot Water and Cold Water) are wired M-Bus meters. These meters are used by the housing association Himmerland for the consumption billing, where the existing data collectors are set to deliver one daily reading for every meter. On top of that, these data collectors need to store data every 15 minutes and upload it to the MOBISTYLE SFTP every 30 minutes.

This situation presented a big challenge, a challenge that still exists in some sense, and will probably continue in a small scale throughout the MOBISTYLE project. To explain more, currently wireless signals are used for all the sensors while having only few wireless data collectors with a high reading frequency. Thus the possibility that some other signals using 868 MHz frequency band will collide is quite high.

3.4.3. Challenges during the set-up of the infrastructure

One of the challenges regarding the new sensor network for MOBISTYLE project was that Wireless M-Bus CO2 sensors with battery was not yet available in the market for an affordable price. Therefore, utility metering company ista Denmark A/S (Varmekontrol) needed to wait for a metering device company to develop them for the MOBISTYLE project purposes. This was a rather slow process and therefore some deadlines with regards to CO2 sensor installation were missed.

Furthermore, for MOBISTYLE purposes the data transfer from the existing wired M-Bus consumption meters (Heat, Hot Water and Cold Water) had to be transformed into wireless data flow to the new MOBISTYLE data collectors. In addition, it had to be done with as few wireless collectors as possible to keep the budget under control.

Another limitation at the Kildeparken housing area is the location of the apartments who participate in MOBISTYLE project. Namely, apartments are spread out from each other in a great area, plus they are located in different housing blocks. This presented a challenge to keep the project budget intact, as mentioned before. It required planning to have more wireless collectors and testing the range of these collectors as there is not a wireless collector in every housing block even though there is a MOBISTYLE participant.

3.4.4. Data transfer to the MOBISTYLE database

M-Bus standard protocol is used to read and transfer the metered data. There are two types of M-Bus sensor units used in this project:

1. Wired M-Bus meters (Heat, Water and Hot Water metering)
2. Wireless M-Bus sensors (CO2, temperature/humidity, motion, window/door opening measurements, outdoor air temperature/humidity).

The data from Wired metering devices is collected with Elvaco data collectors (Elvaco's CMc GPRS Box) which is located in each housing block and is communicated via SMS commands. The Elvaco unit is password protected with a 1-4 level security protection. When data from the wired sensor units are collected, it is then uploaded every 30-60 min as encrypted information to a SFTP server at ista Denmark A/S (Varmekontrol). This SFTP server is only accessible by ista Denmark A/S, AAU and DMO.

Data collection and transfer for the Wireless M-Bus monitoring units is almost identical as described about the wired M-Bus network, except that it is wireless communication and all of the units are all encrypted and only decrypted in the Elvaco unit before uploading to SFTP.

In Danish demonstration case, the data exchange is provided via a server and client based architecture. Initially the FTP protocol was used, however now the SFTP file transfer protocol is used to secure data transfer to MOBISTYLE database (managed by DMO).

3.4.5. Data collection and storage

Type of information	Question	Answer
Data characteristics	Data format	CSV files
	Type of data	Records taken every 15 minutes and transferred to the MOBISTYLE database every 30 minutes.
	Amount of data	There are in total 318 sensor units installed in 18 apartments for the MOBISTYLE project purposes. • Each sensor unit is set to upload data every 30 minutes, containing 2 x 15 min value reports. • Therefore, in total it will be 48 uploads pr. day from one monitoring device. • The file size for each unit for the above described specifications is 1,5 KB – 2,0 KB. • 2,0 KB x 318 units x 48 uploads pr. day = 30.528,0 KB / 1.000 to get MB = 30,528 MB pr. day. • Monthly estimation is: 30,528 MB pr. day x 30 days = 916 MB pr. month. 19 apartments are mentioned in other deliverables like D6.1 and D2.5, however during the writing process of this report one apartment cancelled the participation in MOBISTYLE project.
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	B-Bus protocol is used to read and transfer metered data collected with Elvaco data collectors located in each housing block. Every 30-60 minutes data are uploaded on SFTP Server.
Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	Access to SFTP server via username and password.

Table. 8 – Data characteristics, data storage and interfaces Danish case

3.4.6. Costs for setting up the infrastructure

In deliverable *D2.5: Composition of specific sets of data acquisition for the five study and demonstration cases*, the new measuring device network to register temperature, relative humidity, CO2 concentration, window and door openings, occupancy in the 19 apartments is described. Total installation costs of the new IEQ measuring devices and data collectors are presented below.

Device	No of devices	Price per device, kr.	Costs of devices, kr.	Additional costs* (installation), kr.	Total costs, kr.
Temperature / RH	81	580,00	46 980,00	10 530,00	57 510,00

CO2	81	670,00	54 270,00	10 530,00	64 800,00
Window/Door	81	730,00	59 130,00	10 530,00	69 660,00
Presence	19	1 475,00	28 025,00	2 470,00	30 495,00
Data collector	9	1 870,00	16 830,00	1 170,00	18 000,00
Heat meter	3	1 300,00	3 900,00	390,00	4 290,00
Water meter	3	1 100,00	3 300,00	390,00	3 690,00
Outdoor temperature / humidity sensor	1	1 025,00	1 025,00	130,00	1 155,00
		SUM:	**213 460,00**	**36 140,00**	**249 600,00**

*installation costs are 130 kr. pr. one measuring device.

Table. 9 – Costs setup sensing network Danish case

The total costs associated with the installation process of new measuring devices is 36 140, 00 kr., excluding the sensor price. This number includes only the installation process where the expenses are 130, 00 kr. pr. installing one device.

Some of the working hours, that were registered by the company Varmekontrol to set up and test the new platform for MOBISTYLE are shown in the table below.

Development	No of hours
Mobistyle (Demobv)	6
AAU platform	7
Common FTP	2
Testing of the pilot apartment	5
Dashboard	3
Total No of hours:	**17**

Table. 10 – Personnel involved in MOBISTYLE – Danish case

3.4.7. Contact person responsible for management of the infrastructure

In Danish demonstration case all the people involved in the MOBISTYLE project are mentioned in the following table. Utility metering company Varmekontrol is responsible for setting up the MOBISTYLE sensor network and establishing data transfer to the MOBISTYLE database.

Organization	Contact information
Aalborg University	Per Heiselberg, WP6 leader and Coordinator, E-mail: ph@civil.aau.dk Sandijs Vasilevskis, Research Assistant, E-mail: sv@civil.aau.dk
Himmerland Boligforening	Rasmus Hjorth, Project Manager, E-mail: rh@abhim.dk Sven Buch, Development Manager, E-mail: sbu@abhim.dk
Varmekontrol	Daniel Sørensen, Development & System Analyst, E-mail: daniel.soerensen@varmekontrol.dk

Table. 11 – Personnel involved in MOBISTYLE – Danish case

Contact person responsible for sensors and management of the infrastructure is Daniel Sørensen, Development & System Analyst from company Varmekontrol.

3.5. Dutch case

For this demonstration case, the main goal is to observe the relation between human physiological response to different indoor environmental situations and people's perception and acceptance of it. It has been shown that a more varied indoor climate has a positive effect on our metabolic health (human energy metabolism, glucose metabolism and cardiovascular system [3]. The education of users and easy comprehensible information provision on can increase understanding and users acceptance. The objective is to observe people's psychological response and of some physiological parameters over time to drifting indoor temperatures. With sufficient time period there is body acclimatization (physiological response) which is followed by the user's acceptance (psychological response).

It will be further researched how a drifting temperature, especially in winter besides leading to higher acceptance and increased metabolism (related to health) also leads to building energy savings, see for example Toftum et al, 2018 [4].

It will be observed how people perceive the dynamic temperatures and how do they react to it (via comfort, sensation, alertness, and physical activity assessment).

3.5.1. Technical characteristics of the existing and new infrastructure

Technical description of the existing sensing network

At this moment no sensor network exists. There is one thermostat installed for the general heating system and the heat recovery. The fan coil units have each a temperature sensor for local control of the unit. These sensors are not connected to a database and are working without any internet connection. For the purpose of this demo-case study this is sufficient, because the actual indoor environment and people's responses will be collected by the researchers.

New systems/infrastructure implemented

The only device that will be connected to a database will be the Fitbit. Fitbit offers a public web API for retrieving the data from 8 Fitbit devices. Every physical Fitbit device needs to approve sharing of data. When doing so, the user will retrieve a client id and access token for retrieving the data through the web API. A script was written to collect the health data from the Fitbit server (with Auth0) and to push the data in the HIA azure database through a secure API. Here, the data is stored anonymous. UM can access the anonymous data through a read-only user account for this database. Because of the personal data that is collected by the Fitbit, it is chosen not to add the data to the general MOBISTYLE database. The data is not used or needed in the MOBISTYLE dashboard or MOBISTYLE game, because of the specific goal of this demo case (see above 3.5). Only two MOBISTYLE partners, UM and HIA, need to use the data for analysis'. Parties that don't need to access the data will not be able to access it.

3.5.2. Limitations of the existing and new infrastructure

The fitbit is chosen because of its built-in heart rate monitor that is quite accurate compared to other built-in heart rate monitors. A polar watch was also considered, but this one only had an external heart rate monitor which is considered more invasive for the test-subjects. It is harder to start wearing it during work, because you need to undress and also during the night it is not preferred to sleep with a heart rate band. Although an external heart rate monitor is more precise.

3.5.3. Challenges during the set-up of the infrastructure

A limitation of this way of working with the Fitbits is that Fitbit aggregates its data for local storage on the device when not synchronised frequently. So at least once a week all devices separately should be connected to de Fitbit app to synchronize there data. Otherwise minute data is lost and only aggregated daily values remain. So this is something that has to be taken into account when using the fitbit in a field study.

3.5.4. Data transfer to the MOBISTYLE database

The MOBISTYLE game and dashboard won't be used for this demonstration case because of the different objectives of the demonstration project. Therefore, the sensors won't be connected to the MOBISTYLE database. Instead the sensors will collect their data locally on the devices where after the data will be retrieved by plugging the device into a desktop computer and saved on a local hard drive of HIA. No data will be shared over internet (e-mail or cloud services) with other parties. UM will receive the anonymous data on a USB stick encrypted with a password for analysis.

3.5.5. Data collection and storage

Type of information	Question	Answer
Data characteristics	Data format	Direct connection database to database through 2 API's
	Type of data	Heart rate, activity level, black bulb temperature, air temperature, RH, CO2
	Amount of data	Fitbit: 2 variables x 8 persons x 60 minutes x 24 hours x 14 days 322.560 data points IEQ: 4 variables x 12 times per hour x 24 hours x 14 days 16.128 data points
Local storage	Where data are locally stored for the first time before they are sent to the Mobistyle platform	The fitbit stores his data on the device for a maximum of one week, with synchronization it is stored to the fitbit database. The IEQ sensors store their data locally on the device until retrieved.
Interface	How the Mobistyle database can be connected to the local storage. How we can interrogate the BMS, or any other system you use to collect sensors data	Does not apply

Table. 12 – Data characteristics, data storage and interfaces Dutch case

3.5.6. Costs for setting up the infrastructure

Description	Cost per unit (euro)	Number of devices	Number of week	Costs
IEQ measuring tree	1000/week		6	6000 €
Temperature sensors	90	15		1350 €
Accelerometer (Fitbit)	149	10	5	1490 €
Thermosensors (iButton)	49	40	15	1960 €
Questionaire app (Metric Wire)	100/week			400 €
	Total			9850 €

Table. 13 – Costs setup sensing network Dutch case

3.5.7. Contact person responsible for management of the infrastructure

Loes Visser (HIA) l.visser@huygen.net and Rick Kramer (UM) rick.kramer@maastrichtuniversity.nl

4. MOBISTYLE database setup and data gathering

Chapter 2 provided a description of the 'decision taken' for the development of the MOBISTYLE Platform in the period from M12 (D4.2) and M24. In chapter 3, the specification of the existing and new sensing networks and technologies for each demo cases have been presented. More information about the specific sensors and measurements are available in D2.5 and D6.1.

This chapter 4 presents the technical description and specifications of the MOBISTYLE database, taking into consideration the starting point of D4.2 and the characteristics of the sensing network established in each demonstration case.

The development of the MOBISTYLE database started by understanding the data type, data structure, network system and possibilities for data transfer from each demo case. For this reason, periodic web meetings and weekly contact was necessary to proceed in parallel with the implementation of the MOBISTYLE database and the setup of the sensing network in the different demo cases. In this way, the consortium was able to test, monitor, and improve the network and minimize the risks during the implementation phase.

One of the biggest challenges in this phase was that the data type, data structure and amount of the data and their frequency, and the data transfer possibilities were different in each demonstration case. On the other hand, the second big challenge was related to the fact that it was necessary to take into consideration at the same time both the limitations deriving from the demonstration cases and the requirements and needs of the User Interfaces Applications developed by the other ICT partners, Holonix (responsible for the development of the dashboard) and HighSkillz (responsible for the development of the game).

In this phase DEMO, as responsible partner for the implementation of the MOBISTYLE database, played a key role in managing design and communication between the demonstration cases and the other ICT partners. A lot of tests and agreements were necessary in order to achieve a technical solution generic enough to be used by all partners, guaranteeing though the satisfaction of the requirements defined by the game and the dashboard in terms of performance, data structure, and data transfer.

In the next subchapters, a detailed description of the MOBISTYLE database is presented.

4.1. Technical description of the database

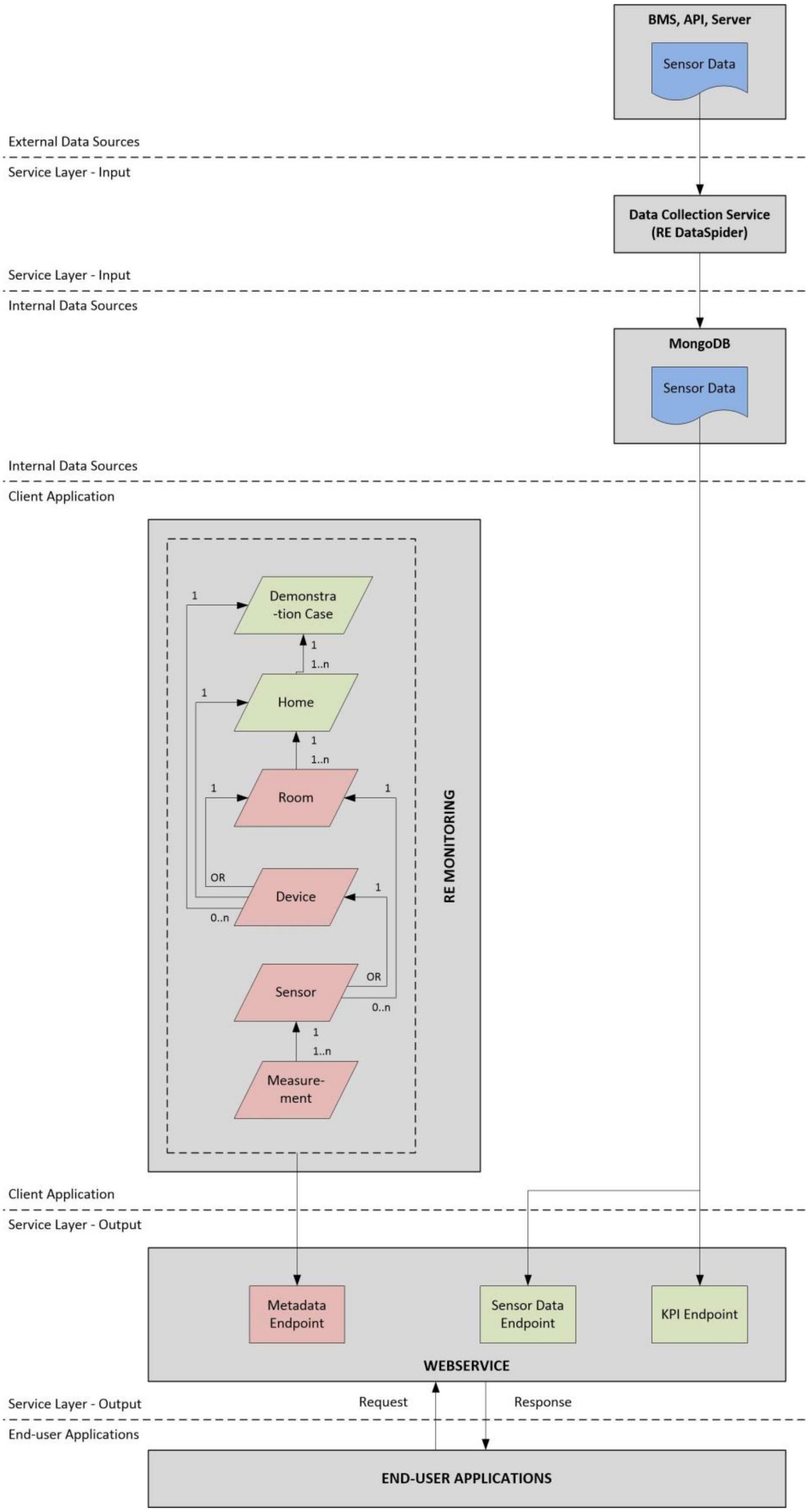


Fig. 8 – Software concept

The image above shows the software concept developed by the consortium partner DMO for data collection, storage and data transfer – starting from the external data source to the end user application.

4.1.1. External data source

As described in Chapter 3, each demonstration case has one or multiple local storages in which sensors data, collected at different frequency, are stored. All the demonstration cases have implemented a pull system as methodology for data transfer to the MOBISTYLE database. All the connections are accessed via IP address, username and password and they are secured (https, SFTP).
The type of connection to guarantee data transfer from the demonstration cases to the MOBISTYLE database is as follow:

Demonstration case	Type of connection
Slovenian case	• SFTP server for data coming from SCADA system • Web service for IEQ data from INAP sensors
Italian case	• Microsoft SQL database
Polish case	• Web service for all energy data from Smart Meters and IEQ sensors • SharePoint CMIS for data collected from the Whirlpool appliances
Danish case	• SFTP server for all sensors data
Dutch case	• No connection to the MOBISYLE database

Table. 14 – Connection demonstration cases to MOBISTYLE

4.1.2. Service layer Input

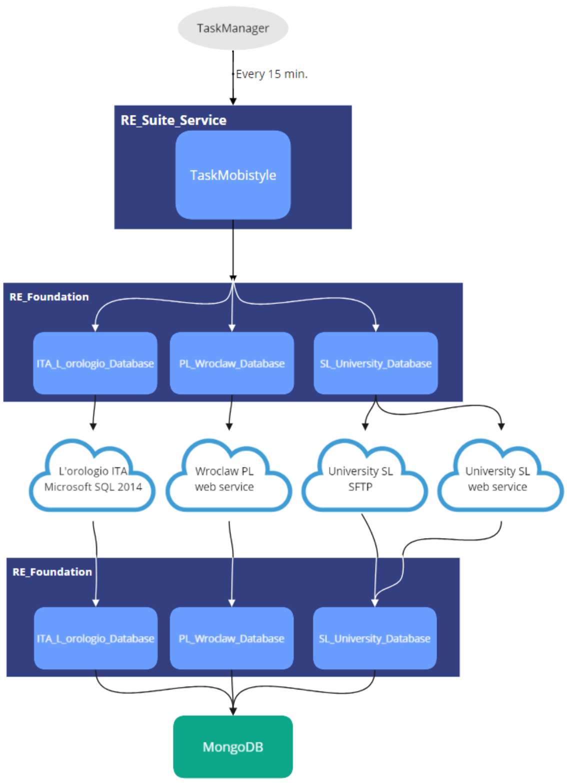


Fig. 9 – Data collection software concept

Data collection is implemented on the side of demo cases as a pull system. So RE Data Spider is executed every 15 minutes in order to pull all the available data from the demo cases.

In the first run when MongoDB is empty it fetches all the historical data which is made available by data providers. To prevent introduction of duplicate records in our database during the next executions of RE Data Spider, which can cause errors in statistics and KPI's calculated on this data, we have added 2 layer protections. The first one is implemented in RE Data Spider. Before fetching the data it looks at date and time of the latest saved data points for each sensor/device from each demo case and pulls the new data beginning from that date and time. The second layer is described in the paragraph below "Performance improvement".

After the data is pulled from the demo cases it is pre-processed. Because the data is gathered from different source types (file servers, web services, SQL databases) and in different formats (CSV files, JSON objects, SQL tables), the data is processed before inserting it in the database in order to generalize the format as much as possible. It includes not only the naming conventions and structure of the records but also amount and type of information per record and ensuring that different types of timestamps of each record are correctly converted to UTC (Universal Time Coordinated) format. But due to diversity of data sources and data collection procedures per demo case there are some specific ways of gathering, storing and restoring data for some cases.

Slovenian case

There are 2 different ways the data is delivered to DMO in 2 different formats and naming versions. It means that 2 different implementations for pulling, pre-processing and storing of the data are required only for this demo case.

The first one is INAP web service which delivers the data in a JSON format. The API makes possible to request the data between given range of time. The format of the pulled data is slightly adjusted; date and time are converted to UTC format, time-stamping Batch ID is added and also SCADA sensor name equivalent of the INAP name is added to the record to increase performance of DMO web service.

The second one is SFTP file server which stores the sensor data in the files of CSV format. The data fetching process is pretty different from web service. A secure connection with the file server is made. Then lookup for all existing files and download them to our local sever. After all the files are downloaded and removed from the SFTP server the connection is closed and the pre-processing of the data begins. The data format differs much more from JSON objects which we receive from web service, so a separate file parser had to be implemented. In contrary to JSON records where we have 1 record for each data point in CSV we have 1 row containing all the data points of all sensors for given date and time. So firstly all the sensor data is extracted to separate records and then date and time are converted from unstandardized format to UTC in ISO 8601. Also as with INAP time-stamping Batch ID is added alongside with INAP sensor name equivalent of the SCADA name is added to all the records to increase performance of our web service.

The Slovenian case requires deleting the files from their SFTP file server after the files are downloaded. In case of corruption of our database all the data can be restored from all demo cases except for these files because they were deleted immediately after data collection. To reduce the risks, data form the

SFTP file server is stored in original format so that, in case of malfunctioning or issues in the MOBISTYLE database, it will still be possible to recollect the data.

Italian case

The only data source here is the Microsoft SQL Server with encrypted connection. There are only 2 tables in the database useful for MOBISTYLE and both of have the same format. Each table row contains data of 1 sensor at some point of time. So each row is converted to standardized format and then the following data is added for each record depending on its sensor address: data type, measurement unit, room, device name, sensor description and time-stamping Batch ID. All this extra information was provided via excel sheet. Date and time of each record is also converted from Central European Standard Time to UTC.

Polish case

This demo case stores all the sensor data every minute or even every second. It results into huge amounts of data which cannot be pulled at ones so it should happen in chunks. Therefore there exists functionality in API which allows requesting data in chunks of maximum 100 records at once. This is used it to get a representative sample every 15 minutes. It means that each time the response for an apartment is a sample of last 100 records of all data which is 200-400 records for 15 minutes interval. The benefit is that a lower the amount of requests is done by RE Data Spider by 2-4 times, so 15 minutes data fetching and historical data fetching become faster by the same factor of 2-4 times. It means also that the memory consumption will not drastically increase over time. And because not too much data are stored in DMO database is 2-4 times lighter and the Web Service will remain quick for delivering the data as it is now.

Polish case API is used to fetch data in JSON format. Although all the data is in the same format it is differently structured for each data type and each type of sensor. So it is possible to get for one data request multiple records which are differently structured and contain different types and different amount of information. To solve this problem, a parser to detect and normalize all the incoming data has been implemented. Besides that, the following information to each record - based on excel sheet provided to us by demo case - was added: apartment number, sensor description, device the sensor is attached to and time-stamping Batch ID.

Danish case

Danish case provides the data only through SFTP file server and like in Slovenian case in CSV format, but in contrary to Slovenian case files are not removed after they are downloaded. It causes a problem of duplicate data. While in the Slovenian case, every time a new data is received with no duplicates, in Danish case always all the data is received. Therefore, a filter has been implemented to make sure that already once downloaded files will not be downloaded again. Another challenge was a different structure and amount of records depending on sensor type, what means that each sensor had to be parsed on a different way. So another parser to normalize the data was implemented for this case. Also the time-stamping Batch ID is added to each record.

Another issue regarding all demo cases which can (and did) arise is the result of unreliability of sensors. The sensors can crash, be unplugged or maintained and during this process no data will be available for us to collect. Sometimes, like in case of crash, it is even unknown to the demo cases that their sensors are not sending the data. It means that all sensors have to be checked repeatedly to make sure that no data is missing. The challenge is the amount of sensors right now and the amount after upscale to 1000 households in Polish case which makes it impossible to perform all this checks manually. So to solve this issue a tool has been implemented. This tool performs automated data check every 3 days to ensure that no data is missing. In case when data is missing it sends automatically an email to the persons responsible with the list of unique identifiers of the sensors which do not send data and the timestamp when they have sent data for the last time.

4.1.3. Internal data storage

As described in chapter 2, a MongoDB database has been used to store sensors data. MongoDB stores documents in a Binary JSON (BSON) format and uses a query syntax resembling JSON. In the enterprise version it supports high volumes of data and can scale horizontally to evenly distribute data, balance load and create redundancy. A single instance of a MongoDB database can easily be deployed on the available Windows servers with minimal configuration. To make sure the database and its connections are secure, we can enable features such as authentication (username and password) and secure links over SSL. All of these features ensure that a solution using MongoDB satisfies the set requirements and allows us to scale the service to more buildings and sensors in the future [5].

MongoDB is a document storage system; these (BSON) documents are stored in collections which can be defined per database. For the platform's database, only one database has been used, containing four collections, one for each demo case: DK_Kilderparken, ITA_L_orologio, PL_Wroclaw and SL_University.

Time based indices for each collection have been created to increase the database performance, which increased speed by ~100 times for fetching raw data.

A second layer of security is created to prevent the duplicate data. This is done with unique index that ensures that any record, which already exists in the database, is rejected from inserting. The index is based not only on date and time of the record but also on other unique values.

4.1.4. Performance improvement

There are different ways to implement data collection and processing, which depends on functional and non-functional requirements. Naturally the best performance-wise implementation is a pre-processing of the data after it is collected and then storage of the processed data next to the raw data. Another possible implementation is processing during the data gathering and storing only processed data. The advantage is that each piece of data is processed only once in the beginning, so the computation complexity and processing time does not affect the data delivery time each time user makes a request for it. The disadvantage is the lack of flexibility for the user because he/she can request data only within the limits of pre-processing. In the MOBISTYLE situation, the most processing is a KPI calculation which is required to be highly flexible for the users. It makes impossible to push KPI calculations to pre-processing. For this reason KPI calculation has been implemented on-the-fly.

According to No Free Lunch Theorems For Optimization [6] if an algorithm does particularly well on average for one class of problems then it must do worse on average over the remaining problems. It means that it has to be identified which problems are acceptable to have on algorithmic level and which definitely must be solved. Problems that are acceptable on algorithmic level are either solvable with hardware or they are just negligible.

The following 3 most important interdependent problems have been identified: speed, storage, data reliability/data precision.

The speed of RE Data Spider and especially the web service is one of the most important and hardest to solve. Data delivery speed is crucial to the data delivery process. If it is not solved the dashboard and the game, which make use of our web service, cannot function properly. Several actions have been taken to increase the speed. The first and simplest part of the solution was adding time indices to the collections in database as described above. Time indices increased querying time so much because data is always requested for some period of time and in chronological order. The second was moving as much as possible data operations from web service to the database. It means not only simple operations like filter and sort but also some complex calculations especially for KPI's. It means that the web service has to generate dynamically MongoDB queries on each user request. It increased the speed of response of our web service up to ~10 times. And the last but not least was moving the software and database to a faster hardware. The decision was to move them to the cloud in Azure.

The storage capacity is another challenge faced during the project. Unfortunately is it not as simple as just getting more hardware to store information, because data increases rapidly over time so, even without considering an upscale, there will be a continuous need of space. Secondly, the more data we have the slower database response will become. But it is still possible to increase performance at the expense of data precision. In this phase of the implementation, a trilemma, which is described in No Free Lunch theorems, is presented: there are 3 favourable options but only 2 options can be chosen. Therefore, it was necessary to prioritize the problems, identifying which are the acceptable ones and to what extent they are acceptable. The speed appeared to be the problem which definitely had to be solved, because slow response would make the information unusable and invaluable when transferred to the end user applications. The problem of the storage was partially solved by moving the database to the cloud. The other part of the solution is not saving all the data. But it had to be done in a way that data still remains reliable on one hand and accurate enough on the other hand. This was done by diving into the data and the requirements of user interfaces. The decision was, for the real time data incoming from Polish case, to sample the incoming data by taking a representative set of records of every 15 minutes. It did not affect the data reliability in any way, data is still accurate enough for all the use cases needed for user interfaces and it drastically improved performance not only for now but also regarding the upscale in the future.

4.1.5. Client Application: RE Monitoring

Sensors data coming from the demonstration cases do not contain in their structure metadata related to their specific location and other information. However, these data are fundamental for the development of the Dashboard and the Game that need to have the association and connection of each record to the specific sensor and its location within the demonstration case.

For this reason, a tree structure has been defined in order to be able to send to the ICT tools the proper information. The tree structure has been developed in different steps, in parallel with the design of the ICT tools.

Step 1 – first metadata collection

During the implementation of the MOBISTYLE database, to each demonstration case holder has been requested to fill in an excel file with additional information related to the location of each specific sensor. In particular it has been requested to specify and categorize:

- Location of the sensors and devices:
    - Apartment > rooms (for residential cases (Poland and Denmark))
    - Rooms (for non-residential cases (Italian and Slovenian))
- Device ID and description;
- Sensor ID and description.

Each demonstration has added additional information when needed. During the implementation phase, all the records were associated to a normalized list of names to facilitate and speed up the process. During the implementation of the database, this association between sensors data and metadata has been done manually. However, for the up scaling phase of the Polish case to 1000 households, the involved partners TAU and DMO are collaborating to try to implement the metadata in the APIs made available by Lerta, company hired by TAU and responsible for the sensing network in the Polish case.

LOCATION	SOURCE ADDRESS	ADDRESS	TABELLA	TYPE OF SENSOR	DESCRIPTION_ADDRESS	DESCRIPTION_ADDRESS_FULL	Normalized list
RECEPTION	1.6.30	9/1/53	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Power kW RECEPTION - FOTOCOPIATRICE	Polito.RF Socket.Active Power kW RECEPTION - FOTOCOPIATRICE	ElecPower
RECEPTION	1.6.31	9/1/54	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Power kW RECEPTION - PC	Polito.RF Socket.Active Power kW RECEPTION - PC	ElecPower
RECEPTION	1.6.30	9/1/73	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Energy kWh RECEPTION - FOTOCOPIATRICE	Polito.RF Socket.Active energy kWh RECEPTION - FOTOCOPIATRICE	ElecCons
RECEPTION	1.6.31	9/1/74	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Energy kWh RECEPTION - PC	Polito.RF Socket.Active energy kWh RECEPTION - PC	ElecCons
RECEPTION	1.1.202	9/2/2	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	CO2 RECEPTION	Polito.CO2 - RH - Temperature.CO2 RECEPTION	CO2
RECEPTION	1.1.202	9/2/12	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	RH RECEPTION	Polito.CO2 - RH - Temperature.RH RECEPTION	Hum
RECEPTION	1.1.202	9/2/22	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	Temperature RECEPTION	Polito.CO2 - RH - Temperature.Temperature RECEPTION	Temp
Room 01	1.1.3	5/0/1	LOG_ALL_TELEGRAMS	Termostato ABB TUX/U 1.1	Temperatura attuale camera 01	Temperature e set point.Temperatura attuale.Temperatura attuale camera 0	Temp
Room 01	1.1.2	2/2/01	LOG_ALL_TELEGRAMS	magnetic switch / BMS	Contatto finestra camera 01	Ingressi.Contatto finestra.Contatto finestra camera 01	WindowSwitch
Room 01	1.1.1	2/0/1	LOG_ALL_TELEGRAMS	magnetic switch / BMS	Micro contatto porta camera 01	Ingressi.Micro contatto porta.Micro contatto porta camera 01	DoorOpen
Room 01	1.1.4	4/2/01	LOG_ALL_TELEGRAMS	Attuatore	Stato FM camera 01	Stati uscite.Stato FM camera.Stato FM camera 01	Occ
Room 01	1.1.3	5/1/01	LOG_ALL_TELEGRAMS	Termostato ABB TUX/U 1.1	Set point camera 01	Temperature e set point.Set point.Set point camera 01	ThermAdj
Room 01	1.1.4	7/2/01	LOG_ALL_TELEGRAMS	Attuatore	Stato v1 fan coil camera 01	v fan coil.Stato v1 fan coil.Stato v1 fan coil camera 01	FanCoil
Room 01	1.1.4	7/3/01	LOG_ALL_TELEGRAMS	Attuatore	Stato v2 fan coil camera 01	v fan coil.Stato v2 fan coil.Stato v2 fan coil camera 01	FanCoil
Room 01	1.1.200	9/0/1	LOG_TELEGRAMS	KES-KNX Energy saver (floor distribution board)	Active energy kWh APT01	Polito.KES energy meter.Active energy kWh APT01	ElecCons
Room 01	1.1.200	9/0/16	LOG_TELEGRAMS	KES-KNX Energy saver (floor distribution board)	Active Power kW APT01	Polito.KES energy meter.Active Power kW APT01	ElecPower
Room 01	1.6.10	9/1/51	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Power kW APT01 - TV	Polito.RF Socket.Active Power kW APT01 - TV	ElecPower
Room 01	1.6.11	9/1/52	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Power kW APT01 - FORNO	Polito.RF Socket.Active Power kW APT01 - FORNO	ElecPower
Room 01	1.6.10	9/1/71	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Energy kWh APT01 - TV	Polito.RF Socket.Active energy kWh APT01 - TV	ElecCons
Room 01	1.6.11	9/1/72	LOG_TELEGRAMS	KNX RF+ Socket with active power measurement	Active Energy kWh APT01 - FORNO	Polito.RF Socket.Active energy kWh APT01 - FORNO	ElecCons
Room 01	1.1.201	9/2/1	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	CO2 APT01	Polito.CO2 - RH - Temperature.CO2 APT01	CO2
Room 01	1.1.201	9/2/11	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	RH APT01	Polito.CO2 - RH - Temperature.RH APT01	Hum
Room 01	1.1.201	9/2/21	LOG_TELEGRAMS	SchneiderCO2, humidity and temp. sensor	Temperature APT01	Polito.CO2 - RH - Temperature.Temperature APT01	Temp

Apartment n	Device ID	Location	SensorID	Sensor Description	Description	Normalized list			
718	Jps7MZXbQeuT0yXWeFH6Bg	bedroom	0015BC002F00248E	Smart plug	IT appliances	ElecPowerITApp	ElecConsITApp	PowerSwitchITApp	TotEnergyProdITApp
718	Jps7MZXbQeuT0yXWeFH6Bg	bedroom	0015BC002F002462	Smart plug	iron	ElecPowerIron	ElecConsIron	PowerSwitchIron	TotEnergyProdIron
718	Jps7MZXbQeuT0yXWeFH6Bg	kitchen	0015BC002F002775	Smart plug	el. kettle	ElecPowerKettle	ElecConsKettle	PowerSwitchKettle	TotEnergyProdKettle
718	Jps7MZXbQeuT0yXWeFH6Bg	living room	0015BC002F00253F	Smart plug	HiFi	ElecPowerHiFi	ElecConsHiFi	PowerSwitchHiFi	TotEnergyProdHiFi
718	Jps7MZXbQeuT0yXWeFH6Bg	living room	0015BC0035000639	Humidity/temperature sensor		HumLivingroom	TempLivingroom		
718	Jps7MZXbQeuT0yXWeFH6Bg	Front door	0015BC001E00372E	Window sensor	door	DoorOpenEntrance	TempEntrance		
718	Jps7MZXbQeuT0yXWeFH6Bg	kitchen	0015BC001E00373B	Window sensor	window	WindowSwitchKitchen	TempKitchen		
718	Jps7MZXbQeuT0yXWeFH6Bg	bedroom	0015BC001E002C58	Window sensor	window	WindowSwitchBed	TempBed		
718	Jps7MZXbQeuT0yXWeFH6Bg	living room	0015BC001E0036CC	Window sensor	window	WindowSwitchLivingroom	TempLivingroom		
719	rPbTWyKDRICxM7GZt4H23g	basement	0015BC002F0026AF	Smart plug	washing machine	ElecPowerWashing	ElecConsWashing	PowerSwitchWashing	TotEnergyProdWashing
719	rPbTWyKDRICxM7GZt4H23g	office	0015BC002F002641	Smart plug	monitor/display	ElecPowerMonitor	ElecConsMonitor	PowerSwitchMonitor	TotEnergyProdMonitor
719	rPbTWyKDRICxM7GZt4H23g	kitchen	0015BC002F002399	Smart plug	el. Kettle	ElecPowerKettle	ElecConsKettle	PowerSwitchKettle	TotEnergyProdKettle
719	rPbTWyKDRICxM7GZt4H23g	living room/kitchen	0015BC002F0024A5	Smart plug	vacuum cleaner - irobot	ElecPowerVacuumClean	ElecConsVacuumClean	PowerSwitchVacuumClean	TotEnergyProdVacuumClean
719	rPbTWyKDRICxM7GZt4H23g	bedroom no.3	0015BC0035000669	Humidity/temperature sensor		HumBed3	TempBed3		
719	rPbTWyKDRICxM7GZt4H23g	bedroom no.1	0015BC001E003686	Window sensor	balcony door	DoorOpenBed1	TempBed1		
719	rPbTWyKDRICxM7GZt4H23g	bedroom no.2	0015BC001E00365E	Window sensor	balcony door	DoorOpenBed2	TempBed2		
719	rPbTWyKDRICxM7GZt4H23g	bedroom no.3	0015BC001E00364D	Window sensor	balcony door	DoorOpenBed3	TempBEd3		
719	rPbTWyKDRICxM7GZt4H23g	office	0015BC001E002E1A	Window sensor	balcony door	DoorOpenOffice	TempOffice		
719	rPbTWyKDRICxM7GZt4H23g	AMI smart meter	106230800880202		total energy consumed (in kilowatt hours)	TotEnergyCons			
719	rPbTWyKDRICxM7GZt4H23g	AMI smart meter	106230800880202		datetime of the measurement (CP48)	Timestamp			
719	rPbTWyKDRICxM7GZt4H23g	AMI smart meter	106230800880202		meter's serial number	MeterNum			
719	rPbTWyKDRICxM7GZt4H23g	AMI smart meter	106230800880202		total energy consumed (in kilowatt hours) for G11 tariff	TotEnergyConsG11			
719	rPbTWyKDRICxM7GZt4H23g	AMI smart meter	106230800880202		maximum energy consumed (inkilowatt hours)	MaxEnergyCons			

LOCATION	SENSOR ID FROM SCADA	DESCRIPTION SL	TRANSLATION ENG	normalized list	LOCATIOI	INAP SENSORS ID	DESCRIPTION	normalized list
K1N0623	FKKT_FRI.K1N0623_AO_DO4_VENTIL_FC_HL.F_(	Ventil konvektor hlajenje	Valve position convector cooling = K1N0623_FC_HL = Fan coil cooling valve	FanCoilCoolValv	K1028	0068224F_ACSCVTHLNPV	relative humidity, CO2 concentration, V	CO2, Hum, VOC
	FKKT_FRI.K1N0623_AO_DO5_VENTIL_RAD_GR.F	Ventil radiator gretje	Valve position radiator heatin K1N0623_RAD_HV = Radiator heating valve	HeatValvPos				
	FKKT_FRI.K1N0623_AV_HITROST_VENTILATOR.F	Dejanska hitrost ventilatorja	Convector ventilator speed [0-3] K1N0623_FC_SPEED = Fan coil speed	FanCoilSpeed				
	FKKT_FRI.K1N0623_AV_NAGIB_ZALUZIJ.F_CV	Položaj žaluzij	External shading position K1N0623_ANGL_BLINDS B= linds slope	SolShad				
	FKKT_FRI.K1N0623_AV_PRISOTNOST.F_CV	Prisotnost preko kartice	Occupancy from access K1N0623_OCC Room occupied	Occ				
	FKKT_FRI.K1N0623_AV_SP_TPR.F_CV	Želena temperatura prostora	Set point temperature K1N0623_SP_TEMPR = Room set point temperature	SetPointTemp				
	FKKT_FRI.K1N0623_AV_SP_ZALUZIJE_AVTO.F_C	Nastavitev kota žaluzij Avtomatsko	Set shading angle automatic (it is not the same: this mode, that determines the angle calculated based on sun position) K1N0623_BLINDS_MODE.F_CV = Blinds mode (AUTO/MANUAL)	SolShadAng				
	FKKT_FRI.K1N0623_AV_TEMP_PROSTORA.F_CV	Temperatura prostora	Room air temperature K1N0623_CV_TEMP = Room temperature current value	Temp				
	FKKT_FRI.K1N0623_AV_TEMP_REZIM.F_CV	Temperaturni režim - GR/HL/IZKL	Temperature regime [Heat/Cool/Off] K1N0623_REG_TEMP = Room temperature regime (HEAT, COOL, OFF)	TempReg				
	FKKT_FRI.K1N0623_BI_DI3_OKNO.F_CV	Okensko stikalo	Window switch K1N0623_WINDOW = Room window (open, close)	WindowSwitch				
	FKKT_FRI.LIGHT_OUT_FKKT_K4_N1C_74.F_CV	K1N0623: luči	Lights use	Light				
	FKKT_FRI.LIGHT_OUT_FKKT_K4_N1C_82.F_CV	K1N0623: luči	Lights use	Light				

Table. 15 – Examples excels for design of tree structure – Italian, Polish, Slovenian case

Step 2 – metadata structure check and validation

The excel files showed above have been compared with the specific requirements coming from the Game (HS) to check if the requested information were already collected by DMO. All the main fields requested by the Game were already available.

Demonstration case > Home > Room > Device > Sensor > Measurement

Starting from this structure, it was fundamental to understand the relations and hierarchies between the different levels before proceeding with the software implementation. For example, the measurement always refers to a sensor. But the sensor can be connected to a device – such as sensors embedded in smart plugs – or to a specific room – such as a window sensor is associated to the window of the kitchen –.

UseCases

ID	Code	Name	Location	Contact	Provider	RetrievalPeriod
1	DK					
2	PL					

Homes

ID	HomeNum	UseCaseID	DateAdded	Enabled
1010	729	1		
1011	730	1		

Rooms

ID	HomeID	RoomNum	RoomType	NumDoors	NumWindows	UserAssignedNa
20001	1011	1	Kitchen	2	1	
20002	1011	2	LivingRoom	1	2	
20003	1011	3	Bedroom	1	1	Joe's room
20004	1011	4	Bedroom	1	1	Children's room

Devices

ID	DeviceID	HomeID	RoomID	DeviceType	Placement	ActiveFrom	ActiveTo	Enabled
1	dwd933434a	1011	20003	Combined-T-C-R	near window			
2	4rf3ewd3g5	1011	(null)	HeatMeter	inside utility cabinet			

Sensors

ID	SensorID	DeviceID	SensorType	SamplingType	SamplingRate	SampleUnits	ValueType	ActiveFror	ActiveTo	Enabled
1001	adc5w34w	dwd933434a	Temp	average	15min	C	float			
1002	adc3vwe55	dwd933434a	RH	average	15min	%	float			
1003	5tfc454ff	dwd933434a	CO2	average	15min	ppm	float			
1004	5c4w56g4	dwd933434a	Door	event	(null)	0/1	int			
1005	g345w5f4	4rf3ewd3g5	DCW	average	15min	m3	float			
1006	4253g645	4rf3ewd3g5	DHW	average	15min	m3	float			
1007	f5w233f5	4rf3ewd3g5	Heat	average	15min	kWh	float			

Users

This is probably kept by HX

ID	HomeID	NameFirst	NameLast	UserEmail	UserPhoneID	DateAdded	IsEnabled
1331	1011	Mary	Simpson	marysimpson@s	+45-123456789	5/22/2018	TRUE

SensorData

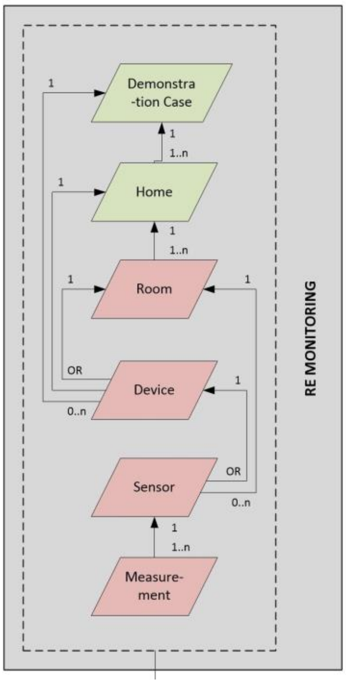
Timestamp is ISO8601; alternative, it can be UNIX time, but needs guarantee of correctly referring to UTC
ImportBatchID is a proposal for a simple way to allow retrieval of only the changed data

ID	SensorID	TStampRef_UTC	Value	ImportBatchID	
19999900	1001	2018-05-12T07:37:16	21.3	444	
19999901	1002	2018-05-12T07:37:16	30.1	444	
19999902	1003	2018-05-12T07:37:16	400	444	
19999903	1004	2018-05-12T07:00:00	1	444	Example of
19999904	1004	2018-05-12T07:10:00	0	444	door
19999905	1004	2018-05-12T07:12:00	1	444	open/close
19999906	1004	2018-05-12T07:13:00	1	444	events
19999907	1005	2018-05-12T07:34:12	0.04	444	DCH
19999908	1006	2018-05-12T07:34:12	0.1	444	DHW
19999909	1007	2018-05-12T07:34:12	12.1	444	

Table. 16 – Fields required by HighSkillz for the development of the Game

Step 3: software development

Once the structure and the connection between the different layers were validated as shown in the image on the left, the third step consisted implementing this structure into software.



DMO has developed RE Monitoring application for the definition and management of metadata. RE Monitoring is implemented in DMO's existing software tool RE Suite, used as a framework to build the application that enables the user to do this by defining a metadata tree structure which subsequently includes, from coarse to fine, demonstration cases, homes or zones, rooms, devices, sensors and measurements.

Fig. 10 – Tree structure logic behind the software development

The metadata structure and the software that manages this, RE Monitoring, provides the end-user with the necessary tools to be able to define this relationship.

RE Monitoring disseminates this metadata, the tree structure(s), to applications developed by other partners, such as the Dashboard and the Game. This will be further elaborated upon in the next chapter, 4.1.6, of this report.

RE Monitoring is also equipped with an interface for sensors data located in the Mobistyle platform. Therefore, RE Monitoring is the starting point for the further development of the Expert Tool, as explained in chapter 5.2.3 of this report.

RE Monitoring is a desktop application primarily written in C# (.NET) and uses a Microsoft SQL (MSSQL) database as the storage for both configuration as well as the metadata structure itself. The application is then made accessible through HTML4/5-compliant web browsers (all modern browsers) by using Myrtille, a gateway that encapsulates an RDP-session in a web browser.

4.1.6. Service layer Output

A web service has been implemented by DMO as an API façade pattern. This pattern is a virtual layer between the interface on top and the implementation on the bottom. It has been created a comprehensive view of what the API should be; most importantly, this view is given from the perspective of the app developer and end user of the apps [7].

Façade pattern mainly consists of two sections. The external interface is DMO's API which is known as the façade and the internal complex subsystem. When a user requests data from the system, the external interface is the component, which directly interacts with the user. The user can communicate with the internal complex system only by sending API requests to the façade. Façade forwards the requests to the suitable subsystem component and translates the user request as suited for the subsystem receiving interfaces. Subsystem components perform the actual work [8].

The data available through the web service can be distinguished into two main categories: sensor data (the actual data derived from measurements) and metadata (descriptive data to identify and relate sensor data). The metadata component of the web service communicates directly with RE Monitoring, providing the end-user with a data tree structure directly derived from the metadata as defined in RE Monitoring . The sensor data component communicates directly with the MongoDB which stores all sensor data. By combining metadata and sensor data the meaning of the sensor data can be attained.

According to user requirements 3 types of sensor had to be made available for each demo case: the latest available data record, range of data records between any two given dates and KPI calculations. So practically we can reduce it in 2 types: raw data as it is received from sensors and KPIs which are the result of calculations based upon the raw data. For this purposes we have built a separate subsystem only for KPI's. In total we have implemented 6 subsystems: 1 for each demo case, output creator and KPI calculation system. KPIs have been defined in collaboration with WP3 and WP6.

Each demo case subsystem is directly connected to the façade. For all incoming API requests façade decides to which demo case it should pass the request parameters. When request data arrives to a demo case subsystem it collects all needed metadata and creates a first part of database query and passes all of it to the output creator subsystem.

At this point different ways to collect the data are available.

For the raw data requests like last data and range data requests queries are finalized in place and sent to database. Response from the database is then converted to the standardized format and sent to user.

When KPI's are requested the output creator makes use of KPI calculation subsystem. While last and range data requests were straightforward to implement there was a challenge regarding KPI calculations in terms of performance and accuracy. If the latter is implemented naively a huge delays in time will arise which will slow down the whole web service. KPI calculator decides which type of calculation has to be done and depending on it lets one of its subcomponents generate a calculation query. Some calculations are so complex for the database that they have to be performed by our web service after data is received from the database. Finally when all the requested KPI information is computed, the data is converted into the standardized output format and sent to the user.

Most of the KPI's calculations make use of the core implementation of KPI query generator. This algorithm quickly generates efficient MongoDB queries which make the calculations on the database side increasing by this the overall performance of the web service. Below is explained what the algorithm actually does.

This core KPI query generator algorithm is suitable for any arithmetic operation and any simple mathematical function which is supported by MongoDB. Let's take for simplicity of explanation a calculation of average temperature as a KPI. To meet the user interface requirements it is not enough to simply calculate an average over the given time range, because it will result in one number while there is need in a list of numbers to visualize them as a diagram. So what we need is a list of time periods to calculate the average over each of them. These time periods should be defined within the abovementioned time range. And we also need to know the frequency of these periods, are they occurring every minute, hour or day. So it defines what the KPI API needs from the user, namely: start and end date to define time range, length of the period to calculate an average over it and frequency of those periods. An example from Table 16. below would result in a KPI as a list of 2 values, first for day 4 and second for day 5, where each value is the average of the past 3 days.

	Day 1	Day 2	Day 3	Day 4	Day 5
1st average calculation	Period 1			1st KPI record	
2nd average calculation		Period 2			2nd KPI record

Table. 17 - Calculation example of an average KPI for the range between the day 4 and day 6, period of 3 days and frequency of 1 day

The table below shows the list of KPIs that will be implemented in each demonstration case. For the Polish and Danish demonstration cases, additional KPIs might be investigated depending on the needs and goal of the MOBISTYLE Game developed specifically for the residential cases.

Demonstration case	Data type	Formula

Poland	Electricity consumption (Apartment level)	$E_{el} = E_{el\,(yesterday)} - E_{el\,(day\ before\ yesterday)}$
	Cost for electricity consumption	$Cost_{E_{el}} = E_{el} * PR_{kWh,el}$
	Electricity consumption (Smart plugs)	$E_{el} = E_{el\,(yesterday)} - E_{el\,(day\ before\ yesterday)}$
	Energy consumption (Appliances)	$E_{elc} = E_{elc\,(current\ washing\ cycle)} - E_{elc\,(previous\ washing\ cycle)}$
	Water consumption (appliances)	$E_{wc} = E_{wc\,(current\ washing\ cycle)} - E_{wc\,(previous\ washing\ cycle)}$
	Temperature	$OP = \frac{\sum_{i=1}^{1} OPi}{N}$
	Relative Humidity	$RH = \frac{\sum_{i=1}^{1} RHi}{N}$
Slovenia	Temperature	$OP = \frac{\sum_{i=1}^{1} OPi}{N}$
	Relative humidity	$RH = \frac{\sum_{i=1}^{1} RHi}{N}$
	CO2 concentration	$Ci = \frac{\sum_{i=1}^{1} CO2i}{N}$
Italy	Total Electricity consumption (Apartment level)	$E_{el} = E_{el\,(yesterday)} - E_{el\,(day\ before\ yesterday)}$
	Cost for electricity consumption	$Cost_{E_{el}} = E_{el} * PR_{kWh,el}$
	Electricity consumption (Smart plugs and appliances)	$E_{el} = E_{el\,(yesterday)} - E_{el\,(day\ before\ yesterday)}$
	Emission of CO2,equivalent	$CO_{2,eq} = \mathrm{E_{el}} * \mathrm{K1} * 365$ K1 = 0,337 kgCO2/KWh
	Number of trees that takes to absorb apartment CO2 emissions	$\mathrm{n} = CO_{2,eq}/\mathrm{K2}$
	Temperature	$OP = \frac{\sum_{i=1}^{1} OPi}{N}$
	Relative humidity	$RH = \frac{\sum_{i=1}^{1} RHi}{N}$
	CO2 concentration	$Ci = \frac{\sum_{i=1}^{1} CO2i}{N}$
Denmark	Temperature	$OP = \frac{\sum_{i=1}^{1} OPi}{N}$
	CO2 concentration	$Ci = \frac{\sum_{i=1}^{1} CO2i}{N}$
	Relative Humidity	$RH = \frac{\sum_{i=1}^{1} RHi}{N}$
	District heating water consumption	$E_{dh} = \sum_{i=1}^{n} E_{dh,i}$
	Costs for district heating water consumption	$Cost_{E_{dh}} = E_{dh} * PR_{kWh,dh}$

Table. 18 – List of KPIs for each demonstration case

Another endpoint implementation which was added on special request of HSz is a raw data request based on time stamped Batch ID. It was required to implement for as well each specific demo case as data of all demo cases at the same time for some time range. This kind of API endpoint forms a threat to the performance of the web service because it allows user easily and unnoticeably for him/her to request a huge amounts of data. To prevent this from happening we added a maximum range limitation of 1 day. It means that it is not allowed to request the data by Batch ID over a longer period of time than 1 day.

5. Further development and interfaces

5.1. Database preparation and requirements to scale up to 1000 households in the polish case

There are multiple bottlenecks which can influence the performance of the RE Data Spider and the web service during the scale up to 1000 households in Polish case. We have identified the following ones most of which did not occur until now but are expected for scale up: amount of requests done through Polish API, data pre-processing speed, database write speed, data storage space, database read speed.

At the moment we are dealing with 22 households with minimal 5 and maximal 15 sensors per household and an average of ~10.9 sensors per household. It means that the scale up to 1000 households will bring us from 239 sensors to ~10900. So we will have to make ~45 times more API requests to the Polish case servers and we will receive ~45 times more data than before.

The calculation above leads directly to the first and second identified bottlenecks.

Amount of API requests

Increased amount of API requests will definitely slow down the working of RE Data Spider. On one side because each HTTP request has a basic time consumption for opening and closing TCP connection each time the request is done. On the other hand the speed of Polish servers, which is the bigger part of time consumption, slows down the data fetching process. It results in a total time consumption of each API request between 20ms and 1500ms. It means that in worst case scenario, if every API request takes 1500ms, we will spend 24 minutes only on waiting for response from Polish case servers and then we will also have to pre-process and insert data in our database. But the total amount of time available to fetch, pre-process and insert all the data for all demo cases is 15 minutes.

Although the expectation based on the tests and practical usage of Polish API shows that a response time of 1500ms occurs only when requesting an amount of 100 records while ~2.5 million data records are available in total. So in practice we can expect 500ms per request in worst case scenario which results in 8 minutes. It is also a huge delay but it is possible to speed it up 4 times or even more by running API requests and processing the response asynchronously.

Besides the abovementioned issue there is a metadata available for each household which among other things includes a unique ID of each household to request the data for it. This information is at the moment delivered in a format of an excel sheet and manually inserted to our software. But it is not feasible to scale up to 1000 households in the same way because of amounts of data which would have to be inserted manually.

This problem will be solved by Polish demo case by making an API available for fetching the metadata of all households. This data will be fetched not every 15 minutes but ones in a while, we could think of few months. And then automatically added and used by RE Data Spider.

Data pre-processing speed and database write speed

For most of the sensor data the pre-processing speed of each record will remain the same so the total pre-processing of those records will increase linearly. On the other hand there is the data of “Electricity meter AMI” which is delivered in different way than data from other sensors. Instead of 1 record containing all the relevant information to a particular point of time we receive multiple records from different meters which have to be brought together by timestamp and meter ID. To combine those records safely, in a way that no data is missing, no duplicate data is introduced and without introducing mixed records which contain information from different sensors, we have to insert and update each of those records individually in our database. It slows down a writing process in to the database compared to bulk data insert which happens for all other sensors.

The scaling problem of most sensors can be solved with parallel processing, by pre-processing multiple data records simultaneously. But the exponentially growing data of "Electricity meter AMI" sensors does not pose a threat to the performance yet because of its amounts, namely it is only a 0.6% of all data we have from Polish case. So it is negligible even after the scaling.

Data storage space

At the moment we have approximately 9 GB of data from Polish case from the beginning of 2018, so expecting the growth of data according to the calculation above we will need additional 190 GB for Polish case for the duration of 12 months after the scale up. It is more space than we have at the moment, considering that the other demo cases data will also consume more space, but because we are using Azure cloud for our database it is easy and simple to increase the storage capacity.

Database read speed

Due to enlarged data sets of Polish demo case we will have to deal with the capacity of a single server, because working set sizes larger than the system's RAM stress the I/O capacity of disk drives. So it will not only influence the RE Data Spider but also the web service when the data is queried by the user. It will result in slow downs and irresponsibleness in the response of the single server.

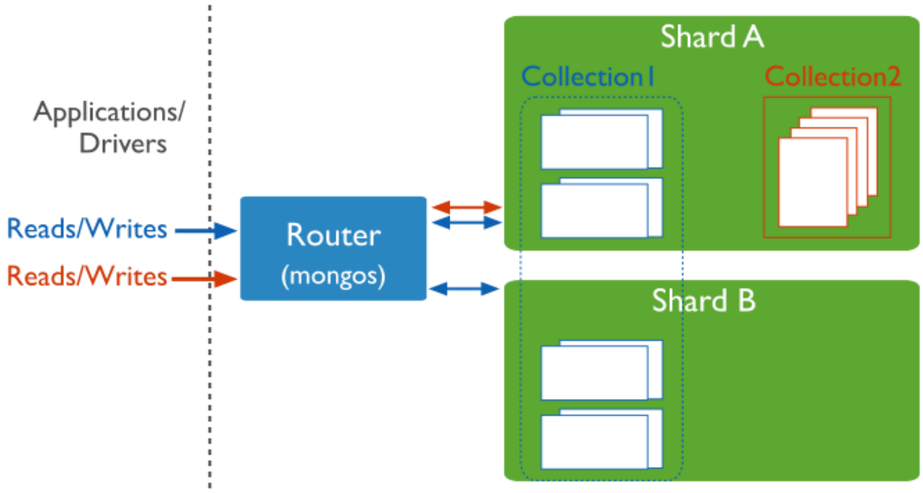


Fig. 11 - Sharding with MongoDB [7]

There are 2 possible solutions available in this case. First one is described in the solution of previous challenge; it is called the Vertical Scaling. It is an acceptable, quick and easy solution in our case regarding the data storage. But because all the operations done with data require better CPU and RAM capacity and Cloud-based providers like Azure have hard ceilings based on available hardware configurations, there is a practical maximum for vertical scaling. In case that we reach a maximum capacity in terms of storage capacity, RAM or CPU or all of them at the same time there is a second solution called Horizontal Scaling. Horizontal Scaling involves dividing the system dataset and load over multiple servers, adding additional servers to increase capacity as required. In contrary to the Vertical Scaling it does not require better hardware; instead it requires much cheaper multiple average servers [9]. It is supported by MongoDB through sharding with different strategies.

5.2. User interfaces

5.2.1. Dashboard

Users of the Mobistyle platform have access through authentication to an online dashboard where they can monitor the values coming from the sensors.

The sensors have been grouped based on the location and provided information in “sensorized entities” (e.g. a hotel room, a classroom, etc.), from which can be retrieved the current status of the entities and their history. The sensorized entities belong to the use case partners, which have the possibility to decide which information to show to the users and how.

There are different types of users and different types of dashboard to visualize based on the role of the user and their associated entities. Users can be classified in three categories:

- Admins: specific for case holders, in charge of the overall management of the entities and creation of the dashboard to be shown to the users. The admins are in charge of the account management and creation or removal of the users related to their organization.
- Managers: specific for case holders, with most of the privileges of the admin, but not for the account management. Their primary role is to give access to their sensorized entities to other users of the platform (e.g. the hotel manager that gives access to the data of a room to a user who booked it).
- Users: consumers of the sensor data. They cannot change the dashboard structure and see data only related to the sensorized entities which a manager gave them access to.

Admins and managers have the possibility to create new sensorized entities and connect them to the sensor interface. Being the owners of the sensorized entities, they’re capable of seeing on the dashboard all the sensorized entities they own. A specific dashboard is available for the managers and admins, and can be seen exclusively by them. Admins and managers belonging to different organization (i.e. case holders) cannot see each other sensorized entities.

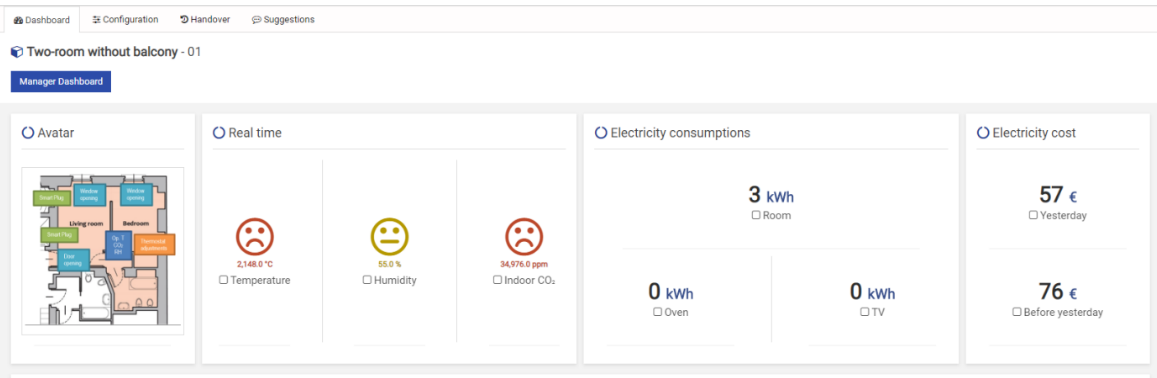


Fig. 12 – Example of manager dashboard

Admins and managers have the possibility to grant access to a specific user dashboard to users of the platform. Permission can be granted only to users, it cannot be granted to any manager or admin belonging to other organization.

Users can receive permission to consume sensor data by multiple admins or managers, meaning that a single user can see data coming from sensorized entities belonging to different organizations (e.g. a hotel room and a classroom). The users can see a specific dashboard for each sensorized entity.

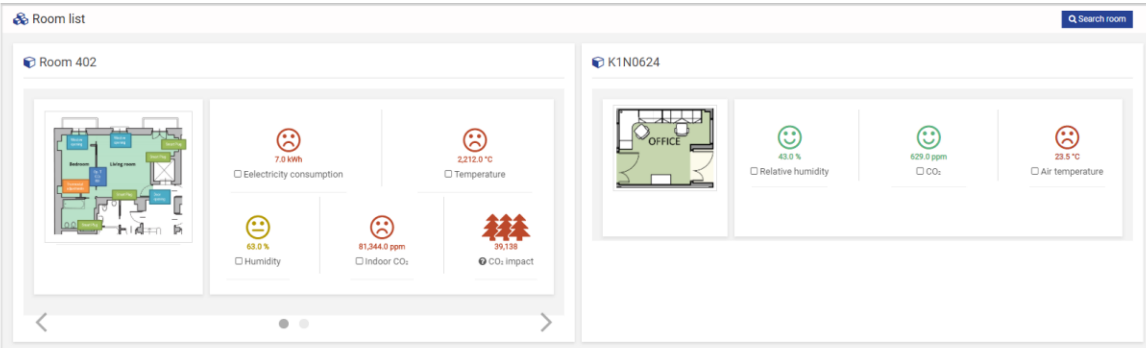


Fig. 13 – Example of list of sensorized entities for the user. Here the user can see a hotel room (Room 402) and a classroom (K1N0624)

A mobile application is available for all the user type of Mobistyle where they can access the same information available on the desktop dashboard. The same dashboard structure and widgets, with same configuration, can be accessed via mobile phone.

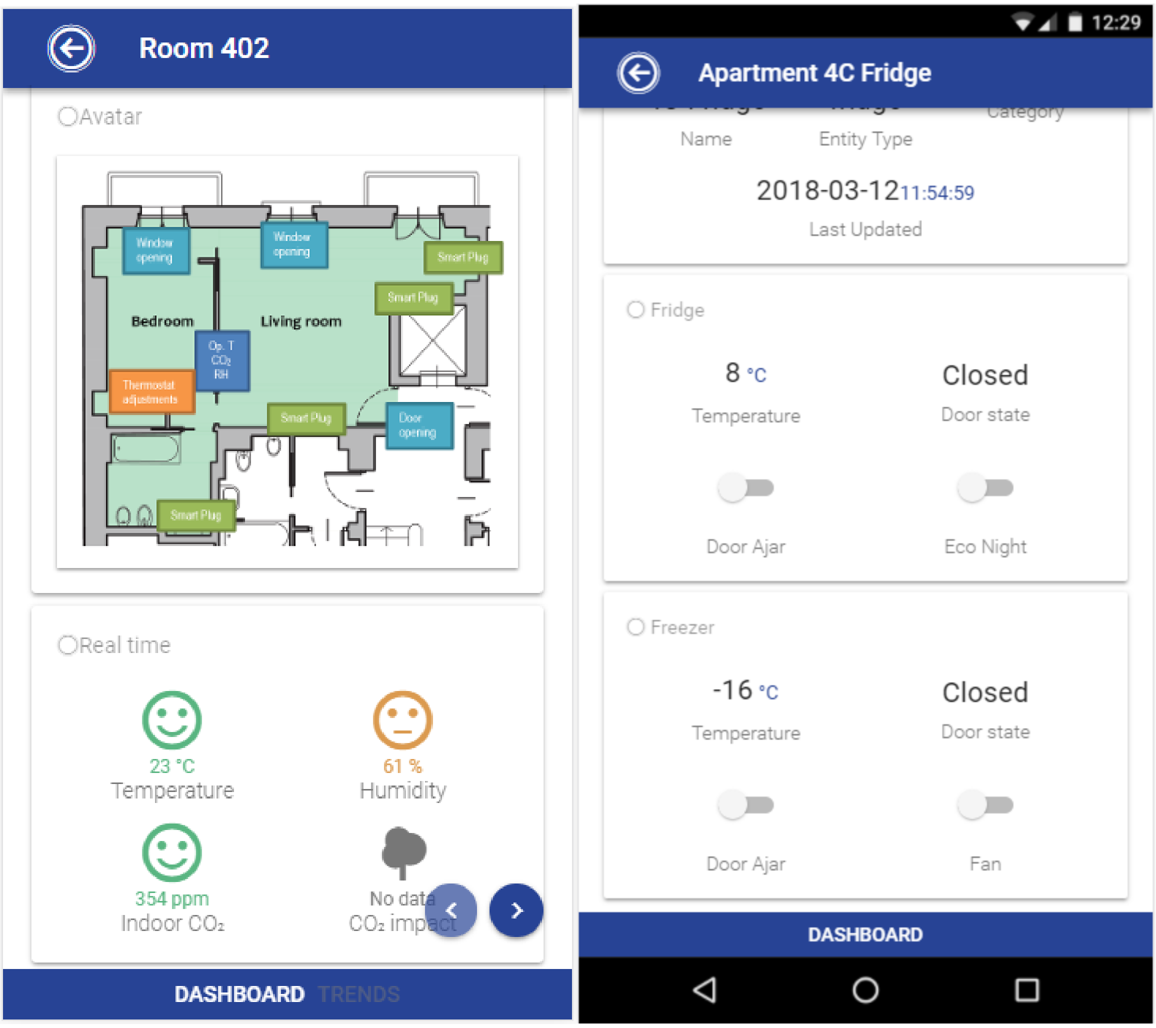


Fig. 14 – Example of mobile dashboard

5.2.2. Game

The MOBISTYLE Game development followed the projects' people centric approach, following the recommendations resulting from WP2 (see D2.3).

The Game relies on the operationalization of the MOBISTYLE behaviours in measurable actions that can be captured by sensors within the environment. The Game identifies different behaviours within the home based on data collated from the sensors available. Then based on the analysis of the data, the Game provides incentives in the form of recognition, achievements and suggestions with the ultimate goal to encourage the users to adopt and sustain particular behaviours towards better energy efficiency and also provide useful health tips.

A co-creation approach was applied in the Game design, features, contents and interface. HighSkillz collaborated closely with subject matter experts within the consortium as well as with end users from both the Danish and Polish use cases. To support the co-creation process, several sets of storyboards were created to enable the discussion on the use of the Game and relevant features to facilitate behaviour change, taking into account the constraints of the Mobistyle platform. A set of goals with associated missions on achieving energy efficiency were defined, in addition a list of health tips was also defined to encourage healthier lifestyles. For each one of the missions, a set of rules was defined to trigger the digital "nudges" that will encourage a person to take action, which complemented by the gamification would encourage the change of behaviour. A fundamental drive in the creation of the rules was the data requirements in terms of the data collated from the sensors in the environment, which was analysed for both residential use cases (DK and PL).

An interactive mock-up of the MOBISTYLE Game was developed to demonstrate the core functionality and features of the game. This mock-up was used to support the co-creation activities and collate feedback to shape the user experience of the initial prototype of the MOBISTYLE game that is currently being developed.

The final version of the Game will be available for mobile devices, following the recommendation on using smartphones (see D2.3).

The main screen of the MOBISTYLE Game allows easy access to the user's notifications and identified issues in need of their attention as well as to useful energy and health related tips (see Fig5). This screen also allows access to the rest of the sections, namely: House Status, Missions, Achievements and Profile.

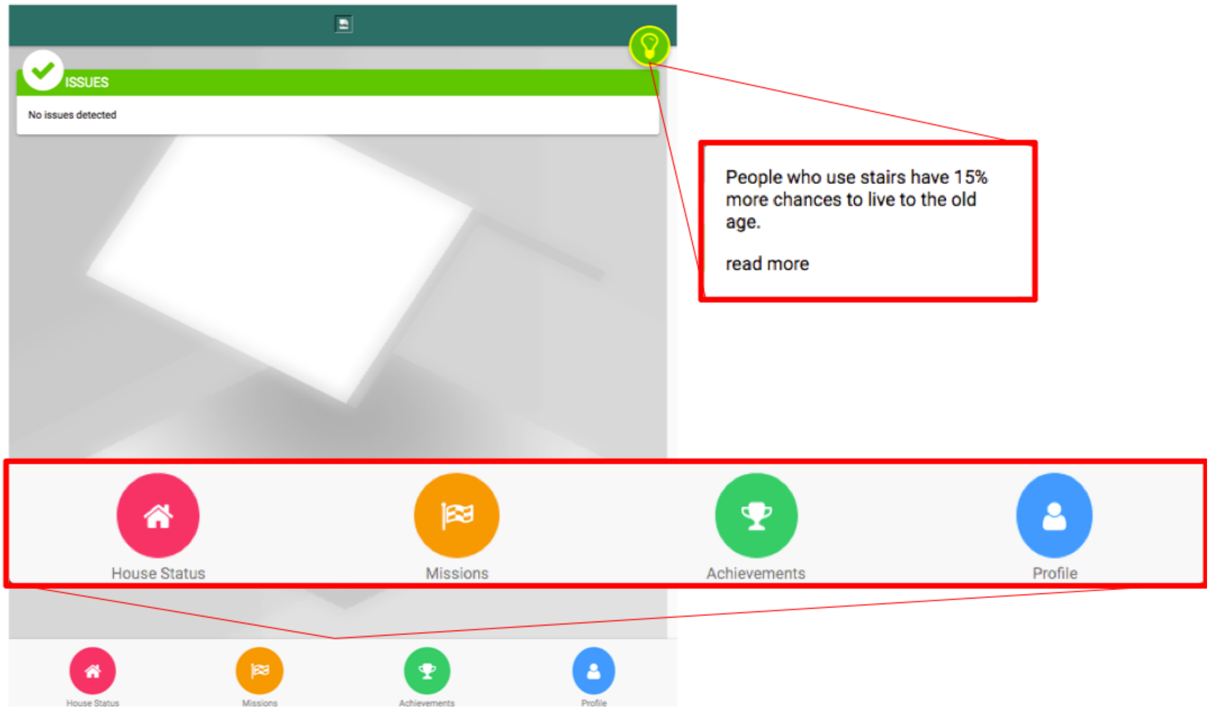


Fig.15 – Game: Home Screen

The "House Status" section gives an overview of all the sensorized rooms of the house. For each one of the rooms there's an indicator of its status. Consequently, the user may quickly identify possible issues in the house (e.g. Room 1 has one problem, see Fig. 16).

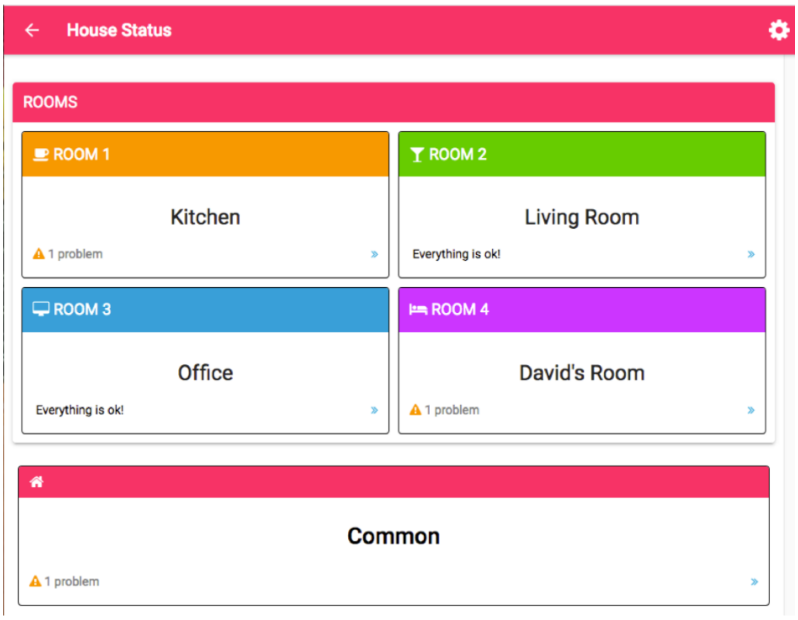


Fig. 16 – Game: House Status

If the user wishes to investigate further on the issues, they are allowed a more detailed view of the rooms. All the aspects of the room tracked by sensors are available for the user to review and act upon if needed (see **Fout! Verwijzingsbron niet gevonden.**17). In more detail, in the provided example the u ser can easily identify that there's an issue with the room's CO_2 levels and also gets a suggested action in order to deal with the issue -he is suggested to open the window to improve the air quality.

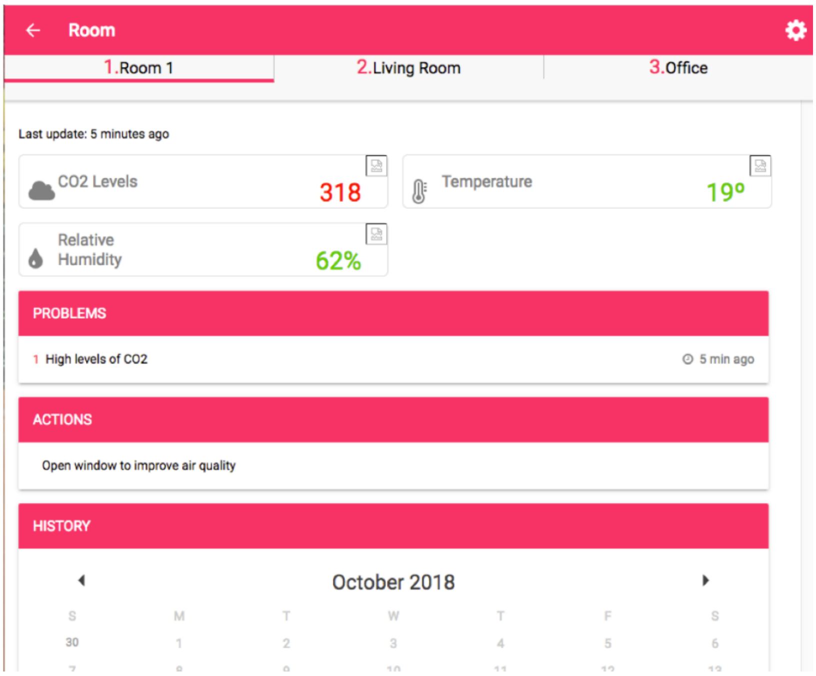


Fig. 17 – Game: Room View

The "Missions" section of the Game is where all selected missions can be found (e.g. sleep quality, air quality, night temperature etc.). In our example, the "sleep quality" mission, aiming to improve sleep quality by increasing the air quality during night time, has identified a problem (see **Fout! V erwijzingsbron niet gevonden.**). The user receives details on the problem and suggested actions. All the missions are linked to the available sensors in each house thus achieving customisation for each case, following the relevant recommendation for development (see D2.3).

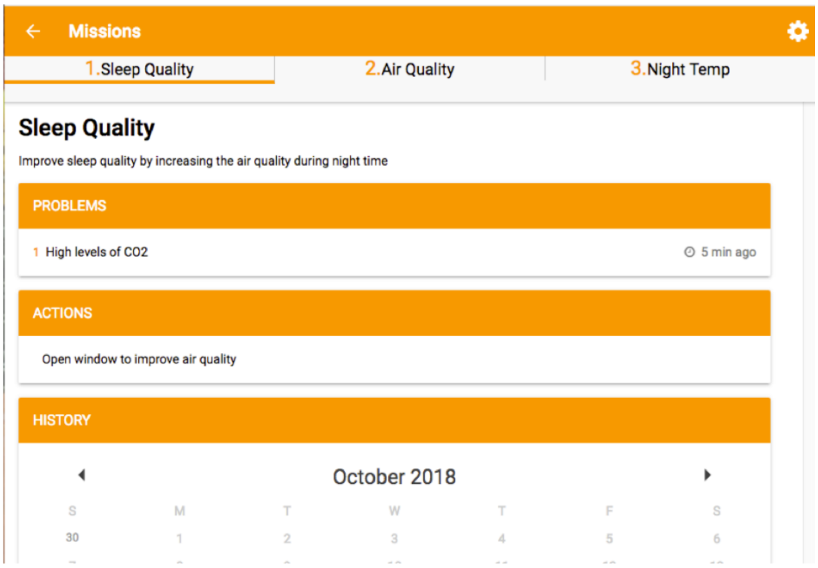


Fig. 18 – Game: Missions

All the actions of the user are monitored through the sensors installed in their house and are encouraged to adopt MOBISTYLE energy related behaviours using gamification mechanisms. The "Achievements" section of the Game refers to the gamification incentives used in terms of rewarding the user's behaviours (see **Fout! Verwijzingsbron niet gevonden.**19). The user receives different type o f medals and awards for his actions. They are also provided with a list of the suggested goals for them to choose from.

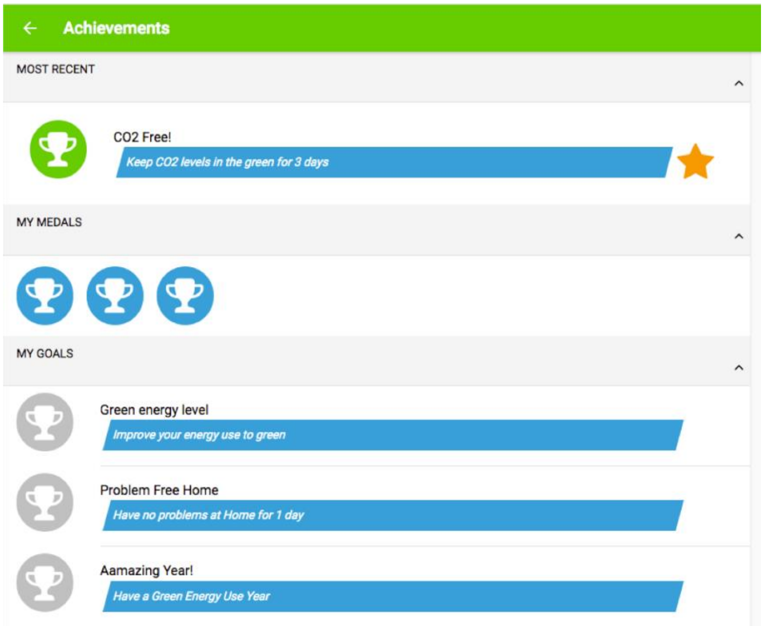


Fig. 19 – Game: Achievements

Further gamification mechanisms are used in the "Profile" section (see Fig.20**Fout! Verwijzingsbron niet gevonden.**). The user can access their points and compare themselves with the broader community, following to recommendation in peer pressure stimulating people to achieve their commitments (see D2.3).

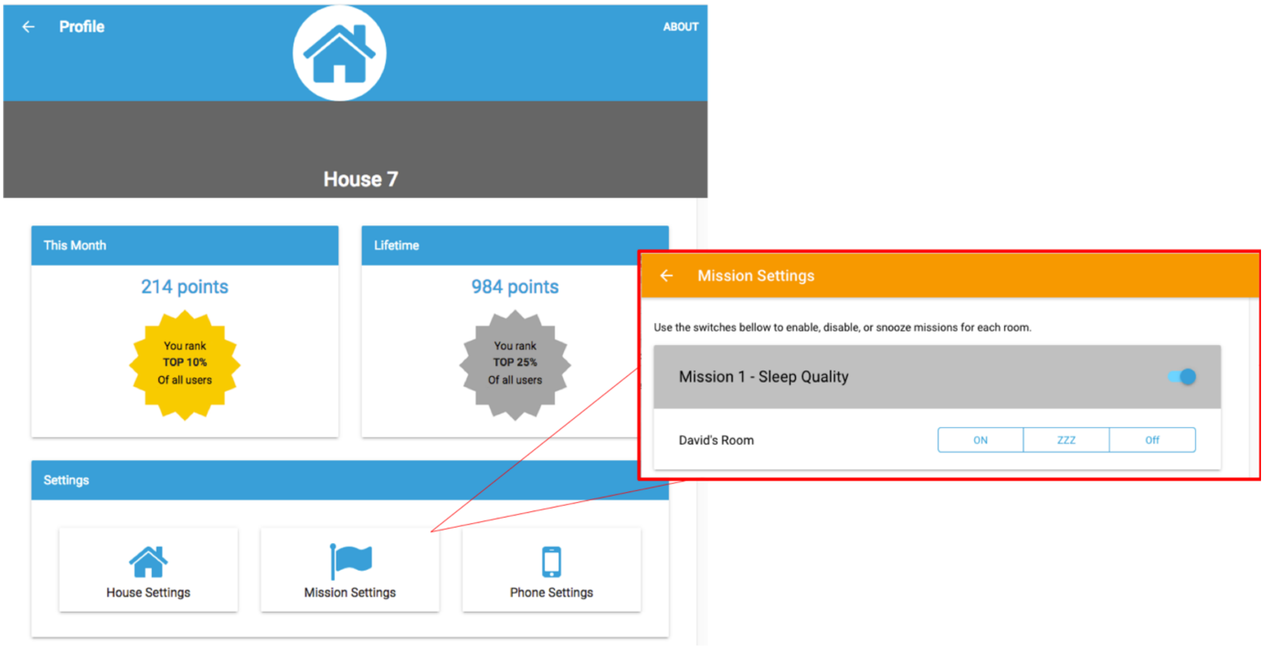


Fig. 20 – Game: Profile

Finally, in this section the user may also edit their settings and choose which notifications to receive by turning them on or off from the "Missions settings", following the recommendation on calm technologies principle (see D2.3).

5.2.3. Expert tool

The third ICT tool developed within MOBISTYLE is the Expert Tool developed by the consortium partner DMO. Within MOBISTYLE, this tool is intended for the experts in WP3, responsible for the analysis and evaluation phase to assess the MOBISTYLE approach is indeed influencing users toward a more energy efficient behaviour.

The need of this tool derive from the fact that there is the need to access, filter, extrapolate sensors data that are in big quantity and not available in a format clear also to non-software developers. For this reason, the Expert Tool has 3 main purposes:

1. Data management: Through the tool, the expert has access to the data for validation, check filtering.
2. KPI calculation: This feature will be further discussed with the experts in WP3 but the intention is first of all to give the possibility to calculate and visualize the KPIs presented in chapter 4.1.6 according to selected data. Secondly, it will be investigated the possibility for the user to modify the parameters and algorithms for the calculation of the KPIS. The KPIs are related to energy, comfort and health and are made available to the end user through the dashboard.
3. Interoperability: The expert tool will support data needs of third party analytical tools. In agreement with the experts in WP3, import / export functions and characteristics will be defined and implemented to allow to import / export sensors data and metadata in specifics format, to be used in other tools for further analysis.

The starting point of the expert tool is the RE Monitoring described in chapter 4. The additional functionality as described above will be implemented in addition to the current metadata functionality. The method of accessing the tool, a desktop application encapsulated in a HTML4/5-compliant web browser will remain the same, and provides easy external access. The images hereunder show the user interface for the expert tool.

The top task bar contains the main commands and configuration option. On the top left, are indicated the main categories coming from the tree structure described in chapter 4 of this document. By clicking on each category, the detailed information and content is available in the main screen. Here it is possible also to visualize the relation between the elements.

At the moment the tool is used for the management of the metadata. The next steps will consist in finalizing the implementation of the connection between the metadata and the sensors data in order to fulfil the first purpose of data management. After that the other features described above will be discussed and investigated in collaboration with WP3.

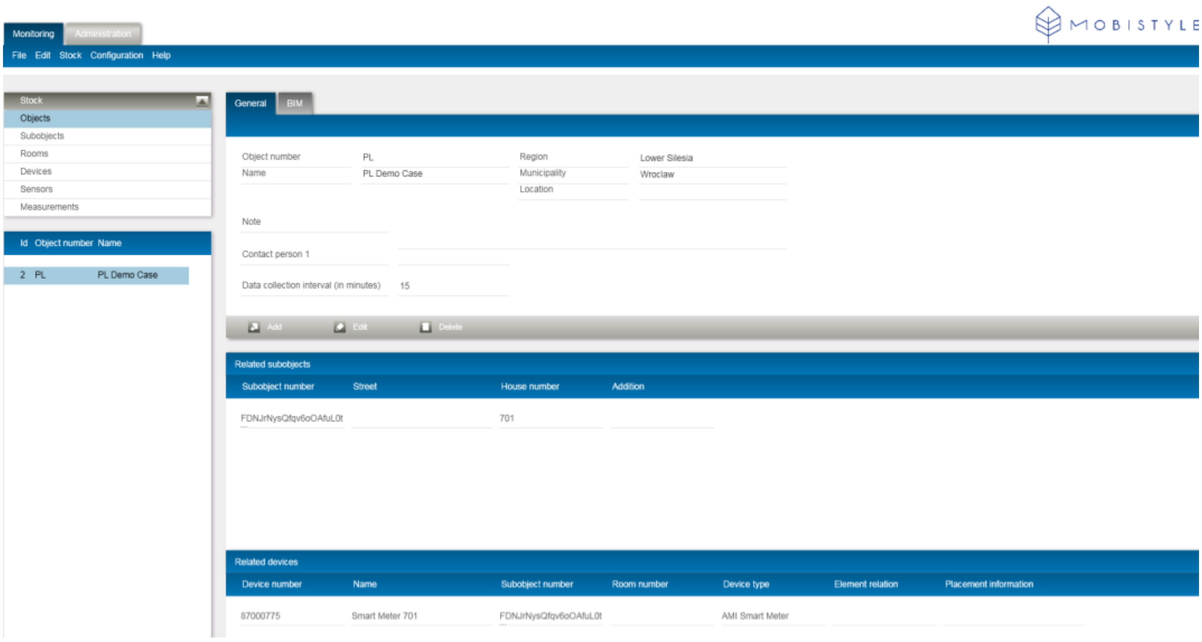


Fig. 21 – Landing page after login showing all the demonstration cases and sub-elements (apartments, rooms, devices, sensors) for each case

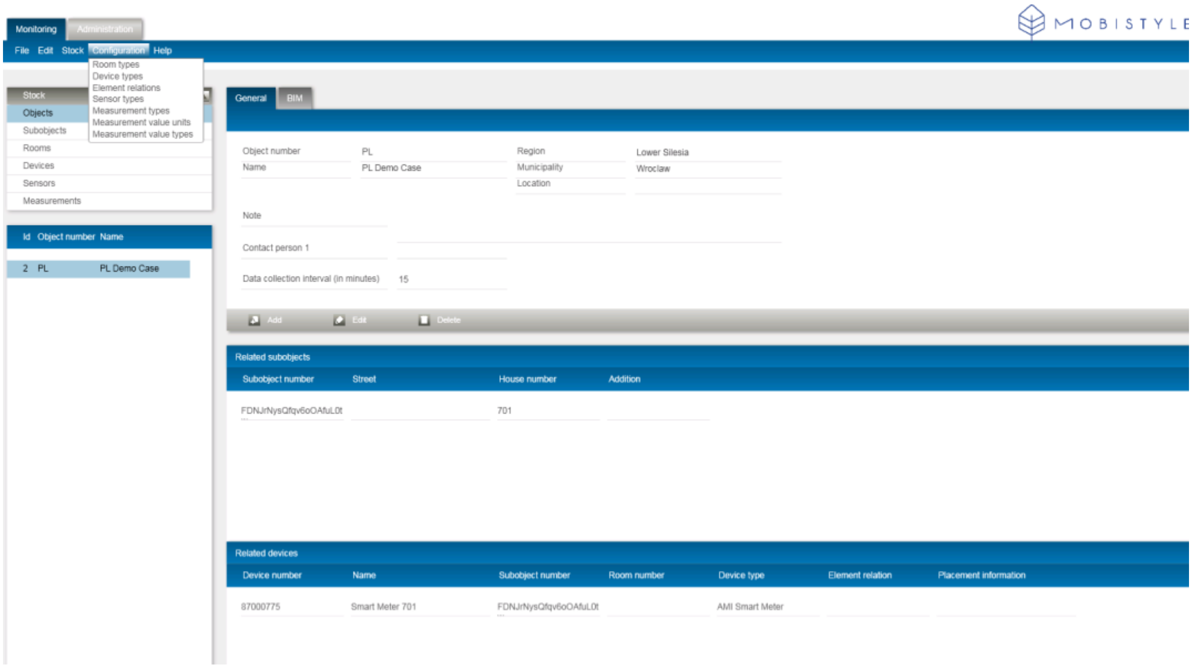


Fig. 22 – Configuration options where it is possible to set up the room type, the device type, the data type, sensor type and the relation within the elements

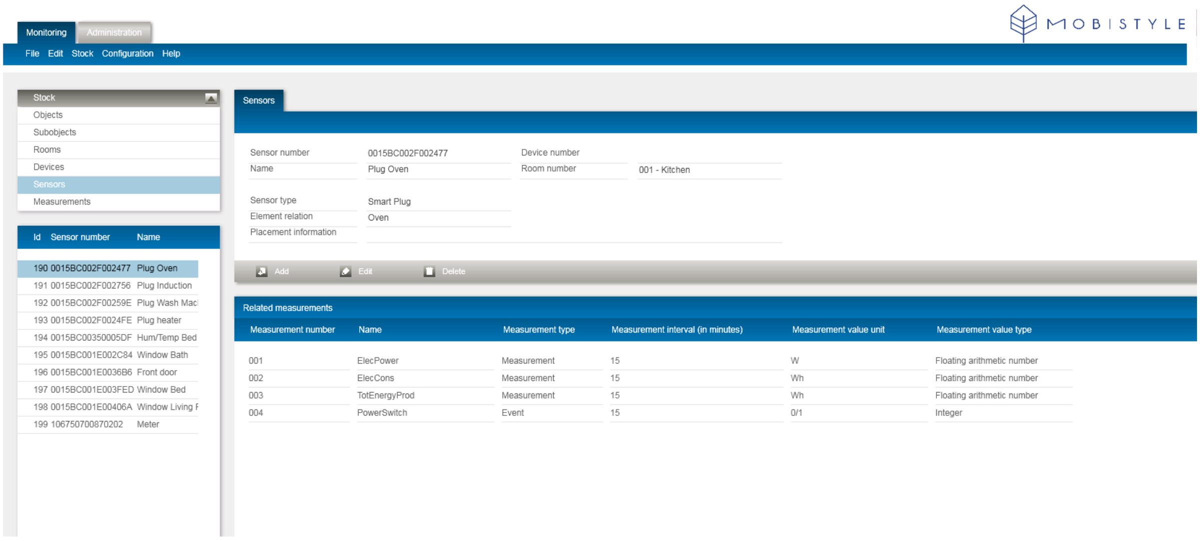


Fig. 23 – Example of list of measurements coming from a specific sensor or device in the Polish case

6. Conclusions

Deliverable 4.1 describes all the work done in WP4 from M12 to M24 in relation to the MOBISTYLE platform, in particular the data collection, processing and storage of sensors data coming from the different demonstration cases. The content of this report is the result of the close collaboration between the ICT partners in WP4 and the demonstration cases in WP6, with the support of the energy experts and anthropologists in WP2 and WP3.

Each demonstration case has an existing sensing network, mainly related in collecting energy data. But, in order to achieve the purposes of MOBISTYLE and according to the input coming from WP2 and WP3, additional sensors, with the aim of monitoring IEQ and occupancy, are installed. The first challenge encountered was to integrate the new sensors in the existing network. The second challenge was related to making the sensors data available and accessible from the MOBISTYLE platform, meeting also the performance requirements defined by the users' applications, WP2 and WP3. Some limitations derived from the structure of the sensing network in the demonstration cases which, in some cases, do not allow data collection so often.

Three different kind of information are made available for the development of the end user applications:

a) metadata regarding the location of each sensor in the room / apartment / demonstration case, the sensor type and what they measure;
b) sensor data available both as last measurement available and measurement within a range of time;
c) KPI values as defined specifically for each demonstration case.

The development of the MOBISTYLE database started by understanding the data type, data structure, network system and possibilities for data transfer from each demo case. One of the biggest challenges in this phase was that the data type, data structure and amount of the data and their frequency, and the data transfer possibilities were different in each demonstration case.

This document provides a description of the set up and characteristics of the MOBISTYLE central database and its connection to the demonstration cases and the user application is given. Measures on how to increase the performance of the database are also given.

While working on the data collection, it was fundamental to take into considerations requirements and constraints given by the Dashboard and the Game developed for the end users. For this reason, a brief presentation of these ICT tools is given.

After M24 the research activity will continue further developing and testing the ICT tools (Dashboard, Game, Expert tool). The results will be presented in a preliminary deliverable D4.3 in M30. Moreover, the scale up of the Polish case will be performed. This will result in researching solutions and further improve the performances of the MOBISTYLE databases and the interoperability between the different software tools used in MOBISTYLE.

References

[1] D. Clements-Croome. Creating the Productive Workplace, E&FN Spon, Taylor &Francis Group, London/New York, 2nd edition, 2006.

[2] L. Jagemar and D. Olsson. The EPBD and Continuous Commissioning. CIT Energy Management AB. Göteborg, October 2007.

[3] W.D. van Marken Lichtenbelt, M.J. Hanssen, H. Pallubinsky, B. Kingma & L. Schellen Healthy excursions outside the thermal comfort zone. Build Res & Inform Online 2017, pp. 1466-4321

[4] J. Toftum, A. Thorseth, J. Markvart, Á. Logadóttir. Occupant response to different correlated colour temperatures of white LED lighting. Building and Environment. 2018;143:258-68.

[5] Lubbers, Carlos. "Designing a Software Platform for the Collection and Storage of Smart-Sensor Data in Buildings" *Rotterdam University of Applied Sciences*, 2017.

[6] D.H. Wolpert W.G. Macready "No free lunch theorems for optimization" *IEEE Trans. on Evolutionary Computation* vol. 1 no. 1 pp. 69-72 1997.

[7] Mulloy, Brian. "API Facade Pattern - A Simple Interface to a Complex System." *Apigee*, 2012.

[8] Tak, T. "What Is Facade Design Pattern?" Java Code Gists, 10 May 2018, www.javagists.com/facade-design-pattern.

[9] "Sharding." In-Memory Storage Engine - MongoDB Manual, docs.mongodb.com/manual/sharding/.