



**Project no.:** 610658  
**Project full title:** eWALL for Active Long Living  
**Project Acronym:** eWALL  
**Deliverable no.:** D6.3  
**Title of the deliverable:** Technical evaluation report

<b>Contractual Date of Delivery to the CEC:</b>	30.04.2015
<b>Actual Date of Delivery to the CEC:</b>	30.04.2015
<b>Organisation name of lead contractor for this deliverable:</b>	University of Zagreb
<b>Author(s):</b>	Dina Simunic, Zeljko Martinovic, Igor Vitas, Andrej Grguric, Miran Mosmondor, Liljana Gavrilovska, Vladimir Atanasovski, Daniel Denkovski, Cristian Bara, Krasimir Tonchev, Georgi Balabanov, Aristodemos Pnevmatikakis, Maria Mitoi, Razvan Craciunescu, Octavian Fratu, Sofoklis Kyriazakos
<b>Participants(s):</b>	P01, P02, P03, P04, P06, P07, P08, P09, P10, P11, P12
<b>Work package contributing to the deliverable:</b>	WP6
<b>Nature:</b>	R
<b>Version:</b>	Final
<b>Total number of pages:</b>	35
<b>Start date of project:</b>	01.11.2013 – 31.10.2016
<b>Duration:</b>	36 months

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610658

Dissemination Level		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

**Abstract:**

This deliverable provides the description of the technical evaluation of the integrated eWALL prototype at the end of the half project's path. Report includes brief description, context, scope and objectives of the technical evaluation and end-to-end testing.

**Keyword list:** integration, interoperability, functionality, methodology, system integration testing.

## Document History

<b>Version</b>	<b>Date</b>	<b>Author (Unit)</b>	<b>Description</b>
0.1	20/03/15	UOZ	Started document
0.2	08/04/15	UOZ/RRD/AIT/AAU	eWALL metaphor
	08/04/15	UOZ	Introduction
0.3	09/04/15	UOZ	System integration testing
0.4	09/04/15	ENT	Inputs to section 5
0.5.	14/04/15	UKIM	Input to section 4
0.6.	14/04/15	RRD	Corrected metaphor Added Sleep application description
0.7	15/04/15	TUS	Added description on updated applications
0.8	16/04/15	UOZ	General update
0.9	17/04/15	AIT	Section 4
0.91	19/04/15	UPB	Section 4.1
0.92	27/04/15	ENT	Section 5
0.93	20/4/15	UOZ	General update
0.94	28/04/15	AIT	Updated section 2.1 with simulator details
0.95	29/04/15	UOZ	Final
1.00	30/4/2015	CTIF/AAU	Final review and approval

# Table of Contents

<b>1 EXECUTIVE SUMMARY</b> .....	<b>5</b>
<b>2 INTRODUCTION</b> .....	<b>6</b>
2.1 ROLE OF THE DELIVERABLE.....	6
2.2 RELATIONSHIP TO OTHER DELIVERABLES .....	6
2.3 STRUCTURE OF THIS DOCUMENT .....	6
<b>3 INTEGRATION STATUS</b> .....	<b>7</b>
3.1 INTEGRATING WP3 COMPONENTS .....	7
3.2 INTEGRATING WP4 COMPONENTS .....	9
3.3 INTEGRATING WP5 COMPONENTS .....	12
<b>4 TECHNICAL EVALUATION METHODOLOGY</b> .....	<b>18</b>
4.1 EVALUATION METHODOLOGY FRAMEWORK .....	18
4.2 SMOKE TESTING .....	20
4.3 USAGE MODEL TESTING.....	21
4.4 END-TO-END TESTING .....	23
<b>5 CONCLUSION</b> .....	<b>25</b>
<b>6 BIBLIOGRAPHY</b> .....	<b>26</b>
<b>ANNEX I : END-TO-END TESTING</b> .....	<b>27</b>
<b>ANNEX II: PLATFORM TESTING</b> .....	<b>32</b>

# 1 Executive summary

Technical evaluation is a very important parameter for every prototype (or product) development. In eWALL the importance is even stronger, as the prototype will be deployed into houses of a sensitive population; therefore the system should be robust and reliable.

Among WP6 activities, the technical evaluation is of major importance and the processes of the methodology applied, as well as the outcome is described in this document. The prerequisite for the technical evaluation is the system integration, which is a process occurring in cycles in WP6. The basic system was delivered (on-time) in M12 and since then it is further developed and optimized.

Progress and additional implementation toward M18 prototype followed the initial eWALL plan as well as suggestions related to M12 prototype, received European Commission. Received suggestions were adopted and realized by careful integration process. Partners from WP3, WP4 and WP5 invested additional effort in order to improve prototype by adding different functionalities. In order to achieve this goal, functionalities with appropriate service bricks, reasoners and applications were improved and developed. Functionalities such as: weather, notification manager, proximity, voice and sound, pulse, blood pressure, oxygen in blood, fall detection, messaging, questionnaires, body motion, motivation messaging, goal-setting, exercising video, sleep monitoring, personal hygiene, visitors services, sanitary visitors services, entertainment content, cooking services, presence detection, air quality, humidity, illuminance, temperature, doors & windows, administration tools and security protocols in platform are at latest stages of development and implementation. Those services are described more detailed in D6.2.2. prototype deliverables. Especial attention was dedicated to the eWALL main screen. Main screen was improved and developed by integration of additional services and applications.

## **2 Introduction**

### **2.1 *Role of the deliverable***

Deliverable D6.3 is a document report that presents the process of the technical evaluation of eWALL system and the outcome of this. The technical evaluation is an activity that follows the integration, which is described in detail in D6.2.2 for Phase 2 of the project (M18).

### **2.2 *Relationship to other deliverables***

It is related to the following eWALL deliverables:

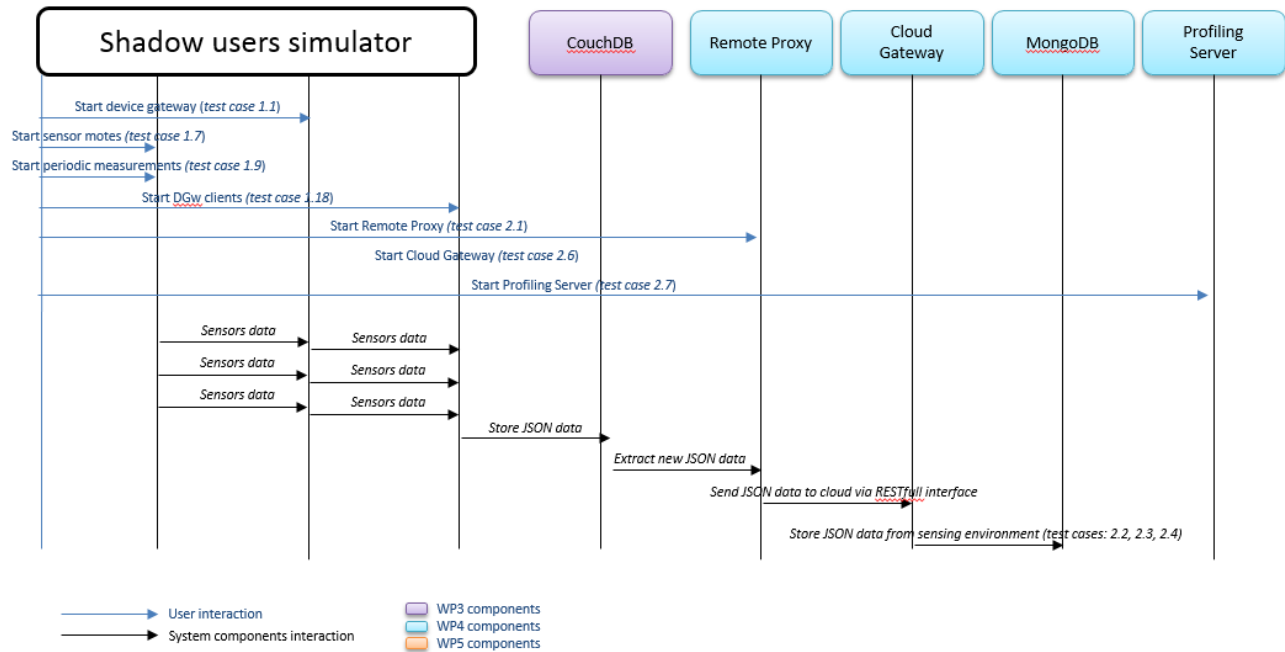
- D6.1 Integration report
- D6.2.2 eWALL prototype /Ph2(previous version)
- D2.7 Final user and system requirements and architecture
- D3.1.2 eWALL Networked Devices
- D3.2.2 Perception from sensors
- D3.3.2 eWALL configurable metadata streams
- D4.1.2 Semantic model of eWALL middleware services
- D4.2.2 Intelligent support system for eWALL
- D4.3.2 Cloud middleware services for eWALL
- D4.4.2 eWALL Platform
- D5.1.2 eWALL metadata from the outside: Filtered and retrieved
- D5.2.2 Lifestyle reasoning: Fusion for activities, situations and their patterns
- D5.3.2 Intelligent context-aware services & applications
- D6.3 Technical evaluation report

### **2.3 *Structure of this document***

The document summarizes its major points in the executive summary in section 1 and presents its scope, dependencies and structure in section 2. The integration status of different components from WP3, WP4 and WP5 is presented in section3. In section 4 we present the technical evaluation methodology that we have applied. Finally, the document sums up to the conclusions in section 5.

### 3 Integration Status

This section describes eWALL M18 Prototype in short. More information about the full achieved eWALL M18 Prototype can be found in D6.2.2.



\*Test cases extracted from deliverable D.6.1.1 "Integration report"

**Figure 1: Testing process**

Testing process is shown on Fig. 1. Process starts from “Shadow Users” simulator which fills CouchDB (database) and ultimately simulates user (Simpson). The data are transferred to eWALL software platform and further processed and displayed in eWALL applications.

#### 3.1 Integrating WP3 Components

In the period between the M12 and M18, eWALL prototype has changed its hardware components and software, which utilises hardware, for extracting the context of the user.

These are outlined in the list below:

- Door/window dry contacts: They have been fully integrated into the environmental mote and the environmental JSON message.
- Bed sensing: Arduino solution has been fully integrated and initial tests with the commercial Beddit have started.
- Activity sensing: Apart from our LilyPad based solution, it is being experimented with the commercial ProMove-3D and the Fitbit.
- Medical sensing: Arduino-based SPO2 measurements are fully integrated into the medical JSON message. The commercial stethoscope solution is ThinkLabs One that gives metadata but is not fully integrated yet. The blood pressure options have been deferred to the second half of 2015.

- Kinect sensing: Audio support is now in place. Both visual and audio processing components are being integrated.

Overall, the status of home sensing is as follows:

**Table 1: Home sensing status**

Category	HW	Device GW	Processing SW	CouchDB	Remote proxy	Simulated	Integration cycle
Visual	Kinect	Not connected	Proximity	N/A	N/A	N/A	M18
Visual	Kinect	Not connected	Face tracker: C++ ready	Yes	Yes	Yes	M18
Audio	Kinect	Not connected	Speaker diarisation: Java ready	Yes	No	No	M24
Audio	Kinect	Not connected	Memory book quiz	No	No	No	M24
Audio	Kinect	Not connected	Cough detection	No	No	No	M24
Medical	SPO2 arduino	Yes	Data forwarding	Yes	Yes	No	M18
Medical	SPO2 NONIN	Not supported	Not implemented	Yes	Yes	No	M21
Medical	Blood pressure arduino	Not connected	Not implemented	No	No	No	M21
Medical	Blood pressure OMRON	Not supported	Not implemented	No	No	No	M21
Medical	Stethoscope Thinklabs 1	Not supported	Not implemented	No	No	No	M21
Fall	Analogue accelerometer	Yes	Under development	Yes	Yes	Yes (always false)	M18
Fall	ProMove3D	Not supported	Under development	Yes	Yes	Yes (always false)	M21
Body motion	Analogue accelerometer	Yes	IMA, ISA, steps	Yes	Yes	Yes	M18
Body motion	ProMove3D	Not supported	IMA, ISA, steps	Yes	Yes	Yes	M21
Exercise	Bluetooth connector & Bike	Not supported	Not implemented	No	No	No	M24
Sleep	Arduino: Pressure, analogue acc	Yes	Pressure on, IMA	Yes	Yes	Yes	M18
Sleep	Beddit, Withings	Not supported	Pressure on, IMA	Yes	Yes	Yes	M21
Environmental	PIR, humidity, gas, dry contact, illuminance	Yes	Data forwarding	Yes	Yes	Yes	M18
Environmental	Smartplug	Not supported	Not implemented	No	No	No	M21
Environmental	Light control actuator	Not supported	Not implemented	N/A	N/A	N/A	M21



Mobility support	Smartphone GPS	Not supported	Not implemented	No	No	No	M24
Mobility support	Smartphone sensors	Not supported	Not implemented	No	No	No	M24

The higher-level components of eWALL utilise metadata from the caring homes in order to reason about the context of the care recipients and to provide personalised services to them. To do so, these components need a large volume of metadata, in terms of:

- Timespan covered
- Diversity of sensors and processing algorithms
- Diversity of users

Currently, the caring home sensing functionality is still evolving. Moreover, its deployments are few and partial in terms of what each site supports. Also, the deployments are rarely functional, meaning that used timespan is sparse: the sensing environments are relatively new and they are not always operational. Finally, the environments are still in our labs; the people sensed are not care recipients, they are researchers working on the systems. As a result, the needed volume of metadata is not available from the actual sensor deployments.

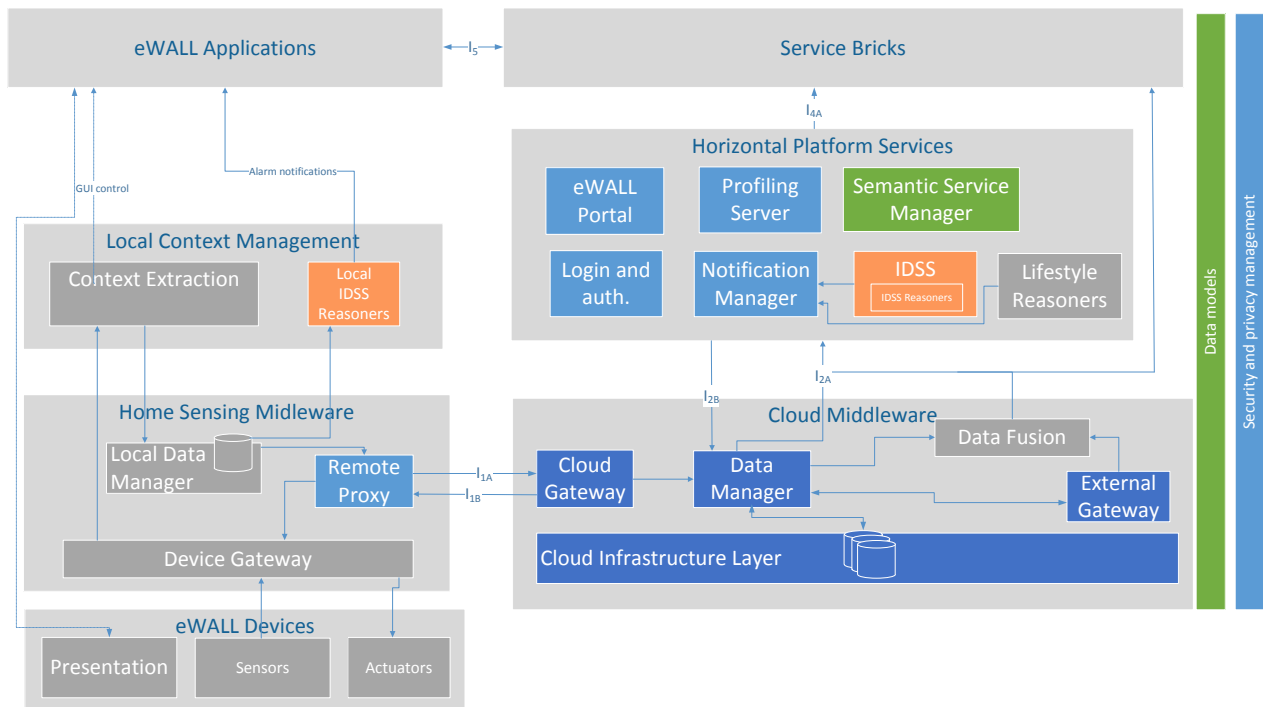
“Shadow User” is a simulation, driven by models for the home, the weather, the sensing environment and the behaviour of the user in it. The home model is fixed at initialisation time, the weather is obtained from an external service, and the sensing environment model comprises conditional probabilities of the measurements given the user state. Finally, the user behaviour model is described by state transition probabilities that vary based on four user driving forces, as discussed at the user state section.

The simulator returns all the metadata expected from an actual home, obtained by processing the measurements of the sensors. The metadata are organised into the same CouchDB databases, in the same JSON format found in the actual deployments. In addition, it returns the user state that resulted to these metadata. The goal of the higher-level components is to infer this state from the given metadata.

Currently the simulator is running at three virtual home environments, home-dev (user Simpson) for development and home-int-level-2 (user Simpson) and home-int-level-2-b (user Bob) for testing. It is exactly the latest two virtual environments that are used for the end-to-end tests in Section 3.4 concerning users Simpson and Bob.

## 3.2 *Integrating WP4 Components*

eWALL system is divided into two main subsystems: eWALL Sensing Environment and eWALL Cloud Environment.



**Figure 2: Illustration of eWALL decomposition model**

#### **eWALL Sensing Environment** includes:

- eWALL Devices (sensors, actuators and presentation devices) responsible for implicit and explicit interaction with the end user(s),
- Home Sensing Middleware responsible for connecting and configuring the end user devices, managing local data (in terms of privacy, etc.) and for sending data (that is allowed to be sent) to the eWALL Cloud,
- Local Context Management responsible for low level (context) data processing and extraction.

#### **eWALL Cloud Environment** includes:

- Cloud Middleware responsible for all cloud communication infrastructure, data management (storing, retrieving, transforming) and interoperability with external systems and data sources,
- Horizontal Platform Services responsible for providing common functionalities needed for development of personalized and intelligent services,
- Service Bricks responsible for utilizing Horizontal Platform Services and providing a set of functionalities to be used by eWALL applications,
- eWALL applications responsible for delivering the functionalities of eWALL system to the end user(s).

Following components have been developed within WP4:

**Table 2: WP4 components**

WP4 component	Description in deliverable	Responsible partner(s)
Remote Proxy	D4.4.2	ENT, AAU
Cloud Gateway	D4.3.2	ENT, AAU
Profiling Server	D4.4.2	ENT
Commons data model	D4.1.2	ENT
IDSS reasoner motivational messages	D4.2.2	RRD
IDSS reasoner automatic goal-setting	D4.2.2	RRD
IDSS reasoner sleep anomaly classifier	D4.2.2	TUS, UOZ
Notification Manager	D4.4.2	UKIM
eWALL Portal	D4.4.2	HP, ENT
eWALL Platform login	D4.4.2	HP

All WP4 components can be found in WP4 project's GitLab<sup>1</sup> server (<http://serv2.radio.pub.ro/gitlab>), which is connected to project's Jenkins<sup>2</sup> Continuous Integration server (<http://ewall.radio.pub.ro/jenkins/job/wp4-ci/modules>) configured to automatically deploy components on the *Platform-dev* cloud environment<sup>3</sup>.

All WP4 components can be found in WP4 project's GitLab<sup>4</sup> server (<http://serv2.radio.pub.ro/gitlab>), which is connected to project's Jenkins<sup>5</sup> Continuous Integration server (<http://ewall.radio.pub.ro/jenkins/job/wp4-ci/modules>) configured to automatically deploy components on the *Platform-dev* cloud environment<sup>6</sup>.

In comparison with M12 delivery following new features and components have been added:

- eWALL Platform login component – initial version provided for M18 (did not exist in previous M12 delivery).
- eWALL Portal component – initial version provided for M18 (did not exist in previous M12 delivery).

<sup>1</sup> <https://about.gitlab.com/gitlab-ce/>

<sup>2</sup> <http://jenkins-ci.org/>

<sup>3</sup> DEV environment is a set of virtual machines for development. *Applications-dev*: vm for service bricks and applications, *Platform-dev*: vm for central system, *Home-dev*: vm for hosting what is installed in a home environment, *Mysql-dev* and *mongodb-dev*: vms for databases.

<sup>4</sup> <https://about.gitlab.com/gitlab-ce/>

<sup>5</sup> <http://jenkins-ci.org/>

<sup>6</sup> DEV environment is a set of virtual machines for development. *Applications-dev*: vm for service bricks and applications, *Platform-dev*: vm for central system, *Home-dev*: vm for hosting what is installed in a home environment, *Mysql-dev* and *mongodb-dev*: vms for databases.

- Remote Proxy component – update of M12 component (most relevant: added AMQP protocol as an addition to HTTP protocol between Remote Proxy and Cloud Gateway component)
- Cloud Gateway component – update of M12 component (most relevant: AMQP support added, new data support added – e.g. mattress pressure, door status, illuminance data, gasses status, physical activity data..).
- Profiling Server component – update of M12 component with respect to M18 requirements, also an addition of RESTful endpoints for storing and deleting user credentials data needed for the authentication with the eWALL system. The passwords are hashed and stored to the separate part of the eWALL MongoDB database. New REST interface for checking if the given user is allowed/authenticated to access the eWALL system was added.
- IDSS reasoner motivational messages component and IDSS reasoner sleep anomaly classifier component - updates respect to M12 version.
- IDSS reasoner automatic goal-setting component - new component (not part of M12 delivery).
- Commons data model - update of M12 component with new ontologies and refinement of existing ones with respect to new requirements for (this) M18 delivery.

As a help to the developers (mostly from WP5) using WP4 components following documentation has been added/updated:

- Swagger interactive documentation (described in D4.4.x) has been added on all components exposing RESTful API (Cloud Gateway, Profiling Server, Notification Manager..).
- Javadoc (described in D4.4.x) has been added for new functionalities as well as improved for old ones so that every class and method is documented properly.

WP4 components wiki pages describing different functionalities and offering guidelines and usage instructions: <http://serv2.radio.pub.ro/gitlab/wp4/wp4project/wikis/home>

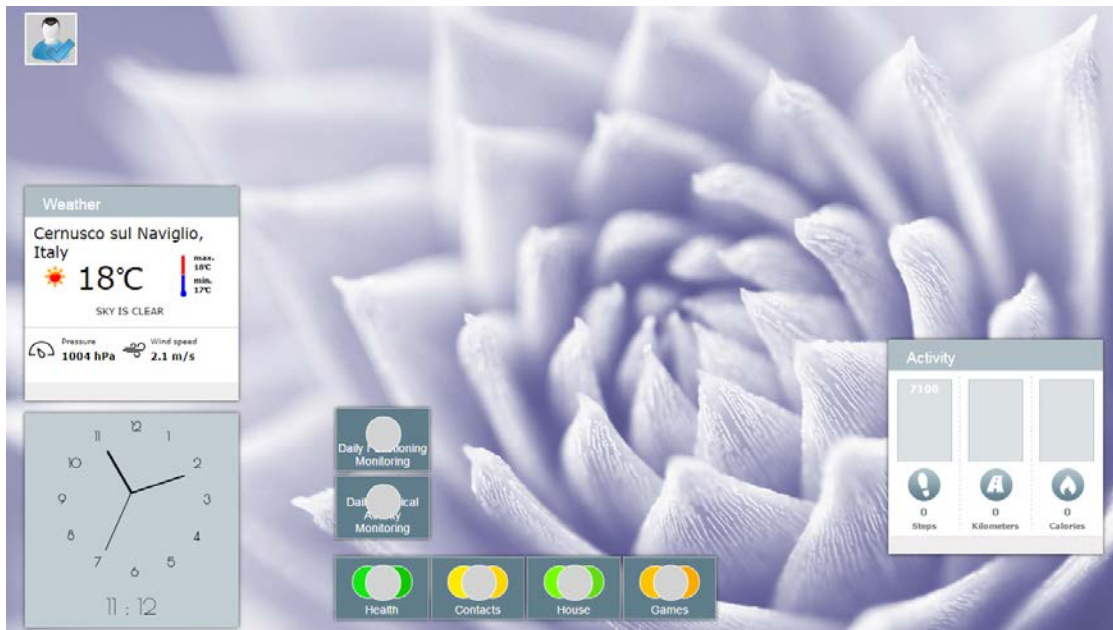
### **3.3 Integrating WP5 Components**

The eWALL GUI comprises a main screen and applications. The main screen plays a central role in the eWALL home installation and supports the user interface, which enables the primary user to operate and consume data provided by the eWALL system.

The new design of the main screen is shown in the following Figure.

In comparison to M12 delivery the following new features and components have been added:

- DPAM Widget – provides to the user in real time the status of its physical activities;
- Updated Weather Widget – better graphical interface;
- Authenticated Login – enables secure connection;
- A New Structure of Service Bricks and Applications – refactored architecture providing flexibility and better responsiveness ;

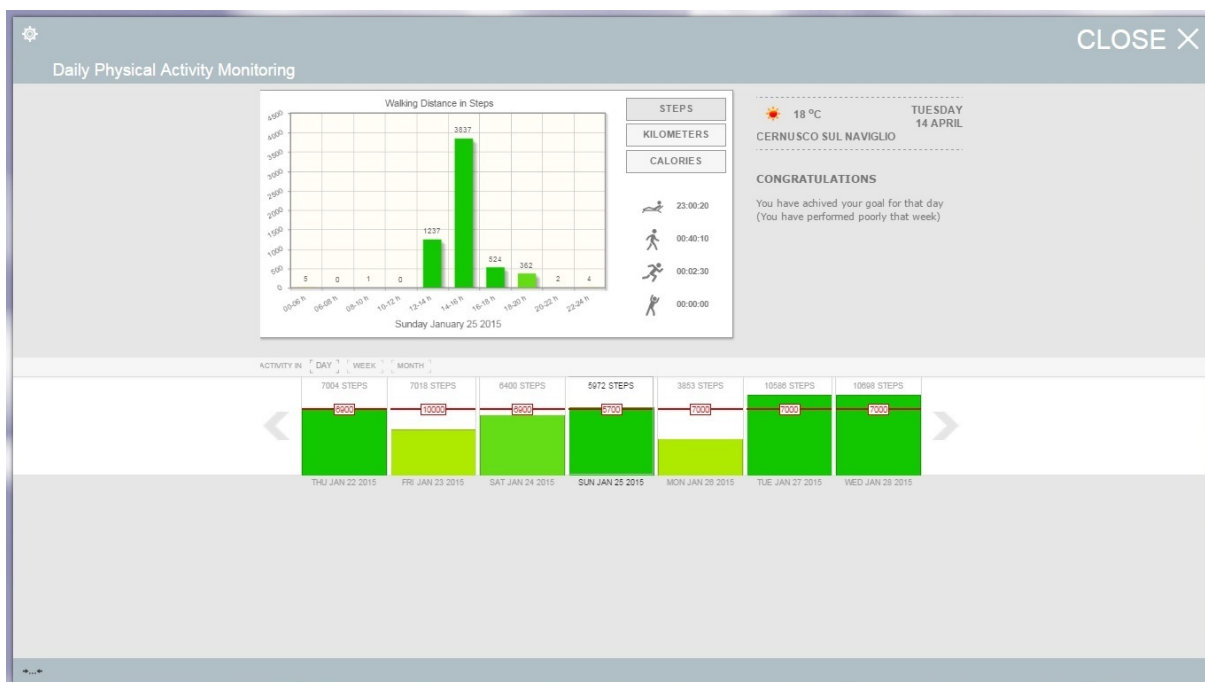


**Figure 3: Redesigned Main screen**

The DPAM Widget displays current activity (Steps, Walked Distance in Kilometres and Burned Calories) to the user. The information is updated every 10 seconds. It also supports “goal line” which is a target that the user has to accomplish for the day.

Compared with M12 the new structure of service brick implements one service brick with several endpoints supporting one application. Three new service bricks were developed:

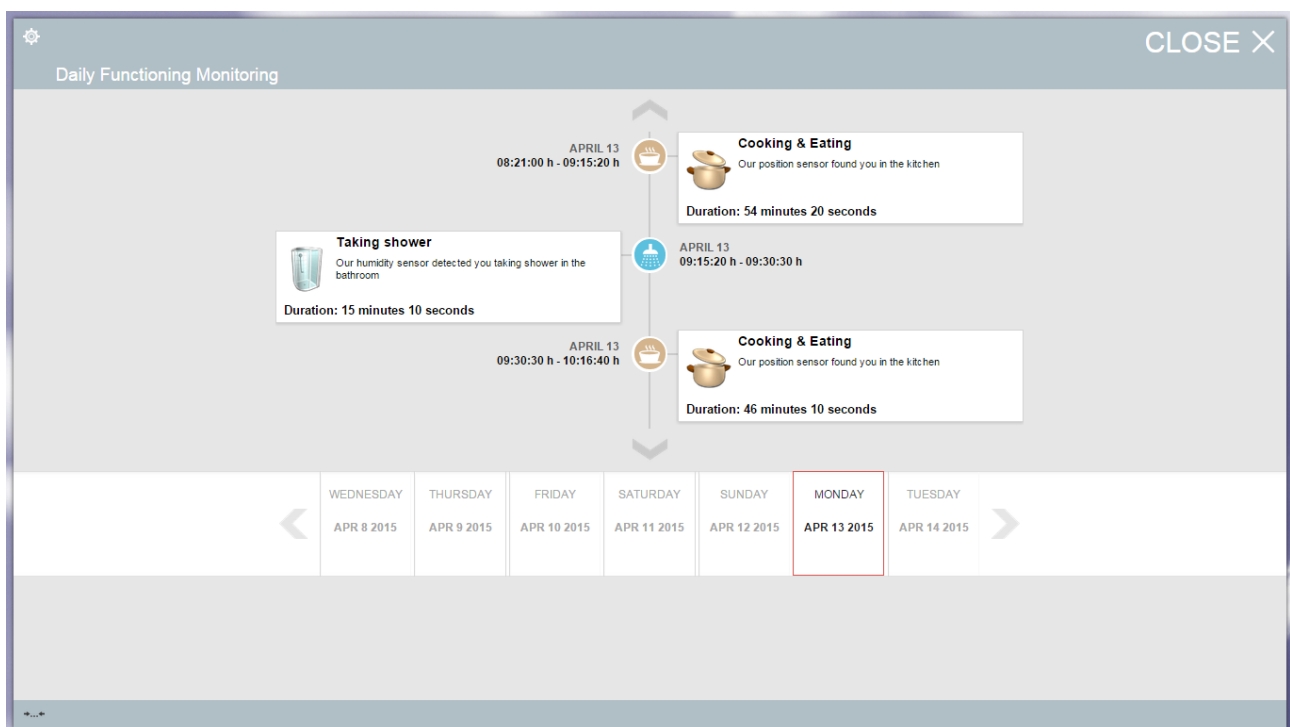
- DPAM Service Brick - delivers information related to user’s physical activities;
- DFM Service Brick – delivers information related to user’s daily functioning



**Figure 4: Updated DPAM application**

The DPAM application utilizes one service brick (DPAM Service Brick) that extracts data related to physical activity (e.g., normal everyday life activities such as walking, cycling, gardening etc.) and the IDSS Automatic Goal Setting reasoned, which provides information about the goal that has to be reached for the day. The application utilizes accelerometer data to analyze daily physical activity. The core of the application measures the daily physical activity of the user, using raw accelerometer data, as well as IMA (integration of accelerometer values) per 10 seconds intervals. Based on these measurements, the application offers a number of end points that deliver various information about the physical activity of the user. Snapshot of updated DPAM’s GUI is shown on the Figure above. In comparison with M12 delivery the following new features and components have been added:

- Threshold with daily goal in the timeline – provides information about the goal in terms of steps that the user have to achieve for the day;
- Activity type and duration section – provides the duration of each physical activity the user was performing. The supported physical activities are: Resting, Walking, Running and Exercising;
- One Service Brick for DPAM application with several endpoints – the refactored service brick utilizes offline calculations which in turns improves dramatically the responsiveness of the application.



**Figure 5: Updated DFM application**

The DFM application includes one service brick (DFM Service Brick) that extracts information relevant to daily functioning activities (e.g. normal everyday functionality such as cooking and

eating, showering, sleeping etc.). The monitoring comprises room occupancy using movement sensors, social activity in the living room using A/V sensors and sleep using bed sensor. A snapshot of application's graphical user interface is presented on Figure 5. In comparison with M12 delivery following new features and components have been added:

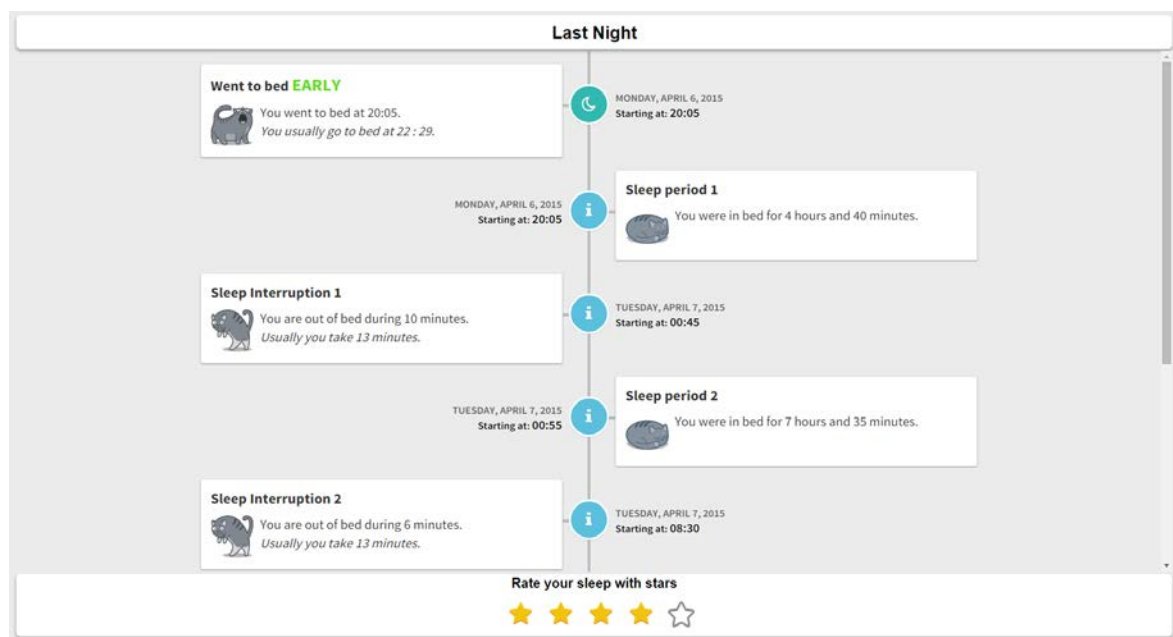
- More accurate presentation of users functionality - each functionality has start time, end time and duration;
- One Service Brick for DFM application with several endpoints – the service brick utilizes algorithms for better estimate of the functioning activities.

The Sleep application has the purpose of presenting quantitative and qualitative interpretation of the user's sleep behaviour based on the home sensing. This version of the application, shows an interpretation of last night sleep data.

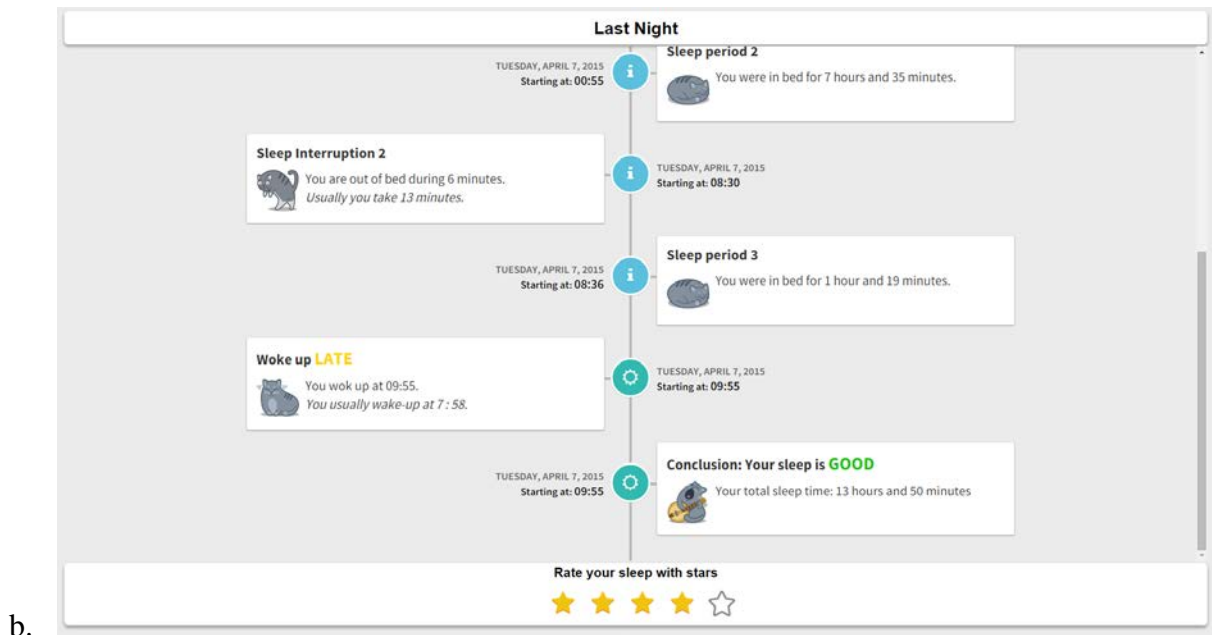
As described in the previous deliverable, the first sleep application prototype presented a short summary of quantitative detection of sleep from last night rendered into an easy to read newspaper like article, a set of tags containing adjectives, describing the falling asleep state of users' mind; the overall sleep quality and factors, which may influence it like outside temperature, dreams or nightmares, and the state of mind of the user when waking up. Also, on the summary page this leap application presented a five star rating system.

The user could choose to see a sleep details page where a rendering of the bed pressure sensor data was displayed in a vertical timeline.

Based on the usability tests performed in all four test sites, several redesign outcomes have been implemented. We will point out the most important changes. Figure 6 depicts the new interface for the sleep application.



a.



**Figure 6: The eWALL Sleep application redesigned interface, based on elderly user and expert user tests. (a) depicts the first view of the application, just after it was launched. (b) depicts the application offering an inferred conclusion about the last night user’s sleep quality.**

The sleep summary was perceived to be too long (some participants referred to it as a boring, tedious read) and difficult to grasp at a glance. This part was eliminated in the redesigned version.

Displaying information in a timeline fashion was highly appreciated due to its clarity and the main input what's the furthest and information encapsulated in each timeline item. We refer to timeline items as cards due to their visual representation. In the redesign session, the information in the cards was changed to:

- display the data interpretation of the last night sleep sensor detection into a meaningful sentence
- when possible, display a meaningful representation of the lifestyle reasoner output corresponding to the card nature.

Figure 7 depicts an example of such information processing, which encapsulates computations on sensor-based data provided by the service brake and computations on the outputs provided by the sleep life style reasoner.





**Figure 7: The “went to bed” card describing service brick data computation, sleep lifestyle reasoned output interpretation, computed quality of sleep related event (the user went to bed late, according to his usual behavior and personal preferences).**

Adapting to users personal preferences and sets of values, which in previous version was handled by the tag set, will be done through a series of daily and monthly questionnaires according to the validated procedure off the sleep diary used in current healthcare practice for evaluating subjective quality of sleep in persons home. This feature will be implemented once the user input mechanism can be handled by the eWALL back-end system.

The icons using this application triggered a generally positive reaction of the evaluated users, some showing emphatic links to them, therefore they were kept.

The outcomes from the service brake computation and the lifestyle reasoner output interpretation are represented through clear text. This reduces the memory load of the users, as opposed to interpreting data charts, or iconic representation of quantitative data. The text is short and concise, reducing the perception of “long boring read” as reported in the first prototype evaluation.

The five star rating system was considered intuitive for most test participants, and therefore it was kept. The star rating will provide input to calculating the overall sleep quality, displayed in the conclusion card (Fig. 7b.)

The integration will encompass daily and monthly questionnaires for assessing subjective parameters of sleep. The outcome of these questionnaires will specialize the lifestyle reasoner’s output, the computed quality of sleep related event and the general sleep overview with subjective measurements, which match today’s healthcare practices for evaluating sleep quality at home: the sleep diary. Further investigation on creating automatic sleep quality measurements including personal user values are to be performed.

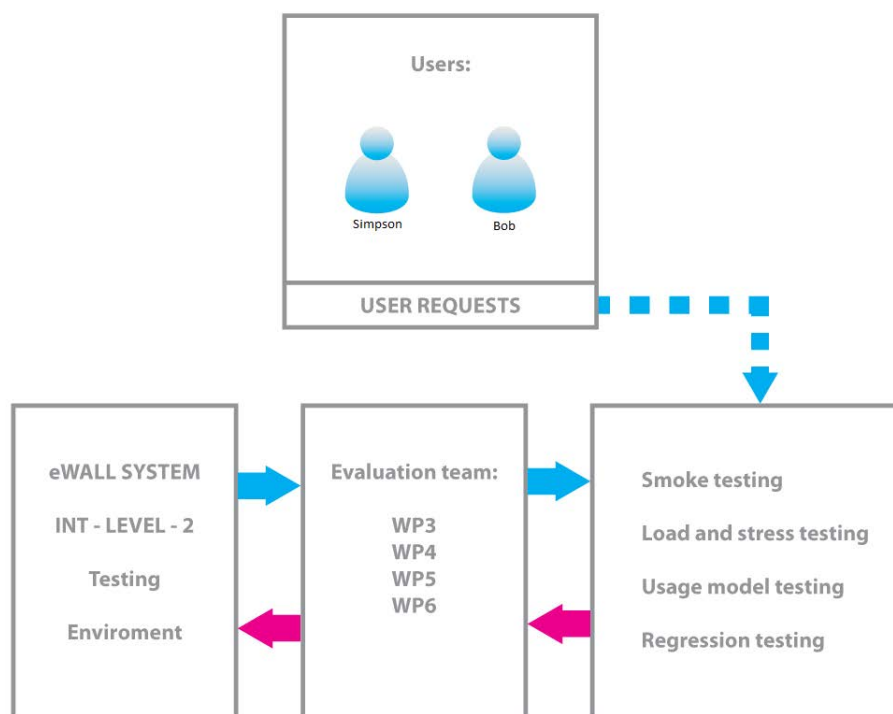
Further investigation in the added value of voice enabled feedback (the application would speak the content of each card when the user taps a speak button) will be performed.

## 4 Technical Evaluation Methodology

Technical evaluation methodology is a process which is required for reaching high quality of evaluation of the developed eWALL system. eWALL evaluation team has to evaluate the eWALL system by monitoring flow of information through the eWALL system. On the determined quality level, the eWALL Consortium has a possibility to improve services of eWALL system.

### 4.1 Evaluation Methodology Framework

Evaluation methodology framework is based on four entities: users, tests, evaluation team and eWALL test system. In the first phase, the virtual users (Simpson and Bob) were created. More users will be created for M24 Prototype. Their activities are monitored at testing environment, especially created for this purpose: virtual test bed int-level-2 environment. Evaluation team consists of team work of four work packages: WP3, WP4, WP5 and WP6. The team evaluated eWALL system according to evaluation methodology framework (shown in Figure 8). Different groups of tests (smoke tests, load and stress tests, usage model test and regression tests) were used for different users.



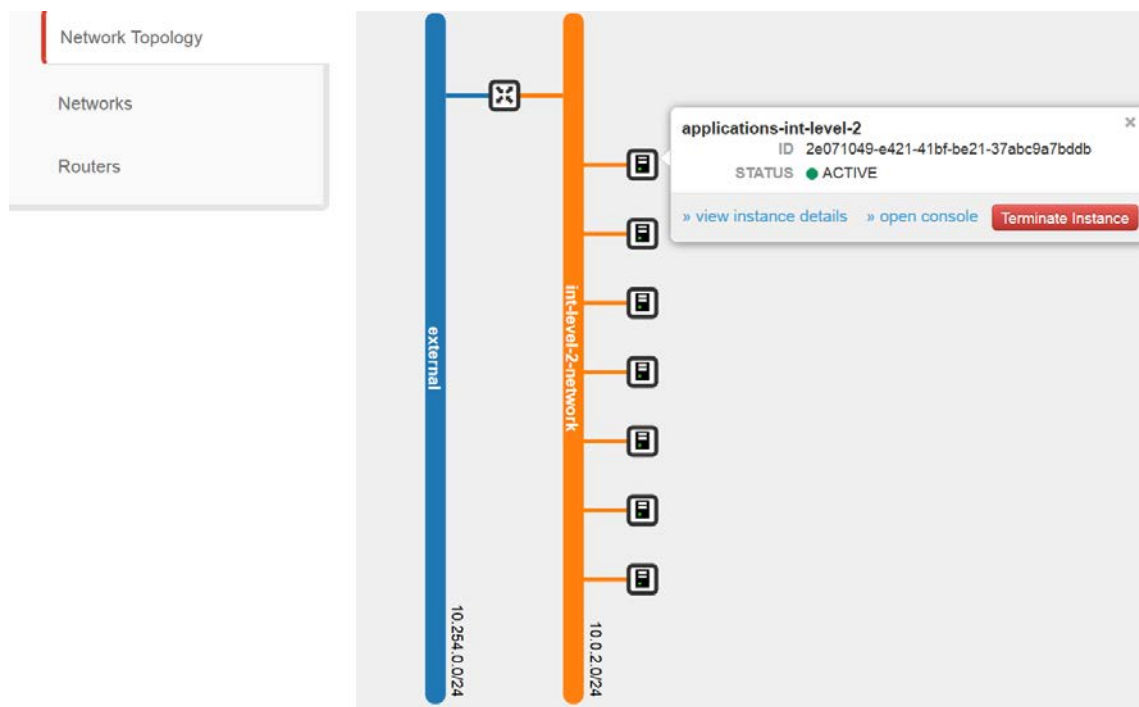
**Figure 8: eWALL evaluation framework**

eWALL OpenStack (Cloud operating system) int-level-2 environment has been established (Fig. 10.). Int-level-2 environment is a dedicated environment that is used for end-to-end testing of eWALL system. eWALL system components from development environment (eWALL OpenStack dev environment) have been deployed on int-level-2 environment to represent currently developed components of eWALL platform and applications in testing environment. On int-level-2 environment there are 7 virtual machines, but current system four of them: mongodb-int-level-2, home-int-level-2, platform-int-level-2 and applications-int-level-2 for end2end testing of eWALL

prototype. Home-int-level-2 virtual machine is used for deployment of device gateway, couchDB and remote proxy components. Platform-int-level-2 virtual machine is used for deployment of cloud gateway, eWALL portal, IDSS components, notification manager, profiling server and lifestyle reasoners. Application-int-level-2 virtual machine is used for deployment of eWALL DFM and DPAM Applications, eWALL Mainscreen, and eWALL portal. Instead of connecting real sensors to int-level-2 environment "Shadow Users" simulator has been used to populate couchDB with Simpson user data. This data is used further in the end-to-end system for verification of data flow from CouchDB to eWALL Applications (DFM, DPAM).

**Table 3: Int-level-2 virtual machines**

Virtual machines
Mysql-int-level-2
Mongodb-int-level-2
memcached-int-level-2
rabbitmq-int-level-2
home-int-level-2 and -b
platform-int-level-2
applications-int-level-2



**Figure 9: OpenStack int-level-2 environment**

## 4.2 Smoke Testing

Smoke tests represent a minimum build verification test and serve as a preliminary acceptance test. The most important functionality is quickly examined to determine if further tests are necessary or even possible. For the system to pass the smoke testing, it needs to be fundamentally functional and successfully started. Conducted scenarios for smoke testing fall into two groups:

- Deployment testing (verify successful deployment of proper versions),
- Functional testing (verify that every component is running and that basic functions are accessible).

**Table 4: Smoke testing**

Platform	
✓	Remote Proxy
✓	Cloud gateway
✓	eWALL portal
✓	IDSS
✓	Notification manager
✓	Profiling server
Applications	
✓	Daily functional monitoring (DFM)
✓	Daily physical activity monitoring (DPAM)
✓	Main screen
✓	Service brick – activity type duration
✓	Service brick – calories
✓	Service brick – inactivity detection
✓	Service brick – personal hygiene
✓	Service brick – room occupancy
✓	Service brick – sleep detection
✓	Service brick – social interaction detection
✓	Service brick – walking distance
✓	Service brick – weather

### 4.3 Usage Model Testing

The biggest group for integration testing is the usage model testing group (a big-bang system testing from the user perspective). The automated testing process assumes to be a regular user and takes all the regular actions that a user can take on the system. These tests are GUI based for the most part, so they can also inspect the GUI responsiveness and accessibility.

**Table 5: Usage model testing**

TEST	TEST CASE DESCRIPTION	TEST SCENARIO INPUT DATA	EXPECTED OUTCOME 1	RESPONSIBLE FOR TEST	TEST conducted
eWALL PORTAL					
✓	LOG IN	username: pAdmin password: admin	eWALL Portal is logged in	UOZ	Y
✓	Adding new region	Name:Zagreb Country:Croatia	New region is visible in eWALL Portal	UOZ	Y
✓	Delete region	Pressing delete/garbage button	region is deleted	UOZ	Y
✓	Available user's	Pressing USER tab	List of users	UOZ	Y
✓	Filling new user form	Pressing add new user	User detail form appear	UOZ	Y
✓	User edit profile	Pressing edit button	User detail form opens	UOZ	Y
✓	Region selection	Choosing region	Region details	UOZ	Y
✓	User role	Choosing role of user	User role administrative page	UOZ	Y
✓	Delete user	Pressing delete/garbage button	User deleted	UOZ	Y
✓	Adding new services	Pressing button new services	List of services	UOZ	Y
✓	Services edit	Pressing edit button	services details	UOZ	Y
✓	Services delete	Pressing delete/garbage button	Services deleted	UOZ	Y
✓	Available applications	Pressing application tab	List of applications	UOZ	Y

✓	Adding new applicationes	Pressing add new button	Application is added	UOZ	Y
✓	Application edit	Pressing edit button	applicationes details	UOZ	Y
✓	Application delete	Pressing delete/garbage button	application deleted	UOZ	Y
✓	Sensing environments	Pressing sensing environment tab	Listo of environment data	UOZ	Y
✓	Deleting/editing data	Pressing edit/delete button	edit/delete form	UOZ	Y
✓	New user creation platform administrator	New user form submission	No respond	UOZ	Y
✓	New user creation region administrator	New user form submission	No respond	UOZ	Y
✓	New user creation caregiver	New user form submission	No respond	UOZ	Y
✓	New user creation informal caregiver	New user form submission	No respond	UOZ	Y
✓	New user creation - primary user	New user form submission	No respond	UOZ	Y
Portal - Main screen Crossing					
✓	DFM application delete	Pressing delete/garbage button at eWALL portal	Delete visible at main screen	UOZ	Y
✓	Adding DFM application	Pressing add new application at eWALL portal	Application visible at main screen (with delay)	UOZ	Y
✓	DPAM delete	Pressing delete/garbage button at eWALL portal	Delete visible at main screen	UOZ	Y
✓	Adding DPAM application	Pressing add new application at eWALL portal	Application visible at main screen (with delay)	UOZ	Y
✓	Delete Daily Physical Activity	Pressing delete/garbage	Widget not visible at main	UOZ	Y

	Monitoring Widget	button at eWALL portal	screen		
	Adding widget DPAM	Pressing add new application at eWALL portal	Error during submit!Conflict		
Main Screen					
✓	DFM data display	Running DFM application	Displayed data related with user activity	UOZ	Y
✓	DPAM data display	Running DPAM application	Displayed data related with user activity	UOZ	Y

### 4.4 End-to-End Testing

End-to-end testing shows and tests functionality of eWALL system from user home environment to eWALL applications through developed platform. In the first testing phase, communication channel between user and profiling server was successfully tested. In the second test phase, communication between profiling server and eWALL portal was successfully tested. Communication between profiling server and DPAM application was successfully tested, as well. In further phases DFM application will be tested, as well as mood reasoner. End-to-end testing progress is shown on Fig. 11.

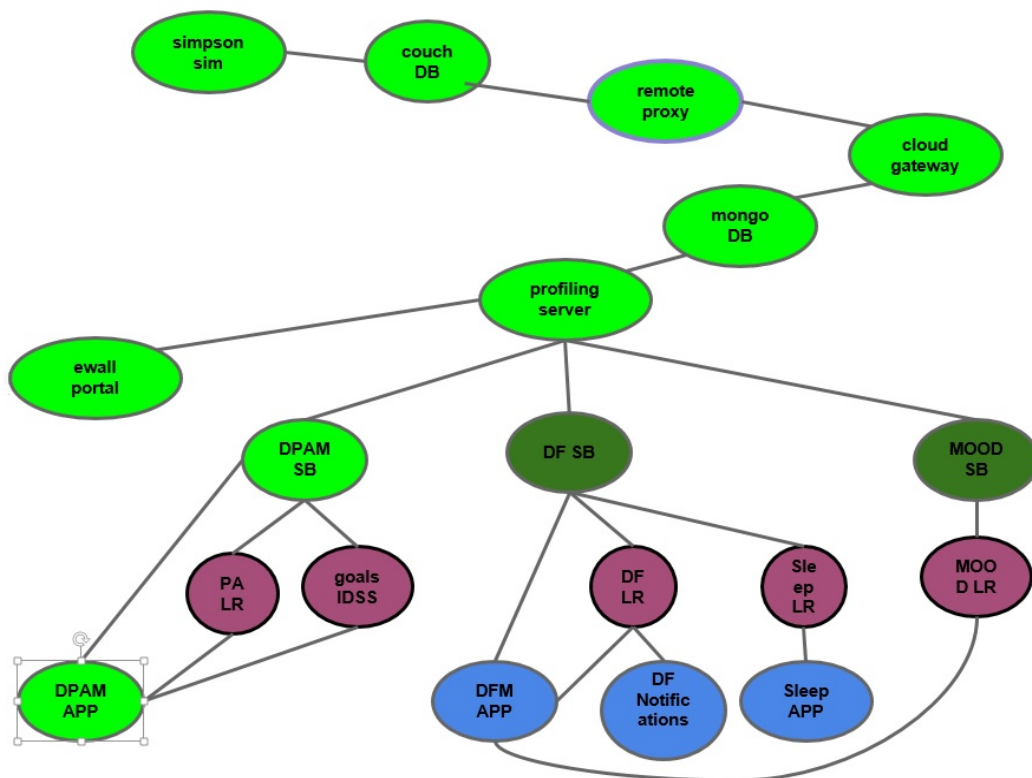


Figure 10: End-to-end testing process

Table 1. shows end-to-end testing that was performed for Simpson user on int-level-2 environment, starting from 2015-03-01T20:00:00.000+01:00 and lasting until 2015-03-02T20:00:00.000+01:00. Data for end-to-end testing was simulated by "Shadow User" simulator and stored in the local CouchDB. Starting date of the "Shadow User" simulator for Simpson user was set to 01.03.2015. This testing was performed for all eWALL components between DPAM Application on eWALL main screen available through Internet HTTP request and CouchDB that is mounted on a computer in eWALL Home Environment. The goal of end-to-end testing was to show consistency of data flow through the whole eWALL system from eWALL sensing environment to data display on eWALL DPAM Application. Consistency of data was checked on every component, which data passes from Sensing Environment to eWALL DPAM Application. Consistency was checked through all components for one chosen time through timespan of one day. End-to-end testing was performed for simulated time period of one day and data was verified through following eWALL components:

- DPAM Application
- DPAM SB
- PA LR
- Goals IDSS
- Profiling Server
- MongoDB
- Cloud Gateway
- Remote Proxy
- CouchDB

In the simulated time period from 01.03.2015 20:00 hours to 02.03.2015 20:00 hours, data in all eWALL components for Simpson user were correct with respect to time and component that produces that data. During eWALL end-to-end testing procedure, every system component was authorized by token generated by eWALL Token Server for Simpson user. This showed security of eWALL components form perspective of multiuser data generation by every eWALL component.

**Table 6: End-to-end data flow verification**

Testing time	Application	Service Brick	Reasoners	Profiling Server	Cloud Gateway	Remote Proxy	MongoDB	CouchDB	User
	DPAM	DPAM SB	PA LR						Simpson
			goals IDSS						
<b>2015-03-01T20:00:00.000+01:00</b>	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	
<b>2015-03-02T20:00:00.000+01:00</b>	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	Data verified	



## 5 Conclusion

Technical evaluation report showed overall status of integration as well as functionalities of the developed eWALL M18 prototype. Progress testing was conducted taking into consideration one user named Simpson. For the second user, named Bob, testing was prepared. However, due to difficulties related to the complexity of eWALL system, namely: difficulties related to database, data transfer, and application data retrieval, testing for Bob is delayed for the next phase, meaning M21.

Technical evaluation methodology framework was examined by using three groups of tests: the first group of tests, named smoke tests, are automated tests, with which the deployment of the components was tested (e.g., eWALL platform components, applications components); the second group of tests, named usage model tests, helped to examine functionality of cross-sections between different components (e.g., eWALL portal-main screen); the third group of tests, named end-to-end tests, confirmed data stream flow from the user, as the starting point, and applications, as the final point in the eWALL communications chain.

The developed “Shadow User” simulator limits was useful for the first series of end-to-end testing; however it limits testing capacity of unpredictable real user environment. Namely, tests were conducted in the given, simulated and controlled conditions. Thus, in the next period (M19-M24) the main task of WP6 is to generate the closest possible environment of the “real user world”, meaning that before small-scale evaluation, it is necessary to replace the “Shadow User” simulator by real hardware and software in the user environment.

Except the previously mentioned three groups of tests, in the next period, the fourth group of tests will be launched, named load and stress testing. This group of tests will be based on creation of multiuser environment in different countries, in order to challenge capacity and throughput of eWALL system.

In parallel with the development of load and stress testing, further end-to-end testing, prepared for user Bob, will be continued. This will include further bug fixing in the eWALL system.

The whole series of tests and further work on eWALL, carried together with integration process, will be reflected in the improved and final M24 prototype, which will enable successful and meaningful small scale evaluation.

## 6 Bibliography

- [1] The eWALL Consortium, “D3.1.2, eWALL devices and networked infrastructure,” eWALL for Active Long Living FP7 project, April 2015.
- [2] The eWALL Consortium, “D3.2.2., eWALL gateway,” eWALL for Active Long Living FP7 project, April 2015.
- [3] The eWALL Consortium, “D3.3.2., Perception from sensors (non-A/V),” eWALL for Active Long Living FP7 project, April 2015.
- [4] The eWALL Consortium, “D3.4.2., Audio/Visual perception,” eWALL for Active Long Living FP7 project, April 2015.
- [5] The eWALL Consortium, “D3.5.2., eWALL configurable metadata streams,” eWALL for Active Long Living FP7 project, April 2015.
- [6] The eWALL Consortium, “D4.1.2., Semantic modelling of eWALL middleware services,” eWALL for Active Long Living FP7 project, April 2015.
- [7] The eWALL Consortium, “D4.2.2., Intelligent support system for eWALL,” eWALL for Active Long Living FP7 project, April 2015.
- [8] The eWALL Consortium, “D4.3.2., Cloud middleware services for eWALL,” eWALL for Active Long Living FP7 project, April 2015.
- [9] The eWALL Consortium, “D3.3.2., Perception from sensors (non-A/V),” eWALL for Active Long Living FP7 project, April 2015.
- [10] The eWALL Consortium, “D4.4.2., eWALL Platform,” eWALL for Active Long Living FP7 project, April 2015.
- [11] The eWALL Consortium, “D5.1.2., eWALL metadata from the outside: filtered and retrieved,” eWALL for Active Long Living FP7 project, April 2015.
- [12] The eWALL Consortium, “D5.2.2., Lifestyle reasoning: fusion for activities, situation and their patterns,” eWALL for Active Long Living FP7 project, April 2015.
- [13] The eWALL Consortium, “D5.3.2., Intelligent context-aware services & applications, ” eWALL for Active Long Living FP7 project, April 2015.

## Annex I : End-to-End Testing

TEST CASE DESCRIPTION	TEST SCENARIO INPUT DATA	EXPECTED OUTCOME	RESPONSIBLE FOR TEST	TEST DATE	TEST conducted
functioning activity controller	username: simpson from: 2015-03-01T20:00:00.000+01:00 to: 2015-03-03T20:00:00.000+01:00 latestevents: 2	"username": "simpson", "timestamp": "2015-03-03T18:02:50.000+01:00", "functioningActivity": "showering"	UOZ	29.4.	Y
inbed controller	username: simpson from: 2015-03-01T20:00:00.000+01:00 to: 2015-03-03T20:00:00.000+01:00 latestevents: 2	No data	UOZ	29.4.	Y
location controller	username: simpson from: 2015-03-03T20:00:00.000+01:00 to: 2015-03-05T20:00:00.000+01:00 latestevents: 2	username: simpson timestamp: 2015-03-05T13:13:40.000+01:00 location: livingroom	UOZ	29.4.	Y
inactivity controller	"username": "", "from": "", "to": "", "latestEvents": 0, "inactivityEvents":	"username": "", "timestamp": "", "inactive": false	UoZ	28.4.	Y
movement controller	"username": "", "from": "", "to": "", "aggregation": { "length": 0, "unit": ""	"movements": [ { "from": "", "to": "", "steps": 0, "kilometers": 0, "burnedCalories": 0 }	UoZ	28.4.	Y
physical activity controller	{ "username": "", "from": "", "to": "", "latestEvents": 0, "physicalActivityEvents": [	{ "username": "", "timestamp": "", "activityType": "" }	UoZ	28.4.	Y
weather					
mood controller	username: simpson from: 2015-03-01T20:00:00.000+01:00 to: 2015-03-	"timestamp": "2015-03-02T22:00:20.000+01:00", "valence": -0.4063,	UOZ	UOZ	Y

	03T20:00:00.000+01:00 latestevents: 2	"arousal": -0.7817			
Socializing controller	username: simpson from: 2015-03-01T20:00:00.000+01:00 to: 2015-03-03T20:00:00.000+01:00 latestevents: 2	"username": "simpson", "from": "2015-03-01T22:00:20.000+01:00", "to": "2015-03-02T22:00:20.000+01:00", "socializingEvents": []	UOZ	UOZ	Y
Accelerometer Measurements Controller	username: simpson from: 1425341510000 to:1425341520000	No data	UOZ	29.4.	Y
Accelerometer Measurements Controller	username: simpson timestamp: 1425341510000	No data	UOZ	29.4.	Y
activity sensing controller	username: simpson from: 1425164480000 to: 1654123124356	"constantQuantityMeasureType": "Accelerometer_measure" "timestamp": 1425168250000, "imaValue": 0.10929538007371187, "isaValue": 0, "steps": 0, "activityType": "RESTING", "fallDetected": false,	UOZ	28.4	Y
Door status controller	username: simpson from: 1425164480000 (ms) to: 1525164480000 room: "kitchen"	"measuredValue": "true", "constantQuantityMeasureType": "DOOR_STATUS_OPEN", "timestamp": 1425168000000, "indoorPlaceName": "kitchen",	UOZ	28.4.	Y
Environment sensing controller - HUMIDITY	username: simpson from: 1425164482133 to: 2099349222828 room: livingroom, kitchen,	"measuredValue": "55.5", "constantQuantityMeasureType": "HUMIDITY_MEASURE", "timestamp": 1425168000000, "indoorPlaceName": "kitchen",	UOZ	28.4.	Y
Environment sensing controller - Illuminance	username: simpson from: 1425196800000 to: 2325196800000 room: bathroom, livingroom, bed room, kitchen	"measuredValue": "610.6283743124628", "constantQuantityMeasureType": "ILLUMINANCE_MEASURE", "timestamp": 1425196830000, "indoorPlaceName": "livingroom",	UOZ	28.4.	Y
Environment sensing controller - Movement	username: simpson from: 1425196800000 to: 4234543543545 room: bathroom, livingroom, kitchen	"measuredValue": "false", "constantQuantityMeasureType": "MOVEMENT_MEASURE", "timestamp": 1425284700000, "indoorPlaceName": "bathroom",	UOZ	28.4.	Y

Environment sensing controller - Temperature	username: simpson from: 1425196800000 to: 3525284700000 room: bathroom, livingroom, kitchen	"measuredValue": "19.757084925999937", "constantQuantityMeasureType": "Temperature_Measure_celsius", "timestamp": 1425196850000, "indoorPlaceName": "bathroom",	UOZ	28.4.	Y
Gases Measurements Controller - Carbon Monoxide	username: simpson from: 1425196800000 to: 4021312392132 room: bathroom, livingroom, kitchen, bedroom	"measuredValue": "0", "constantQuantityMeasureType": "Carbon_Monoxide measurement", "timestamp": 1425205200000, "indoorPlaceName": "bedroom",	UOZ	28.4.	Y
Gases Measurements Controller - LPG	username: simpson from: 1425196800000 to: 4425232400000 room: bathroom, livingroom, kitchen, bedroom	"measuredValue": "0", "constantQuantityMeasureType": "Liquified_Petroleum_measurement", "timestamp": 1425196850000, "indoorPlaceName": "bathroom",	UOZ	28.4.	Y
Gases Measurements Controller - NG	username: simpson from: 1425232400000 to: 35231231231231 room: bathroom, livingroom, kitchen, bedroom	"measuredValue": "0", "constantQuantityMeasureType": "Natural_Gas measurement", "timestamp": 1425242780000, "indoorPlaceName": "bedroom",	UOZ	28.4.	Y
GPS coordinates controller	Username: simpson	No data C2 phase	UOZ	28.4.	Y
GPS coordinates name	username: simpson name:	No data C2 phase	UOZ	28.4.	Y
GPS coordinates timestamp	username: simpson timestamp: 1425232400000	No data C2 phase	UOZ	28.4.	Y
Health sensing controller	username: simpson from: 1425232400000 to: 35231231231231	No data C2 phase	UOZ	28.4.	Y
Health sensing heart rate	username: simpson from: 1425232400000 to: 35231231231231	No data C2 phase	UOZ	29.4.	Y
Health sensing oxygensaturation	username: simpson from: 1425232400000 to: 35231231231231	No data C2 phase	UOZ	29.4.	Y
Humidity measurement controller timestamp	username: simpson from: 1425197400000 to: 2425197400000 room: bedroom, kitchen, livingroom, bathroom	"measuredValue": "50.88437169226227", "constantQuantityMeasureType": "HUMIDITY_MEASURE", "timestamp": 1425220730000, "indoorPlaceName":	UOZ	28.4.	Y

		"bathroom",			
Humidity measurment controller timestamp/timest amp	username: simpson timestamp: 1425298400000	"measuredValue": "53.69761760590162", "constantQuantityMeasureType ": "HUMIDITY_MEASURE", "timestamp": 1425298400000, "indoorPlaceName": "livingroom"	UOZ	29.4	Y
Illuminance measurment control timestamp	username: simpson from: 1425197400000 to: 2425197400000 room: kitchen, bedroom, living room, bathroom	"measuredValue": "83.3379197736935", "constantQuantityMeasureType ": "Illuminance_measure", "timestamp": 1425350580000, "indoorPlaceName": "bedroom",	UOZ	29.4.	Y
Illuminance measurment control timestamp/timest amp	username: simpson Timestamp: 1425197400000	{ "measuredValue": "628.4005278988827", "constantQuantityMeasureType ": "ILLUMINANCE_MEASURE ", "timestamp": 1425197430000, "indoorPlaceName": "bathroom",	UOZ	29.4.	Y
Presure mattress sensor controller	username: simpson from: 1425196800000 to: 2621321312123 room	No respond	UOZ	29.4.	Y
movement measurmenets controll timestamp	username: simpson from: 1421321312123 to: 4621321312123 room: kitchen, bedroom, living room, bathroom	"measuredValue": "false", "constantQuantityMeasureType ": "MOVEMENT_MEASURE", "timestamp": 1425341320000, "indoorPlaceName": "livingroom"	UOZ	29.4.	Y
movement measurmenets controll timestamp/timest amp	username: simpson timestamp: 1425341520000	"measuredValue": "false", "constantQuantityMeasureType ": "MOVEMENT_MEASURE", "timestamp": 1425341520000, "indoorPlaceName": "kitchen",	UOZ	29.4.	Y
temperature measurment controller timestamp	username: simpson from: 1425341520000 to: 3425341520000 room:	"measuredValue": "18.561585738039508", "constantQuantityMeasureType ": "Temperature_measure_celsius" "timestamp": 1425348400000, "indoorPlaceName": "bedroom",	UOZ	29.4.	Y
temperature measurment controller timestamp/timest amp	username: simpson timestamp:1425341510000	"measuredValue": "20.399932556721666", "constantQuantityMeasureType ": "Temperature_measure_celsius"	UOZ	29.4.	Y

		"timestamp": 1425341510000, "indoorPlaceName": "bedroom",			
visual sensing controller	username: simpson from: 1425341510000 to: 1425341520000	No data	UOZ	29.4.	Y
ActivityWeekPattern	userid (string) date (string)	{ "user": "", "weekDay": 0, "weightedAverage": 0, "category": "", "updated": { "chronology": { "zone": { "fixed": false	UOZ	29.4.	Y
LastUpDate	userID	"value": { "user": "", "date": { "chronology": { "zone": { "fixed": false } }, },	UOZ	29.4.	Y
LastUpDate	userID	no respond	UOZ	29.4.	Y
sleep week pattern	userid (string) date (string)	no respond	UOZ	29.4.	Y
activity goal	userid (string) date (string)	no respond	UOZ	29.4.	Y
activity goals	userid (string) date (string)	no respond	UOZ	29.4.	Y

## Annex II: Platform testing

TEST DESCRIPTION	TEST SCENARIO INPUT DATA / PREREQUISITES	EXPECTED OUTCOME 1	EXPECTED OUTCOME 2	EXPECTED OUTCOME 3	TESTING ENVIRONMENT	RESPONSIBLE FOR TEST	TEST CONDUCTED
Starting Remote Proxy and checking successful authentication and registration	RemoteProxy properly configured in config.properties file (proper sensing environment id, root, and local couch db parameters). Used sensing environment and user must be provisioned at eWALL system via eWALL Portal. Local CouchDB started. Entered valid user credentials in console after starting Remote Proxy	OK: "Authenticated successfully." and "Successfully registered sensing environment x:x:x:x to eWALL system y:y:y:y" message in console, or by sens env status through eWALL Portal			Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub.ro/platform-dev/).	ENT	Yes, April 2015
Adding new environmental sensing data to local Couch DB, data adapted and sent to cloud Mongo DB via Remote Proxy and Cloud Gateway, data access via Profiling	Remote Proxy successfully authenticated and registered at Cloud GW, and listening to local Couch DB changes. New document with environmental sensing data added to local Couch DB, for example: { "_id" : "2014-09-01T00:00:00.000Z", "_rev" : "1-b34306f2f0344672d653f5b5c7df711c", "movement" : false, "illuminance" : 2.0464407112347436, "temperature" : 19.40782989444393, "humidity" : 52.07199253060395, "NG" : 1, "CO" : 2, "LPG" : 1, "door_open" : true,	OK: response status 201 in console	OK: New data entry at cloud MongoDB in "tempMeasCollection", "movementMeasurementsCollection", "illumMeasCollection", "humidityMeasurementsCollection", "ngMeasCollection", "lpgMeasCollection", "coMeasCollection", "doorStatusCollection",	OK: Data visible via Profiling Server environmental data API	Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub.ro/platform-dev/). Tested both directly (with simulated sensor data from simulator, and indirectly (by manually	ENT/AIT	Yes, April 2015



Server	"timestamp" : "2014-09-01T00:00:00.000Z" }				adding data to local CouchDB)		
Adding new wearable sensing data to local Couch DB, data adapted and sent to cloud MongoDB via Remote Proxy and Cloud Gateway, data access via Profiling Server	<p>Remote Proxy successfully authenticated and registered at Cloud GW, and listening to local Couch DB changes. New document with environmental sensing data added to local Couch DB, for example:</p> <pre>{ "_id": "2014-09-01T12:29:10.000Z",   "_rev": "1-156f011ef2ecd6643a089eb61bc0b24e",   "activity": {     "IMA": 0.1112141600593139,     "ISA": 0.1112141600593139,     "steps": 5473,     "physicalActivity": "WALKING" },   "fall": false,   "timestamp": "2014-09-01T12:29:10.000Z" }</pre>	OK: response status 201 in console	OK: New data entry at cloud MongoDB in "accMeasCollecton"	OK: Data visible via Profiling Server activity sensing API	Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub.ro/platform-dev/). Tested both directly (with simultoed sensor data from simualtor, and indirectly (by manually adding data to local CouchDB)	ENT/AIT	Yes, April 2015
Adding new visual sensing data to local Couch DB,	<p>Remote Proxy successfully authenticated and registered at Cloud GW, and listening to local Couch DB changes. New document with environmental sensing</p>	OK: response status 201 in console, new data entry at cloud MongoDB in "VisualSensing", data visible via Profiling Server			Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud	ENT/	Yes, April

data adapted and sent to cloud Mongo DB via Remote Proxy and Cloud Gateway, data access via Profiling Server	data added to local Couch DB, for example: { "_id": "2014-09-01T12:29:10.000Z", "_rev": "1-156f011ef2ecd6643a089eb61bc0b24e", "people": [ { "trackID": 0, "x": 400, "y": 340, "width": 40, "height": 40, "positionConf": 0.9, "gender": "MALE", "genderConf": 0.9, "age": 70, "ageConf": 0.8, "emotion": "NEUTRAL", "emotionConf": 0.7 } ], "timestamp": "2014-09-01T12:29:10.000Z" }	visual sensing API			Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub.ro/platform-dev/). Tested both directly (with simualtoed sensor data from simualtor, and indirectly (by manually adding data to local CouchDB)	AIT	2015
Adding new furniture sensing data to local Couch DB, data adapted and sent to cloud Mongo DB via Remote Proxy and Cloud Gateway, data access via Profiling	Remote Proxy successfully authenticated and registered at Cloud GW, and listenining to local Couch DB changes. New document with environmental sensing data added to local Couch DB, for example: { "_id": "2014-09-01T00:00:10.000Z", "_rev": "1-80ede81818d8a45211212921ae6749a7", "pressure": true, "IMA": 0.08626862285226562, "timestamp": "2014-09-01T00:00:10.000Z" }	OK: response status 201 in console, new data entry at cloud MongoDB in "pressSensingCollection", data visible via Profiling Server mattress sensing API			Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platfrom components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub.ro/platform-dev/). Tested both directly (with simualtoed sensor data from simualtor,	ENT/ AIT	Yes, April 2015

g Server					and indirectly (by manually adding data to local CouchDB)		
Adding new medical sensing data to local Couch DB, data adapted and sent to cloud Mongo DB via Remote Proxy and Cloud Gatewa y, data access via Profilin g Server	Remote Proxy successfully authenticated and registered at Cloud GW, and listening to local Couch DB changes. New document with environmental sensing data added to local Couch DB, for example: { "_id": "2014-09- 01T00:00:10.000Z", "_rev": "1- 80ede81818d8a4521121 2921ae6749a7", "SPO": 15, "BPM": 60, "timestamp": "2014-09- 01T00:00:10.000Z" }	OK: response status 201 in console, new data entry at cloud MongoDB in "HeartRateMeasure ment" and "OxygenSaturation Measurement", data visible via Profiling Server healthsensing API			Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub .ro/platform- dev/). Tested by manually adding data to local CouchDB.	ENT	Yes, April 2015
Stoppin g Remote Proxy	Remote Proxy successfully authenticated and registered at Cloud GW	OK: sens env status through eWALL Portal will be offline.			Remote Proxy, local CouchDB, deployed on home PC. eWALL Cloud Platform components, namely Cloud Gateway, Mongo DB and Profiling Server deployed on Open Stack environment running on UPB server (ewall.radio.pub .ro/platform- dev/).	ENT	Yes, April 2015