"This project has been funded under the fifth AAL call, AAL-2012-5. This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein"

PROJECT Nº: AAL-2012-5- 232

D3.1 SALIG++ COMPONENTS TEST

Start Date of Project : 01/06/2013 **Duration :** 36 months

PROJECT FUNDED BY THE AAL JOINT PROGRAMME	
Due date of deliverable	M10, M21, M33
Actual submission date	2014-11-30
Organization name of lead contractor for this deliverable	PIA
Author(s)	PIA
Participant(s)	
Work package	WP3. System Integration, testing and evaluation
Comments	
Classification	
Version	Version 1
Total number of pages	21

DISCLAIMER

List of Authors

Partner	Authors
PIA	Arkadiusz Perski, Jacek Mickiewicz

Table of Contents

1. Introduction

The present document describes results of the tests from initial integration work done within scope of activity of WP3. The objective of this deliverable is to present results of so far tested components of SALIG++ system, especially functionalities of:

- “Safety at home” subsystem,
- Videoconferencing subsystem,
- Pose recognition subsystem,

Subsystems mentioned above were installed and configured in PIAP facility especially dedicated for preparation of system integration. The facility was equipped with required IT infrastructure (1 Gbps local network, 20 Mbps uplink/downlink Internet connection) and PC computers. For “Safety at home” scenario testing, provided by PIA set of safety sensors (main control unit, movement detectors, fire detector, leaking gas detector, temperature detectors, “open-door” detectors together with remotely controlled water valve) were set up. Preconfigured by DIV equipment (Raspberry Pie units, HD USB cameras, others) was installed and served as video conferencing subsystem. Provided by TUD equipment for pose recognition (MS Kinect sensor) was also installed. All subsystems were appropriately tested.

2. Objectives and methods

The components test aims to verify the basic functionalities of SALIG++ subsystems for their compliance with the requirements described in D1.2 “Technical specification and interaction design”. Properly working and interacting between themselves subsystems will allow for realization of End User scenarios fully defined in D1.1 “Scenarios, Use Cases, and Requirements”. Tests done during Year 1 of the Project were conducted to:

- Obtain basic functionality of subsystem for “Safety at home” scenario realization, for ensuring personal safety related to external risks (i.e. detected fire) within senior apartment,

- Set up working locally video conferencing subsystem, enabling audio/video communication in both directions: from caregiver web portal to senior TV set as well as in the opposite direction,

- Acquire working pose recognition subsystem, providing to the system information about actual pose of senior and his/her both potentially dangerous activity (i.e. fall detection) and routine activity (i.e. taking medicines on time, eating)

Performed tests have put stress on the basic functionality of each separate component. More comprehensive tests including subsystems performance, comprising also interaction functionality between different subsystems, will be performed in the next year of the project.

2.1 “Safety at home” subsystem test

Subsystem was installed in PIA premises laboratory consisting two rooms. PIA provided all elements of the subsystem, including:

- PC server (with preinstalled VMware environment) and laptop with Windows 7 64bit

- Set of sensors:

    - Wireless Fire detector

    - Wireless Leaking gas detector

  - Wireless Flood detector
  - 3 pcs. of wireless movement detectors
  - 2 pcs. of wire movement detectors
  - “Open-door” detector
  - Wireless movement detector with embedded camera
  - Wrist emergency button
  - Wall programmable button
  - Two-channel temperature sensor
- Wireless water valve – serving as an example of subsystem actuator
- Main Control Unit – equipped with wireless modem, GSM/UMTS modem, Ethernet card, USB 2.0 connector
- Moxa Nport server (4xRS-232/485, 1x Ethernet)

Subsystem elements were mounted on the laboratory walls, ceiling and on water installation. Then, after pre-configuration done in dedicated software, they have become part of fully operated safety subsystem.

For appropriate maintenance of information coming from separate sensors, a Java application has been developed by PIA and tested locally. The most important functionalities of the application were presented below:

- Querying Main Control Unit about list of active sensors and actuators
- Processing information with status of each sensor
- Providing messages to Main Control Unit for controlling connected to the system actuators
- Ensuring protocol translation between local Distributed Sensors platform and MediaSense platform common for each elements of SALIG++ systems (JSON format)

Pictures of PIA Integration laboratory were presented below. Red arrows present some elements of the subsystem:

Fig. 1. PIA Integration Site – Room 1 – main entrance to the laboratory, wireless flood detector, wireless movement sensor, wire movement sensor, open-door detector.

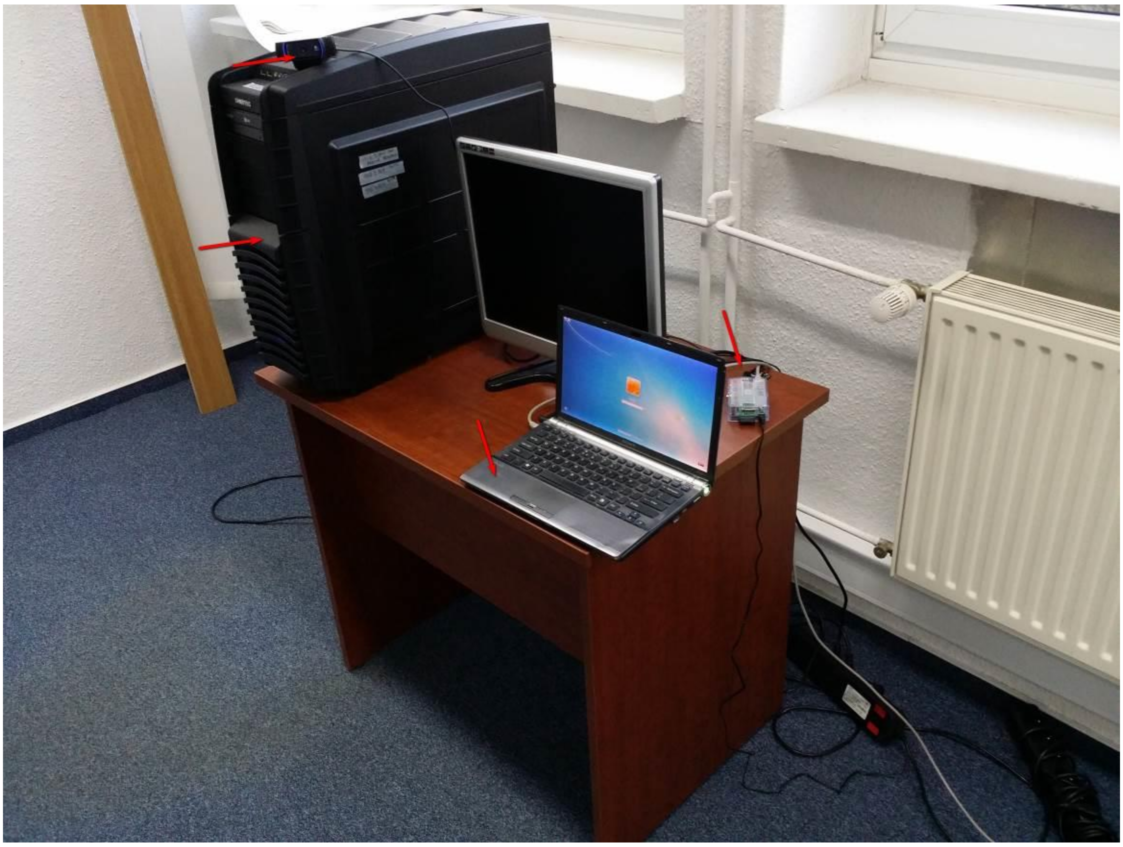


Fig. 2. PIA Integration Site – Room 1 – video server, caregiver laptop, Raspberry Pie unit, Logitech HD camera.

Fig. 3. PIA Integration Site – sink equipped with flood detector working with wireless transmitter.

Fig 4. PIA Integration Site – Room 2 – Pose recognition server, Main Control Unit, Moxa Nport server, wireless & wire movement detectors, MS Kinect, Raspberry Pie unit, Logitech HD camera, wrist emergency button, wall programmable button.

Some elements of "Safety at home" subsystem communicate with the Main Control Unit with the use of wired bus, however majority of sensors utilize wireless communication. Both of them require pre-configuration process, available in dedicated software.

Pictures below presents some print screens done during pre-configuration procedure:

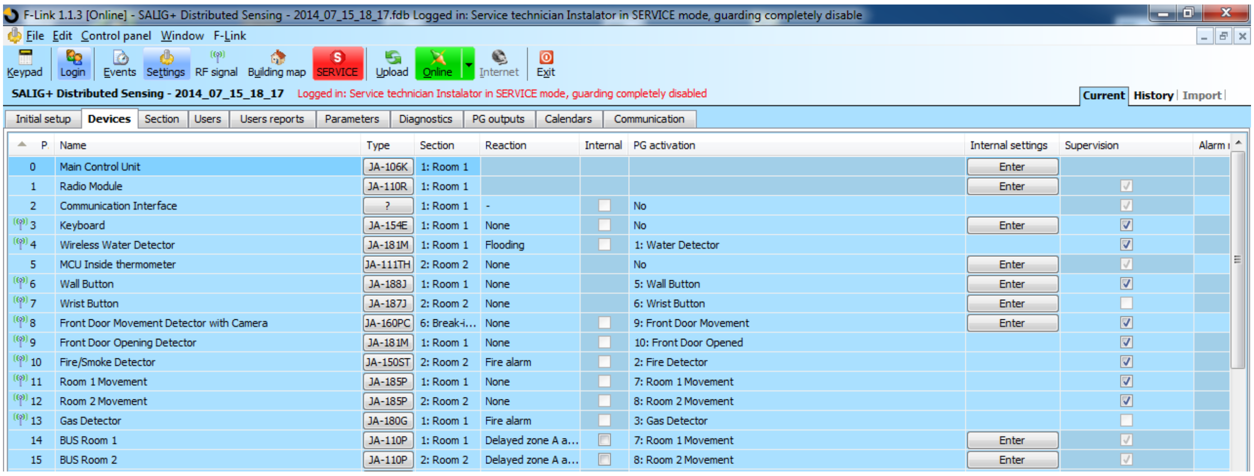


Fig. 5 List of available sensors and actuators in the laboratory.

Fig. 6 Diagnostic window – with operational status of each element.

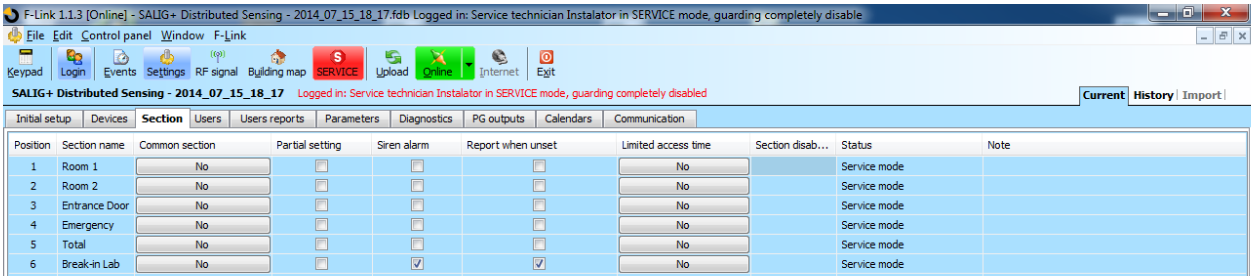
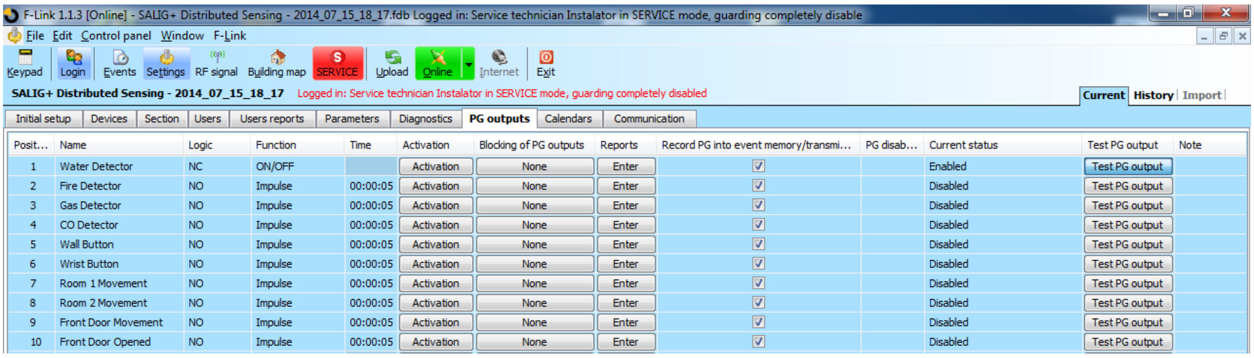


Fig. 7 List of section in laboratory – alarms from each section provide different information to the system.



Posit...	Name	Logic	Function	Time	Activation	Blocking of PG outputs	Reports	Record PG into event memory/transmi...	PG disab...	Current status	Test PG output	Note
1	Water Detector	NC	ON/OFF		Activation	None	Enter	☑		Enabled	Test PG output	
2	Fire Detector	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
3	Gas Detector	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
4	CO Detector	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
5	Wall Button	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
6	Wrist Button	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
7	Room 1 Movement	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
8	Room 2 Movement	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
9	Front Door Movement	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	
10	Front Door Opened	NO	Impulse	00:00:05	Activation	None	Enter	☑		Disabled	Test PG output	

Fig. 8 List of pre-programmed actions for threats detected by each sensor.

According to predefined actions separately prepared for particular scenario, functionalities of subsystem were investigated. Some actions connected with sensors/actuators specific functionality were tested and confirmed. Due to predefined actions, gathered information is provided to MediaSense platform, where DB Reasoner processes it and make appropriate decisions for caregivers (like presentation of pop-up on caregiver portal). Independently from taken by Reasoner decisions, the "Safety at home" subsystem offers instantaneous both: notification of caregiver by i.e. SMS text messages or e-mails and actions for installed actuator (like switching off valve water). This functionalities were also tested during Year 1 of project.

As an example, actions for several algorithms were presented below:

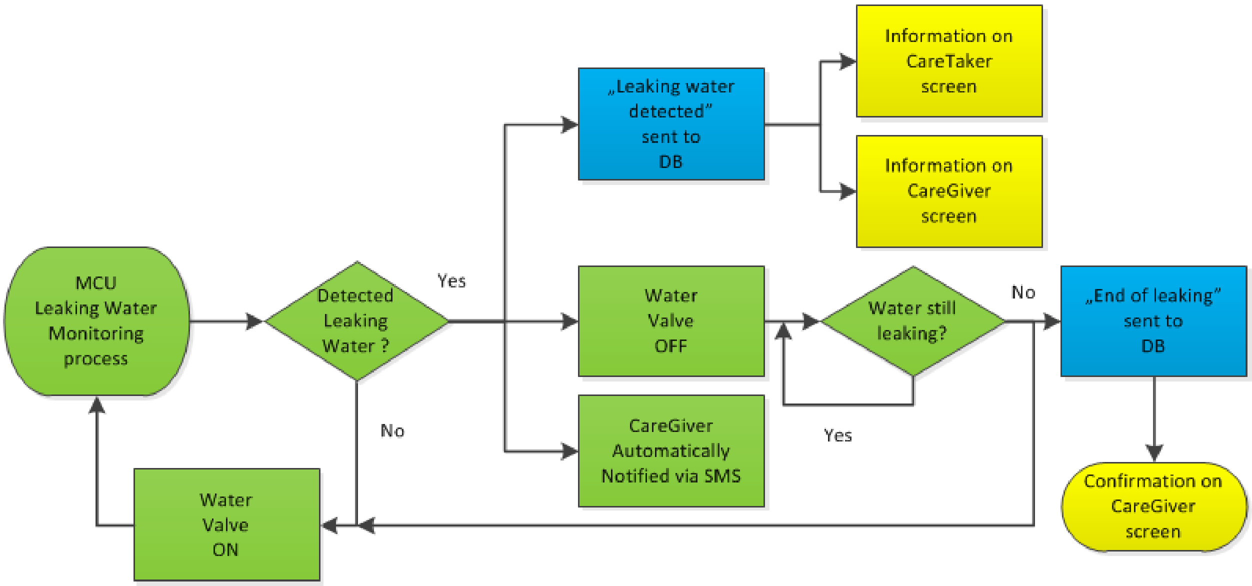


Fig. 9 Actions dedicated for detected flood in senior apartment.

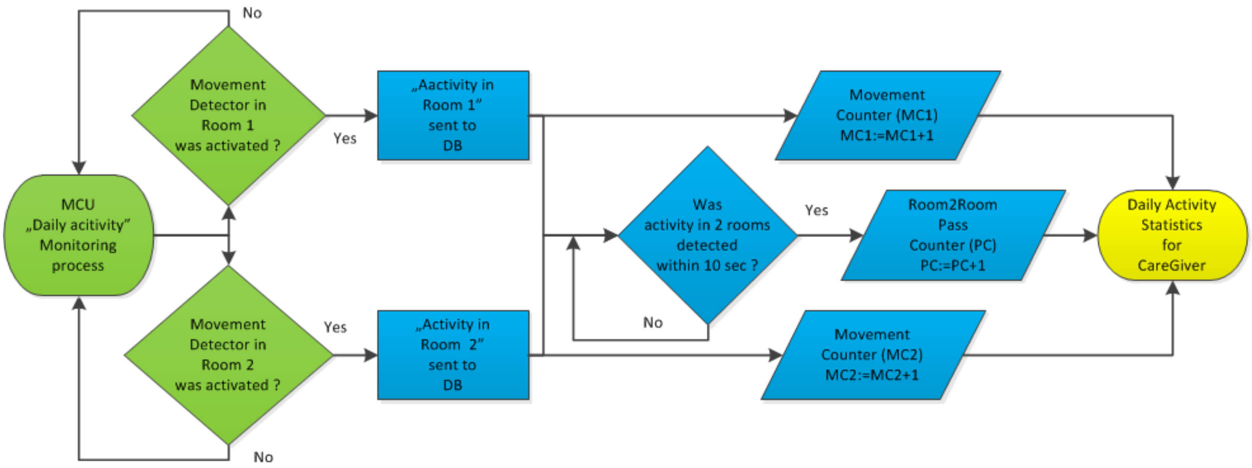


Fig. 10 Action dedicated for activity statistics preparation.

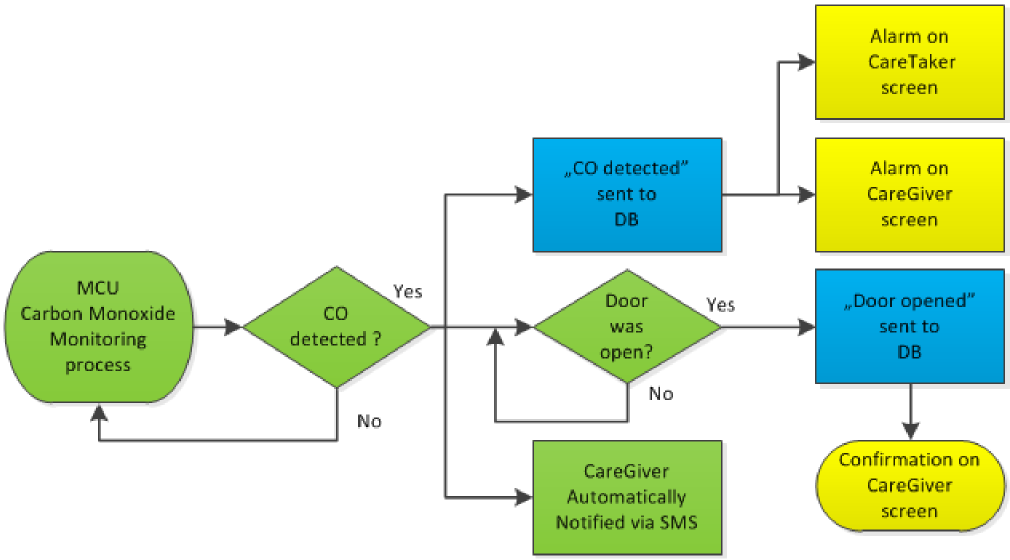


Fig. 11 Action dedicated in case of Carbon Monoxide/propan-butan detected.

Algorithm elements marked green color were taken into account during test of SALIG++ components.

2.2 Video conferencing subsystem test

Specific elements of the subsystem were provided by DIV. The list of elements is presented below:

- Entone STB box + dedicated remote control
- 2 pcs. of Raspberry Pie units
- 2 pcs. of HD Logitech cameras
- Pre-configured images of DIV software ready to deploy under VMware environment
- Pre-configured software provided on SD cards (one for each Raspberry Pie unit)
- RTSS hub – protection dongle

After DIV video subsystem deployment on provided by PIA VMware server, the initial configuration were conducted by DIV with the use of remote desktop, to obtain presented below scheme of architecture:

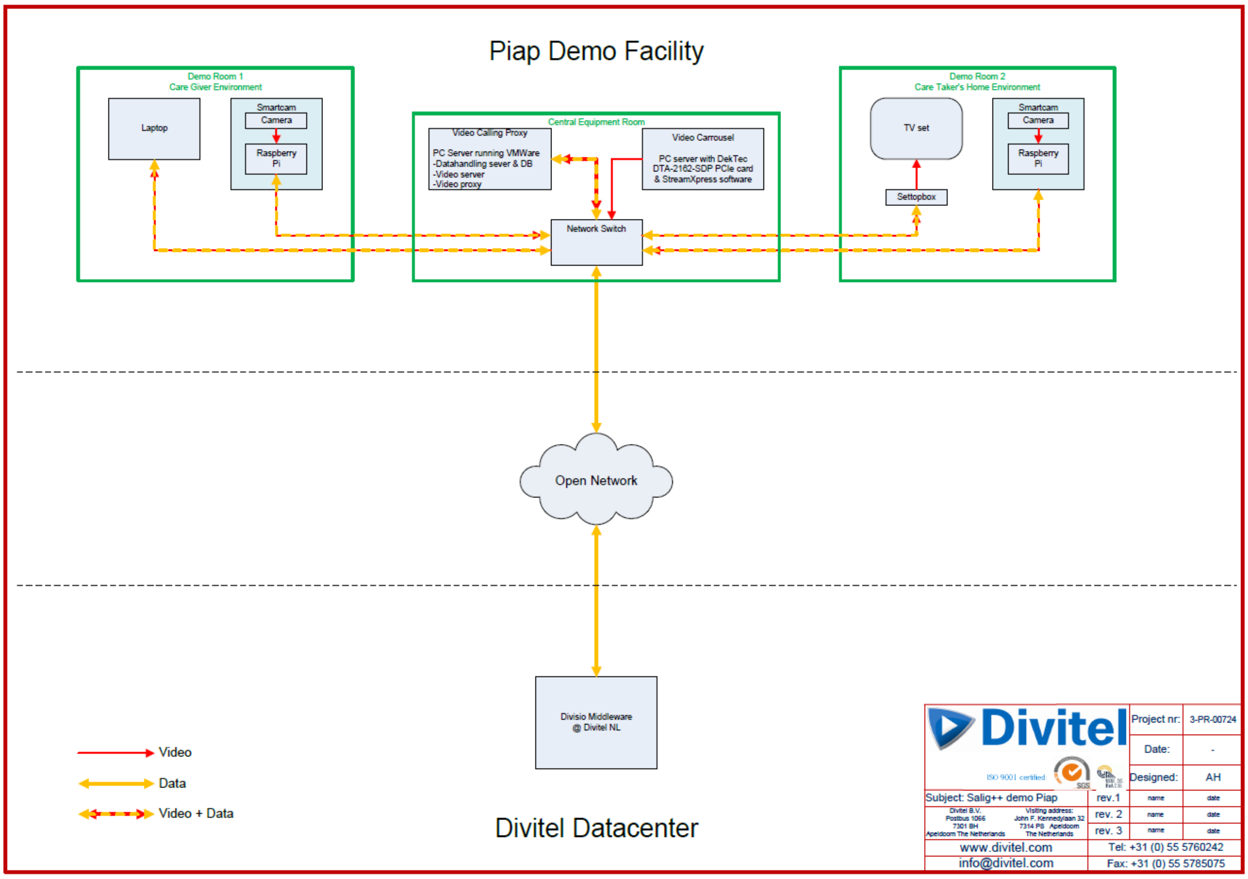


Fig. 12 Video subsystem architecture built in PIA SALIG++ Integration site.

Entone STB box suits as a dedicated equipment connected at senior apartment:

- via HDMI cable to senior TV set,
- via Ethernet cable to IT infrastructure.

With the use of STB remote control, senior can operate the system, particularly:

- switch TV screen from live program to SALIG++ portal,
- initiate video call,
- accept incoming from caregiver video call,
- reject incoming from caregiver video call.

All presented above operations are easy and intuitively available by pressing buttons on remote control, respectively:

- STB button,
- OK button,
- Green button,

- Red button.

After video subsystem tests of separate elements, the overall performance was tested, according to provided specification:

a. STB to PC communication

What	Register
Setup	- Two smart-cameras (Logitech HD Pro Webcam 920 connected to Raspberry PI (Model B) running Salig++ software) connected to a LAN. - One STB (Entone Kamai) connected to the LAN (all equipment is connected to the same LAN). - One computer connected to the LAN. - TV channels are made available by connecting the LAN to a live broadcasting infrastructure running "Fokuson" middleware by Nordija.
Reproduction Steps	Note: if the procedure below doesn't work as described, first exit the Salig++ video on the TV if it is still showing by pressing the red button on the remote control. Close the Salig++ menu interface if it is open by pressing the "exit" button on the remote control. Then restart the STB client portal (press the "menu" button on the remote control, navigate to and select the following options using the arrow buttons and the "OK" button on the remote control: "Settings" > "System Info"> "Restart" > "Reload portal"). Continue the procedure from step two. 1. Turn on the TV, STB and computer 2. Wait until the TV shows a TV channel. 3. Open a browser window on the computer and navigate to the following address: "http://10.6.3.103:9000" (or refresh the browser window if the webpage is already open). 4. Log in using the following credentials: user "web1" and password "salig314". 5. Press the blue button on the remote control to open the Salig++ interface on the TV. 6. The TV shows the Salig++ interface including a contact list with registered subscribers. Subscribers that are logged in are listed with a green light indicator preceding their name. Subscribers that are not logged in are listed with a red light indicator preceding their name. Wait until a green indicator appears preceding the user "Web Client 1". 7. Navigate to "Web Client 1" (using the arrow keys on the remote control) and press "OK" on the remote control. 8. A message appears on the TV ("Calling | Web Client 1 | Hang Up"). A message appears in the browser window ("Incoming Call | STB Client 1 | Accept Call | Reject Call"). Click on "Accept Call" in the browser window to establish a connection. 9. After a few seconds the computer displays two video windows (a small window within a big one). The big window shows the smart-cam connected to the STB, the small window shows the smart-cam connected to the computer. The TV shows the video feed from the computer. 10. Check if the lag, volume and image quality are of good quality by talking to another user. 11. Stop the call from the STB, by pressing the red button on the remote control. 12. Both the TV and the computer return to the Salig++ menu interface. 13. Press "exit" on the remote control to return to the channel the TV was showing.
Expected Result	A two way video communication should have been established, with good image and audio quality, and with a latency of less than 0.5 seconds.

b. PC to STB communication

What	Register
Setup	- Two smart-cameras (Logitech HD Pro Webcam 920 connected to Raspberry PI (Model B) running Salig++ software) connected to a LAN. - One STB (Entone Kamai) connected to the LAN (all equipment is connected to the same LAN). - One computer connected to the LAN. - TV channels are made available by connecting the LAN to a live broadcasting infrastructure running "Fokuson" middleware by Nordija.

Reproduction Steps	Note: if the procedure below doesn't work as described, first exit the Salig++ video on the TV if it is still showing by pressing the red button on the remote control. Close the Salig++ menu interface if it is open by pressing the "exit" button on the remote control. Then restart the STB client portal (press the "menu" button on the remote control, navigate to and select the following options using the arrow buttons and the "OK" button on the remote control: "Settings" > "System Info"> "Restart" > "Reload portal"). Continue the procedure from step two. 1. Turn on the TV, STB and computer. 2. Wait until the TV shows a TV channel. 3. Open a browser window on the computer and navigate to the following address: "http://10.6.3.103:9000" (or refresh the browser window if the webpage is already open). 4. Log in using the following credentials: user "web1" and password "salig314". 5. The browser window shows a contact list with registered subscribers. Subscribers that are logged in are listed with a green light indicator preceding their name. Subscribers that are not logged in are listed with a red light indicator preceding their name. Wait until a green indicator appears preceding the user "STB Client 1". 6. Click on "STB Client 1" in the browser window. The "Call" button in the browser window becomes active (it changes from grey to blue). Click on the "Call" button to establish a connection. 7. The browser window shows a new message indicating that a call is being set up ("Calling | STB Client 1 | Hang Up"). A pop up appears on the TV screen ("Incoming Call | Web Client 1 | Accept call | Reject Call"), press the green button (Accept Call) on the remote control to establish a connection. 8. After a few seconds the computer displays two video windows (a small window within a big one). The big window shows the smart-cam connected to the STB, the small window shows the smart-cam connected to the computer. The TV shows the video feed from the computer. 9. Check if the lag, volume and image quality are of good quality by talking to another user. 10. Stop the call from the computer by clicking on the "Hang Up" button in the browser window. 11. The videos in the browser window disappear, the contact list remains. The STB returns to the TV channel it was showing.
Expected Result	A two way video communication should have been established, with good image and audio quality, and with a latency of less than 0.5 seconds

Conducted tests confirmed video subsystem basic functionality. Pictures below presents results gathered from the test procedures:

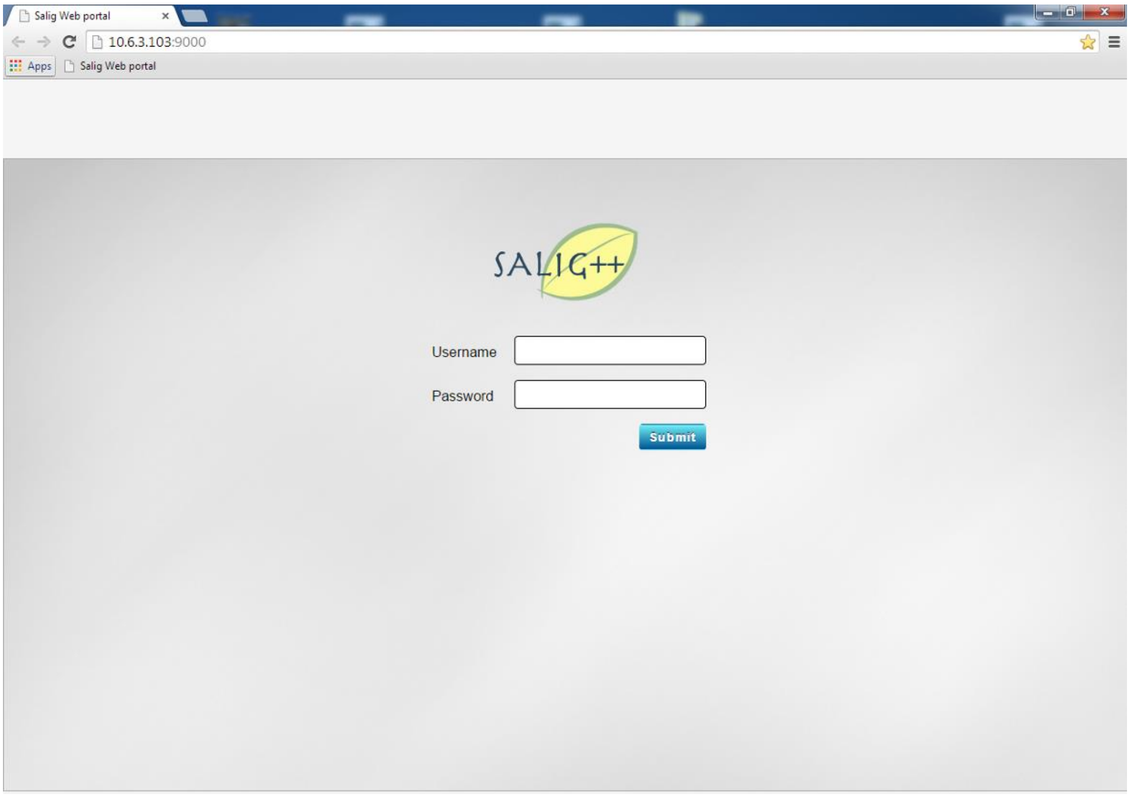


Fig. 13 Caregiver Web portal - initial screen with logon fields

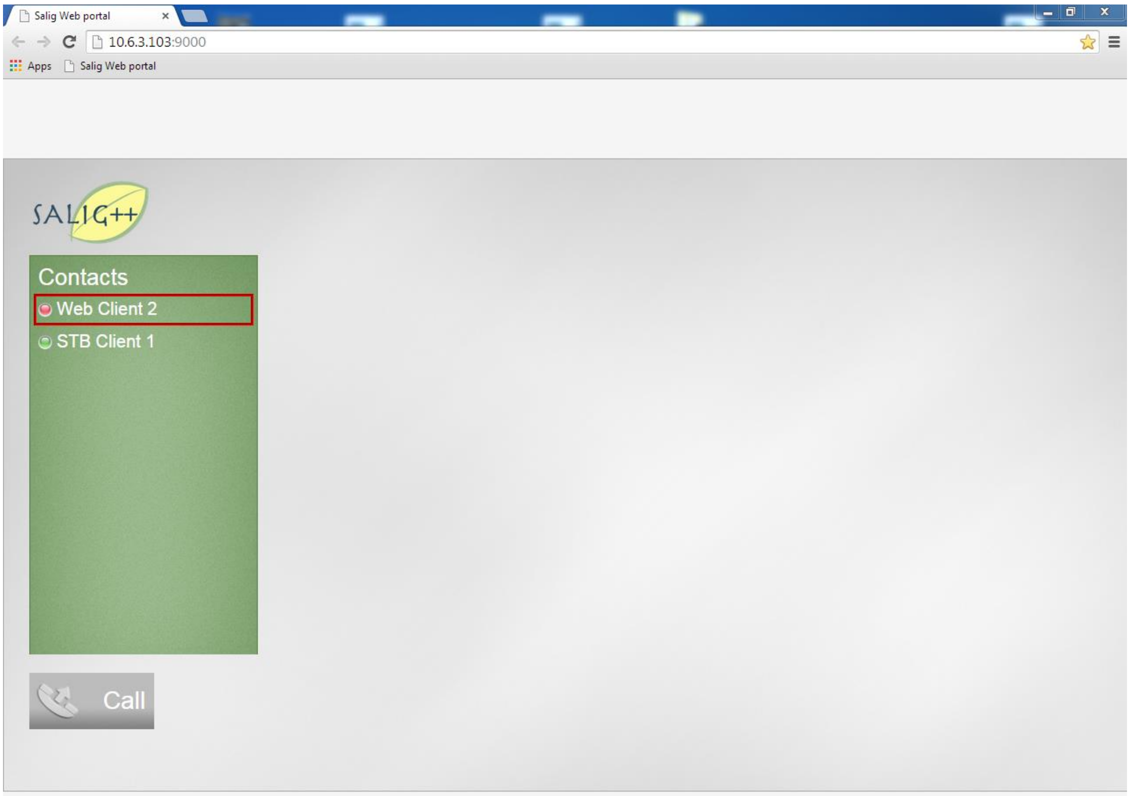


Fig. 14 Caregiver Web portal – screen visible after successful logon. Green indicator next to user name indicates that user is available for calling.

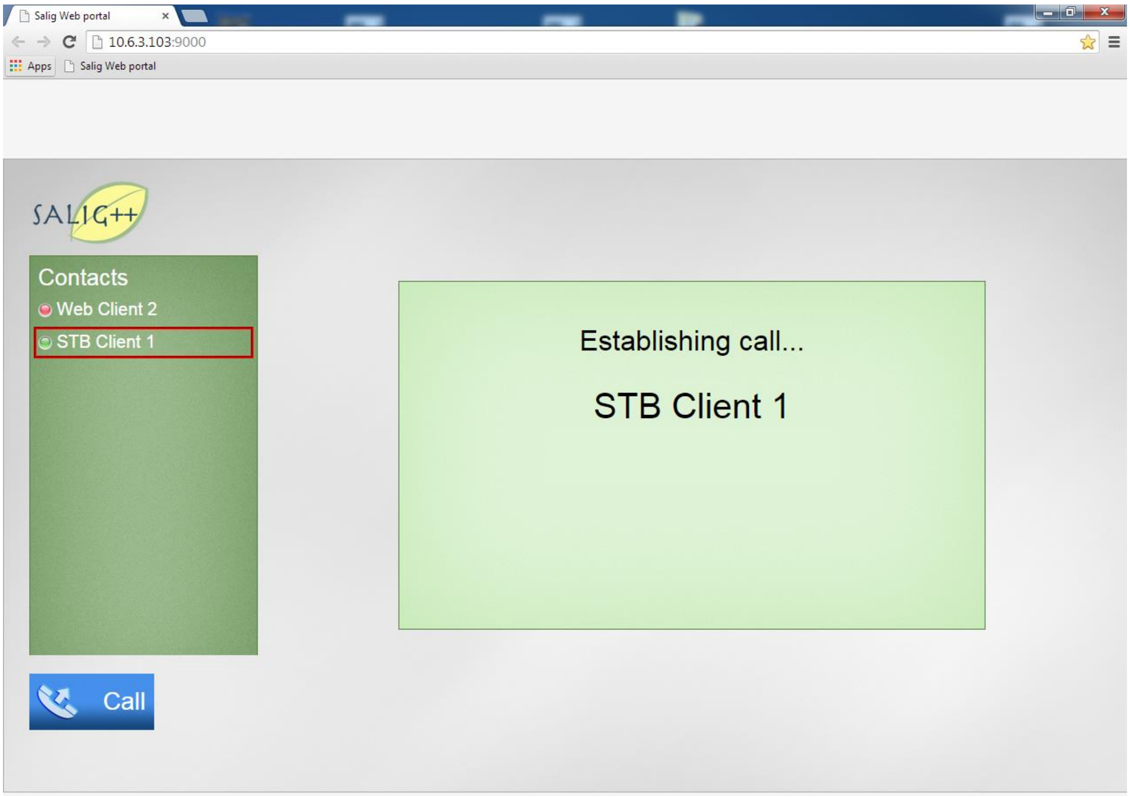


Fig. 15 Caregiver Web portal – screen visible during estabilishing call to caretaker STB box.

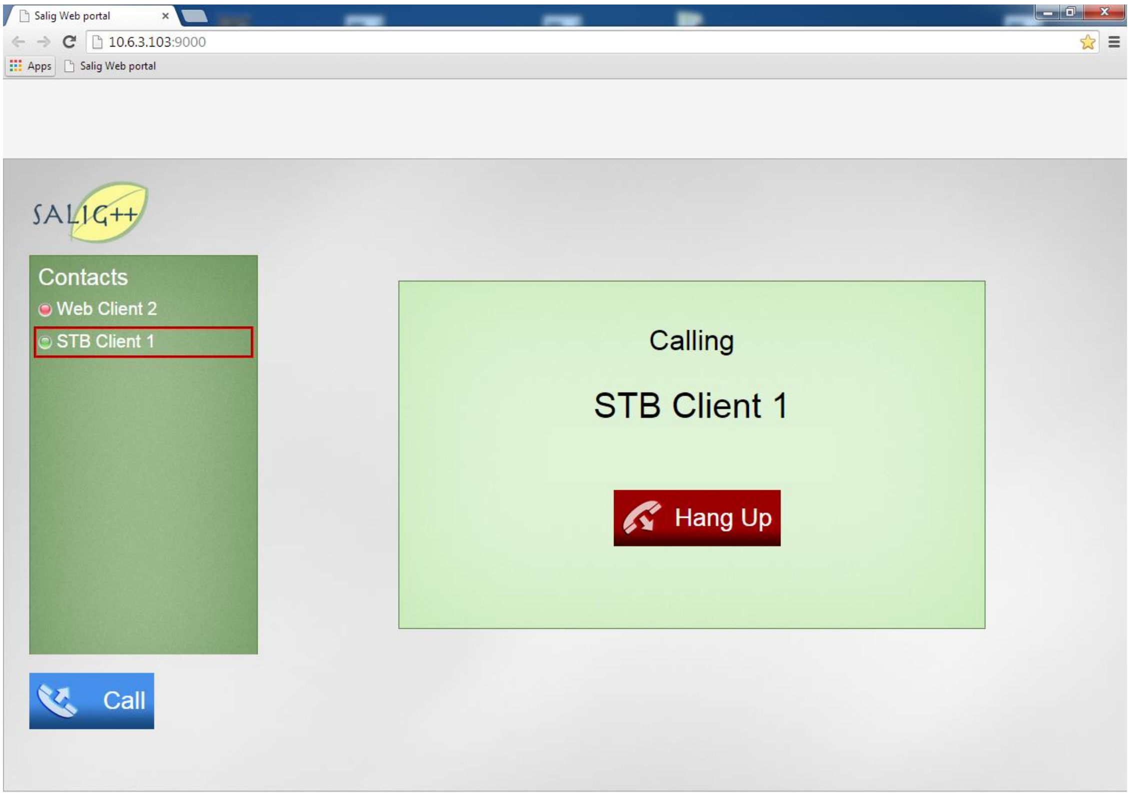


Fig. 16 Caregiver Web portal – screen visible during call attempt to caretaker STB box.

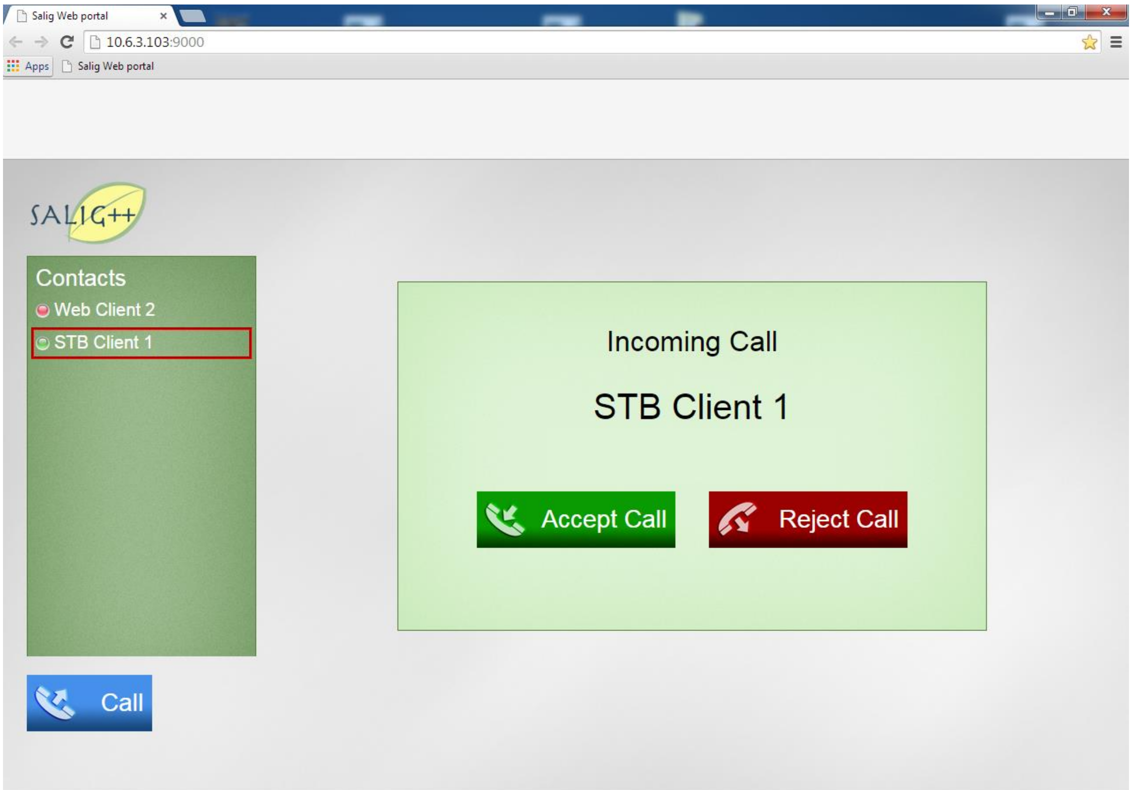


Fig. 17 Caregiver Web portal – screen visible during incoming call from caretaker STB box.

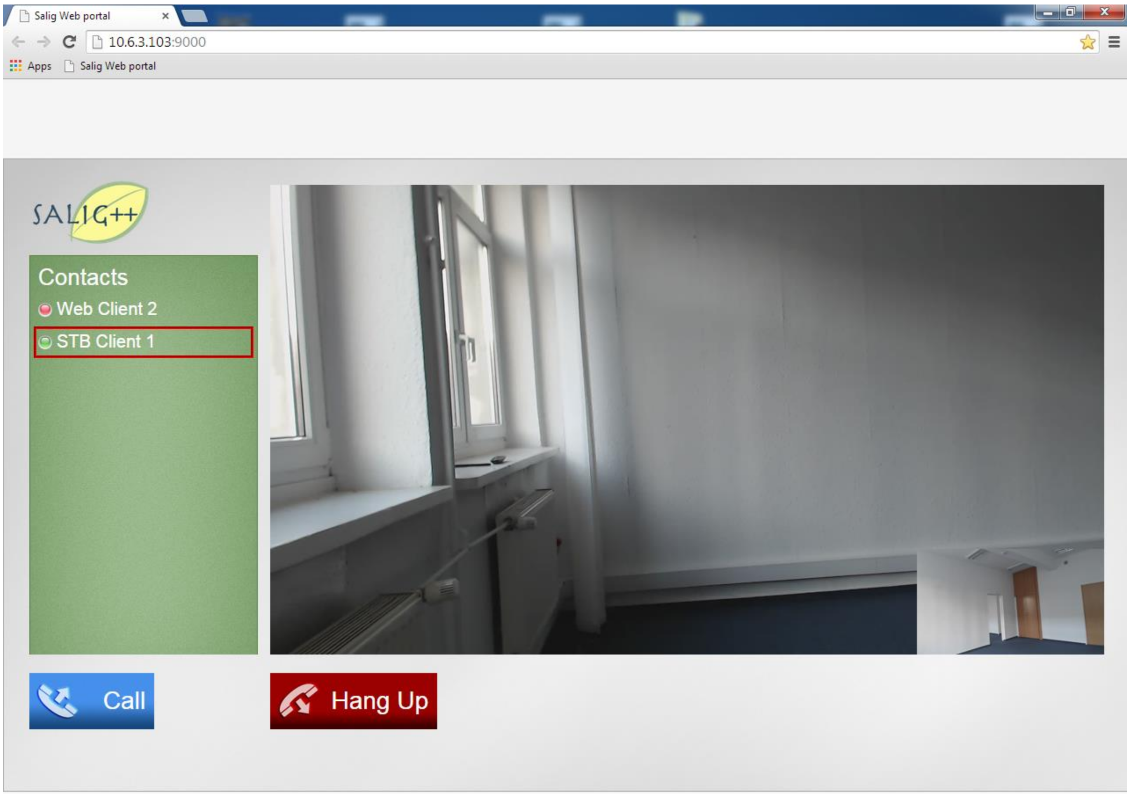


Fig. 18 Caregiver Web portal – screen visible during call with caretaker STB box. On the right bottom corner, caregiver has Picture-in-Picture view from their own camera.

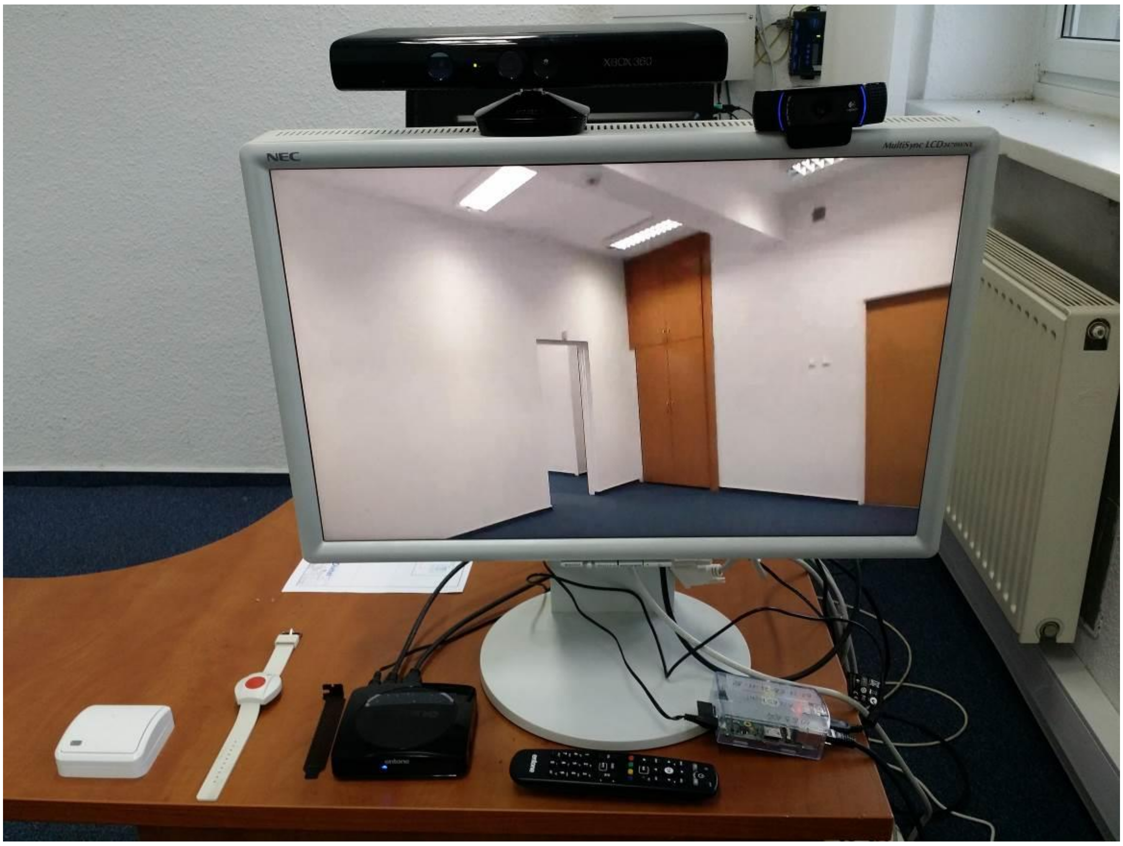


Fig. 19 Caretaker TV set – picture of screen done when video call with cargiver is estabilished. On the bottom of the picture, the wall programmable button and wrist emergency button are presented.

Initial tests of video subsystem confirmed that their baisic functionalities work well with good performance. Quality of video and latancy time are on acceptable level ensuring good communication channel between caregiver and caretaker.

2.3 Pose recognition subsystem test

The subsystem for pose recognition was prepared by TUD together with ALM. Software, including:

- The executable file of eating and fall-down detection implementation by C++,
- MongoDB,
- Python,
- PyMongo - Python distribution tool for MongoDB
- Sleepy.Mongoose - MongoDB REST Server
- cURL tool

Listed above software were installed on provided by PIA server with Windows 7 64-bit system.

Sent by TUD hardware (MS Kinect sensor) were then attached to the server giving functional system for pose recognition.

Figure below presents working application:

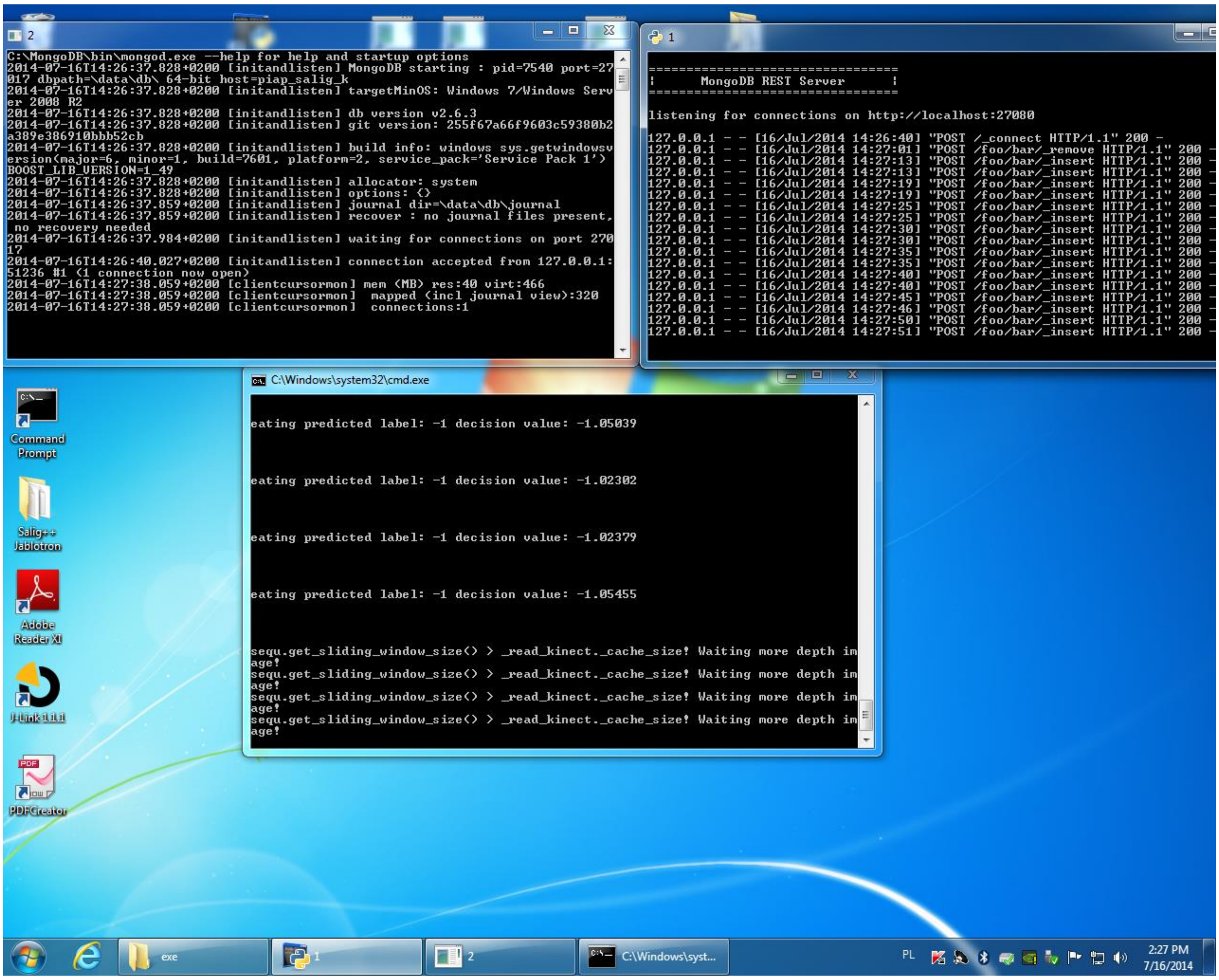


Fig. 20 Working applications ensuring pose recognition functionality.

The table below with logs is an example of output from the applications. Recognized predicitions of senior postures (like eating, falling down, etc.) are stored in local MongoDB for further processsing.

C:\MongoDB\salig_project\exe_file\exe>LocalHON4D_eat_falldown_kinect http://localhost:27080 5000
rest port: http://localhost:27080
output time slot: 5000
Start to set up:
size of training_min_max: 228000
load training_min_max success!
Loading training model....
load training model success!
{"ok" : 1}
set up finish!
start to monitor...
sequ.get_sliding_window_size() > _read_kinect._cache_size! Waiting more depth image!
_cache_size: 10
_cache_size: 20
sequ.get_sliding_window_size() > _read_kinect._cache_size! Waiting more depth image!
_cache_size: 30
sequ.get_sliding_window_size() > _read_kinect._cache_size! Waiting more depth image!
_cache_size: 40
_cache_size: 50
_cache_size: 60
eating predicted label: -1 decision value: -1.15065

_cache_size: 70
eating predicted label: -1 decision value: -1.17443
eating predicted label: -1 decision value: -1.13979
sequ.get_sliding_window_size() > _read_kinect._cache_size! Waiting more depth image!
_cache_size: 80
_cache_size: 90
eating predicted label: -1 decision value: -1.06912
_cache_size: 100
{"oids": {"$oid": "53c66d79e53a28105051912e"}}
non-eating!
{"oids": {"$oid": "53c66d79e53a28105051912f"}}
Normal (no fall down)!
_cache_size: 110
eating predicted label: -1 decision value: -1.10275
eating predicted label: -1 decision value: -1.05773
_cache_size: 120
eating predicted label: -1 decision value: -1.04831
eating predicted label: -1 decision value: -1.05656
sequ.get_sliding_window_size() > _read_kinect._cache_size! Waiting more depth image!

5. Results and conclusions

The results from the initial test of SALIG++ components show that expected basic functionalities of each tested subsystem have been fully achieved. They confirmed that crucial for SALIG++ subsystems work with satisfactory performance.

As further steps of testing foreseen during the next stage of the project, listed below aspects have to be taken into account:

- interactions between all SALIG++ subsystems working together on common platform,
- overall performance of the system working under real conditions at the selected Pilot Site,
- results, suggestions and comments gathered during evaluation by End Users.