



**ICT Policy Support Programme
Call 3 objective 1.3 ICT for ageing well / independent living**

Grant Agreement No. 250505

inCASA

**Integrated Network for Completely Assisted Senior citizen's
Autonomy**

D5.4 Descriptions of installation and extensions

Author(s)

CNET, TID, NTUA, REPLY, SIG, BRUNEL, INSERM

Project start date: 1st April 2010

Duration: 30 months

Published by the inCASA Consortium
Coordinating Partner: SANTER REPLY Spa

15-03-2013 – version 1.0

Project co-funded by the European Commission
within the CIP ICT-PSP Programme

Dissemination Level: Public

Document file: D5-4_Description_of_installation_and_extensions_v1.docx

Work package: WP5

Tasks: Task 5.3 – 5.4

Document responsible: CNET

Document history:

Version	Author(s)	Date	Changes made
Draft01	Stefan Asanin (CNET)	01/02/13	Table of Contents
Draft02	Stefan Asanin (CNET)	04/02/13	Added Continua subchapter
Draft03	Stefan Asanin (CNET)	20/02/13	Changed ToC after discussion
Draft04	Liudmila Dobriakova, Paola Dal Zovo, Andrea Prestileo (REPLY)	25/02/13	SPP content and ATC pilot.
Draft04	Jordi Rovira (TID)	04/03/13	Contribution for SARA sections
Draft05	Stefan Asanin (CNET)	06/03/13	Restructured ToC
Draft06	Malcolm Clarke (BRUNEL)	08/03/13	Added detail on CHC installation
Draft07	Andrea Prestileo (REPLY)	12/03/13	Added detail on ATC and SKIVE.
Draft08	Alexandre Arbaud (INSERM)	12/03/13	Added detail on INSERM.
Draft09	George Lamprinakos, Andrew Kapsalis (NTUA)	12/03/13	Contribution on Consumer Applications Sections and Greek pilot technical procedures.
Draft10	Stefan Asanin (CNET)	15/03/13	Corrected review comments and finalised document.

Peer review history:

Reviewed by	Date	Comments
George Lamprinakos (NTUA)	14/03/13	
Matts Ahlsén (CNET)	14/03/13	Approved with comments
Liudmila Dobriakova (REPLY)	14/03/13	Comments and corrections are tracked by Track Changes

Index

Executive Summary	6
1 Introduction	7
1.1 Purpose and Content of this Deliverable	7
1.2 Outline of this Deliverable	7
2 Generic installation procedures	8
2.1 Overview of inCASA platform	8
2.2 Data collection at home base station	9
2.2.1 Activity Hub	9
2.2.2 SARA Client	22
2.2.3 LinkSmart Middleware	27
2.2.4 Chorleywood Platform	33
2.3 Remote Service Provider	43
2.3.1 Data management at Smart Personal Platform	43
2.3.2 Mediator	44
2.3.3 Reasoner	45
2.3.4 Electronic Patient Record	46
2.4 User interfaces and Extensions	46
2.4.1 Consumer Applications (NTUA)	46
2.4.2 Extensions (CNET)	52
3 Description of inCASA platform installations	53
3.1 KGHNI pilot	53
3.1.1 Greek procedures	53
3.1.2 Experiences	55
3.1.3 Solution extension	56
3.2 INSERM pilot	57
3.2.1 French procedures	57
3.2.2 Experiences	60
3.2.3 Solution extensions	60
3.3 FHC pilot	61
3.3.1 Spanish procedures	61
3.3.2 Experiences	61
3.3.3 Solution extensions	61
3.4 ATC pilot (REPLY)	61
3.4.1 Italian procedures	62
3.4.2 Experiences	62
3.4.3 Solution extensions	62
3.5 SKIVE Danish Transferability Model	63
3.5.1 Danish procedures	63
3.5.2 Solution extensions	64
3.6 CHC pilot	75
3.6.1 Chorleywood (UK) procedures	75
3.6.2 Experiences	75
3.6.3 Solution extensions	76
4 Conclusion	77
5 Glossary	78
6 References	79

Figures

Figure 1: inCASA architecture iteration 2 (high level view)	8
Figure 2: inCASA Activity Hub network set up.....	10
Figure 3: Opening the AH by removing the screws	11
Figure 4: Inserting GSM SIM card.....	12
Figure 5: SIM card slot from the side view	12
Figure 6: ZigBee Antenna	12
Figure 7: GSM Antenna	12
Figure 8: AH antenna connectors.....	13
Figure 9: AH power supply	13
Figure 10: AH power supply connector and power switch	13
Figure 11: AH LAN cable connection	14
Figure 12: AH identification plate with factory default IP address	14
Figure 13: Accessing AH configuration page with a web browser	15
Figure 14: AH ID setting.....	15
Figure 15: GSM/GPRS settings	15
Figure 16: AH settings for the communication with the LinkSmart base station	16
Figure 17: Store the settings	16
Figure 18: Sensor List link on AH web page.....	17
Figure 19: Sensor list loaded on AH.....	17
Figure 20: Activity Hub status page.....	18
Figure 21: Z-B01C motion sensor binding key	18
Figure 22: Z-B01C motion sensor front	19
Figure 23: Z-302A window/door sensor.....	19
Figure 24: Z-711 Temperature/Humidity sensor.....	20
Figure 25: Z-801 WLS plate connection example.....	20
Figure 26: Z-801 WLS water detection sensor	21
Figure 27: Power Outlet Device	21
Figure 28: Activity Hub sensor event log	22
Figure 29: Windows Bluetooth icon	23
Figure 30: Searching Bluetooth devices.....	23
Figure 31: Detecting our sensors	24
Figure 32: Checking the connection with the sensor	24
Figure 33: New patient inclusion form	25
Figure 34: Choosing treatment.....	25
Figure 35: Choosing inCASA id.....	26
Figure 36: SARA's configuration tool at patient's home.....	26
Figure 37: UI showing and displaying updating patients' application	27
Figure 38: Showing measures associated to a specific patient	27
Figure 39: LinkSmart NetworkManager status page.....	28
Figure 40: Naming the LinkSmart instance.....	29
Figure 41: Setting Java path in Environment Variables	30
Figure 42: Adding service to OSGIServieWrapper	31
Figure 43: Automating the OSGIServiceWrapper at start-up.....	32
Figure 44: Overall Architecture.....	33
Figure 45: Layered Approach.....	34
Figure 46: ZigBee Device Architecture.....	36
Figure 47: inCASA Devices.....	37
Figure 48: CHC Home Gateway.....	37
Figure 49: Clinician using the Clinician Portal	40
Figure 50: CHC homepage view	41
Figure 51: Equipment's page for UK clinicians	41
Figure 52: Tabular data overview.....	42

Figure 53: CHC graphical data representation	42
Figure 54: Reasoner's Windows service	46
Figure 55: SPP – CA Integration	47
Figure 56: Glassfish configuration	48
Figure 57: MySQL Database Configuration.....	49
Figure 58: Default CA Database schema	49
Figure 59: Data source creation under Glassfish	50
Figure 60: Additional Data source Properties	50
Figure 61: JDBC Resource configuration	50
Figure 62: CA Deployment.....	51
Figure 63: Web Portal start page	51
Figure 64: KGHNI pilot architecture	53
Figure 65: Interface to submit new questionnaire score under a patient.....	57
Figure 66: Questionnaires graph - the red circle is showing an alerting score that triggered a psychologist's intervention	57
Figure 67: French form for assigning new patient.....	58
Figure 68: French GUI when registering new treatment	58
Figure 69: Choosing an inCASA id for the INSERM pilot.....	59
Figure 70: SARA GUI for the INSERM pilot.....	59
Figure 71: Patient measurement choices as provided by SARA.....	60
Figure 72: The Actigraph AML file handler as webpage and where latest activity data is on top for easier retrieval by the clinicians.	61
Figure 73: Zilant hub	62
Figure 74: New connection in Bluetooth Settings	64
Figure 75: Toshiba Bluetooth device connection Wizard	65
Figure 76: Select device to pair in connection Wizard. (1.) Two AND Medical device found, Blood Pressure and Weighing Scale.	65
Figure 77: Bluetooth Security	65
Figure 78: Nonin Pulse Oximeter found by Connection Wizard	66
Figure 79: Bluetooth Security, insert PIN code	66
Figure 80: GlucoFacts, Database Setup Dialog	67
Figure 81: GlucoFacts, new meter detected.....	67
Figure 82: GlucoFacts, settings and manage connection	68
Figure 83: StartGlucoFacts shortcut into Start\Startup	71
Figure 84: Msconfig/Startup and Toshiba Bluetooth stack.....	73
Figure 85: LIVA setup	73
Figure 86: Daily smoke report (1 is smoke ON, -1 is smoke OFF) with hidden patient name	75
Figure 87: Calories burned graph (sub-measurement of Activity monitoring graph)	75

Tables

Table 1: GPRS settings.....	16
Table 2: IAS Zone Registration	21
Table 3: inCASA listening ports.....	44
Table 4: inCASA Windows services	44

Executive Summary

This deliverable intends to provide the general public (i.e. any organisation or care domain interested in adopting the progress delivered by the inCASA project and solution) with detailed descriptions on the installations and extensions that were implemented and deployed by the inCASA pilots with the help of the technical expertise in the consortium. The descriptions solely provide inCASA adopters with insight and understanding on how to assemble the inCASA platform subsystems into an executable whole.

D5.4 reviews a generic approach when assembling the inCASA platform and its subsystems from the home gateway to the remote service provider at the backend and to different Consumer Applications. This is done by naturally stepping into pilot installation procedures in the next coming subchapters that gives deeper insight of real world deployments.

The goal of D5.4 is to enhance any plausible impact that the inCASA platform might have in different telehealth and telecare settings and business models throughout Europe. Further, this deliverable intends to strengthen the exploitation aims and resources of each partner and provide means of regenerating the inCASA subsystems into a complete and coherent whole that is able to satisfy plausible end customer needs and workflows by providing detailed practical installation and deployment guidelines with references.

1 Introduction

The inCASA project has developed a system that supports the aging population and facilitates them to stay longer and more healthily in their own home, by means of the following specific objectives:

- By providing the means to profile the everyday behaviour of elderly people in their own home, through unobtrusive monitoring using motion and contact sensors, as part of a Smart Personal Platform with embedded Behaviour Analysis determining unusual behaviour and sends alerts via a base station to selected actors
- By providing elderly people (and patients with special needs) with the means to monitor their health conditions outside traditional healthcare environments, and more specifically while they are at home, by using state of the art personal health systems and integrated telemedicine services
- By providing doctors and health professionals with more comprehensive monitoring data to understand the social, physical and/or psychological condition of the person and so allow early decision for personalised care
- By enabling continuity of care through a wider interaction between elderly people or patients and caregivers, especially to include not just health specialists but also relatives or people who have close social relations with the user

1.1 Purpose and Content of this Deliverable

WP5 objective is the definition of the solution for the inCASA modules integration into one advanced monitoring system. Modules and components to be integrated have been defined in WP3 and implemented in WP4 adhering to the pilot's requirements and derived specifications. Since the different inCASA modules are defined by different providers (i.e. each with proven competences on a specific subject), integration is seen as a core activity for the project and as cohesion between the several components must be guaranteed by means of a proper integration strategy.

Deliverable D5.4 is focused on the detailed integration specifications for the two macro-modules identified in WP3: Home Base Station and Remote Service Provider but also integration between the inCASA modules such as the SPP-CA via PCD-02 and PCD-04 [2][1]. Integration is described in terms of interfaces, where they are modelled as services and data flows. Data definition and manipulation is fundamental throughout the integration since the user habits may be iteratively improved by data collection (measurements at Base Station level and aggregated in models at Service Provider level). The data formatting and representation is the key for the interoperability (HL7 standard). Moreover, the use of standard-based interfaces enables the integration of additional components, such as the Consumer Applications, that can bring added value to the inCASA platform.

1.2 Outline of this Deliverable

This document is structured into the following chapters:

- Chapter 2 – provides a generic overview of the installation procedures that are undertaken for the inCASA pilots. All pilots will refer back to this chapter of its subchapters.
- Chapter 3 – covers descriptions over all separate pilot installations, their experiences and any solution extension (if any).
- Chapter 4 – provides a conclusion with minor discussion on the D5.4 aims and benefit.

2 Generic installation procedures

2.1 Overview of inCASA platform

The inCASA architecture is strongly oriented to service integration. The two main subsystems, the Home Base Station and the Smart Personal Platform (SPP), needs to be integrated in order to collect data related to the patient and his/her environment and to model habits. In the same fashion, the SPP needs to be integrated to the outside world (namely, to the Consumer Applications) in order to allow caretakers and other involved people to monitor the situation at patient's premises. See Figure 1.

Modules must offer interfaces in order to exchange messages with other modules and with the ecosystem. Messages are sent in response to events and information is provided by means of standard formats.

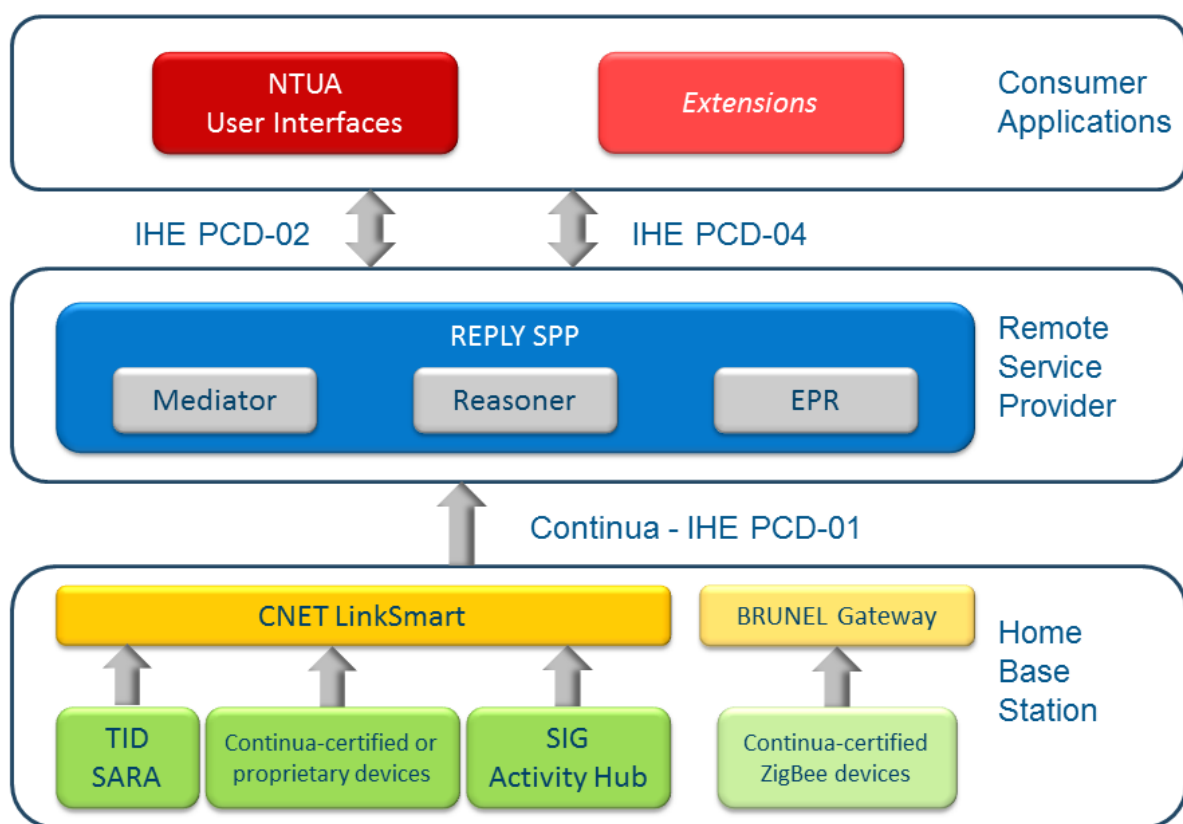


Figure 1: inCASA architecture iteration 2 (high level view)

The inCASA architecture can be considered as an *event-driven architecture*. According to [6], an event-driven architecture (EDA) is a software architecture pattern promoting the production, detection, consumption of, and reaction to events.

An *event* is a notable (meaningful) thing that happens inside or outside a domain. An event (business or system) may signify a problem or impending problem, an opportunity, a threshold, or a deviation [7]. Usually, events are generated by a source and then propagated to a number of registered downstream subscribers. This integration pattern is known as *publish & subscribe*. The interested subscribers evaluate the event, and optionally take action. The event-driven action may include the invocation of a service, the triggering of a business process, and/or further information publication [7].

The triggering of processes within the inCASA system depends on several clients communicating with a central server and where each of these comprises a multitude of different sensors. All of the sensors, independently, generate events that by using the HL7 nomenclature are made comprehensible to upper layers (i.e. SPP). Each inCASA integration process can therefore be mapped to one of the EDA styles (simple, stream or CEP):

- sensors at the HBS level generates messages which are collected by either the Activity Hub or the SARA client; these latter represent the subscribers for event generated by the house/patient by means of the several devices; the Activity Hub and the SARA client report messages to the LinkSmart¹ client agent (*simple event processing*²)
- the LinkSmart client agent converts data in HL7 formats and sends events as streams of measurements to the SPP (*stream event processing*³)
- the Brunel Gateway is able to communicate with Continua-certified ZigBee-based devices and send HL7 messages to the SPP.
- the SPP receives events and updates its internal habits model through the reasoning process, then it generates alerts towards the Consumer Applications; this enables CEP (*Complex Event Processing*⁴).

All of the different modules in the inCASA architecture are decoupled by means of asynchronous standard interfaces, such as SOAP-based web services. That means, messaging between components is implemented through asynchronous data flows, which are most recommended in EDA architecture.

2.2 Data collection at home base station

This chapter will describe the integration at Base Station level which is pivoted on the LinkSmart component. This provides the main integration backbone between the Base Station and the Service Provider Platform. The Base Station inscribes the patient's environment and allows the end user to interact with the inCASA system.

2.2.1 Activity Hub

This section is an installation instruction for the inCASA Activity Hub (AH) and the sensor network. This covers the installations for the pilot phase of the inCASA project. The AH for the pilot installations come with a new firmware compared to the AH's for the pre-pilot installations. In this firmware, the list of sensors that an AH supports is managed by the inCASA base station, maintained through CNet by the LinkSmart application server.

This central approach simplifies greatly the management of installations. Furthermore it adds a new level of manageability to the installations also in terms of remote monitoring of the correct functioning of the AHs.

The basic network setup is shown in the figure below. This figure corresponds only to ATC pilot while the Greek and Danish pilots use LinkSmart running locally at the patient's PC (i.e. no GSM/GPRS needed) doing the same thing.

¹ LinkSmart is the middleware component formerly known as "Hydra Middleware".

² Events concerning any specific and measurable changes of condition.

³ Enables in-time decision making by using the real-time flow of information in and around infrastructures.

⁴ Allows pattern evaluation on simple events and actions on these.

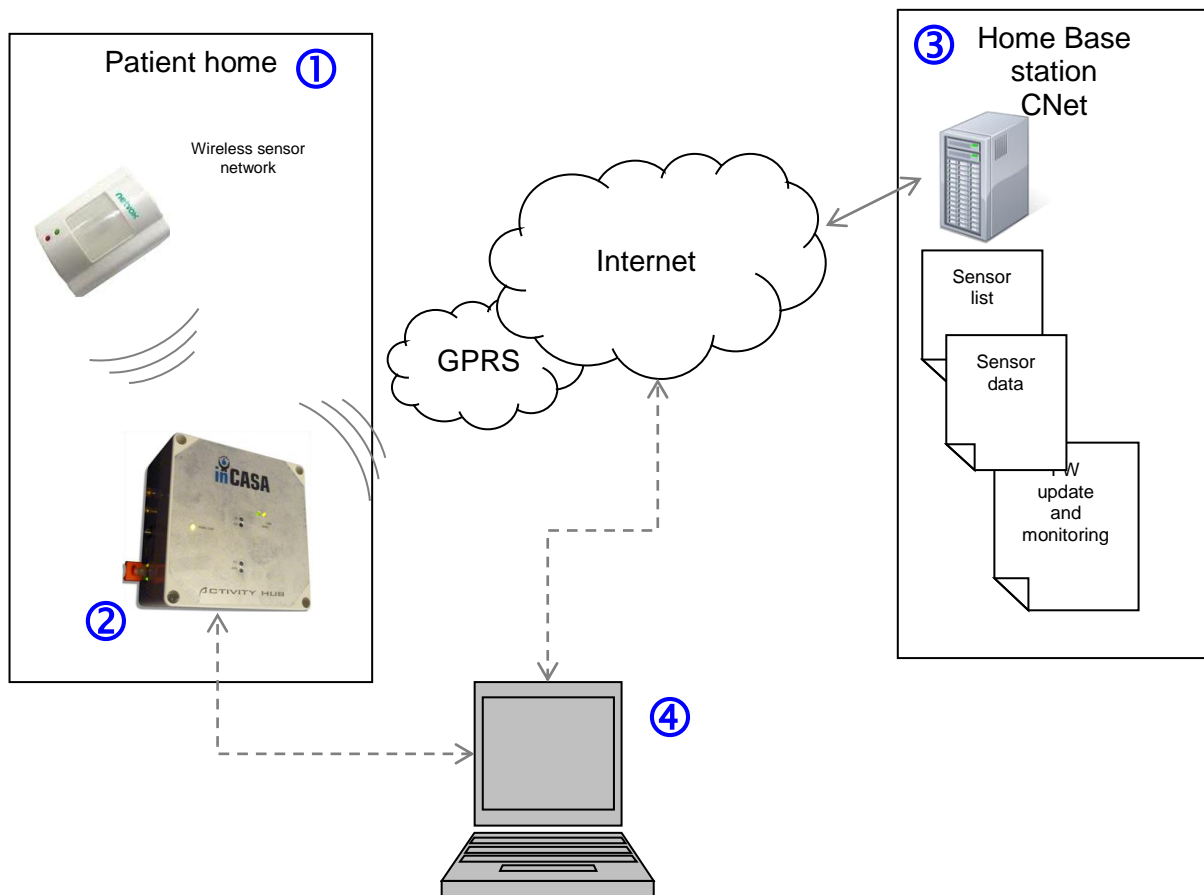


Figure 2: inCASA Activity Hub network set up

1. In the patient home there is a wireless sensor network installed.
2. The inCASA Activity Hub (AH) collects the data from the wireless sensor network and forwards it over GPRS and the Internet to the base station at CNet. Furthermore it receives the configuration for the wireless sensor network from the base station. Every AH has a unique ID assigned in order for the base station to map the communication of the AH to the corresponding patient.
3. The inCASA base station is located at CNet. It is connected to the Internet. Three different datasets are managed by the base station
 - Sensor list: every sensor in the wireless sensor network is listed. The AH checks this list periodically and upon changes it refreshes the sensor network in order to allow new sensors to be added or a sensor to be replaced
 - Sensor data: the base station receives the data sent by the sensors and forwards it to the other services within the inCASA platform
 - Firmware (FW) update and monitoring: this is a monitoring interface to check whether the AH is running correctly. Furthermore it allows to deploy new firmware on the AH.
4. During the initial setup, the installer requires access to both, the AH and the web frontend of the base station to set up an AH. Configuration can be carried out in a laboratory. The set-up consists of:
 - creating the AH ID at the base station and adding the corresponding sensors
 - setting this ID to the AH and setting the GSM/GPRS settings
 After that, the AH is ready to be installed at the patients home.

This section is a step-by-step guide for the configuration of an AH and putting it into operation. The pre-requisites are:

- AH with Antennas and power supply
- GSM SIM card supporting Internet by GPRS
- A number of sensors to be added to the AH
- Ethernet LAN cable
- PC with a current web browser. It must be possible to access the configuration web page of the AH as well as the base station management interface at CNet through Internet.

The steps presented in this chapter are:

1. Add the sensors to the base station
2. Putting the AH into operation (chapter 2.2.1.1)
3. Start-up and add the wireless sensor network (chapter 2.2.1.4)
4. Test the communication of the wireless sensor network and the communication with the base station

2.2.1.1 Putting the AH into operation

2.2.1.1.1 Insert SIM card

The GSM SIM card must be inserted while the AH is powered off. The SIM card holder is placed inside the enclosure of the AH. The lid must be removed for inserting the SIM card. There are four screws that need to be removed first:



Figure 3: Opening the AH by removing the screws

Once the AH is opened, the SIM card is inserted with the contact side showing upwards.

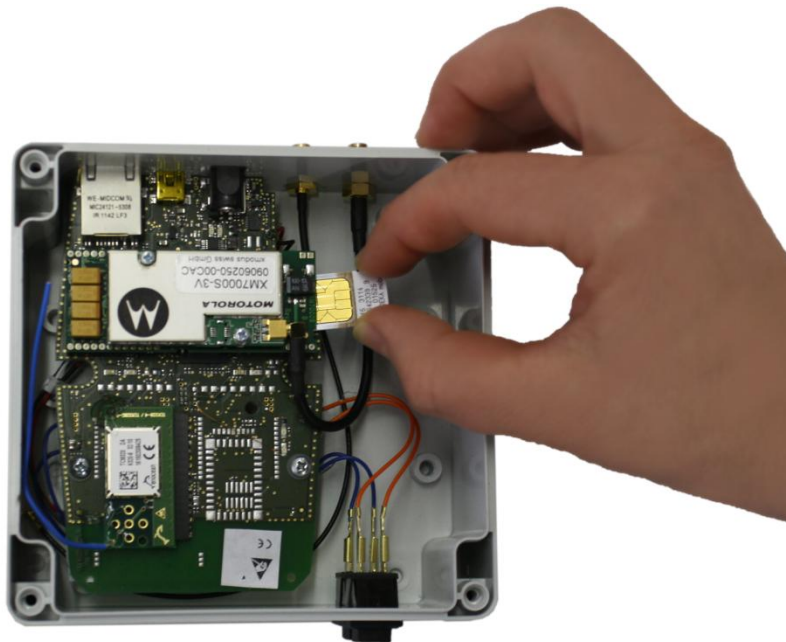


Figure 4: Inserting GSM SIM card

The SIM card holder is not clearly visible from the top. It is located underneath the GSM module (Motorola). The following picture gives an idea how the SIM card holder is located.

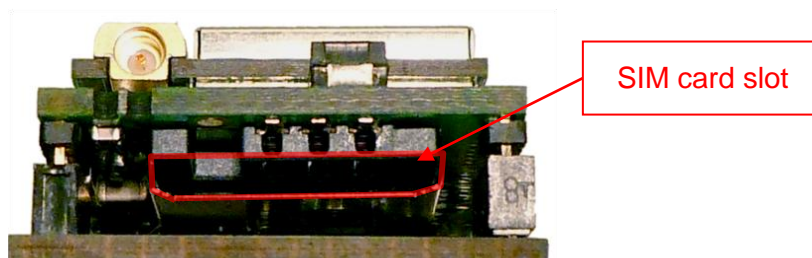


Figure 5: SIM card slot from the side view

Once the SIM card is inserted correctly, put the lid back on the AH.

2.2.1.1.2 Antenna connections

The two antennas shipped with the AH must be connected:

- ZigBee Antenna
- GSM Antenna



Figure 6: ZigBee Antenna



Figure 7: GSM Antenna

The figure below shows the positions of the connection of the antennas:

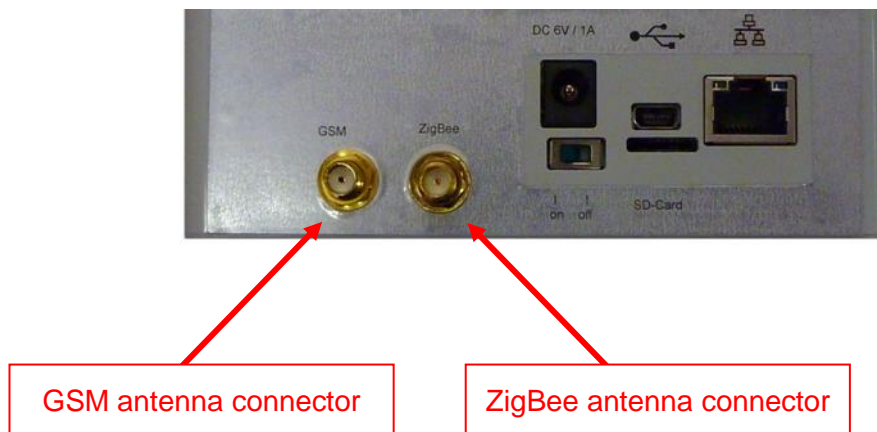


Figure 8: AH antenna connectors

2.2.1.1.3 Power connection

A power supply 230V AC/ 6VDC-1A is shipped with every AH.

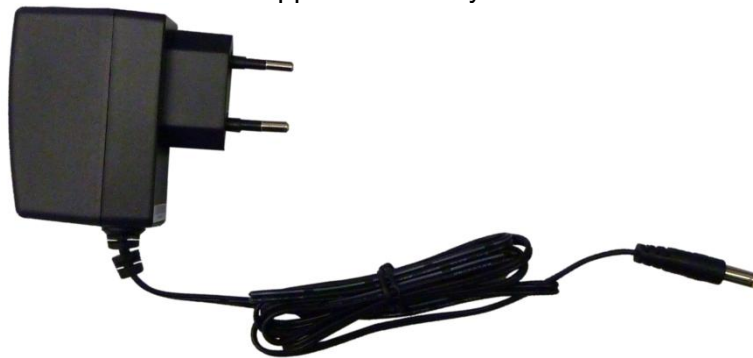


Figure 9: AH power supply

Connect the power supply to the AH and power it up by the switch.

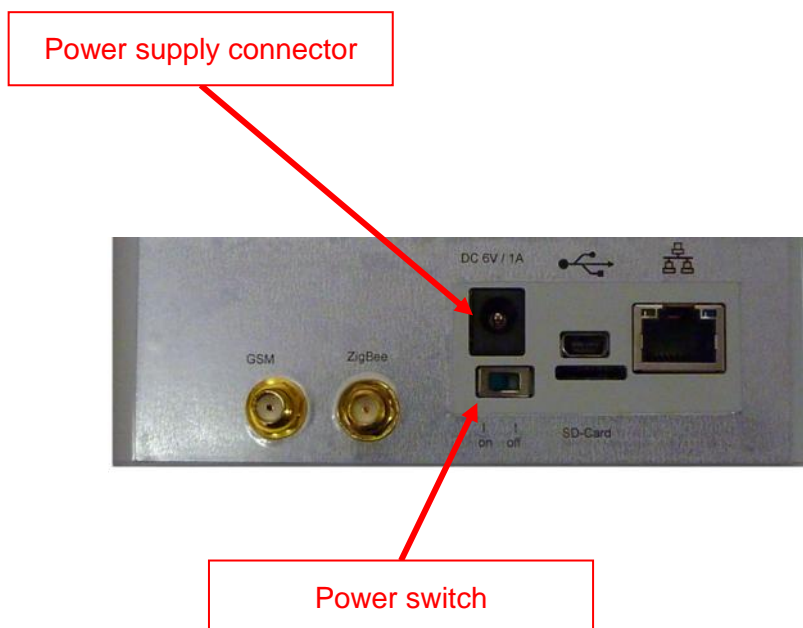


Figure 10: AH power supply connector and power switch

Please note: powering off the AH when ZigBee sensors are already associated to the AH require a special procedure.

2.2.1.1.4 Connect LAN for the configuration of the AH

To configure the AH later with a web browser (see chapter 2.2.1.2) it is required to connect the AH to a LAN. A twisted pair Ethernet cable is required for this purpose.

It is possible to connect the AH directly to a PC or to an Ethernet switch or hub. Any Ethernet cable should be OK.

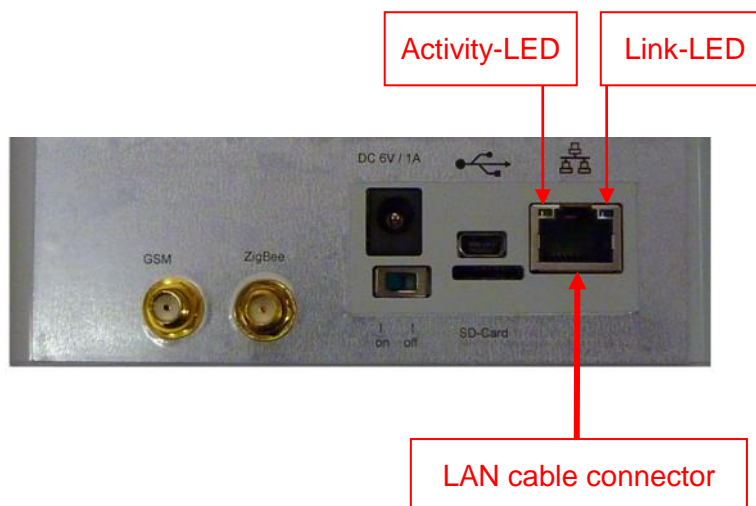


Figure 11: AH LAN cable connection

If the LAN cable is connected correctly and the cable is connected to a PC or Ethernet Switch, the Link-LED will be on and the Activity LED blinks upon network activity.

2.2.1.2 Configure the AH

2.2.1.2.1 Accessing the configuration page

The AH is configured using a web frontend. To access the configuration web page use a standard browser such as Mozilla Firefox, Google Chrome or Microsoft Internet Explorer. JavaScript must be enabled to use the web page.

The default IP address can be found on the identification plate of the AH. The identification plate is located on the bottom side of the AH.



Figure 12: AH identification plate with factory default IP address

In addition, the factory default subnet mask is 255.255.255.0.

If the factory default IP address is not suitable for accessing the AH, then it can be changed using the console. If the IP address is correct, enter it in the address bar of your web browser.

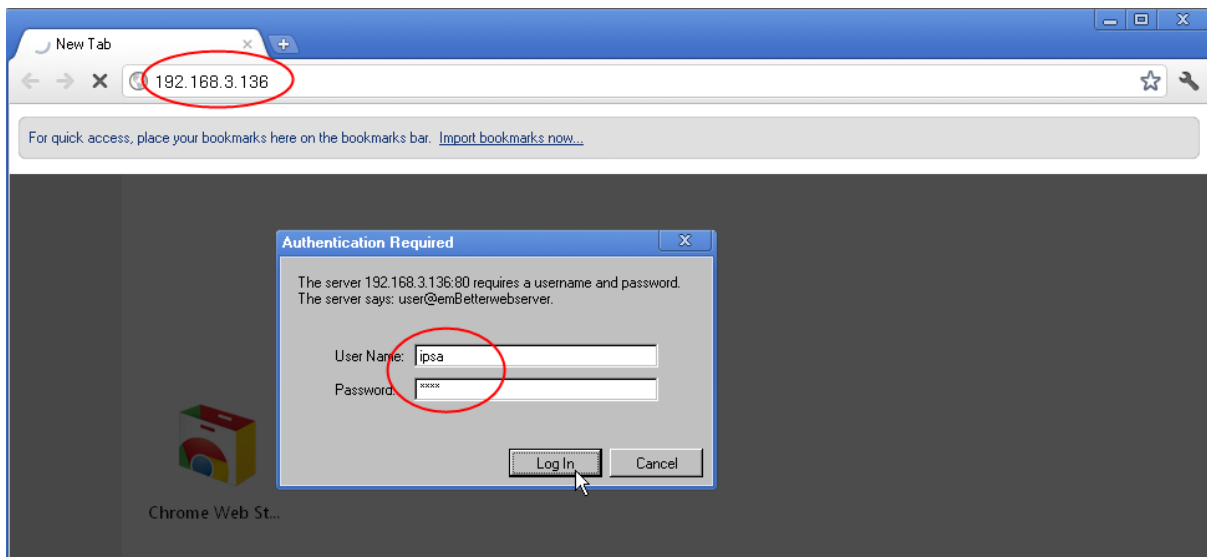


Figure 13: Accessing AH configuration page with a web browser

The user name is “ipsa” and the password is “1ps4”.

2.2.1.2.2 Settings for the pilot installations

The following settings for the pilot installations must be configured and verified for every AH:

1. AH ID
2. GSM/GPRS settings
3. Sensor list update URL
4. Sensor data URL
5. Firmware update URL

The AH ID was generated during the base station set-up. This ID must be entered in the field “ActivityHub ID”:

ActivityHub Settings	
Name	capt2collect
ActivityHub ID	SIG_2

Figure 14: AH ID setting

Next configure the GSM/GPRS settings according to the requirements of your provider:

2.2.1.2.3 Connection via GSM/GPRS

GSM Settings	
APN	web.abc.de
PIN	1234
Username	user
Password	password

Figure 15: GSM/GPRS settings

Settings name	Description
APN	The APN that the GPRS provider requires for outgoing connections. Please be very careful, that the correct APN is selected. When the wrong APN is selected, the GPRS modem is able to connect but the TCP connection to the server may not work.
PIN	Four digit PIN number of the SIM card. In case the PIN on the SIM card is deactivated, the value of this field does not matter. In case the PIN is wrong, this is indicated by the AH by setting the field automatically to the value “!wrong:abcd” In this case the correct pin can be entered. If the PIN was entered wrongly three times, the field is set to “!PUK required”, if no PUK is available. Then the SIM card must be removed from the AH and unlocked with another device such as a mobile phone.
Username	The user name the provider requires for authentication. Most providers do not require a user name for GPRS.
Password	The password required for GPRS communication. Most providers do not require a password.

Table 1: GPRS settings

The next settings concern the URL's for the communication with the LinkSmart base station. The sensor list update, the sensor data and the firmware update URL must be correctly specified. By default, these settings are correctly set by default in the firmware. Nevertheless it is required that the settings are verified. The following image shows the correct settings to be used for the inCASA pilot phase.

The screenshot displays three configuration sections for the AH settings:

- Sensor data upload:**
 - Server: hydra.cnet.se
 - Port: 8090
 - Page name: /
- Sensor list update:**
 - Server: hydra.cnet.se
 - Port: 8091
 - Page name: /sensorlist
- Firmware update:**
 - Server: hydra.cnet.se
 - Port: 8092
 - Page name: /

Figure 16: AH settings for the communication with the LinkSmart base station

Finally store all settings by pushing the button “Store”. This will save the settings in non-volatile memory to remain also during a power-cycle of the AH.

This screenshot shows the 'Firmware update' section of the settings interface. The 'Store' button is circled in red, and a mouse cursor is pointing at it. Below the 'Store' button is a 'Base station test' button.

Figure 17: Store the settings

2.2.1.3 Loading the sensor list

With the AH configured so far, it must be possible that the sensor list from the server can be loaded.

To verify whether the sensor list was loaded, open the page “Sensor List” on the AH:

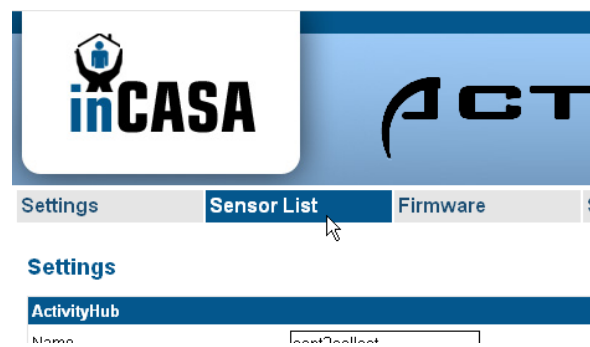


Figure 18: Sensor List link on AH web page

Initially it might be that the sensor list is empty. It takes a while until the GPRS connection is established and active. When the GPRS connection is established, the GSM and the GPRS LED's are lit. If an error occurs the LED ERR blinks. Reload the sensor list page, if no sensors are listed. After 1-2 minutes the sensors as configured must appear.

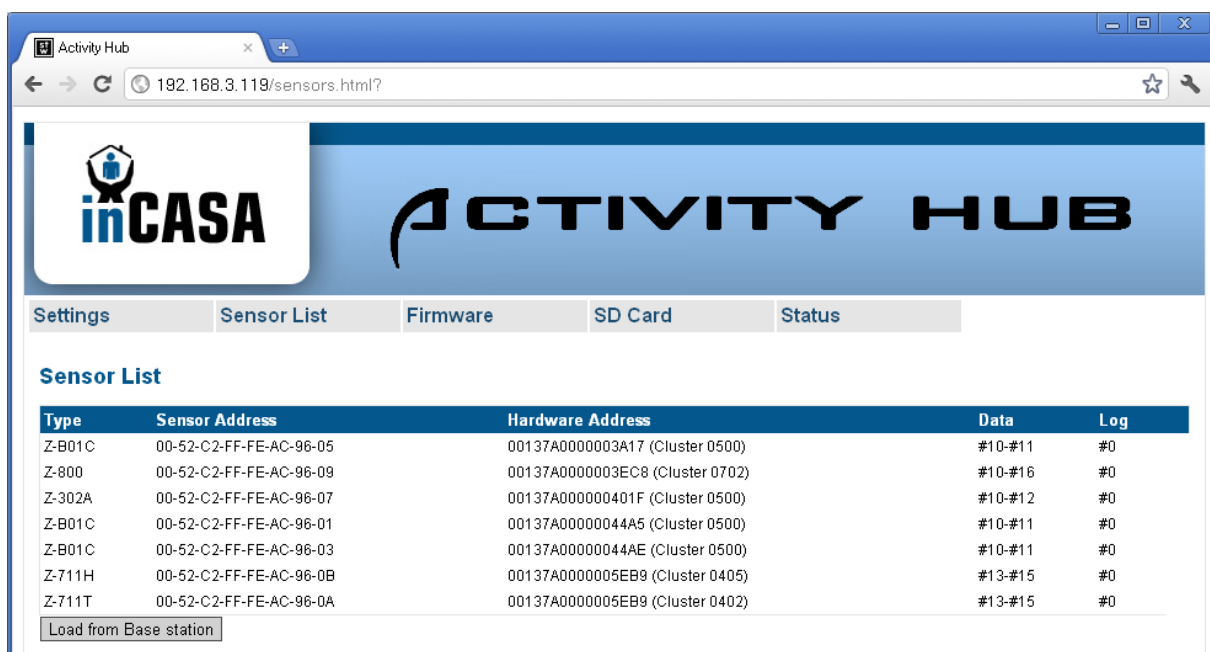


Figure 19: Sensor list loaded on AH

The status of the Activity Hub is on the status page. This is useful for checking for errors of the configuration.

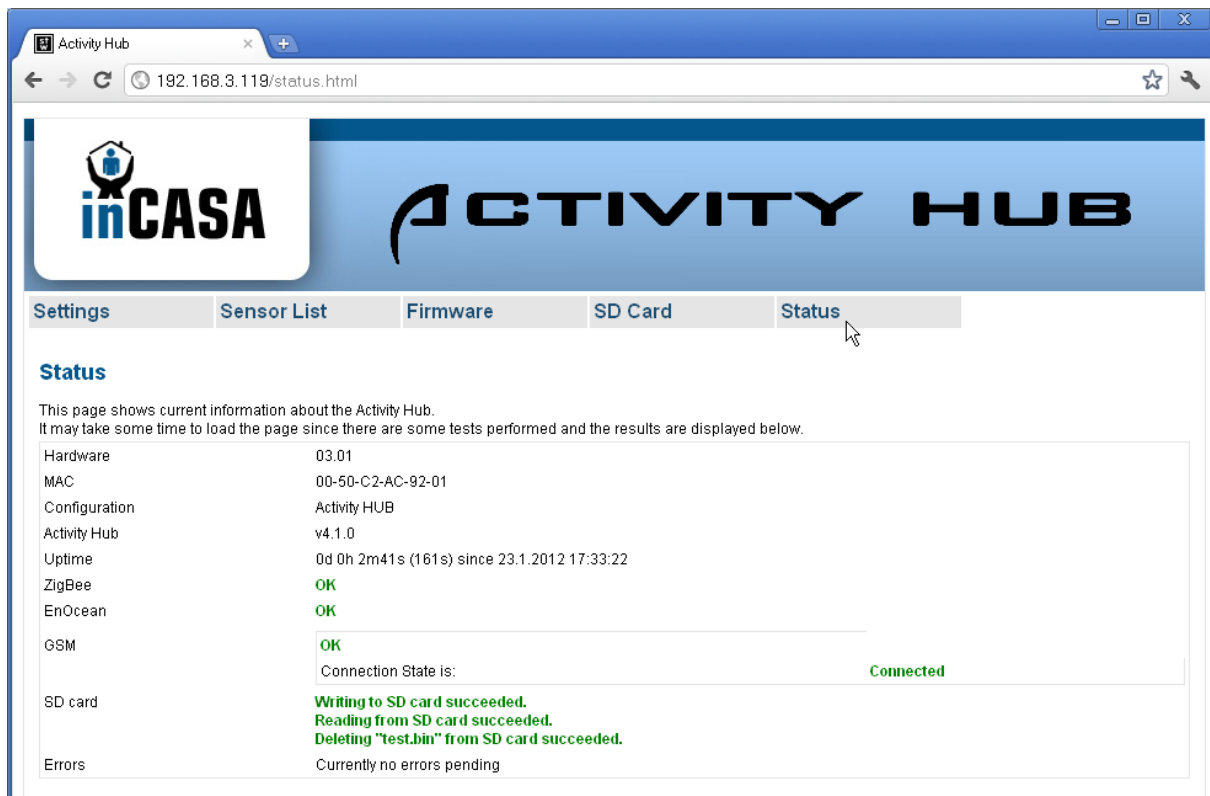


Figure 20: Activity Hub status page

The status indicates whether the configuration is set properly. The example given in Figure 20 shows the test page of a successfully configured and running AH. No errors are listed.

2.2.1.4 Adding ZigBee sensors to the network

2.2.1.4.1 Introduction

This chapter describes what needs to be done for every sensor to associate it with the AH. It describes first what needs to be done to reset a sensor to factory default settings. This is needed only, when the sensor was associated to another device already. In case these are new sensors, the sensor does not need to be reset. These are the sensors:

2.2.1.4.2 Z-B01C Motion Sensor

2.2.1.4.2.1 Reset the sensor

Remove the batteries or the power supply. Press the binding key while restoring power. If it is restored correctly, the green LED flashes 10 times at high frequency (~ 3 Hz).



Figure 21: Z-B01C motion sensor binding key

Note: at least two out of 5 sensors of this type showed problems with the power connector. They have a loose contact and do not power up correctly. In case the sensor is not working, please check whether it is powered up correctly or use batteries instead of the power supply.

2.2.1.4.2.2 Associate to the AH

Power cycle the sensor. When the sensor is powered again, the “Network indicator” LED flashes 5 times to indicate successful association with the AH. If it flashes only 2 times, association failed.

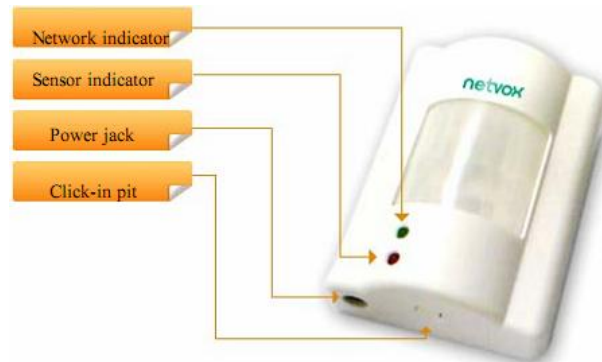


Figure 22: Z-B01C motion sensor front

2.2.1.4.3 Z-302A Window/door sensor

2.2.1.4.3.1 Reset the sensor

Press and hold the auxiliary key first for at least one second and then ADDITIONALLY press the binding key until you see the red LED flash 2 times and after a pause 10 times.



Figure 23: Z-302A window/door sensor

2.2.1.4.3.2 Associate to the AH

Press the binding key once. If the sensor correctly associates with the AH, the green “Status indicator” LED flashes 6 times.

2.2.1.4.4 Z-711 Temperature/Humidity sensor

2.2.1.4.4.1 Reset the sensor

Remove the batteries. Press the binding key while restoring power. If it is restored correctly, the green LED flashes quickly (~3 Hz).

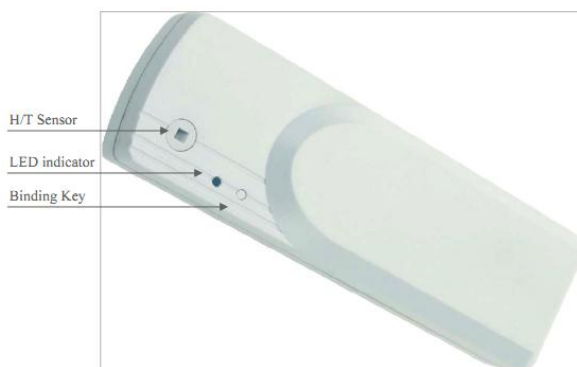


Figure 24: Z-711 Temperature/Humidity sensor

2.2.1.4.4.2 Associate to the AH

Power cycle the sensor. Once it is associated to the AH, the “LED indicator” flashes 5 times.

2.2.1.4.5 Z-801 WLS water detection sensor

2.2.1.4.5.1 Sensor configuration

Before using the Z-801 WLS device, you need to add a sensing plate. Note that only port 1 is active, port 4 – 5 are not activated. It might be possible to add multiple plates in parallel to port 1. However, this has not been tested. Please note that you have to connect GND to the black wire of each plate. In Figure 25, an example of a possible connection of a single plate is shown. The Z-801 WLS will generate an alarm event if the plate gets wet (short-circuit). You can test the behaviour by short-circuit port pin 1 with the GND pin using a wire.

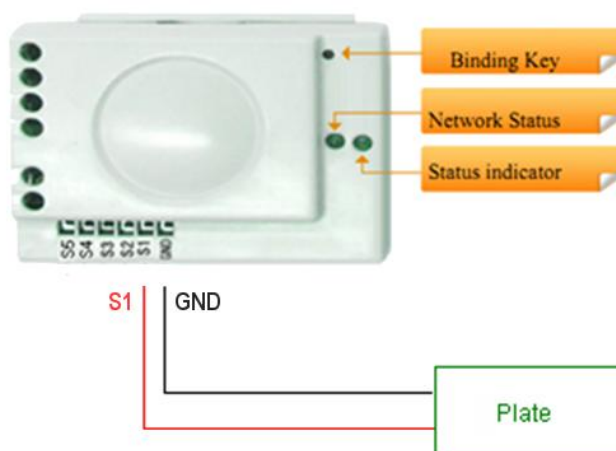


Figure 25: Z-801 WLS plate connection example

2.2.1.4.5.2 Reset the sensor

Open the cover in the back of the sensor and remove the batteries. Press the binding key while restoring power. If it is restored correctly, the green “Network Status LED” flashes quickly (~3 Hz).

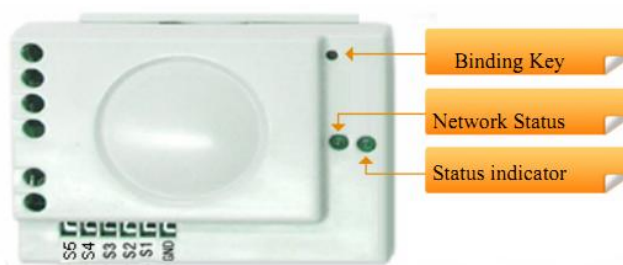


Figure 26: Z-801 WLS water detection sensor

2.2.1.4.5.3 Associate to the AH

Power cycle the sensor. The “Network Status” LED flashes 5 times.

2.2.1.4.6 Z-800 Power Outlet sensor

2.2.1.4.6.1 Reset the sensor

Remove the Z-800 from the power socket. Press the binding key with a non-conductive pin while plugging the sensor into the power socket again. If the device is restored correctly, the green LED flashes quickly. To set the device in operation mode, please power cycle it.



Figure 27: Power Outlet Device

2.2.1.4.6.2 Associate to the AH

Put the sensor into a power plug. The green “indicator light” will be turned on when successfully associated with the AH. If the “indicator light” is blinking, association to the AH was not possible.

2.2.1.4.7 Troubleshooting

For troubleshooting the association of a sensor to the network, the registration indication through the LED needs to be observed. The following table lists the indications:

Device	On success	Failed	No Zone device found
Z-B01C	6 flashes	4 flashes	2 flashes
Z-302A	6 flashes	4 flashes	2 flashes
Z-801	6 flashes	2 flashes	2 flashes

Table 2: IAS Zone Registration

Error cases:

1. The IAS Zone registration was not successful. Solution(s):
 - a. Wait for the next try of the Netvox sensor to register. This happens immediately after the first try. The green LED should flash as described in Table 2.
 - b. This happens if a timing problem occurs. Please register only one NetVox device simultaneously.
2. No IAS Zone device (activity hub) was found. Solution(s):

- a. Restart the sensor.
- b. Restart the hub.
- c. Check if the problem exists with another sensor of the same manufacturer and the same type, too.

2.2.1.5 Adding EnOcean based sensors to the network

For the pilot installations the following two EnOcean based sensors are supported

- Chair sensor from Funkstuhl
- Matcontrol (bed mattress sensor)

These sensors do not need any association process to the AH network. If the sensor is within the radio range of the AH, the sensor events are logged by the AH.

It is important that they are listed with their correct hardware address and are added to the sensor list.

For finding out the hardware address of a new sensor, since it is normally not printed on the sensor itself, generate an event and open the page “Sensor List” of the Activity Hub in the web browser. The page then shows the hardware address of the latest event that was received.

The correct operation can be verified by generating an event (e.g. pressing the chair sensor) and to check whether events are logged on the sensor list (see chapter 2.2.1.6).

2.2.1.6 Check sensor data on the AH

If you added a device correctly, you will receive data logs. You can see them at the activity hub web page. Please note, that you either need to wait the report interval or for an event to receive logs.

Sensor List

Type	Sensor Address	Hardware Address	Data	Log
Z-302A	00-52-C2-FF-FE-AC-9E-19	00137A0000004036 (Cluster 0500)	#10-#12	#3
E-CS	00-60-00-00-00-00-00-01	001F5E37	#1-#2	#4

Load from server

Figure 28: Activity Hub sensor event log

2.2.2 SARA Client

The SARA client is one of the Telehealth clients used in some of the pilots. This is integrating within a portable PC together with the Activity Hub (i.e. Telecare client) and the LinkSmart (i.e. Middleware) in order to provide a compact solution for the patient.

2.2.2.1 SARA Installation

SARA client is a Windows executable file that can be installed in a straightforward way. Once the program is installed in a PC, there are some configurations that must be done before the inCASA system can be used.

2.2.2.2 Sensors Attachment

First of all, we must activate the Windows Bluetooth. In order to guarantee that Bluetooth is activated is by checking that the BT icon appears in the right corner on below the screen. The next picture shows how it looks like:

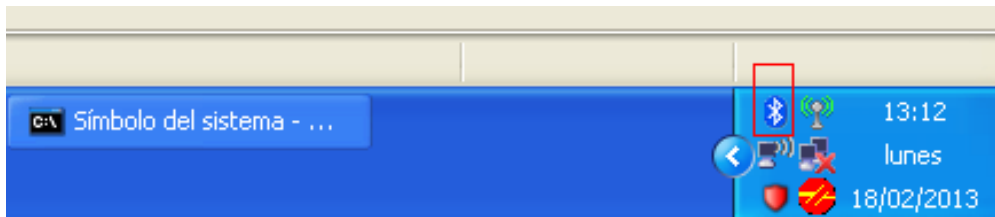


Figure 29: Windows Bluetooth icon

Once we have SARA installed in the patient's processing unit (e.g. PC), the next step is associating the Bluetooth device. In order to do this task, we use the Bluetooth Manager provided by Windows. First of all, the sensor must be switched on so that the manager can recognise it. The next picture shows how Windows tries to search all the Bluetooth devices surrounding the PC:

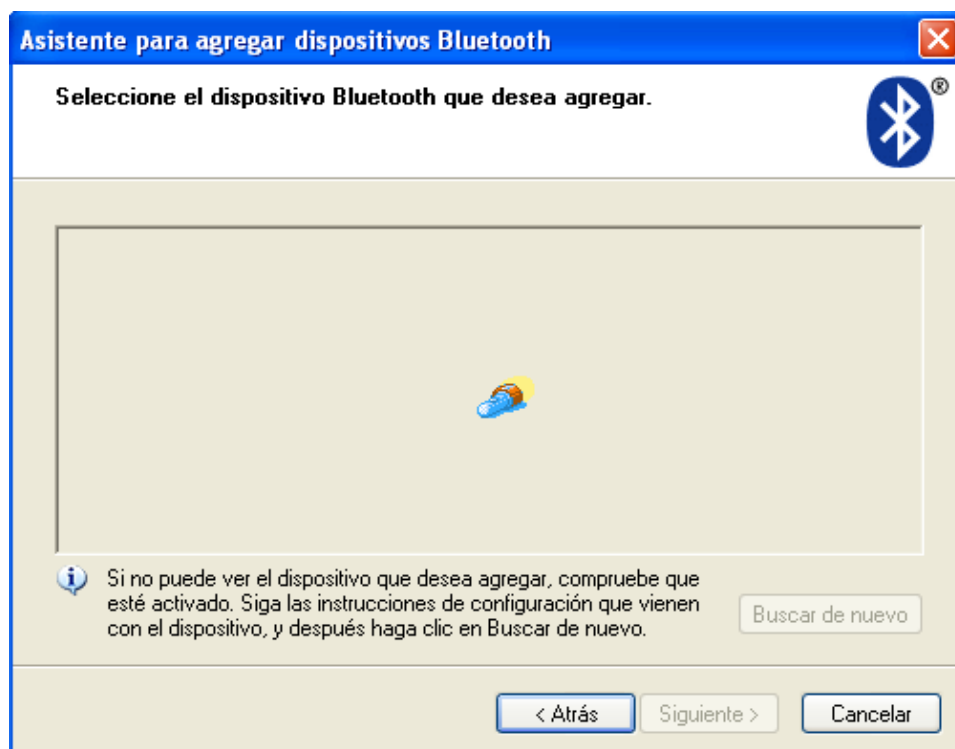


Figure 30: Searching Bluetooth devices

Eventually, the Bluetooth Manager finds our sensor and we can set it up once for all so that SARA can use it.

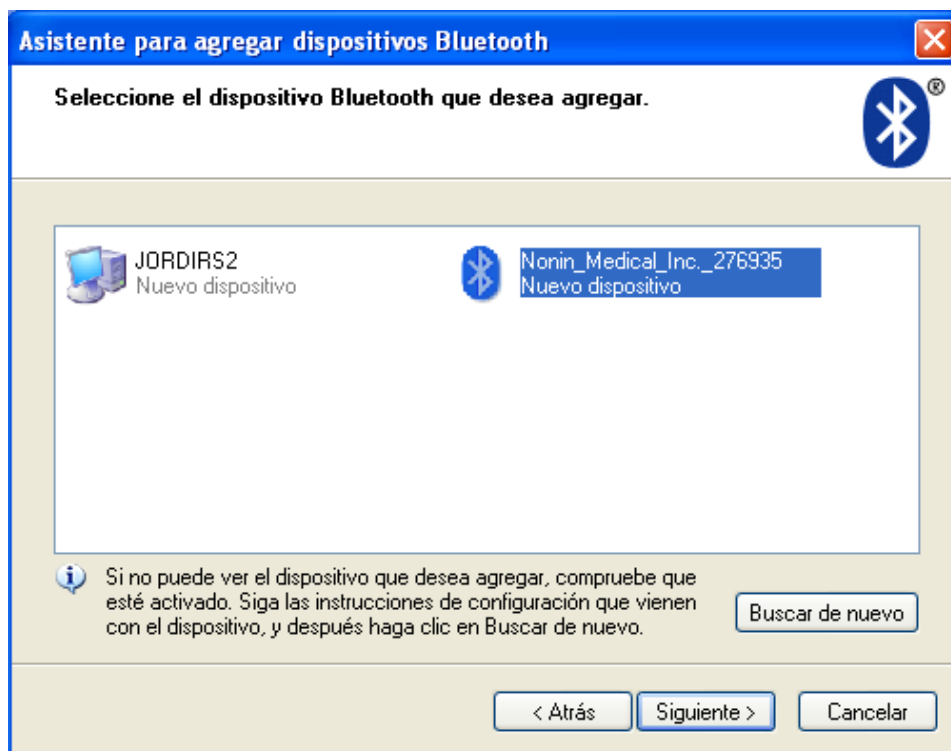


Figure 31: Detecting our sensors

Once done that, we check out that the Bluetooth communication works fine with the sensor. In order to do that, we use our own tool (i.e. sensor dll) in order to force a communication with the sensor. The way to do is by simply executing the dll in a DOS screen while taking a measure with the sensor. If the sensor is set up correctly, measures should start to appear in the screen. The following picture shows what the result of this operation:

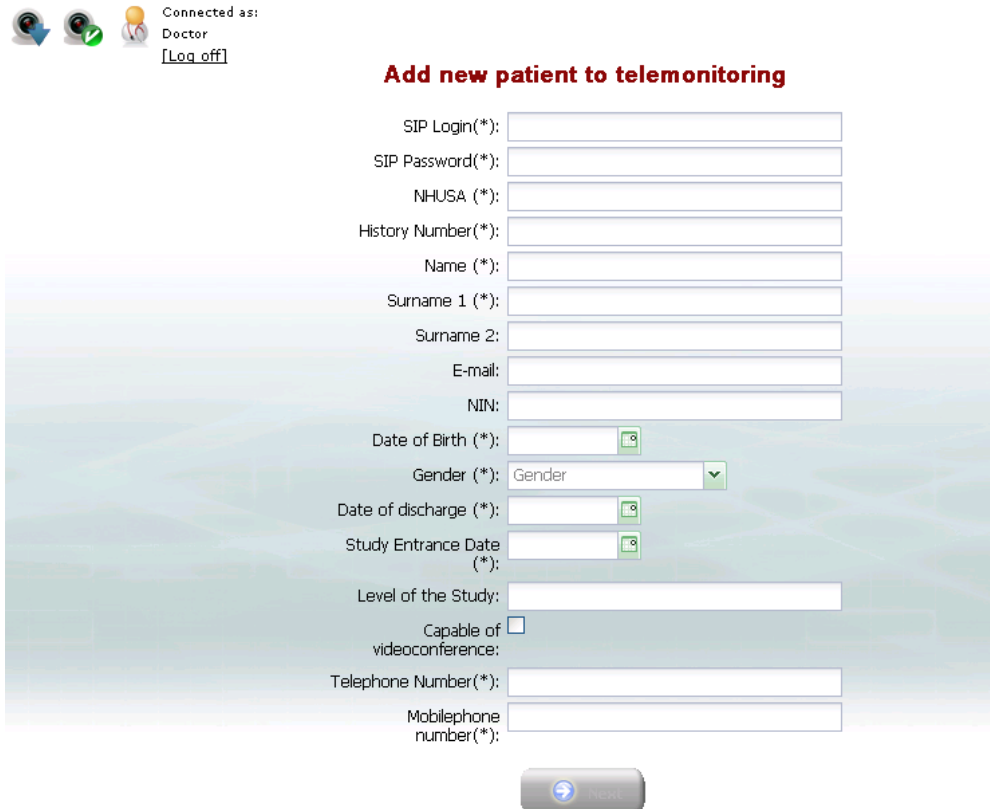
```
C:\enviar>win32 gtnonincustom.xml
GTLIBTest v1.0
Connection Type: BT REMOTESERVICE (DeviceName)
Remote Device Found: Nonin_Medical_Inc._276935
Remote Device Address: (00:1c:05:00:59:02)
Windows Version: 5.1
Pin Authentication Callback started...
Socket Result: Connect OK
Sensor 0: Conexión Inicializada, esperando datos.
<- DATOS Sensor: 0 tramas: 1
Trama: 0 Attributes: 4 Type:0
Name=customheartrate type=1 value=76
Name=customspo2 type=1 value=94
Name=lowbat type=1 value=0
Name=custombatterypo type=1 value=100
<- DATOS Sensor: 0 tramas: 1
Trama: 1 Attributes: 4 Type:0
Name=customheartrate type=1 value=76
Name=customspo2 type=1 value=94
Name=lowbat type=1 value=0
Name=custombatterypo type=1 value=100
```

Figure 32: Checking the connection with the sensor

If we see what measures are transmitted then it means that the communication between sensor and PC is established successfully. The next step is setting up the patient's details so that information is sent to the inCASA system. That is explained in the next section.

2.2.2.3 Patient Set up

Before setting up a patient, we must make sure that they are already created in the inCASA system. This work must be done by a clinical professional. To add a new patient, it is necessary to fill out the next form:



Connected as:
Doctor
[\[Log off\]](#)

Add new patient to telemonitoring

SIP Login(*):

SIP Password(*):

NHUSA (*):

History Number(*):

Name (*):

Surname 1 (*):

Surname 2:

E-mail:

NIN:

Date of Birth (*):

Gender (*):

Date of discharge (*):

Study Entrance Date (*):

Level of the Study:

Capable of videoconference: ☐

Telephone Number(*):

Mobilephone number(*):

Figure 33: New patient inclusion form

The fields marked by an asterisk are mandatory. As well as this form, the clinical professional must indicate the treatment for the patient (i.e. medication, measures intake, etc.) and the frequency patients must follow (e.g. every 8 hours, every 12 hours, etc.).



inCASA

Connected as:
DoctorFR
[\[Déconnexion\]](#)

Enregistrer nouveau Traitement

Treatment configuration

Pulse Oximetry Control

Pulse Oximetry Control

Treatment component	Dernière modification	Cor
Blood Pressure Control		
Weight Control		

Figure 34: Choosing treatment

And finally a KIT identifier is chosen for the patient. This identifier will be the inCASA id that must be set up in the patient's application.



Figure 35: Choosing inCASA id

Now when the patient has been created, we are able to set up the equipment at patient's home. The only tasks we have to perform as well as installing SARA and synchronize the Bluetooth sensor is adding the inCASA id (i.e. Kit user) and other relevant information (i.e. url services, password for security, etc.) in the SARA's configuration tool.

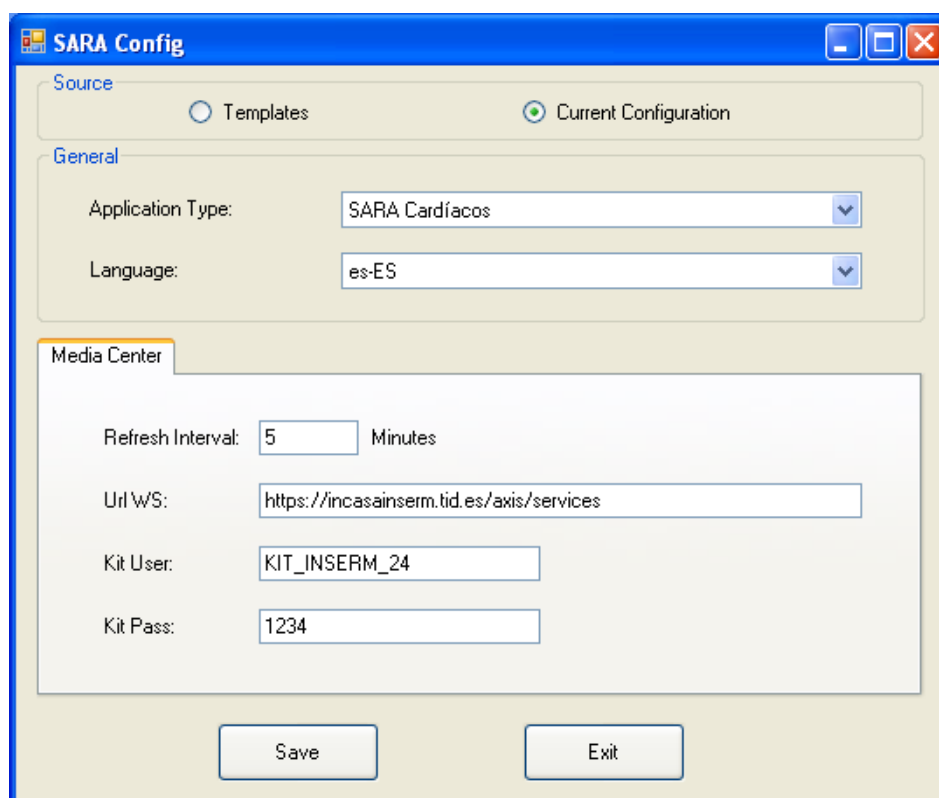


Figure 36: SARA's configuration tool at patient's home

Once the patient has been set up, we are ready to start up the SARA application. The first thing will appear is a screen updating the configuration for that specific patient (i.e. daily medical agenda and types of measures she/he has to take).

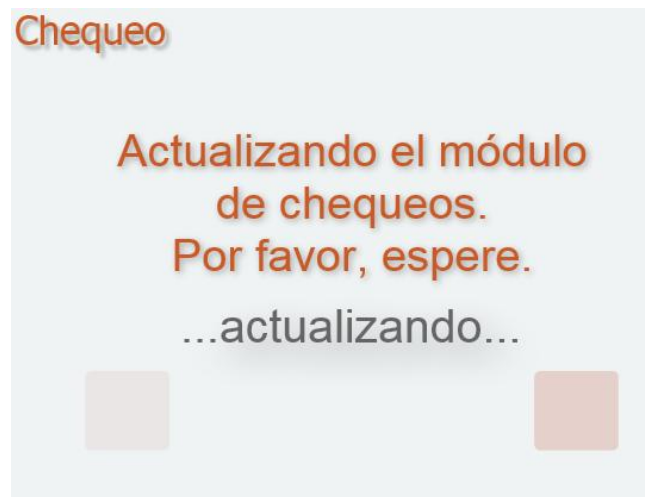


Figure 37: UI showing and displaying updating patients' application

Once the information of that patient is updated, the measures associated to them are shown in the screen (i.e. weight, blood pressure, pulse-oximetry, etc.). The next picture shows the patient measurements of their weight (i.e. Peso in Spanish).

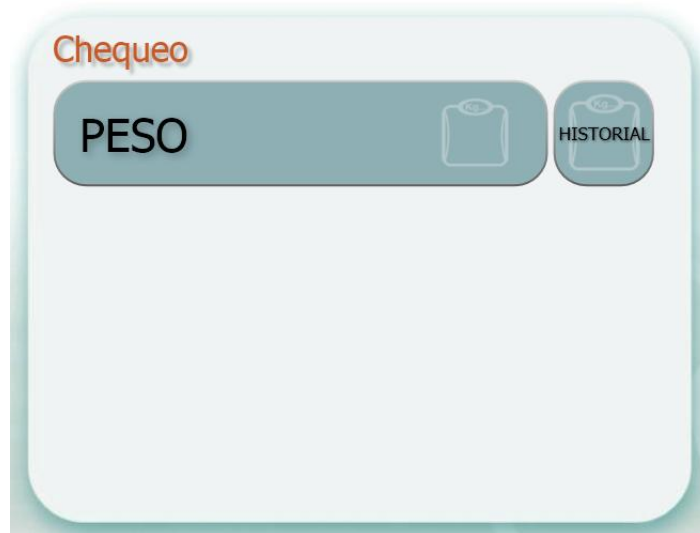


Figure 38: Showing measures associated to a specific patient

From this point on, the system is ready to send telehealth information from the telehealth client (i.e. SARA client) to the inCASA system. However, in order to complete the setup, the middleware must be installed too to merge contributions from the telehealth client (i.e. SARA) and the telecare client (i.e. Activity Hub).

2.2.3 LinkSmart Middleware

This subchapter will explain how to set up a running LinkSmart middleware environment. The description applies to both for client and backend installations where applicable.

2.2.3.1 Java

1. Install Java v 1.6 32-bit.
2. Copy local_policy.jar and US_export_policy.jar to "C:/Program Files(x86)/Java\jre6\lib\security"
3. Make sure the environment variables Path and JAVA_HOME points to your Java installation folder (e.g. ...\\Java\jre6\bin). Locate the environment variables through "/Control Panel/System/Advanced system Setting/Advanced/Environment Variables".

Otherwise create a new environment variable (Variable = JAVA_HOME, Value = "C:\Program Files (x86)\Java\jre6\bin").

2.2.3.2 LinkSmart NetworkManager

1. Move the whole LinkSmart folder to preferable path on "C" drive which would look like this:

- C:\LinkSmart\LinkSmart
- C:\LinkSmart\IoTSmartControlPoint
- C:\LinkSmart\OSGIServiceWrapper
- C:\LinkSmart\LinkSmartDotNetServices

2. Run clean.bat first to make sure the LinkSmart is fresh. Edit the run_LinkSmart.bat And check if the paths are correct for the installation (both LinkSmart and Java).

```
cd "C:\LinkSmart\LinkSmart"
"C:\Program Files (x86)\Java\jre6\bin\java.exe" -jar
org.eclipse.osgi_3.6.2.R36x_v20110210.jar -console
```

Then run run_LinkSmart.bat and wait for a while. To be sure that LinkSmart started go to <http://127.0.0.1:8082/NetworkManagerStatus>

NETWORK MANAGERS	DESCRIPTION	HOST	ENDPOINT
138.10.155.16	NetworkManager:BOWIE	192.168.9.199	http://localhost:8082/axis/services/NetworkManagerApplication
122.310.130.120	NetworkManager:PeterRosHome	192.168.9.91	-

LOCALHIDS	DESCRIPTION	HOST	ENDPOINT
0.0.0.4206494034288035084	Des = TrustManager; SID = TrustManager PID = TrustManager;	192.168.9.199	http://localhost:8082/axis/services/TrustManager
0.0.0.423967622293605181	Des = OntologyManager; SID = OntologyManager PID = OntologyManager;	192.168.9.199	http://localhost:8082/axis/services/ApplicationOntologyManager
138.10.155.16	NetworkManager:BOWIE	192.168.9.199	http://localhost:8082/axis/services/NetworkManagerApplication

HID	DESCRIPTION	HOST	ENDPOINT
0.0.0.6896167717173301557		192.168.9.199	http://localhost:8082/axis/services/LinkSmartConfigurator
0.0.0.4206494034288035084		192.168.9.199	http://localhost:8082/axis/services/TrustManager

Figure 39: LinkSmart NetworkManager status page

On this machine the Network Manager is set to BOWIE. If running LinkSmart pointed to a super node other Network Manager will be visible also pointed to that super node.

3. When running LinkSmart for the first time go to <http://127.0.0.1:8082/LinkSmartStatus> and choose [eu.LinkSmart.network](http://127.0.0.1:8082/LinkSmartStatus)

Scroll down to "Describe your Network Manager Instance" and name your NetworkManager, e.g. NetworkManager:MyNetworkManager. Make sure it is unique in the network.

Type new HID for your NetworkManager by using regular expression $^{\wedge}.\d{1}.\d{1}.\d{1}.\d{1}$ (e.g. 12.34.56.78)

Scroll to bottom of the page and press "Update Configuration" button.

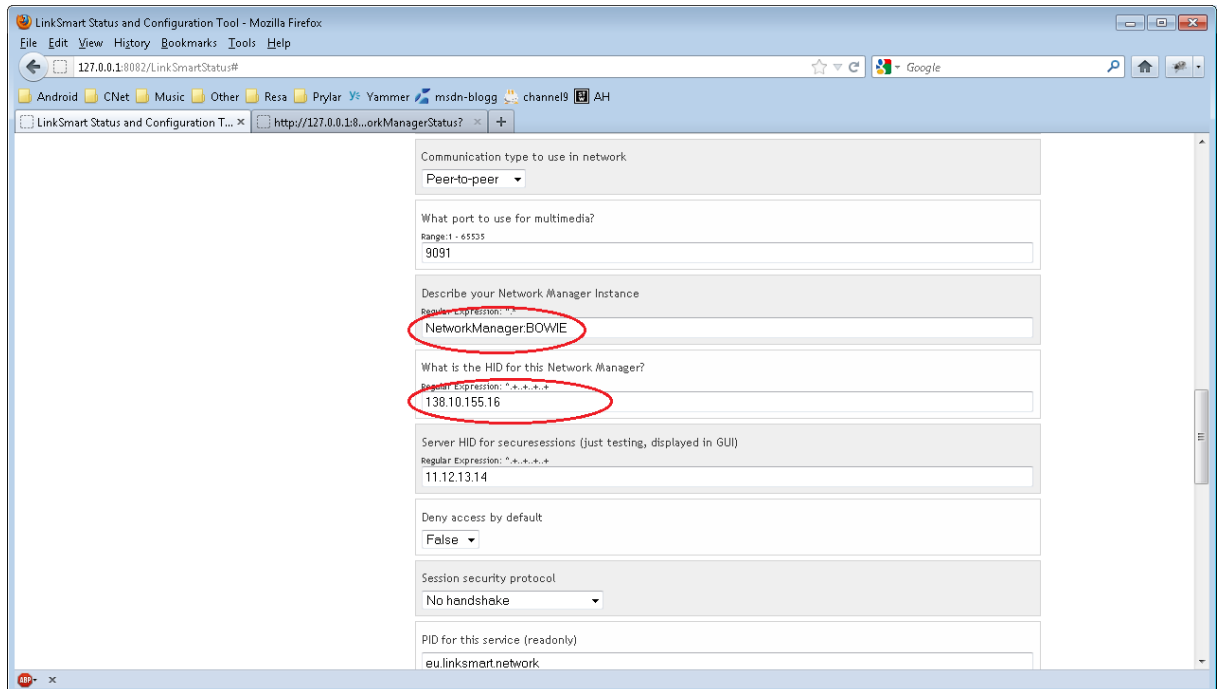


Figure 40: Naming the LinkSmart instance

4. When LinkSmart is running start the IoTSmartControlPoint.exe. This helps finding other NetworkManager and IoT-Device in the network.
5. To change the super node to communicate with or only run locally. Change seeds.txt located at C:\LinkSmart\LinkSmart\NetworkManager\config

The super node located at CNet has
 tcp://212.214.80.136:9101
 http://212.214.80.136:9100

to run locally just replace the above with

tcp://127.0.0.1:9101
 http://127.0.0.1:9100

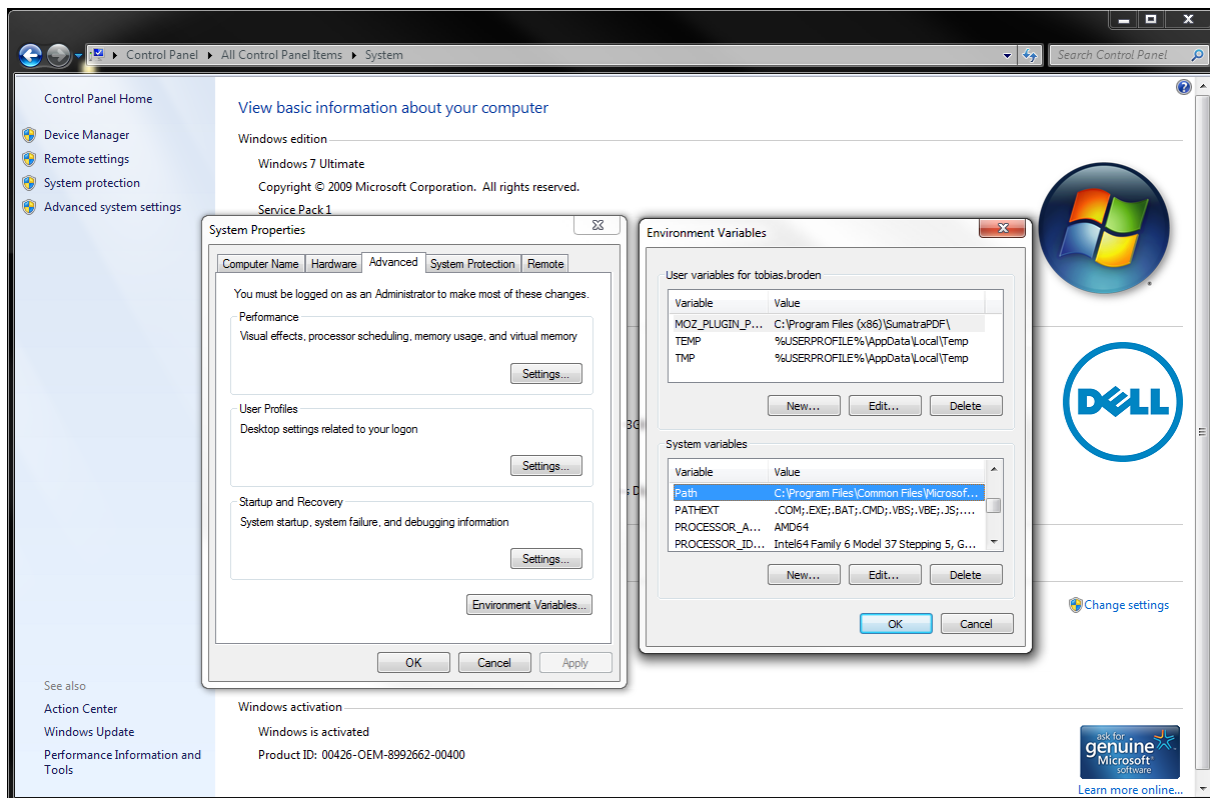


Figure 41: Setting Java path in Environment Variables

2.2.3.3 Auto start of LinkSmart and applications

To enable LinkSmart, IoTSmartControlPoint and applications to start automatically when the computer starts and keeps running when user logs out, OSGIServiceWrapper and LinkSmartDotNetServices can be used. The two services are added as Window Service and start the other executables. To install the services a tool named installutil.exe is need, this program can be located in the Microsoft.Net Framework (C:\Windows\Microsoft.NET\Framework\v4.0.30319⁵).

2.2.3.3.1 OSGIServiceWrapper

1. Open the command prompt as administrator and change directory to the folder for OSGIServiceWrapper ("C:\LinkSmart\OSGIServiceWrapper").
2. Run `C:\Windows\Microsoft.NET\Framework\v4.0.30319\installutil OSGIServiceWrapper.exe`

⁵ Version depending on installed framework on the system.



```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug

C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug>C:\Windows\Microsoft.NET\Framework\v4.0.30319\installutil OSGIServiceWrapper.exe
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.1
Copyright (c) Microsoft Corporation. All rights reserved.

Running a transacted installation.

Beginning the Install phase of the installation.
See the contents of the log file for the C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe assembly's progress.
The file is located at C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.InstallLog.
Installing assembly 'C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe'.
Affected parameters are:
  logtoconsole =
  logfile = C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.InstallLog
  assemblypath = C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe
Installing service OSGIServiceWrapper...
Service OSGIServiceWrapper has been successfully installed.
Creating EventLog source OSGIServiceWrapper in log Application...

The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe assembly's progress.
The file is located at C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.InstallLog.
Committing assembly 'C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe'.
Affected parameters are:
  logtoconsole =
  logfile = C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.InstallLog
  assemblypath = C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug\OSGIServiceWrapper.exe

The Commit phase completed successfully.

The transacted install has completed.

C:\Users\tobias.broden.CNET\My Projects\LinkSmart_Installation_Kit\OSGIServiceWrapper\Debug>

```

Figure 42: Adding service to OSGIServiceWrapper

3. To see if the service has been added to the Windows service, open *Control Panel\Administrative Tools\Services* and locate OSGI Service Wrapper. Change start-up type from manual to automatic by right-click on the service and click properties and locate the drop down menu for Start-up Type.

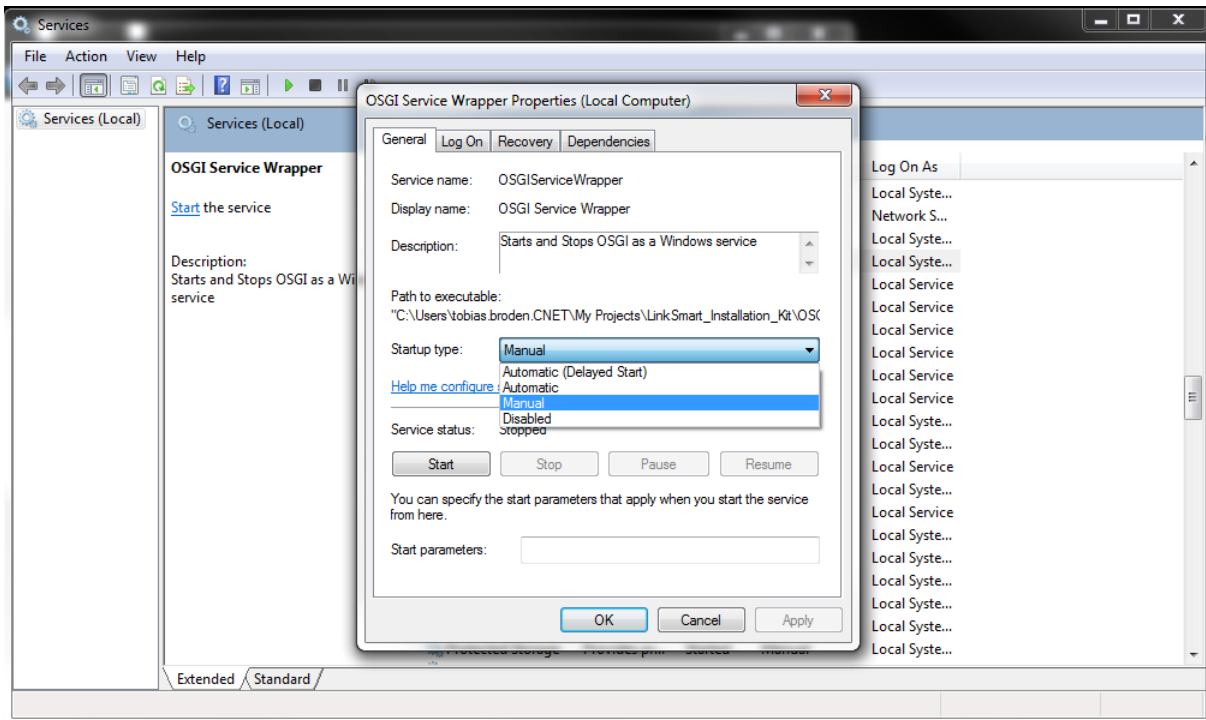


Figure 43: Automating the OSGIServiceWrapper at start-up

4. Locate OSGIServiceWrapper.exe.config in the OSGIServiceWrapper folder and make sure the configuration points to the java.exe folder and LinkSmart installation and .jar file.

```
<setting name="JavaHome" serializeAs="String">
  <value>C:\Program Files (x86)\Java\jre6\bin\</value>
</setting>

<setting name="BundleNameAndPath" serializeAs="String">
  <value>"C:\LinkSmart\org.eclipse.osgi_3.6.2.R36x_v20110210.jar"</value>
</setting>
```

2.2.3.3.2 LinkSmartDotNetService

Open the command prompt as administrator and change directory to the folder for LinkSmartDotNetService ("C:\LinkSmart\LinkSmartDotNetService").

Run *C:\Windows\Microsoft.NET\Framework\v4.0.30319\installutil LinkSmartDotNetServices.exe*

Change Start-up type in the *Control Panel\Administrative Tools\Services* by locating LinkSmartDevices Dot Net Service.

Locate LinkSmartDotNetServices.exe.config in the LinkSmartDotNetServices folder and make sure the configuration is pointed to the IoTSmartControlPoint, IoTSmartControlPoint.exe and LinkSmartDeviceProcesses.xml.

```
<setting name="SmartControlPointHome" serializeAs="String">
  <value>C:\LinkSmart\IoTSmartControlPoint\</value>
</setting>
<setting name="SmartControlPointExe" serializeAs="String">
  <value>IoTSmartControlPoint.exe</value>
</setting>
<setting name="HydraDeviceProcessXML" serializeAs="String">
  <value>C:\LinkSmart\LinkSmartDotNetServices\LinkSmartDeviceProcesses.xml</value>
</setting>
```


Configure the LinkSmartDeviceProcesses.xml to start other application such as MedicalDevice.exe. Name the executable and the path. To add more executables add one more process with id = 2, 3...

```
<?xml version="1.0" encoding="utf-8" ?>
<processDefs>
  <process id="1" exe="LiveMedicalDevice.exe"
homeDirectory="C:\LinkSmart\Applications\LiveDeviceDiscoveryManager"
sleepTimeAfterStartSeconds="10"/>
</processDefs>
```

2.2.3.3.3 Debug: Extend with more errors

Problem: java.security.KeyStoreException

Error: "java.io.IOException: Error initialising store of key store: java.security.InvalidKeyException: Illegal key size java.security.KeyStoreException: java.io.IOException: Error initialising store of key store."

Solution: Copy local_policy.jar and US_export_policy.jar to "...\\Java\\jre6\\lib\\security"

Problem: Dual instances of LinkSmart running? "ERROR 36 [SCR] Exception while activating instance eu.LinkSmart.network.impl.NetworkManagerApplicationSoapBindingImpl@1829d01 of component NetworkManager java.lang.reflect.InvocationTargetException"

"net.jxta.exception.PeerGroupException: Only a single instance of the World Peer Group may be instantiated at a single time."

Solution: Make sure not two instances of LinkSmart are running.

2.2.4 Chorleywood Platform

The overall architecture (Figure 44) of the platform installed to support the pilot in Chorleywood Health Centre (CHC) is based on the reference architecture of Continua Alliance, and exploits the IEEE 11073 PHD standards to define the LAN interface between devices and the home gateway, and IHE-PCD01 to define the WAN interface between home gateway and the Observation WS. Continua Alliance guidelines profile specific use of the standards and interactions between components over the interfaces. This section describes specific implementation details of the components deployed in the inCASA project.

2.2.4.1 Overall Architecture

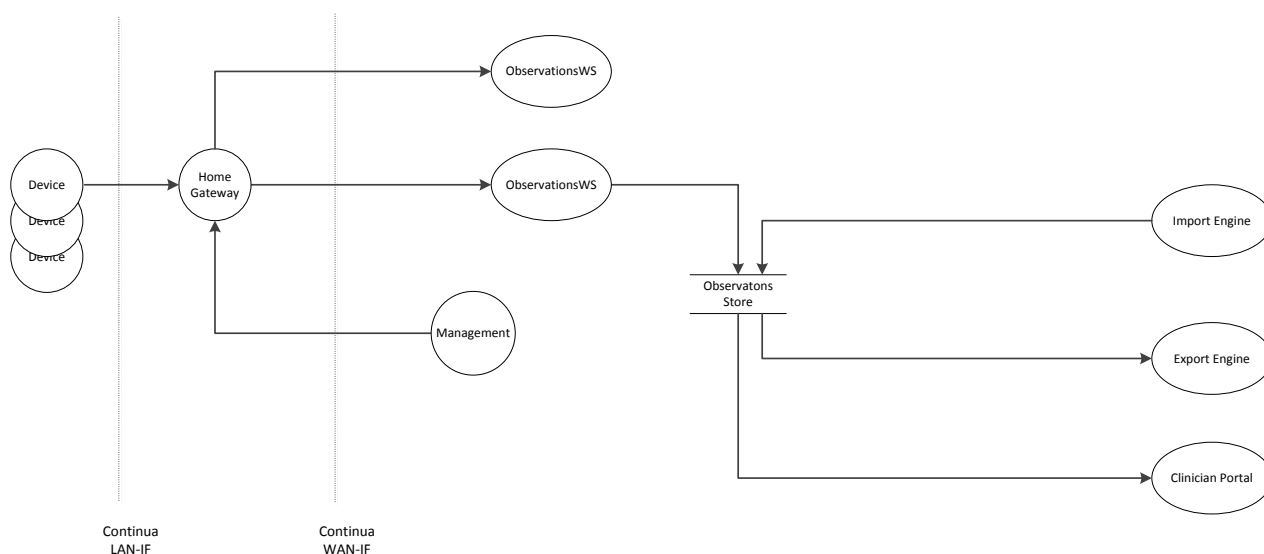


Figure 44: Overall Architecture

The architecture is extended at the backend to include user applications to support: the clinician with patient and data management and to provide tools for patient risk management. This is accomplished through development of a number of specific applications and the integration of a number of components.

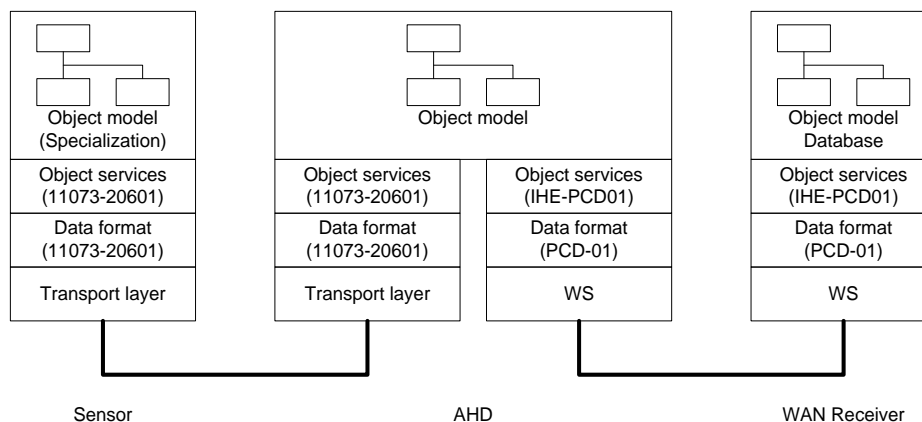


Figure 45: Layered Approach

The architecture is based on the layered approach of Figure 45. Devices are modelled as a technology independent object model using a restricted set of classes and attributes as defined in IEEE 11073-20601, and all aspects are codified using the standard nomenclature of IEEE 11073-10101. Object services and data format are also defined in IEEE 11073-20601. The IEEE 11073-20601 standard is designed to be independent of transport technology, however Continua Alliance define use of Bluetooth Health Device Profile (BT HDP), Bluetooth Low Energy (BT-LE), USB and ZigBee Health Care Profile (ZHCP). At the IEEE 11073 layer, all are interoperable. These are optimised for low power, wireless operation. ZigBee is employed in the platform in CHC in order to support both the telehealth devices (BPM, SpO2, weight, glucose) and independent living (telecare) devices (motion, bed/chair sensor, medication dispenser).

The Application Hosting Device (AHD) (e.g., home gateway) performs a transformation between the IEEE 10073-20601 protocol and IHE-PCD01 used for the WAN interface. IHE-PCD01, based on HL7, is preferred for transmission to the health enterprise as it is transaction based. Both utilise a common underlying object model of the device and its data.

The database is designed around the same object model as IEEE 11073, which allows new types of devices to be added without need for change to the database model. Furthermore, as IEEE 11073 is designed as a plug and play architecture (devices self-describe on association), and the transformation is well defined at object and attribute level in the AHD, there is no need for firmware upgrade to the AHD when a new type of device is introduced to the platform. Change is limited to the applications that utilise the data.

Although applications could be made to run on the AHD to utilise the data, the platform is designed to perform all management of data at the backend in order to support integration of data from other sources (e.g. EPR). This approach also leads to very simple design of the AHD, so that low cost home gateways with no installation can be developed.

2.2.4.2 ZigBee Devices

To meet the needs of the Chorleywood pilot, a range of interoperable devices had to be developed, as these were either unavailable commercially, or presented on individual proprietary interfaces, which would necessitate complex and costly bespoke integration in a hub or back-end. Such an approach might also result in a platform that would be time-consuming and costly to install with each patient, thereby reducing likelihood of adoption in the market. We also took the opportunity to develop devices and home gateway that would be extremely simple to use and thereby suitable for

use by all patients and suit lifestyle. It was most important to develop an interoperable platform able to support monitoring of health and independent living, as changes of habit may indicate changes in health status.

The devices are all based on IEEE 11073-20601 and the respective IEEE 11073-104xx specialisation. So far BPM, weight, SpO2, glucose, PIR motion, bed/chair sensor and medication dispenser have been produced. In order to support both health and environment devices on a single wireless, ZigBee is adopted, and all devices are based on the ZigBee Health Care Profile. The devices are integrated to the home gateway and each device has been tested and certified to be Continua and ZigBee compliant.

ZigBee has numerous advantages:

- It is a mesh network architecture, and so can span an entire home through the use of routers to extend range.
- It is purpose designed to be very low power, and early indications are that battery life of devices such as scales will be years, the radio module falling to 5uA in sleep.
- It uses a robust modulation scheme and has low data rate so has inherently greater link range than Bluetooth.
- It can support practically unlimited devices connected concurrently to the network (BT has a limit of 7 active devices).
- It has excellent network management functionality defined in the standards.
- It is well supported widely available.
- It has been selected to support smart meters.
- Pairing is robust and can be managed remotely.
- Devices can be configured and firmware updated over the air (OTAU).
- A large number of devices can be supported concurrently on the same network.
- ZigBee offers frequencies other than 2.4 GHz (868MHz) which have better propagation properties

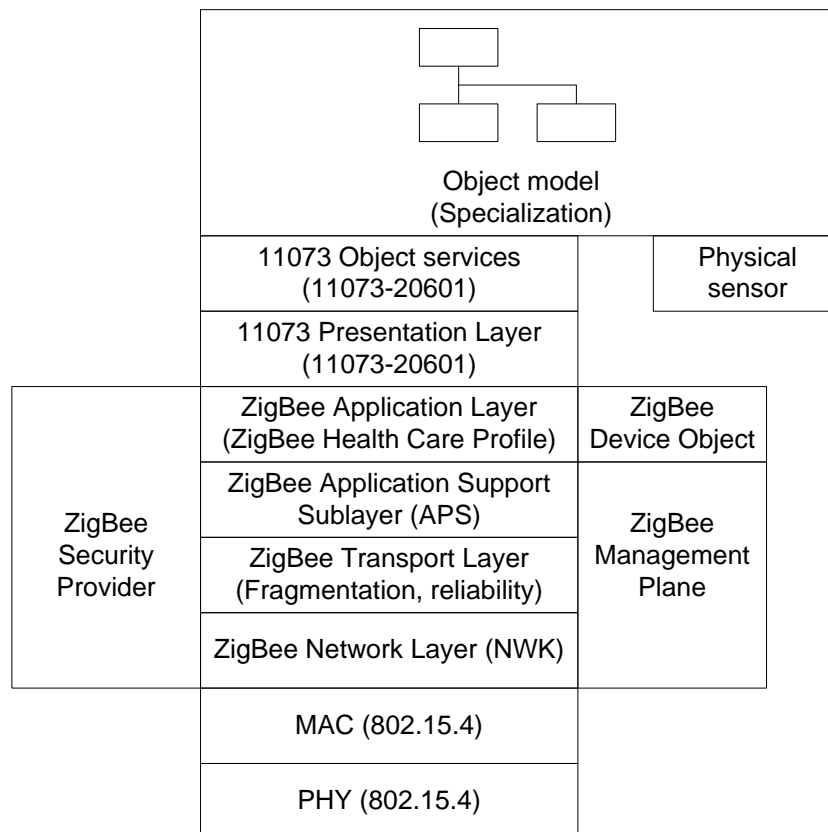


Figure 46: ZigBee Device Architecture

The full architecture of the devices based on ZigBee is shown in Figure 46. Each device contains a full generic implementation of the IEEE 11073 object model, object services (application layer) and presentation layer (MDER). This allows each device to be developed by simple configuration of the objects and attribute values. A ZigBee API allows attributes to be managed dynamically.

The interface to the physical sensor must be developed, and this normally consists of interpreting changing digital signals (switches, etc.) or capturing an interpreting serial data streams.

The configuration and interface routines are constrained to a single file and are largely consistent across devices, making it quick to develop for new sensors in the platform. Generally new object models and simple physical interface may be developed in hours.

The full set of sensors developed to support the project is shown in Figure 47.



Figure 47: inCASA Devices

The devices and gateway that have been developed by Brunel came from another project and have been developed over a 3 year period.

We have included positive indication of data transmission on all the devices to help the user and this has involved adaptation of all the commercial devices include an LED.

2.2.4.3 ZigBee Gateway



Figure 48: CHC Home Gateway

The ZigBee gateway is purpose designed to be simple and rapid to deploy, and this should be sufficiently simple that it could be self-installed by the patient. It is a self-contained unit that plugs directly into the wall socket in the patient home. No further configuration is required. It implements a full AHD with ZigBee wireless for LAN and GPRS module for WAN. WAN connection exploits the TCP/IP stack of the GPRS module.

All inCASA home gateways for the UK pilot are operated on a private M2M mobile network (Arkessa) that connects directly and exclusively with the server. The M2M network ensures security, both authentication (SIM) and encryption through the GSM network and VPN connection.

Future versions will include PCB mounted SIMs to increase physical security of the device. The M2M operator also provides roaming SIM that will work on any network in the UK. This assists with operating in areas of poor signal. The network is operated as private (10.0.0.0) and all gateways have static IP address.

The home gateway was subjected to the “friends and family” test protocol [8]. It was also tested in areas of known poor signal to ensure long term reliable performance.

We monitor devices by use of a “ping” tool on the server that can be configured to send a “ping” to specified devices at regular intervals and log the results. This can help identify trends and patterns in performance of the mobile network and assist in trouble shooting, such as relocating the home gateway in the patient home to a place with improved signal. We have since obtained a GSM meter to assist in finding suitable location in the home.

The home gateway provides a simple red/green light to indicate state of connection to the mobile network and to the server network.

The home gateway acts as the coordinator of the ZigBee network and includes support for all ZigBee network management functionality. This allows remote management of the devices such as status, pairing, connecting, operating parameters, reset, OTAU.

The gateway currently supports a simple TCP/IP serial connection for remote management that interfaces to a command menu. The gateway also supports an internal serial (logic RS232) and a cable may be connected for management and diagnostic purposes.

The gateway may be configured to work with any GSM/GPRS network and for this the APN, username and password can be configured through the command menu. All parameters associated with the observations WS can also be configured in order to connect to different servers. This has been tested by sending data to the Reply server.

2.2.4.4 Commissioning and managing devices

We have developed a simple model for commissioning devices. Devices are delivered in a factory default state and unpaired. When batteries are inserted into the device, it will search for nearby networks and join the first available. Therefore care must be taken during the pairing process that only the desired network (gateway) is operational. Once paired, the setting is stored in non-volatile memory and the device will only connect to that network in future (see later on commissioning PAN).

Devices may be reset through holding an internal reset button for 10 seconds, which returns the device to factory default.

Alternatively, devices may be managed through the use of a “commissioning PAN”. This is a reserved ZigBee PAN id that can be used for commissioning. When a device is powered it will search for all nearby networks. If the commissioning PAN is discovered, then the device will preferentially join this network. A “commissioning manager” may then be used with ZigBee commands to manage the device, including returning the device to factory default and performing over the air upgrade of firmware (OTAU).

2.2.4.5 Backend integration

The Observation WS of the Chorleywood Platform implements the Continua Alliance profiled version of IHE-PCD01 and HL7 WAN-IF receiver. The Observation WS has been redesigned several times to support different versions of the database model. The current version is stable and

is designed around an object model description of the database that can accept new types of device without modification.

The Observation WS of inCASA has been developed using C# and .NET within Visual Studio. It uses Microsoft SQL as database server.

The Observation WS interfaces directly to the Microsoft SQL database. Currently the platform is deployed on a Microsoft Hyper Server based on a HP G5 server with 22Gb RAM and 800Gb of RAID 6 disk. Each virtual server is configured with 1 processor core and 2Gb RAM. The WS is deployed through use of a script.

2.2.4.5.1 Prerequisites for v1.11:

- .Net Framework 2.0 or later
- IIS 6.0 or later
- MS SQL Server 2008
- inCASA database tables and stored procedures

2.2.4.5.2 Installation instructions:

- Open the file web.config and edit the connection string, giving it server and database name. An example is embedded in the same file.
- Install the web service on IIS server: Copy the folder ObservationsWS1_11 to the server, add Application and point it to the directory where the web service is located on the server.
- Web service can be accessed using:
<http://server:port/directoryname/ObservationsWS.asmx> and WSDL:
<http://server:port/directoryname/ObservationsWS.asmx?wsdl>

2.2.4.5.3 Creating tables and stored procedures used by inCASA portal:

- Use the latest SQL scripts to create the database tables and stored procedures.
- Open the SQL server management studio and run the SQL files in the following order:
 - Create_tebles_SPs.sql
 - Insert_Device_types.sql
 - Insert_HealthProfessional.sql
 - Insert_Measurement_types.sql
 - InsertUnit_types.sql
 - Insert_UserAccount.sql

2.2.4.6 Clinician portal

The clinician portal has been developed through extensive user requirements gathering, specification and iterative feedback from users. It has been designed by incorporating best of breed features from other systems. Development continues with experience of the users and new functionality is defined.

The portal is web based to simplify deployment to users across multiple organisations. It has been developed using C# and .NET within Visual Studio. It uses Microsoft SQL as database server. Figure 49 shows typical use by a clinician while Figure 52 and Figure 53 shows display of data through tabular and graphical interface.

The clinician portal is deployed through use of a script on the server. It is web client based on the workstations.

2.2.4.6.1 Prerequisites for v 2.15:

- .Net Framework 4.0
- IIS 6.0 or later
- MS SQL Server 2008
- inCASA database tables and stored procedures

2.2.4.6.2 Installation instructions:

- Open the file web.config and edit the connection string, giving it server and database name. An example is embedded in the same file.
- Install the portal on IIS server: Copy the folder incasaPortal to the server, add Application and point it to the directory where the portal is located on the server.
- Run the application in any web browser. Use the existing database administrator login: Username: *admin*, Password: *test*
- More user accounts can be created using:
<http://server:port/directoryname/Useraccounts.aspx>
- Make sure that the web site uses the correct application pool (ASP.NET v4.0) and the website has permissions to access the database.

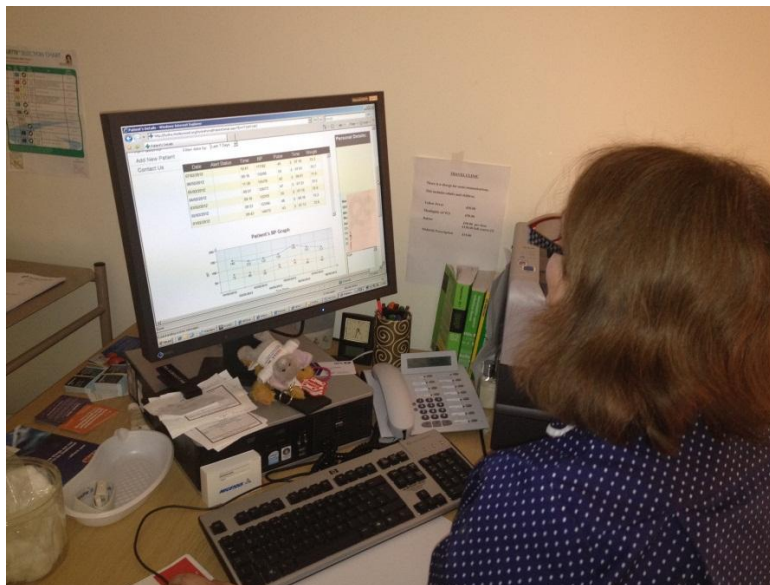




Figure 49: Clinician using the Clinician Portal



Integrated Network for Completely Assisted Senior Citizen's Autonomy

	First Name	Last Name	Date	Blood Sugar	Systolic	Diastolic	Weight (Kgs)	SPO2(%)	Habits	Reviewed
Home			14/03/2013		134	78				<input type="checkbox"/>
Add New Patient			14/03/2013		154	70				<input type="checkbox"/>
Referrals			13/03/2013		156	94				<input type="checkbox"/>
			12/03/2013		173	86				<input type="checkbox"/>
All Patients			12/03/2013		134	76				<input type="checkbox"/>
			12/03/2013		134	76				<input type="checkbox"/>
Equipment			14/03/2013					97		<input type="checkbox"/>
			14/03/2013					87		<input type="checkbox"/>
Back			14/03/2013					93		<input type="checkbox"/>
			14/03/2013					94		<input type="checkbox"/>
Logout			14/03/2013					96		<input type="checkbox"/>
			13/03/2013					94		<input type="checkbox"/>
Project Homepage			13/03/2013					94		<input type="checkbox"/>
			13/03/2013					92		<input type="checkbox"/>
			13/03/2013					87		<input type="checkbox"/>
			13/03/2013					90		<input type="checkbox"/>
			14/03/2013				75.0			<input type="checkbox"/>
			13/03/2013				75.1			<input type="checkbox"/>

Figure 50: CHC homepage view



Integrated Network for Completely Assisted Senior Citizen's Autonomy

	Add New Device:			
Home	Device ID:	<input type="text"/>	Select Device type:	Select Device Type
Add New Patient	Device manufacturer:	<input type="text"/>	Device model:	<input type="text"/>
Referrals	<input type="button" value="Reset"/> <input type="button" value="Save Device"/>			
All Patients	Device Management: Search by:: Hash code or Device ID: <input type="text"/> OR Patient Name: <input type="text"/>			
Equipment	Dev ID or Hash code	Device Type	Device Sub-type	Patient
Back		Weighing Scale		
		Sensors	PIR sensor	
Logout		Sensors	PIR sensor	
Project Homepage		BP Monitor		
		BP Monitor		
		Weighing Scale		
		Sensors	Bed sensor	
		Sensors	PIR sensor	
		Oxymeter		
		BP Monitor		
		Sensors	PIR sensor	
		Sensors	Bed sensor	
		Sensors	PIR sensor	
		BP Monitor		

Figure 51: Equipment's page for UK clinicians

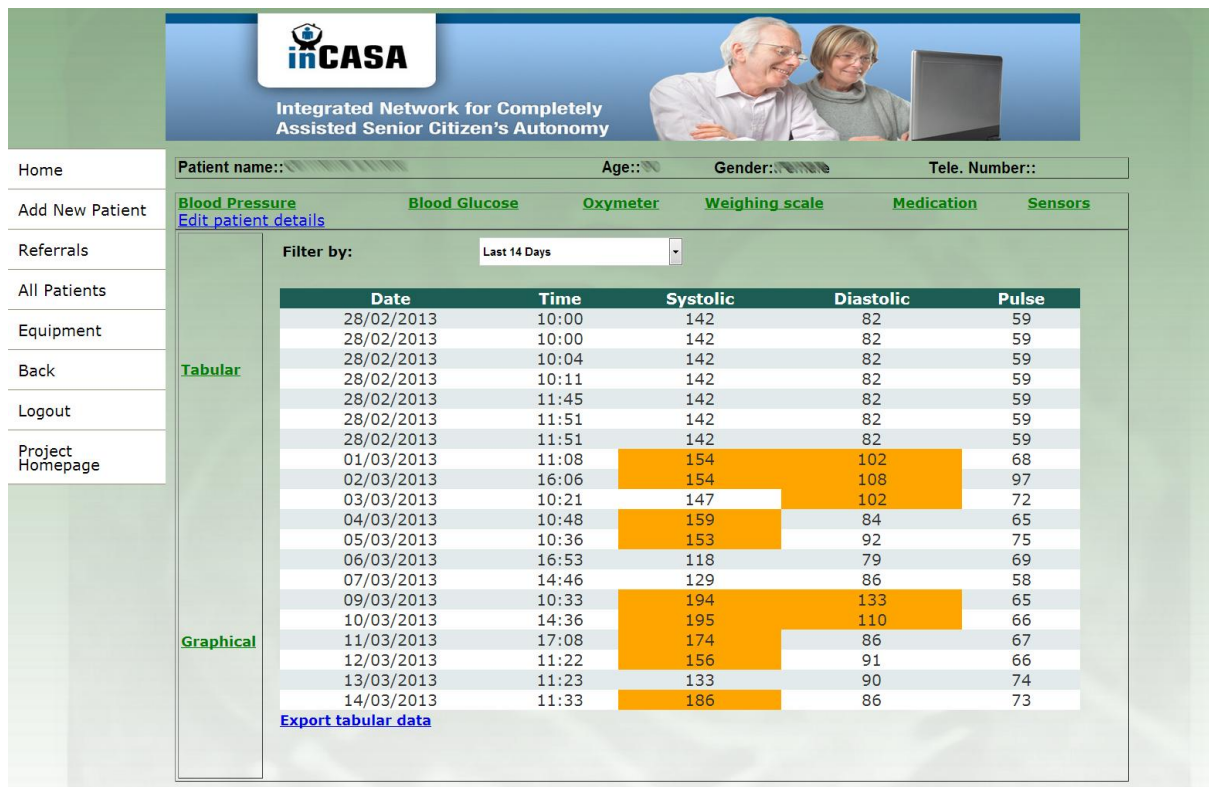


Figure 52: Tabular data overview

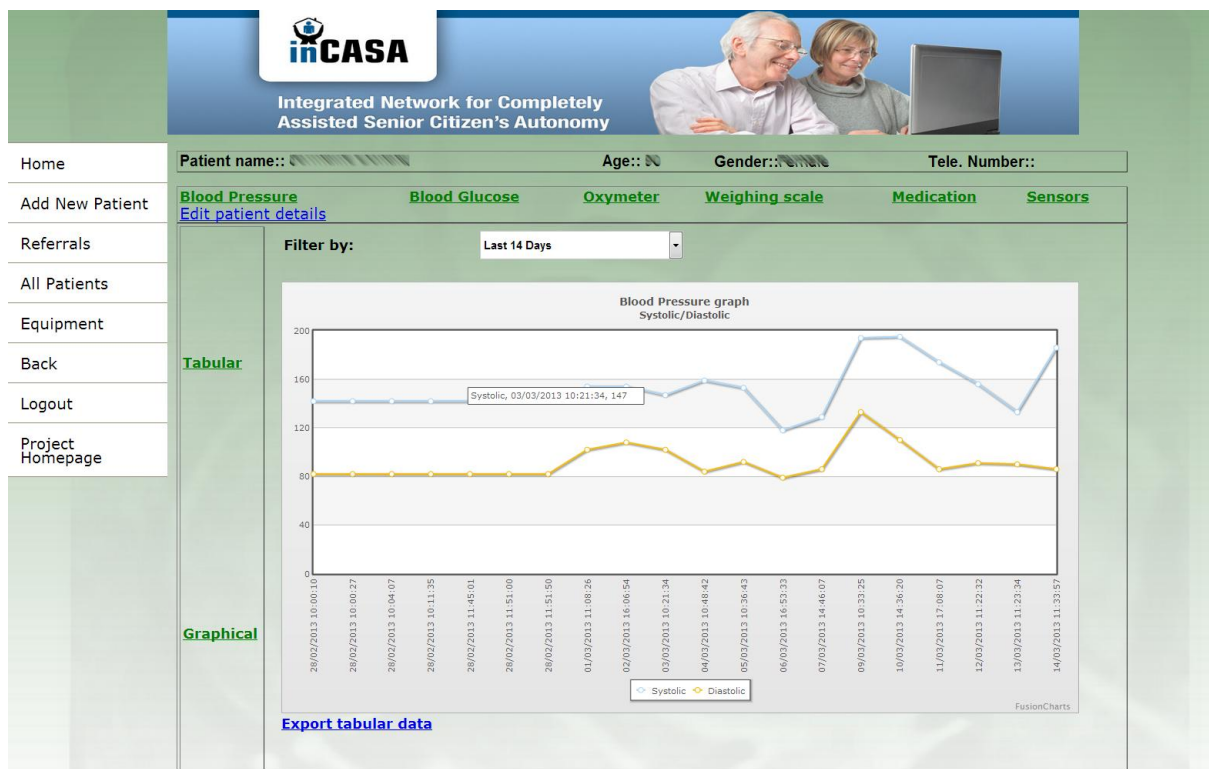


Figure 53: CHC graphical data representation

2.3 Remote Service Provider

2.3.1 Data management at Smart Personal Platform

All the inCASA modules are installed in the same server so they can refer to each other using **localhost** as address. However, LinkSmart is not in the server side necessarily. In addition, the Consumer Applications, SPP and especially the LinkSmart to SPP communication are in a sense decoupled since they communicate via WS calls. So, when installing the Remote Service Provider there is no need to change the code (only configurable endpoints as these modules may communicate from different machines). In the framework of the project, they were installed in the same server (CA and SPP) but this does not mean that is mandatory.

The following table contains the list of the ports each application is listening on:

inCASA block	Module	Listening port	Open on firewall for external access
Common	MySQL	3306	NO
Consumer Applications	Consumer Application (Web Portal)	8083	NO
	Alert Web Service	8081	NO
	Glassfish asadmin	4848	NO
LinkSmart	ApplicationDeviceManager	8080	NO
	NetworkManager	8082	NO
	OSGi	9278	NO
	Activity Hub post data	8090	YES
	Sensor List update 8091 (handle sensor configuration of the AH from the configuration database)	8091	YES
	Firmware update (need to be there for the AH to be able to run)	8092	YES
	HL7 message Post (currently not developed yet, but will be needed soon)	8093	YES
	inCasaATCSuperNode TCP	9101	YES
	inCasaATCSuperNode HTTP	9100	YES
SPP	EPR – WebService	7080	NO
	EPR – Tomcat AJP	7009	NO
	EPR – Tomcat shutdown port	7005	NO
	Mediator – exposes the ServiceRegistry service in Dual Net TCP mode	9690	NO
	Mediator – exposes the ServiceRegistry service in Dual Http mode	9620	NO

	Mediator – exposed services for data exchange with CA and LinkSmart clients and to expose the ServiceRegistry service in Rest mode	8000	NO
	Mediator/Reasoner – expose the general log service, that can be accessed by any external machine, via net tcp dual binding	9500	NO
	Mediator – to host dual binding web service (net tcp and/or http bindings according to config parameters)	9696	NO

Table 3: inCASA listening ports

All the inCASA software components are installed as Windows Services in order to be automatically run at the server start-up.

The following is the recommended sequence for the services to start up in the correct way:

1. EPR – Tomcat
2. LinkSmart (internal modules sequence have to be defined by CNET)
3. Mediator
4. Glassfish with Consumer Application.
5. Reasoner

The following table lists the services in use.

inCASA block	Module	Service name	Depends on
Common	MySQL	MySQL	
CA	Glassfish	domain1 Glassfish Server	MySQL Mediator
LinkSmart	OSGi Service Wrapper	OSGiServiceWrapper	
	LinkSmartDotNetServices	LinkSmartDotNetServices	OSGiServiceWrapper
SPP	EPR – Tomcat	Tomcat7	
	Mediator	IoTMediatorWinService	EPR
	Reasoner	IoTReasonerWinService	Mediator

Table 4: inCASA Windows services

Software pre-requisite for Mediator and Reasoner is the .NET 4.0 Framework or greater which can be downloaded and installed for free from the Microsoft web site [<http://www.microsoft.com/en-us/download/details.aspx?id=22833>].

2.3.2 Mediator

Mediator is a self-hosted .NET web service which provides the communication channel between all the SPP modules and performs the verification and parsing of all the messages received at <http://host:port/PCDDData?wsdl>.

Within the SPP, the Mediator is the component which bridges different service requestors and providers. It communicates with the LinkSmart server component, with the Reasoner, the EPR and the Consumer Applications.

Reply.Platform.IoTConsoleApplication folder contains .NET executable module (Reply.Platform.IoTConsoleApplication.exe) with all the necessary libraries. This is an intermediate module which is necessary to pass all the incoming data to the SPP and to communicate with Consumer Application.

Reply.Platform.IoTMediatorWinService folder contains the executable to launch the .NET project as Windows service at the server start-up. This service can be installed running from command line the following .NET Framework utility:

C:\WINDOWS\Microsoft.NET\Framework\v4.X.XXX\InstallUtil.exe ServiceName.exe.

After this utility has been successfully executed the service will appear in the list of Windows's services as shown in the figure below; the service is configured in order to start automatically.

2.3.3 Reasoner

Reasoner is a self-hosted .NET web service which is listening for the changes on the system and reacts in predefined way based on the rules defined.

Reasoner reads its configuration from the ReasonerConfig.xml file which contains the following parameters:

```
<ConfigurationOptionsEnabled>DebugOntology;AlertCleanUpAtRestart;HealthEnabled;</ConfigurationOptionsEnabled>
<!-- Define the time for periodic habits check and data storing, e.g. 07:59:00 -->
<HabitsCheckAndStoringTime>15:50:00</HabitsCheckAndStoringTime>
<!-- Define the time for periodic health check, e.g. 07:59:00 -->
<HealthCheckTime>22:00:00</HealthCheckTime>
```

The HabitProfilingPhase can be updated from the emulator/development tool only. This should be installed separately in case of necessity. The Reasoner plugin in charge of computing habit profiles can be replaced at run time without stopping the Reasoner using the same emulator/development tool.

The Reasoner is provided in an archive containing executable files and libraries, which need to be unpacked in a folder on the server.

Reply.Platform.IoTReasonerConsole folder contains .NET executable module (Reply.Platform.IoTReasonerConsole.exe) to launch the Reasoner.

Reply.Platform.IoTWinService folder contains the executable to launch the .NET project as Windows service at the server start-up. This service can be installed running from command line the following .NET Framework utility:

C:\WINDOWS\Microsoft.NET\Framework\v4.X.XXX\InstallUtil.exe ServiceName.exe.

After this utility has been successfully executed the service will appear in the list of Windows's services as shown in the figure below; the service is configured in order to start automatically.

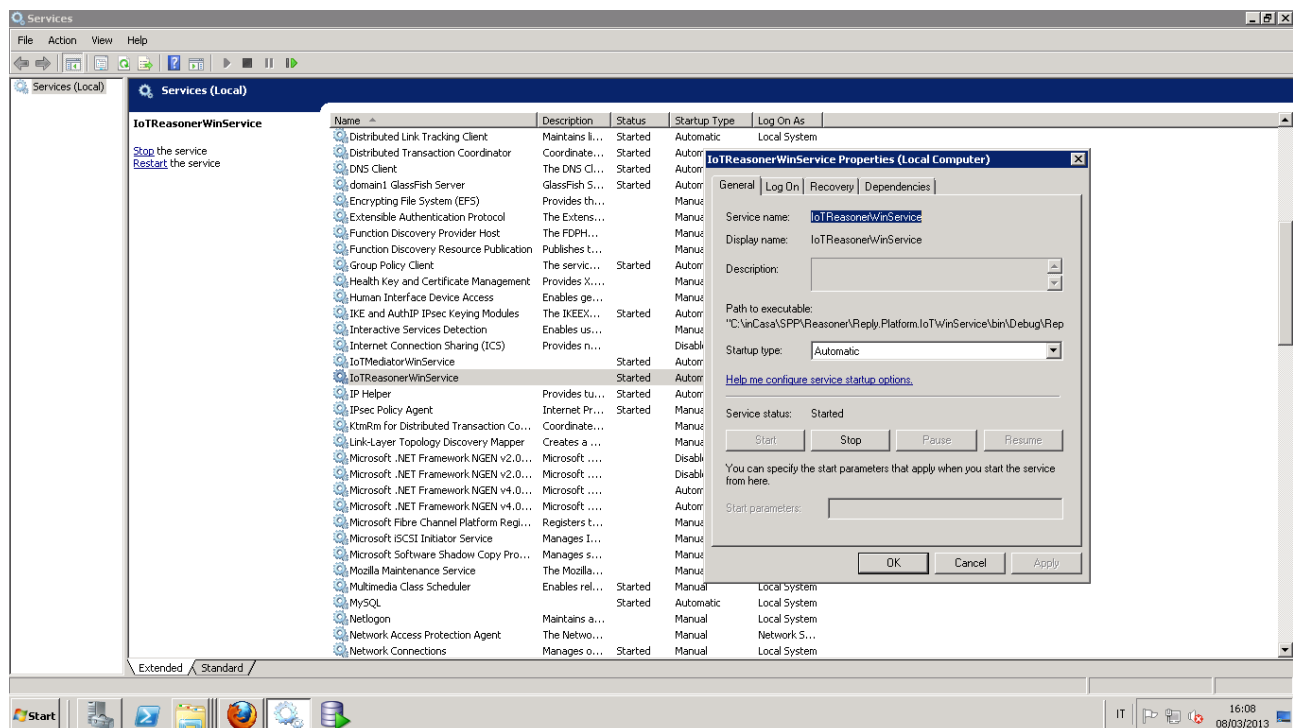


Figure 54: Reasoner's Windows service

The habit profiles are computed by a plugin of the main Reasoner. This plugin can be configured or replaced at run time without stopping the Reasoner using the emulator/development tool, which is a .NET executable module (Reply.Platform.IoTSimulator.exe) that can also be used to test the Reasoner functionality in a comprehensive way.

2.3.4 Electronic Patient Record

Here Electronic Patient Record (EPR) requires an Oracle (version 10g or above) database instance; the amount of disk space needed (tablespace) can be estimated knowing that an average of 100 measurements per hour require approximately 10 Mb per day.

EPR web services are released as a Web Application (EPR-WS.war) deployed to Tomcat 7.0.22. Oracle DB manager and Tomcat Web Server are started automatically on the server start up.

2.4 User interfaces and Extensions

This chapter will describe the different pilot user interfaces (also known as Consumer Applications) as well as the extensions made, if any, at each pilot site.

2.4.1 Consumer Applications (NTUA)

The Consumer Applications (CA) stands as the single point of access to the inCASA platform for the professional inCASA Pilot stakeholders (clinicians, operators, social workers, psychologists, etc.). inCASA Consumer Application is a Web Portal where operators can view patient's socio-medical data and alerts, store notes to the system under a patient, add patient questionnaires' scores and perform also various other actions that are customized in each Pilot site using the application according to their requirements. It is important to note that a Role-Based Access Control (RBAC) is being applied in order to distinguish the views and the allowed actions among the various Professional Stakeholders.

At a generic architectural level, the Consumer Applications (CA):

- expose a web service interface for receiving alerts and

- consume two web service interfaces for
 - Querying the SPP and
 - Sending alert updates, that contain operator comments, back to the SPP

All interfaces follow the IHE guidelines and the template of <http://host:port/PCDDData?wsdl> WSDL file as described in [4], Section 8.3. Therefore, by following the standard recommendations, inCASA Consumer Applications help towards solution's future extensibility and further integration with third party healthcare systems. Since interfaces are the same, what differentiates the inner functionality of the above interfaces is the HL7 message content.

The interfacing component of the Consumer Applications is the SPP and their architectural integration is described in the following picture:

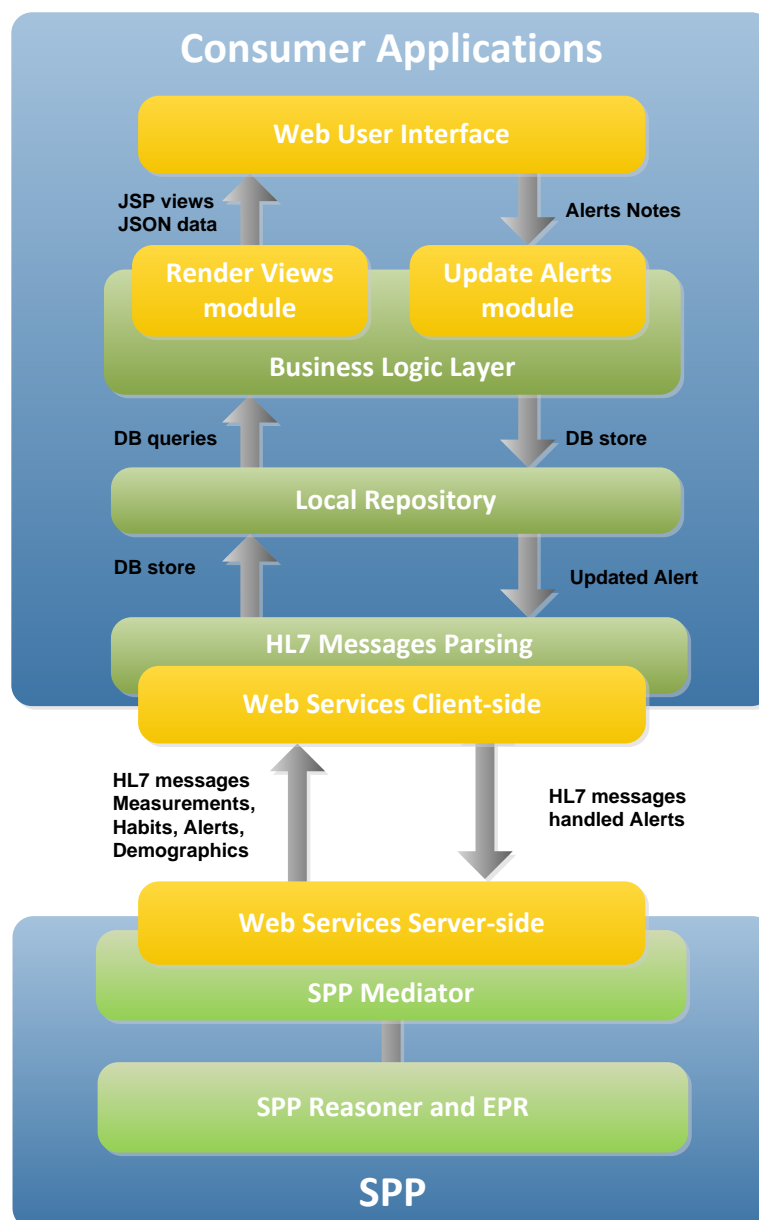


Figure 55: SPP – CA Integration

In the following sections, we describe the software packages needed for the successful deployment of the Consumer Application functionality and the way they should be installed and configured at the Remote Server machine.

2.4.1.1 Glassfish Installation

The Glassfish application server hosts the Consumer Application. The Open Source version 3.1.2 is available for download from the official Glassfish download site [<http://glassfish.java.net/>], and is supported from both Linux and Windows platforms. Once the installation file is retrieved (.exe,.bin), the user follows the standard installation procedure, where at first is prompted to select the installation path. Next, the administrator user creates a standard domain where he configures specific parameters of the application server platform, such as the platforms' domain name and the standard port number where the server will be listening. Finally, the administrator chooses whether the server instance will be run as a service. In order for the server to start upon system start up, the relevant flag should be switched to automatic. This can be done from the Services window on a Windows Machine.

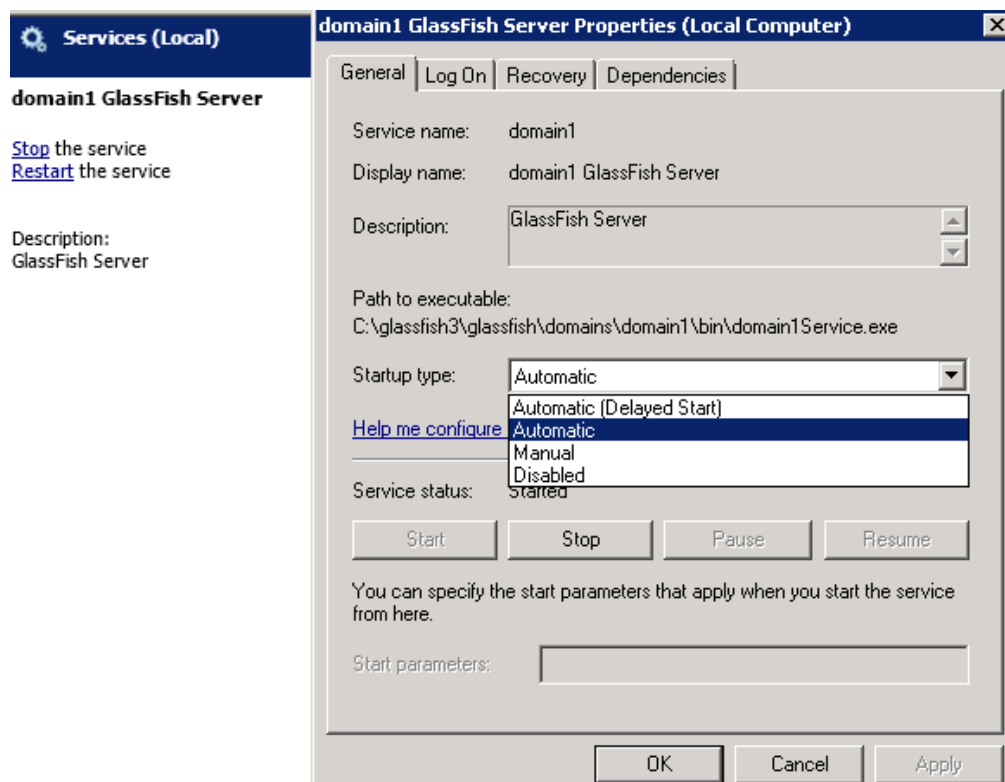


Figure 56: Glassfish configuration

2.4.1.2 MySQL Database Server Installation

The MySQL Community Database server is an open-source relational database management system, which is the world's most popular database system among the open-source ones. In our case, it holds the schema which contains all the HL7 specific data that are extracted from the CA.

The Open Source 5.6.10 community edition can be downloaded from the official MySQL site [<http://www.mysql.com/>]. The installation procedure is rather simple. Firstly, the installation requests for an installation path. Next, the user must provide specific configuration instructions, and finally install the MySQL server as a Windows Service.

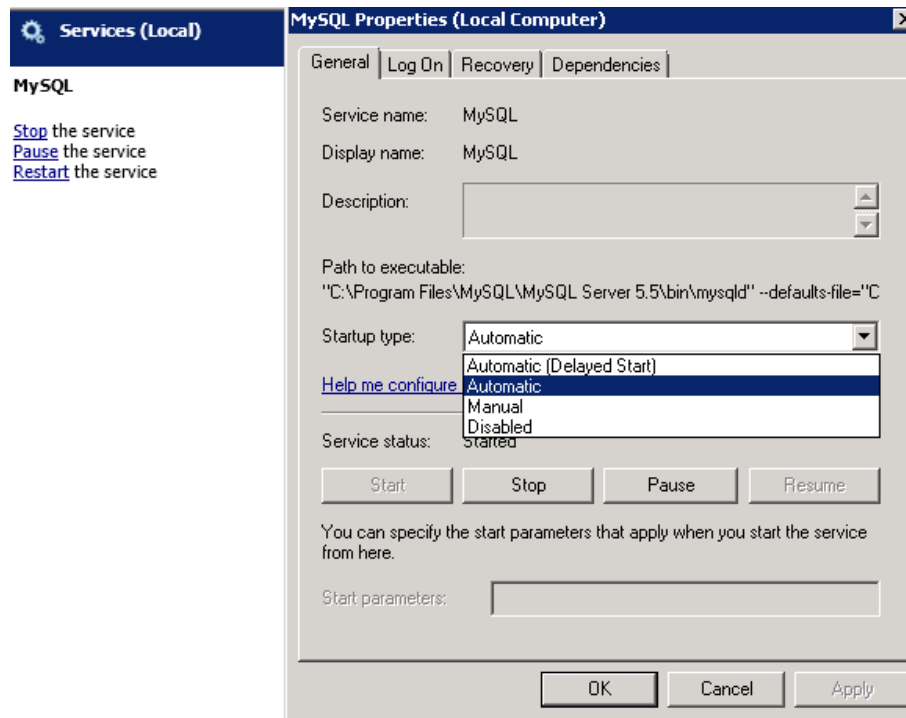


Figure 57: MySQL Database Configuration

The database server contains the schema that stores all the data that is handled by the CA. The generic schema is shown in the picture below, where relevant tables store the information that is useful for the inCASA front-end functionality:

1. Patient name, ID, address and other demographic data
2. Patient measurements
3. Patient habitual profile
4. Patient alerts
5. Custom functionality support, like psychological tests scoring storage.

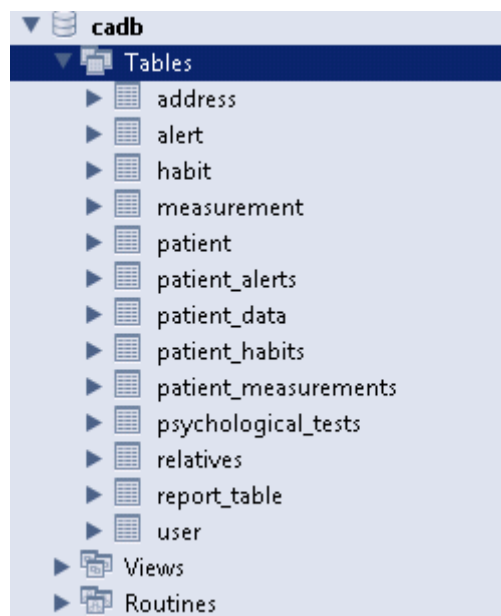


Figure 58: Default CA Database schema

2.4.1.3 Data Source Creation

The application server creates a data source layer between the application and the database server, providing an API in order to simplify the implementation of the Create, Read, Update, Delete (CRUD) operations requested by the application. To create a data source, we use the Glassfish administration console, and we choose the JDBC -> JDBC Connections Pools tab, in the Common Tasks section. We choose to create a new connection pool, and then we specify the pool name and the resource type and the database vendor.

Figure 59: Data source creation under Glassfish

On the next page, we specify some additional properties as shown in the picture below

	Name	Value
<input type="checkbox"/>	driverClass	com.mysql.jdbc.Driver
<input type="checkbox"/>	databaseName	skiveca
<input type="checkbox"/>	serverName	localhost
<input type="checkbox"/>	portNumber	3306
<input type="checkbox"/>	Password	cal@#\$
<input type="checkbox"/>	User	root

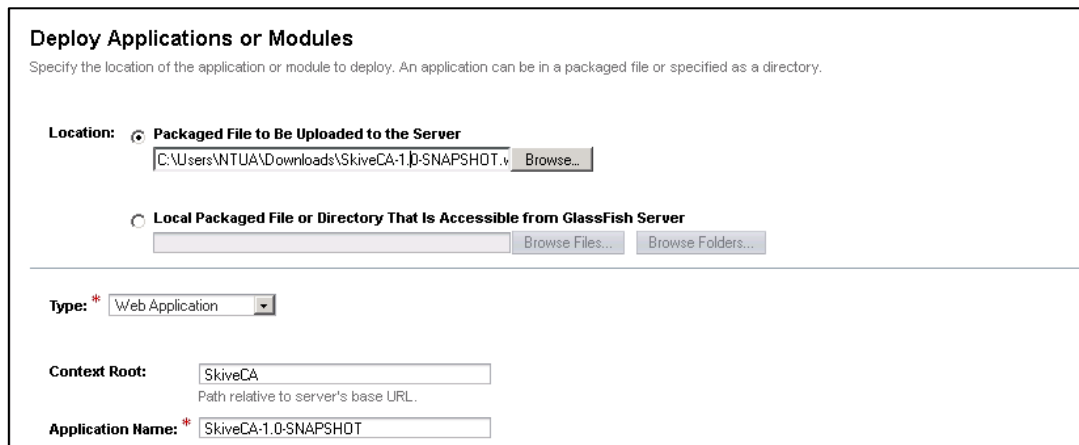
Figure 60: Additional Data source Properties

Finally to complete the data source installation we must create a JDBC resource. This can be done by navigating to the JDBC -> JDBC Resource tab, and clicking New. Then we specify the data source name and the connection pool to be used.

Figure 61: JDBC Resource configuration

2.4.1.4 Application Deployment

The Consumer Application can be deployed from the Glassfish Administration console Web interface. The web interface can be accessed from any browser. The administration console by default listens to the 4848 port. To deploy an application, the Applications tab, on the Common Tasks section, must be selected. Then the user must choose to upload a packaged file. Finally, the user must specify the application name and the application context root. By hitting the OK button, the administrator chooses to deploy the application to the server.



Deploy Applications or Modules
Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

Location: ☒ **Packaged File to Be Uploaded to the Server**
C:\Users\NTUA\Downloads\SkiveCA-1.0-SNAPSHOT.v Browse...

☐ **Local Packaged File or Directory That Is Accessible from GlassFish Server**
Browse Files... Browse Folders...

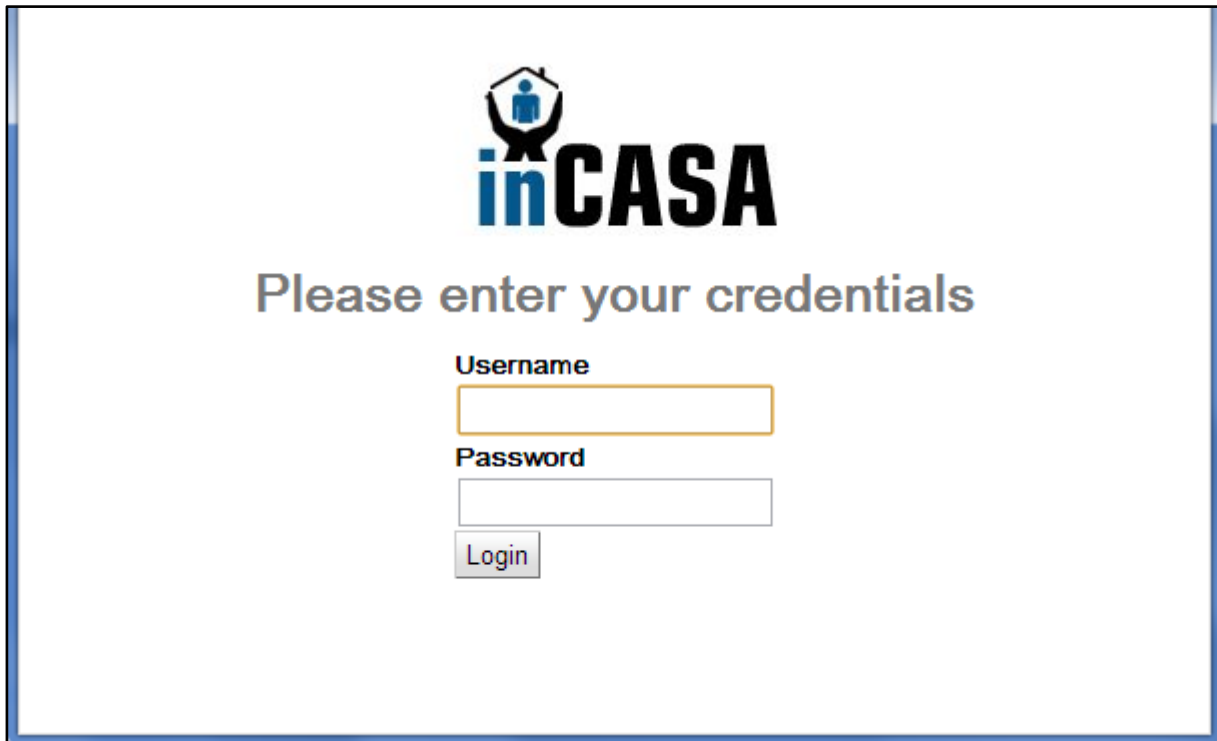
Type: * Web Application

Context Root: SkiveCA
Path relative to server's base URL.

Application Name: * SkiveCA-1.0-SNAPSHOT

Figure 62: CA Deployment

Since the Application is deployed successfully, it can be reached through the Web. Its starting page is shown below, where correct credentials are needed to be provided in order to enter into the Portal and its exposed functionality.



inCASA

Please enter your credentials

Username
[Text Input Field]

Password
[Text Input Field]

Login

Figure 63: Web Portal start page

2.4.2 Extensions (CNET)

If any of the inCASA pilots have requested any extension that normally is not part of the inCASA solution then each one of them will be described for each pilot site. Extensions can be devices, GUIs, formats, interfaces or even data flow deviations. For the KGHNI pilot look in chapter 3.1, for INSERM chapter 3.2.3, FHC chapter 3.3.3, ATC chapter 3.4.3, Skive chapter 3.5.2, and the CHC pilot in chapter 3.6.3.

3 Description of inCASA platform installations

Chapter 3 will provide a full description on each of the pilot sites that have been active during the inCASA project and their installations. Some pilot descriptions may be repetitive but then the authors have used cross-reference to limit the number of pages and figures taking space.

3.1 KGHNI pilot

3.1.1 Greek procedures

The KGHNI pilot is an inCASA Pilot adopting the reference architecture since among its objectives are the combined health, social and psychological monitoring. This means that both the Telehealth and the Telecare Gateways are needed for KGHNI, turning therefore mandatory the presence of the LinkSmart Middleware. Greek Pilot aims also at habits profiling and alert generation, so the presence of SPP is also mandatory. Finally, the integrated view of the Consumer Applications fits perfectly to the aims of the Pilot and allows the combined health and psychological monitoring of CHF patients followed by the Cardiology Clinic of the hospital.

Summarizing the above analysis, the components used in the KGHNI and their integration is shown at the following figure which is totally compliant with the inCASA reference architecture [3].

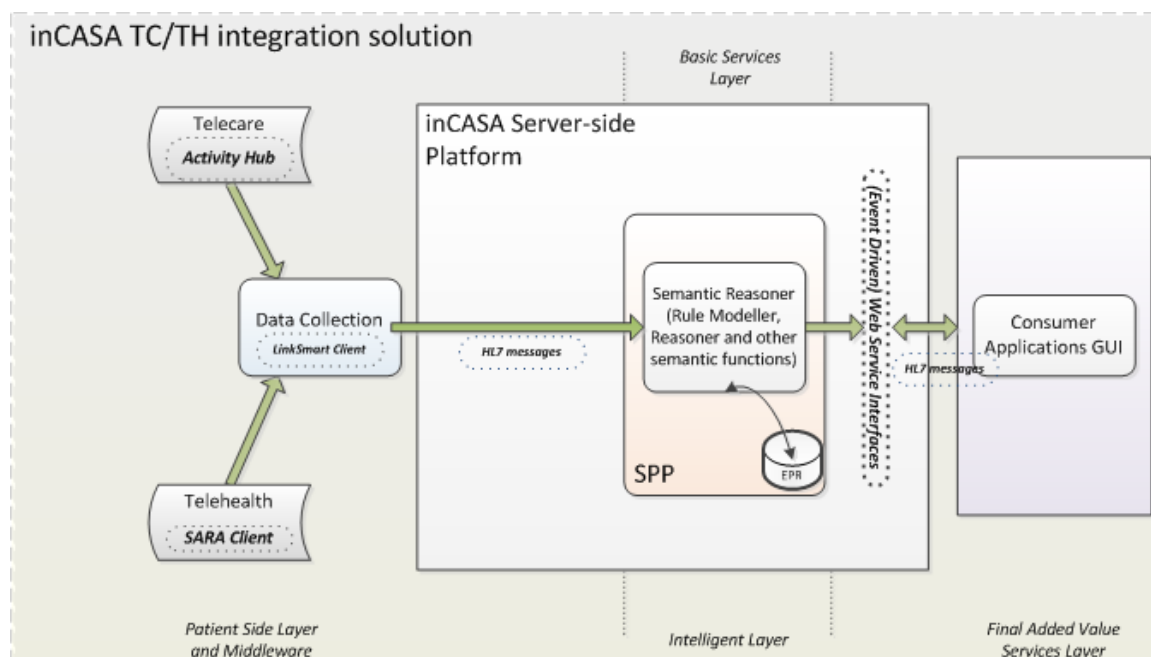


Figure 64: KGHNI pilot architecture

In order for the data collection to be realized, the following components are needed to be configured accordingly with respect to the above figure.

1. Telehealth Gateway – SARA Client: It is installed in the patient's PC in accordance with the guide provided in the section 2.2.2 of this document.
2. Telecare Gateway – Activity Hub: It is installed in the patient's environment and set up in accordance with the guide provided in the section 2.2.1 of this document
3. LinkSmart Middleware: It is installed in the patient's PC in accordance with the guide provided in the section 2.2.3 of this document
4. SPP (Mediator / Reasoner) and EPR: These components are installed at the server side of the platform which is hosted by NTUA. The installation is performed as explained in the section 0 of the document.

5. Consumer Applications: They are installed at the server side of the platform and deployed in order to be reachable over the Web by the inCASA professional users. The installation is performed as explained in the section 2.4.1 of the document.

As long as Software / Hardware requirements are concerned, the following data can be provided:

- The all-in-one PCs provided to the patients should run Windows 7 Operating System. The PCs were selected to have touch screens in order to be easily managed by the elderly people who are often not at all familiarized with modern technologies. In the Greek Pilot, the PC model provided to all patients is SHUTTLE ALL-IN-ONE BB X50 (<http://us.shuttle.com/X50.aspx>)
- The remote machine located at NTUA and hosting the inCASA server side (SPP, EPR, Consumer Applications) has the below specifications:
 - CPU : Intel® Core™2 Duo CPU E8400 @ 3GHz
 - RAM : 8GB
 - Hard Disc size : 150 GB
 - O/S : Windows Server 2008 R2 standard, Service Pack 1 (64-bit)

3.1.1.1 Telehealth Devices

All Telehealth Devices are paired via Bluetooth communication with the SARA Client Software as analysed in the section 2.2.2 of this document. The Telehealth devices used by the KGHNI pilot are the following:

1. Weight scale A&D Medical UC-321-PBT



2. Blood pressure monitor A&D Medical UA-767-PBT



3. Pulse Oximeter Nonin Onyx II 9560



3.1.1.2 Telecare Sensors

The telecare sensors are paired with the inCASA Activity Hub as explained in the section 2.2.1 of this document.

The sensors used in the Greek pilot are the following:

1. Chair Sensor: Funkstuhl Transmitting Chair. The sensor communicates with the Hub via the Wireless Communication Protocol EnOcean.



2. Power Sensor Netvox Z-800. The sensor is connected to the TV in order to send ON/OFF signals when TV is switched on or off respectively. The sensor communicates with the Hub via the Wireless Communication Protocol ZigBee.



3. Motion and temperature sensor Netvox Z-B01C. The sensor communicates with the Hub via the Wireless Communication Protocol ZigBee.



3.1.2 Experiences

At the time of this document writing, more than 18 months have passed since the Pre-Pilot activities began. A lot of experience has been gained at the technical field too. The Pilot faces no major technical issues even if the needed technical support is not negligible mainly due to the big number of devices used.

As a synopsis, the collected technical experience can be stated as follows.

1. The weight scale has caused no technical issues. All measurements were performed normally and their transmission via Bluetooth to the SARA Client was always successful.

2. The blood pressure monitor has caused no technical issues. All measurements were performed normally and their transmission via Bluetooth to the SARA Client was always successful.
3. The pulse-oximeter had no issues concerning the measurement performance but it has presented some Bluetooth connectivity issues with the SARA Telehealth Gateway. At this point, it is important to note the functionality of SARA Client which allows to the patients to enter manually their measurement when there is a failure in the Bluetooth communication in order that no measurement is lost from the server-side of the platform allowing in this way the continuous monitoring of the patient. As a workaround, during the technical installation at the patient's home, it is suggested to them to perform the oximetry measurement close to the PC's Bluetooth dongle, eliminating significantly in this way the above mentioned defect.
4. The chair sensor had no technical issues at any time. The difficult part on this scenario is to choose the right place for its positioning; it should be placed under the chair where the elderly patient usually sits. It is observed in practise that this assumption is bringing fruitful results since the elderly people really tend to follow the same routine, so they almost always prefer their "favourite" chair.
5. The power sensor had initially some problems with the triggering threshold. The power threshold was too high and the power consumption by an ordinary new generation TV was not passing it. KGHNI ordered a new version of Netvox Z-800 power sensors which finally worked successfully.
6. The motion/temperature sensor is the most data consuming sensor (we may have more than 2K measurements per patient). As an issue, around 10 times during the Pilot life, a reset of the sensors was needed in order to unblock them since they were not responding.
7. The Activity Hubs were causing initially some problems since they were stopping sending measurements after an amount of time (around 1 week) and a restart of them was mandatory by the technical support team. Since the latest version of the firmware has been released (beginning of November 2012), the Activity Hubs work continuously without experiencing any problem during their operation and their interconnection with the sensor or the Middleware.
8. The SARA Client Software was always working as expected without causing any problem. This can be also explained by the fact that it was mainly a prior to inCASA developed product that had already been extensively tested.
9. Concerning the Software components developed during the Project's life cycle (SPP, LinkSmart modifications, Consumer Applications) the main issue had to do with their integration and their HL7 interchange messages. This is quite logical considering that inCASA is a demanding integration project. Regarding this issue, the close follow up of the involved partners – NTUA, CNET, REPLY – provided solutions and almost immediate bug fixing.

3.1.3 Solution extension

In the Greek pilot the LinkSmart Middleware is running as a service in the patient's PC and can be executed either remotely or locally.

3.1.3.1 Professional Users Interface

The Consumer Applications in the Greek pilot follow the reference design of combined Telecare and Telehealth data and alerts visualization.

As an extension to the basic set, an extra interface was provided to the professional operators where they can register the scores of the psychological monitoring questionnaires administered to the patients. This data flow is not triggered by electronic devices, so it cannot follow the path Middleware → SPP → Consumer Applications. The extra interface developed allows the operators to store the relevant scores under each patient and at the same page observe the graphical trend of their scoring which may indicate possible psychological worsening or depression signs.

The screenshots below explain the aforementioned extended functionality.

The screenshot shows a web form titled 'Test type'. It contains a dropdown menu with 'BDI' selected, a text input field for 'Score', a text input field for 'Date' containing '2013 January 2', and a 'Submit Score' button.

Figure 65: Interface to submit new questionnaire score under a patient

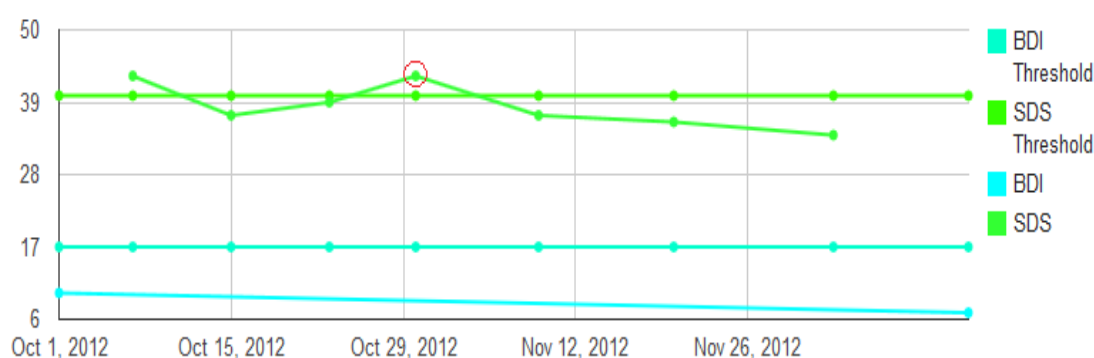


Figure 66: Questionnaires graph - the red circle is showing an alerting score that triggered a psychologist's intervention

3.2 INSERM pilot

3.2.1 French procedures

The INSERM pilot uses the SARA Gateway for body weight measurement (Bluetooth weight scale), rest-activity rhythms recording (Infrared Actigraph) and symptoms assessments (MDASI questionnaire).

As described in section 2.2.2.3 the first step of the installation is the patient profile creation on the web portal.. The weight measurement function is added and configured during the patient set up, however, as the Actigraph and questionnaire are experimental functionalities, they need to be manually added and configured by TID.

The PC platform is configured to hibernate after 15 min of inactivity and to automatically start SARA client at computer start-up.

Before installing the equipment in the patient's home, all the functionalities are tested in order to ensure adequate transmission..

At the patients' home, the PC platform is connected to the Internet using the Windows network manager.. After checking the Internet connection, the system is fully operational and the patient is trained to use it..

The setup is as described in chapters 2.2.2 and 2.2.3. Below is some of the SARA client's customised GUIs used in the French pilot:

Connected as:
DoctorFR
[\[Déconnexion\]](#)

Add new patient to telemonitoring

SIP Login:

SIP Password:

NHUSA (*):

Numéro dossier(*):

Nom (*):

Prénom 1 (*):

Prénom 2:

E-mail:

NIN:

Date de naissance (*):

Sexe (*):

Date de fin (*):

Date d'entrée (*):

Professionnels de santé:

Capable of videoconference: ☐

Numéro de téléphone(*):

Téléphone portable(*):

Suivant

Figure 67: French form for assigning new patient

Connected as:
DoctorFR
[\[Déconnexion\]](#)

Enregistrer nouveau Traitement

Treatment configuration

Weight Control

Add

Treatment component	Dernière modification	Commentaires
Weight Control		

Figure 68: French GUI when registering new treatment

In Figure 69, KIT identified is chosen for the patient. This identifier will be the inCASA id that must be set up in the patient's application.



Figure 69: Choosing an inCASA id for the INSERM pilot



Figure 70: SARA GUI for the INSERM pilot



Figure 71: Patient measurement choices as provided by SARA

3.2.2 Experiences

As for KGHNI, the method to associate a patient profile to a SARA gateway is very simple because it is only necessary to configure the Kit number per patient.

3.2.3 Solution extensions

3.2.3.1 Actigraph

The Actigraph represents basic sleep estimation, high resolution and analog data collection, and simultaneous environmental data collection and is worn on the user's wrist.

The SARA client needs to be installed and running in order to use the Actigraph Motion Logger. Do the following:

1. Unzip the SARAMedicalDevice.zip.
2. Configure the patient ID and residence ID in configuration.xml.

```
<IdNumber>999</IdNumber>
<UniqueHomeID>AP999</UniqueHomeID>
```

This will be used when creating the HL7 ORUR01 Message.
3. Make sure LinkSmart is running <http://127.0.0.1:8082/NetworkManagerStatus>.
4. Run SARAMedicalDevice.

The SARA Gateway uses the Actigraph Motion logger from Ambulatory Monitoring, Inc. and in order to clinicians to be able to extract correct AMI file (non-conversable format) and view it in one of the software provided to do so, the SARA clinical GUI need to have a mechanism to do this. It is solved by SARA and LinkSmart cooperating to reach common goal which is to enable data representation of transferred Actigraph activity data. The LinkSmart has a specific Actigraph IoTDevice that shares the data when a patient downloads new Actigraph data from the Motion Logger to the SARA Gateway. The inCASA consortium has developed a web page that presents correct AMI files for a certain Kit ID that is put in the SARA Clinical GUI. By doing so, clinicians are able to click on a specific timestamp and AMI file and open it in e.g. the Watch Ware or the ActionW software.

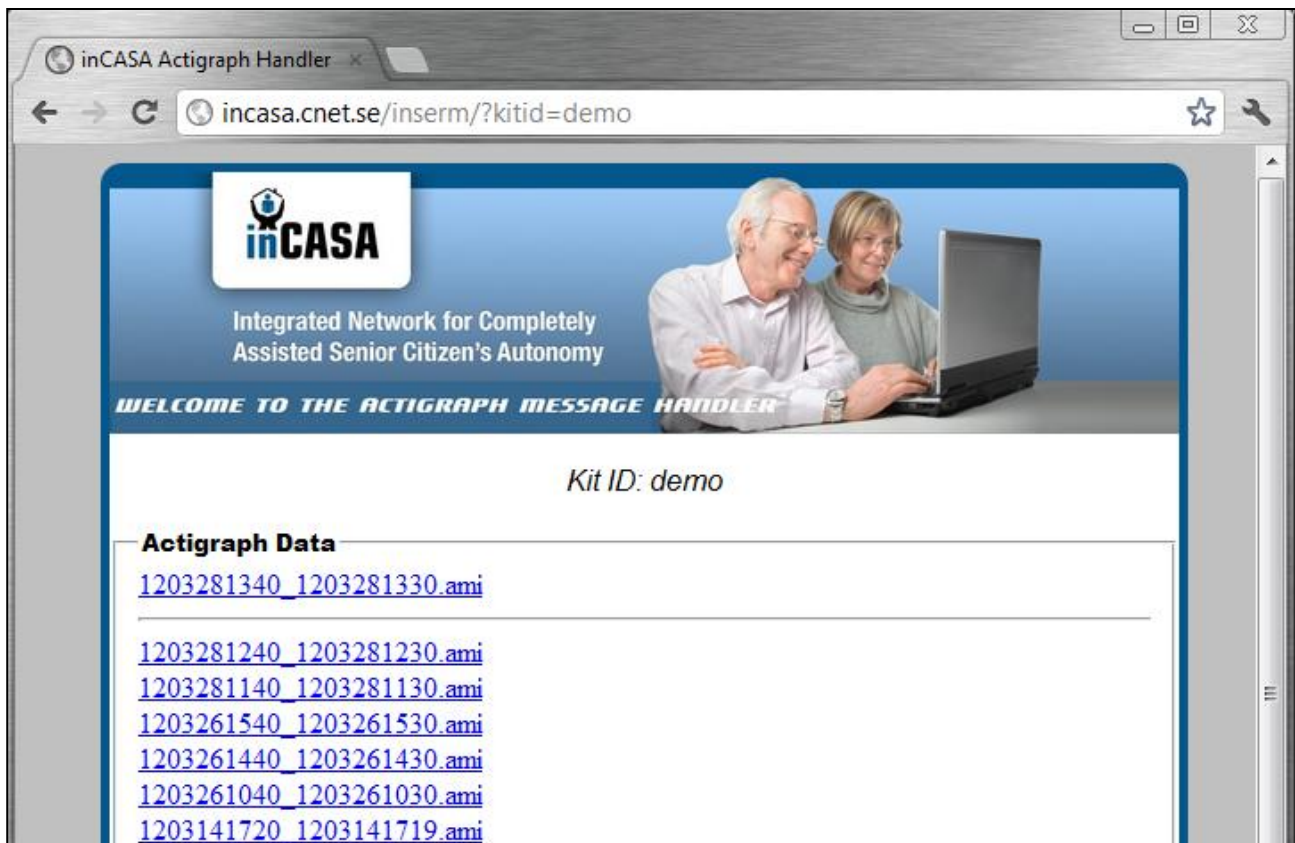


Figure 72: The Actigraph AMI file handler as webpage and where latest activity data is on top for easier retrieval by the clinicians.

3.3 FHC pilot

3.3.1 Spanish procedures

The Spanish pilot at FHC is identical to the one that was carried out at INSERM. Therefore it is best to refer to chapters 2.1, 2.2.2 and 3.2.1

The fact that we are able to refer back to old installation procedures shows on the ideal approach given by the inCASA solution where the platform fulfils some basic repetitive scenarios and needs on both devices and workflows.

3.3.2 Experiences

As for the INSERM pilot, the method to associate a patient profile to a SARA gateway is very simple because it is only necessary to configure the Kit number. However the necessity to request TID to manually enable Actigraph and questionnaire functions for each patient complicated the procedure.

3.3.3 Solution extensions

Please refer to chapter 3.2.3 about the LinkSmart extension for the Actigraph.

3.4 ATC pilot (REPLY)

The server side of the inCASA solution has been installed on a Windows 2003 Standard Server 64 bit bi-processor with 6 GB of RAM and 120 GB of disk space; the virtual server is hosted by X86 VMware ESX 4.0Vsphere.

3.4.1 Italian procedures

In the Italian pilot most of the inCASA reference architecture's components are installed except for SARA which is not required:

- At the Home Base Station the SIG's Activity Hub is installed as described in chapter 2.2.1.
- LinkSmart is installed only inside the Remote Service Provider as described in chapter 2.2.3.
- SPP Mediator and Reasoner are installed as described respectively in chapters 2.3.2 and 2.3.3.
- EPR version 1.1.0.6 is installed as described in chapter 2.3.4; its database schema is hosted by an ATC laboratory instance of Oracle Database 10g Enterprise Edition 10.2.0.5.0 64 bits.
- Consumer Applications user interface is employed in this pilot and it is installed as described in chapter 2.4.1.

3.4.2 Experiences

The Italian pilot started when the inCASA development phase was in its first iteration. Since only the Telecare use cases were implemented it was not necessary to upgrade the solution to the version which was the outcome of the second iteration of development, dedicated to the integration between Telecare and Telehealth use cases.

One big issue that was faced during the pre-pilot installation was due to problems with SIG's Activity Hub, at the time unable to send measurements continuously over time. A backup solution was built using a commercial off-the-shelf hub (Zilant) able to integrate the same medical devices already employed.



Figure 73: Zilant hub

As a consequence, some patients' apartments were provided the SIG's Activity Hub and others (the majority) the commercial off-the-shelf hub.

3.4.3 Solution extensions

The need to integrate Zilant's hub instead of SIG's Activity Hub gave us the chance to demonstrate the openness of the inCASA platform. An adaptor was built, with a small effort, which translates the output of the commercial off-the-shelf hub into the same data format generated by SIG's Activity

Hub: in this way the use of one or the other of the hubs was made transparent to LinkSmart, which receives the data in the same way.

The SPP EPR's Oracle schema is hosted by an ATC laboratory instance of Oracle Database 10g Enterprise Edition 10.2.0.5.0 64 bits.

EPR Web Application (EPR-WS.war) version 1.1.0.6 is deployed to Tomcat 7.0.22.

One of the extensions that are important to mention is the development of the SMS Service used by ATC Pilot in the "Open Door" scenario for monitoring the habits profile of the users.

In fact, when user leaves home (or stays at home) but door remains open for more than 30 minutes, inCASA platform sends an alert to NTUA Consumer App.

The SMS Service consists of an email sent by NTUA Consumer App to ATC Consumer App (developed by ATC) when an alert is occurred.

SMS Service transforms this email in SMS that is sent to the user to warn him/her that the door is opened. If no action have been done after this (door still open), the alert escalation will be performed.

3.4.3.1 Professional users interface (ATC Consumer Application)

As an extension to the basic set of Consumer Application functionality, the functionality deployed in the ATC pilot offers some extra features adapted to the ATC needs.

At first, there is a custom interface where the operators can register notes concerning the patient related to their social, familiar status. This aspect is strongly related to the ATC pilot which focuses on the social support of frail elderly people. Furthermore, there is a distinction on the notes per patient, as first and second level, where the second level notes contain private sensitive information. There is therefore a Role-Bases Access Control mechanism, which allows only to the users that are logged in as Social Service members to view/modify the second level data while logging-in as an operator permits only to view/modify the first level data. Another important ATC specific implementation refers to the addition of notes under a specific alert by the operators (e.g. actions taken to resolve an alerting message). This note is also propagated by the Consumer Application to the SPP/EPR according to the inCASA alert life-cycle specifications [3].

3.5 SKIVE Danish Transferability Model

3.5.1 Danish procedures

SPP Mediator and Reasoner are installed as described respectively in chapters 2.3.2 and 2.3.3. EPR version 2.2.1.0 is installed as described in chapter 2.3.4; its database schema is hosted by an instance of Oracle Database 11g Express Edition (Oracle Database XE), which is an entry-level, small-footprint database based on the Oracle Database 11g Release 2 code base.

The Consumer Application Web Portal is also installed in accordance with what described in chapter 2.4.1.

3.5.1.1 Data collection in home base station

Data in the SKIVE Danish pilot are collected from the following sources:

1. Weight Scale device
2. Blood pressure device
3. Blood Glucose device
4. Blood oxygen device
5. Pedometer
6. Cigarette smoke detector
7. Windows Open/close detector

3.5.2 Solution extensions

For the SKIVE pilot a Danish designed User Interface was preferred before the usual platform Consumer Applications. Also, choosing the LIVA GUI saved the project developing efforts and time as LIVA already was tested and run by In-JeT.

3.5.2.1 LinkSmart modifications

LinkSmart is flexible in that way that it offers developer resources to quickly and efficiently modify device communication according to different needs.

CNet disposes a Device Connectivity Kit that allowed the SKIVE pilot to address what medical device types were of interest and what formats should be supported. Below is a description of the procedures for this particular case.

3.5.2.1.1 Toshiba Bluetooth

1. Install the Toshiba Bluetooth stack *V71001_XpVistaWin7*.
2. After the installation right click on the Bluetooth icon and chose option. Make sure to
 - Check, General/PAN Network Service
 - Uncheck, Options/IT Admin/Bluetooth Assistant Settings
 - Uncheck, Options/Other/Healthcare Device Settings
3. The devices need to be paired to the Bluetooth stack. Right click the Bluetooth icon and chose Bluetooth Settings.

3.5.2.1.2 AND Medical Devices

The process to repair Blood Pressure and Weighing Scale device is the same.

1. Remove battery from the device and press the button *START* a several times to discharge the device.
2. Put back the batteries. Now the device should be detectable for 60 s.
3. Press new connection, see Figure 74, and follow the Toshiba Bluetooth Pairing Wizard.

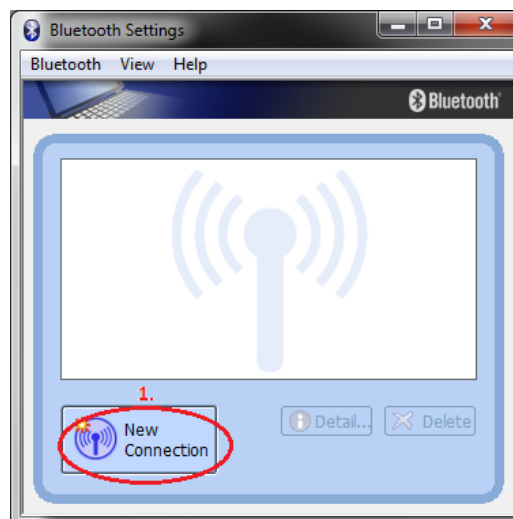


Figure 74: New connection in Bluetooth Settings

4. Choose express mode and click next, see Figure 75

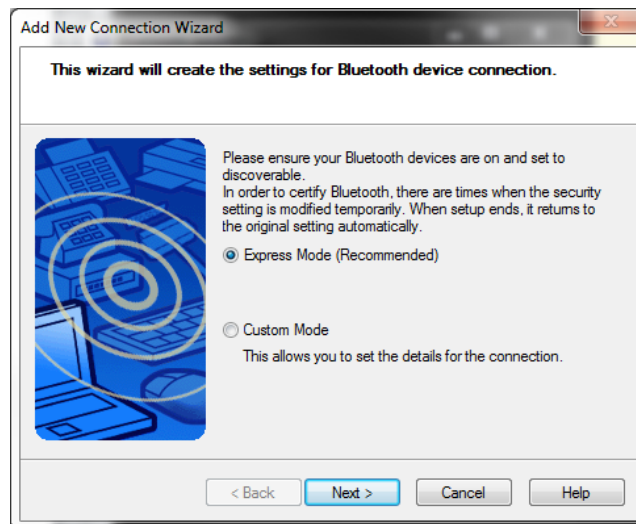


Figure 75: Toshiba Bluetooth device connection Wizard

5. If the device is found by the *Connection Wizard* it will appear under Bluetooth device, see Figure 76. Choose the device and press next. If it is not appearing, press refresh and wait for the list to update. If the device is not appearing or it fails later on, restart the process with 1 and 2 and press refresh.

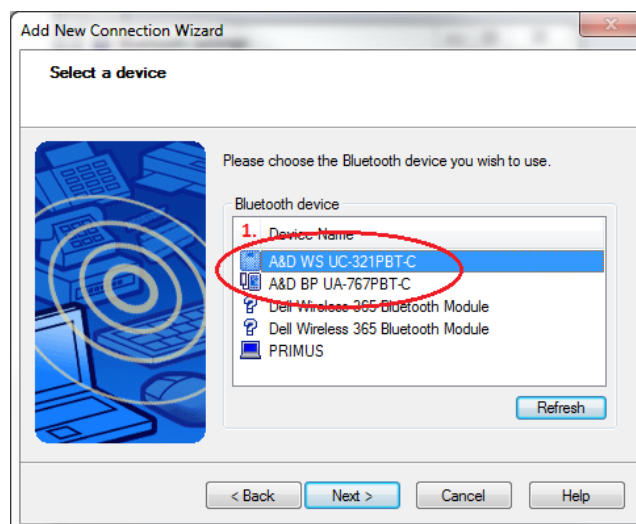


Figure 76: Select device to pair in connection Wizard. (1.) Two AND Medical device found, Blood Pressure and Weighing Scale.

6. Allow the device to connect and if a PIN code is requested use 123456, see Figure 77.

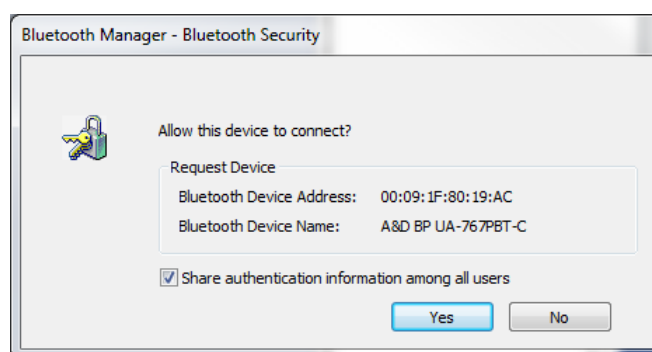


Figure 77: Bluetooth Security

7. When a device is successfully paired it will be showed in the Bluetooth Settings.

3.5.2.1.3 Nonin Pulse Oximeter

1. Insert a finger into the Nonin Pulse Oximeter. The device starts and is discoverable by the *Connection Wizard*.
2. Press new connection, see Figure 74, and follow the Toshiba Bluetooth Pairing Wizard.
3. When the *Connection Wizard* finds the device it will be showed under Bluetooth device, see Figure 78. If the device is not discovery press *Refresh* and wait. If it still doesn't shows up remove the Pulse Oximeter and wait for the display to show *OFF*. Then try again from 1.

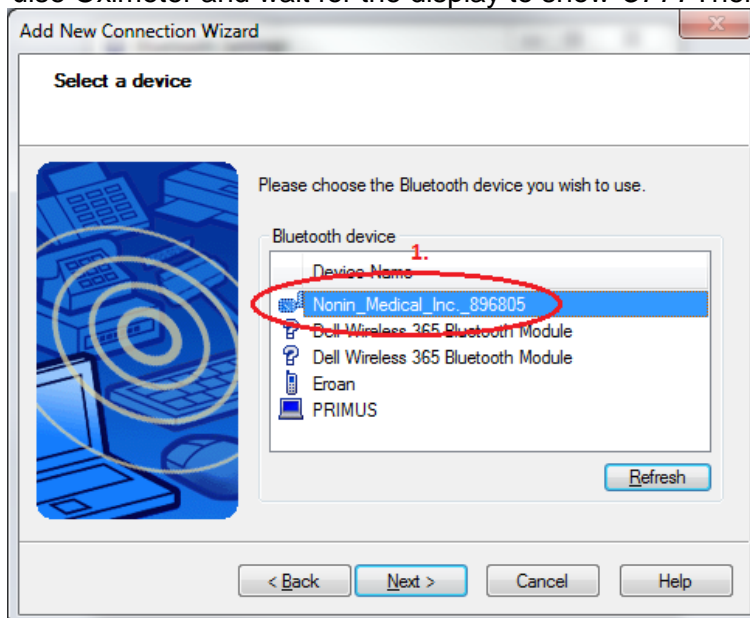


Figure 78: Nonin Pulse Oximeter found by Connection Wizard

4. The device will ask for a PIN code, which is displayed as the last number in the device name, see Figure 79 (1.), or on the side of the device itself. This can occur several times during a pairing. Write the PIN code into the empty field; see Figure 79 (2.).

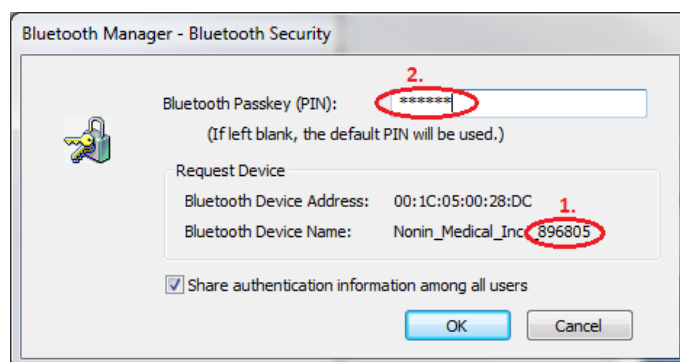


Figure 79: Bluetooth Security, insert PIN code

5. When a device is successfully paired it will be showed in the Bluetooth Settings.

3.5.2.1.4 Glucofacts

In order to get data from the Bayer Glucose Meter the Bayer software *GlukoFacts* must be installed.

1. Unzip "*GLUCOFACTS%20DELUXE.zip*" and install the software.
2. Start "*GlukoFacts.jar*" located in *C:\Program Files (x86)\Bayer HealthCare\GLUCOFACTS Deluxe*.
3. An error will be showed, specifying that no Bayer database is found.

4. Create a new database by clicking “Create a new database and choose the location” and chose the location to C:\Bayer and press create.

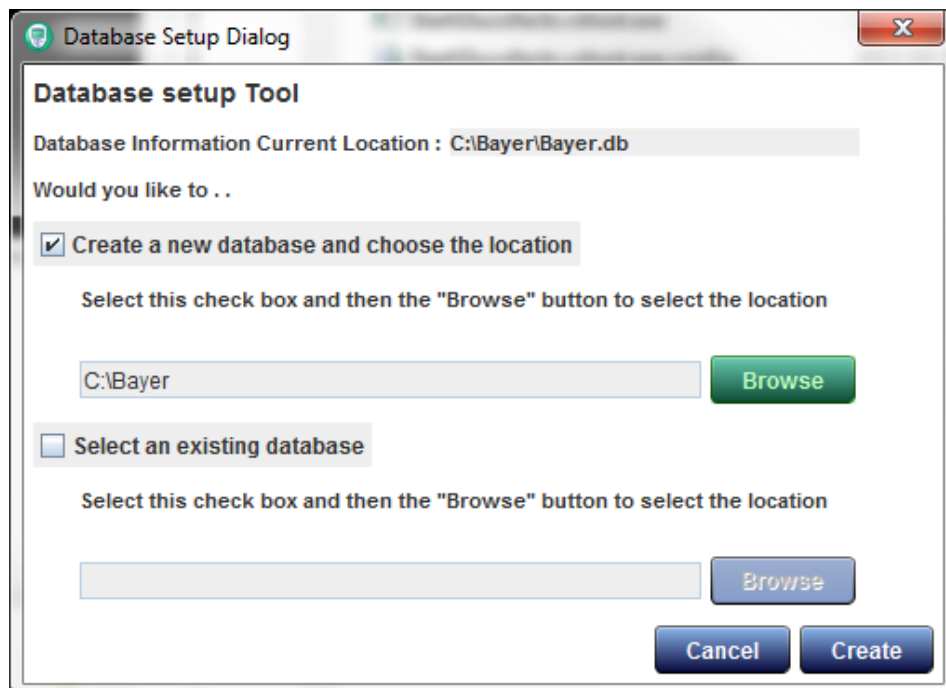


Figure 80: GlucoFacts, Database Setup Dialog

5. In order to get data out of the Bayer USB device, the device needs to be mapped to a person and a meter in the database. Insert the Bayer USB Device and GlucoFacts will notice a new device.

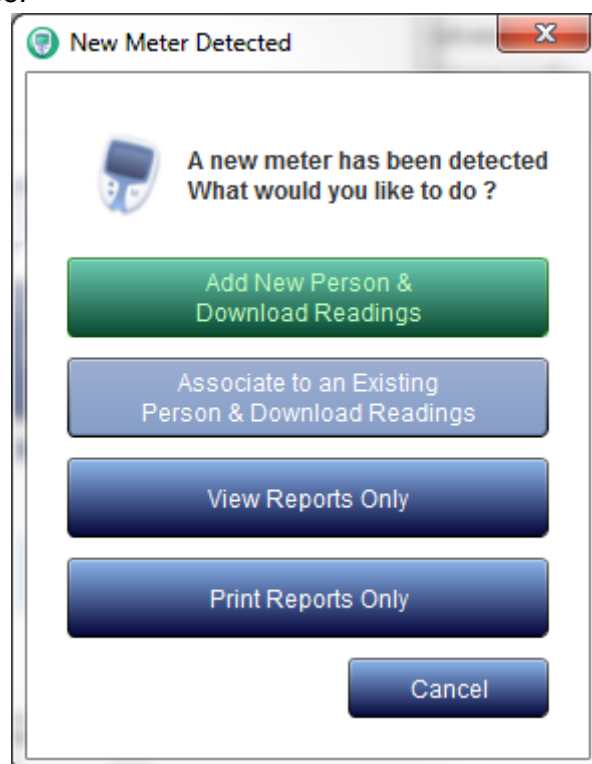


Figure 81: GlucoFacts, new meter detected

Click add new person and fill in:

- First Name = Test
- Last Name = Test

This is only dummy values and will not be used later. Create the person and chose no synchronize between GlucoFacts and the Bayer USB device.

- Click settings (see Figure 82, point 1.) and chose Manage Connection (see Figure 82, point 2.) and disable serial port communication (see Figure 82, point 3.).
- Click settings (see Figure 82, point 1.). Check radio button “Automatically Download Readings” under *If a meter is recognized* (see Figure 82, point 4.).
- Bayer USB device is now ready to be used.
- Each time the Bayer USB device is inserted the measurements will be downloaded to the Bayer Database. Telehealth will then retrieve and convert the measurements not sent and send them to SPP in HL7 format.

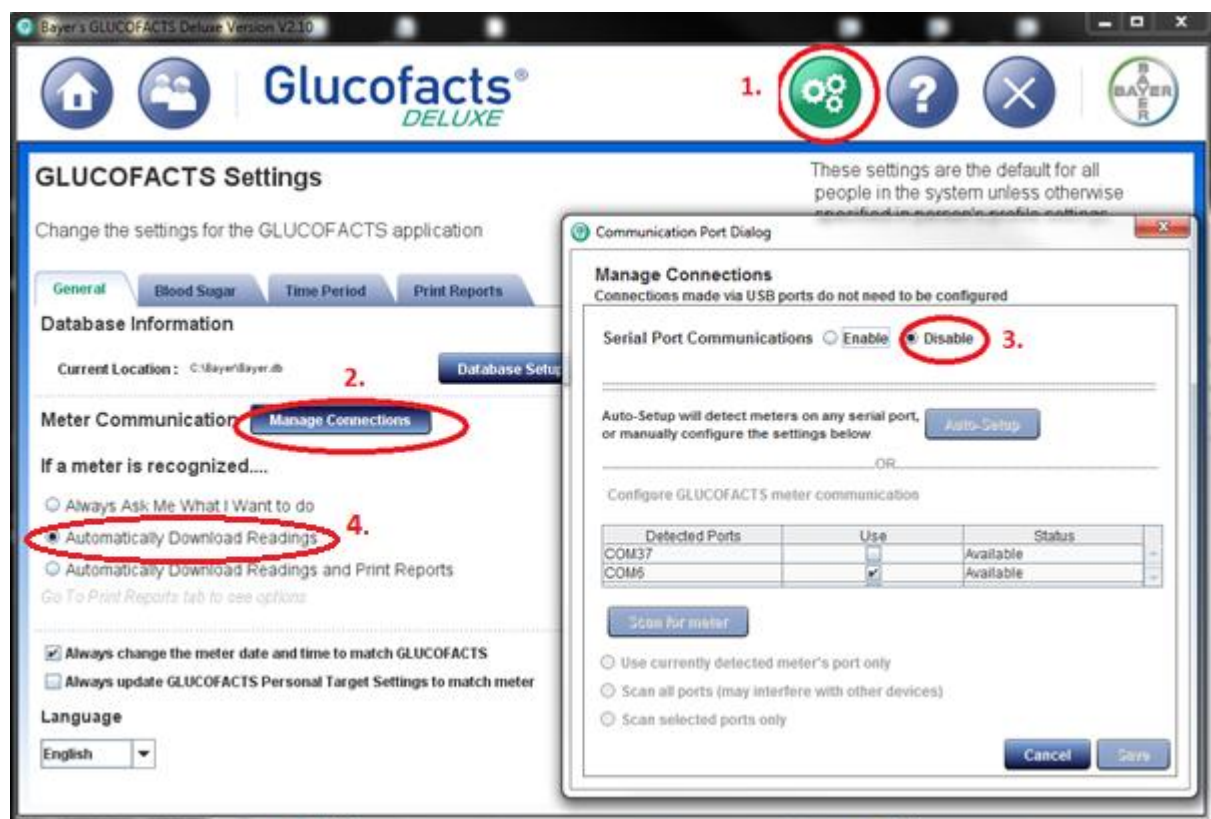


Figure 82: GlucoFacts, settings and manage connection

3.5.2.2 Device and service enabling software

Software and LinkSmart software for use in SKIVE Pilot:

- TeleHealth – handles Blood Pressure Monitor, Weighing Scale, Pulse Oximeter and Glucose Meter
- Telecare – handles the Smoke detector
- CharderPedometer – handles the pedometer data and displays notification of value.
- ObservationClient – Get local stored measurements and sends them to SPP.
- StartGlucofacts – Start Glucofacts software (handles Bayer glucose data) and suppress their windows.
- WSPProcessAutostart – Start programs as windows services.

In other word one need to make the different software work together. This is done by creating the three folders listed below:

1. Create folder “*IoTDeviceDBs*” direct on C, making the path like C:\IoTDeviceDBs.
2. Create a subfolder called “*SKIVE*” in IoTDeviceDBs, making the path like C:\IoTDeviceDBs\SKIVE.
3. Create folders “*LinkSmart*” and “*CommunicationFiles*” direct on C, making the path like C:\LinkSmart\CommunicationFiles

3.5.2.2.1 TeleHealth part

This handles data from Weighing Scale, Blood Pressure Monitor, Pulse Oximeter and Bayer Glucose Meter. Both Weighing Scale and Blood Pressure devices also communicates to LIVA and waits for a confirmation before sending the HL7 message to SPP. Pulse Oximeter and Bayer Glucose Meter stores HL7 message in C:\IoTDeviceDBs\SKIVE\MessageStorage.db3 where ObservationClient fetch and sends them to SPP.

Unzip the TeleHealth.zip into desired path, example: C:\LinkSmart\SKIVE\TeleHealth\

Configuration

There are two configuration files, *configuration.xml* and *TeleHealth.exe.config*

1. In the configuration.xml there are xml elements dealing with patient information. <SetPID> and <SetPV1> set to 1 if they should be used in the HL7 message. <IdNumber> and <UniqueHomeID> represent the Patient ID and Home ID mapped in the EPR on the server.

```
<PatientInformation>
  <SetPID>1</SetPID>
  <!--0 = not use | 1 = use-->
  <IdNumber>12345677890</IdNumber>
  <!-- Patient visit -->
  <SetPV1>1</SetPV1>
  <UniqueHomeID>AP999</UniqueHomeID>
  <AssignAuthority></AssignAuthority>
</PatientInformation>
```

2. In the TeleHealth.exe.config there is xml element <TeleHealth.Properties.Settings> containing information about common path where information will be stored or gathered.

3.5.2.2.2 TeleCare part

This handles data from the Smoke detector. When smoke is detected an event is raised and a HL7 message is stored in C:\IoTDeviceDBs\SKIVE\MessageStorage.db3 where ObservationClient fetch and sends them to SPP. Due to that no data is sent when there is NO SMOKE there is a timer counting down before creating the NO SMOKE message. This timer can be configured in DeviceInformation.xml, see below.

Configuration

There are two configuration files, *Device\DeviceInformation.xml* and *TeleCare.exe.config*

1. In the DeviceInformation.xml there is an xml element <Patient> where patient information needs to be set. <PatientID> and <ResidenceID> represent the Patient ID and Home ID mapped in the EPR on the server. An important xml element is <TimerInterval> describes amount of time in ms before an NO_SMOKE signal is triggered.

```
<Patient>
  <PatientID>999</PatientID>
  <ResidenceID>APIC16002</ResidenceID>
</Patient>
```

```

<Device>
  <TypeID></TypeID>
  <Type>SmokeDetector</Type>
  <EquipmentID></EquipmentID>
  <Address></Address>
  <SerialNumber>123456789</SerialNumber>
  <Manufacturer>Cigarette</Manufacturer>
  <Model>CDA-GDV</Model>
  <TimerInterval>5000</TimerInterval>
</Device>

```

2. The TeleCare.exe.config has information about the location of the MessageStorage.db3

3.5.2.2.3 CharDerPedometer

Handles data from the CharDer Pedometer and stores HL7 message into C:\IoTDeviceDBs\SKIVE\MessageStorage.db3 where ObservationClient fetch and sends them to SPP.

When the device have downloaded its measurements there will be a notification showed for the user.

Unzip the CharDerPedometer into desired path, example: C:\LinkSmart\SKIVE\CharDerPedometer.

To start CharDerPedometer automatically, drag and drop a shortcut into Start\Startup (see Figure 10). Next time the PC starts the CharDerPedometer will start.

Configuration

There are two configuration files, *configuration.xml* and *CharDerPedometer.exe.config*

1. In the configuration.xml there are xml elements dealing with patient information. <SetPID> and <SetPV1> set to 1 if they should be used in the HL7 message. <IdNumber> and <UniqueHomeID> represent the Patient ID and Home ID mapped in the EPR on the server.

```

<PatientInformation>
  <SetPID>1</SetPID>
  <!--0 = not use | 1 = use-->
  <IdNumber>12345677890</IdNumber>
  <!-- Patient visit -->
  <SetPV1>1</SetPV1>
  <UniqueHomeID>AP999</UniqueHomeID>
  <AssignAuthority></AssignAuthority>
  <WeightKg>75</WeightKg>
  <Stride>70</Stride>
</PatientInformation>

```

There are also two other xml elements. First is the user approximated weight <WeightKg> and the length of the stride in centimeter <Stride>. This is information used by the CharDerPedometer to calculate distance and calories burned and cannot be fetched from the real device.

2. The CharDerPedometer.exe.config has information about the location of the MessageStorage.db3

3.5.2.2.4 StartGlucoFacts

Software that automatically starts the GlucoFacts Deluxe on windows start-up. The software also suppresses the window and hides it for the user.

Unzip the StartGlucoFacts.zip into desired path, example: C:\LinkSmart\SKIVE\StartGlucoFacts\

To start GlucoFacts Deluxe automatically, drag and drop a shortcut into Start\Startup (see Figure 83). Next time the PC starts the StartGlucoFacts will start GlucoFacts Deluxe.

Configuration

Make sure that StartGlucofacts.exe.config (can be edited with notepad) points to the right GlucoFacts installation folder. There is also an xml segment to determine if the GlucoFacts windows should be suppressed or not by setting the showWindow <value> True/False.

```
<setting name="fullPath" serializeAs="String">
  <value>C:\Program Files (x86)\Bayer HealthCare\GLUCOFACTS Deluxe</value>
</setting>
<StartGlucofacts.Properties.Settings>
  <setting name="ShowWindow" serializeAs="String">
    <value>True</value>
  </setting>
</StartGlucofacts.Properties.Settings>
```

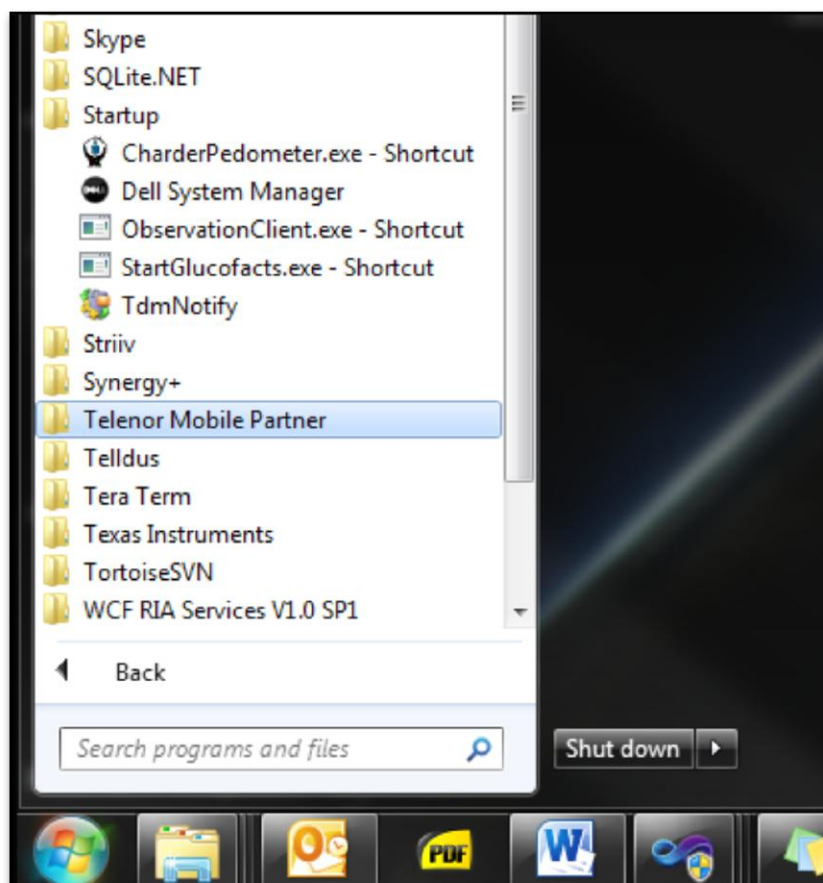


Figure 83: StartGlucoFacts shortcut into Start\Startup

3.5.2.2.5 ObservationClient

The Observation Client application is used to fetch stored HL7 messages in C:\IoTDeviceDBs\SKIVE\MessageStorage.db3 and sends them to SPP.

Unzip ObservationClient into desired path, example: C:\LinkSmart\SKIVE\ObservationClient

To start ObservationClient automatically, drag and drop a shortcut into Start\Startup. Next time the PC starts the ObservationClient will start.

Configuration

There is one configuration file, "*ObservationClient.exe.config*".

- DeviceDbPath – path to MessageStorage.db3
- IsSending – if stored messages should be sent
- InvokeTime – how often in minutes ObservationClient will try and resend messages.
- RemoveObservation – True/False, determine if the message will be removed it has been sent to SPP
- ShowWinow – True/False, determine if the console window should be suppressed and hidden.

3.5.2.2.6 WSPProcessAutostart

Telehealth, Telecare and Toshiba Bluetooth stack has been tested and can run as Windows Service. This means that the different software will always run on the computer.

This is used if one does not want to use the LinkSmartDotNetServices as auto start.

Unzip the WSPProcessAutostart.zip and into desired path, example: C:\LinkSmart\SKIVE\WSPProcessAutostart.

Configuration

Open the WSPProcessAutostart.exe.config and make sure that the segments "LogFolder" and "ProcessXml" has the right path for the files, due to the above unzip.

Open the WSPProcessAutostartProcesses.xml, check the paths and extend the processes to start. To extend the processes to start fill with:

```
<process id="n" exe="NameOfTheExe.exe" homeDirectory="Fullpath"
sleepTimeAfterStartSeconds="1"/>
```

Installation

1. Open command prompt as administrator.
2. Locate the WSPProcessAutostart.exe location
(C:\LinkSmart\SKIVE\WSPProcessAutostart\WSPProcessAutostart.exe)
3. Locate InstallUtil.exe within the Microsoft .Net framework.
(C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe)
4. Execute the command in one go, with "".
"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
"C:\LinkSmart\SKIVE\WSPProcessAutostart\WSPProcessAutostart.exe"
5. Confirm the installation by looking in Services that the WS Process Autostart is installed (Control Panel/Administrative Tools/Services)
6. Change the startup to automatic
7. If TosBtMng.exe should start as Windows Service, make sure that the default Start\Startup is unchecked. This can be done by use "Run" (Windows button + r) and write msconfig. Under the tab startup, uncheck Bluetooth stack for Windows by Toshiba.

To check if the applications is running open the Windows Task Manager. There you should find TosBtMng.exe, TeleHealth.exe and TeleCare.exe is running as processes for the SYSTEM users.

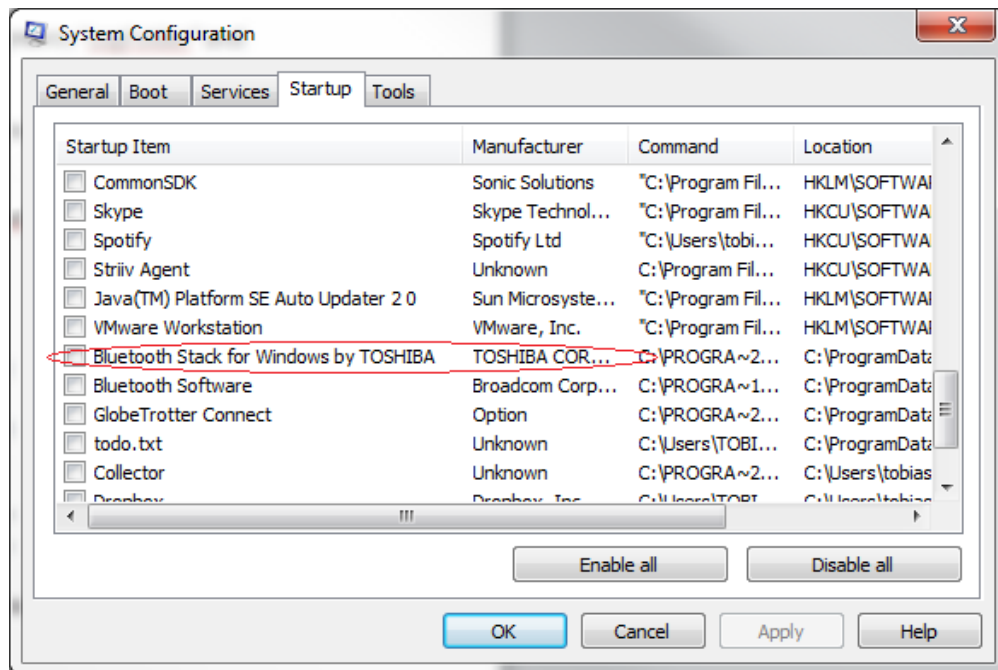


Figure 84: Msconfig/Startup and Toshiba Bluetooth stack

3.5.2.3 LIVA as Home Base station GUI

The LIVA Home Base station is based upon a standard Windows7 PC with a touch sensitive screen. The setup is as described in the figure below

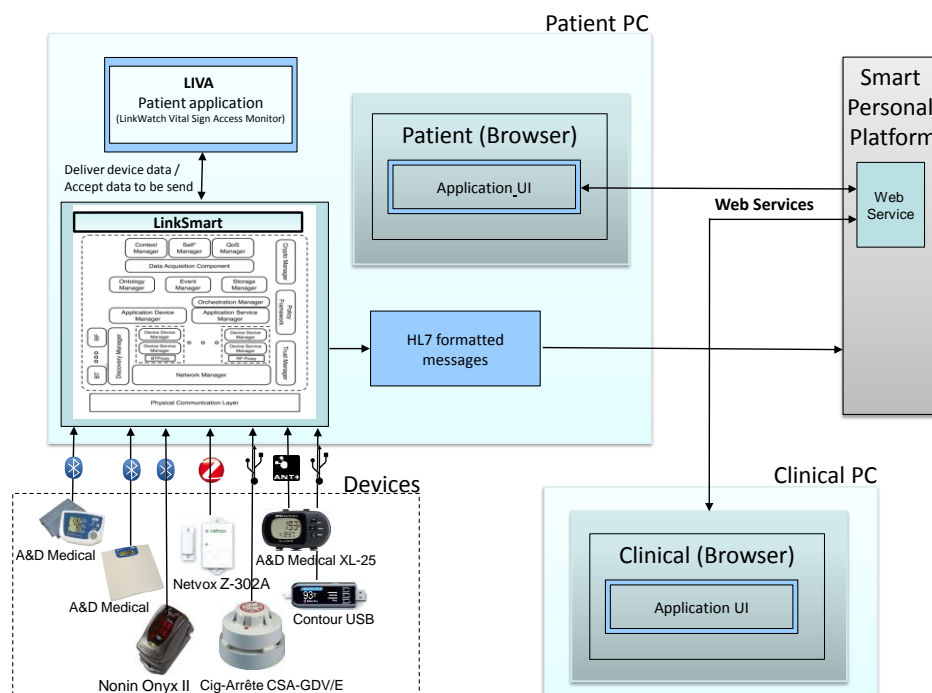


Figure 85: LIVA setup

On the patient PC, the following components are installed:

1. The LinkSmart Engine
2. The LIVA Patient Application
3. The Patient Browser

Connected to the PC are the devices in which all are connected either wirelessly or wired to the PC.

3.5.2.4 Data management on client side

Data from all of the devices are transmitted to the LinkSmart engine and processed by LinkSmart. Depending of the source, the data are transmitted in the following sequence:

1. **Event data from the Cig-Arrête**

The Cig-Arrête device is connected by a USB cable to the PC. When smoke is detected the signal is transmitted to the PC and handled by the LinkSmart engine. The message is directly transmitted in a HL7 formatted message by LinkSmart to the SPP, Smart Personal Platform. The patient does neither see any message on the screen or any signal from the Sensor.

2. **Window (Netvox) sensor**

The Windows sensor device is connected by a wireless ZigBee connection to the PC. When the window open/closed event is detected a signal is transmitted to the PC and handled by the LinkSmart engine. The message is directly transmitted in a HL7 formatted message by LinkSmart to the SPP, Smart Personal Platform. The patient does neither see any message on the screen or any signal from the Windows sensor.

3. **Pedometer reading**

When the Pedometer is connected to a USB connection on the PC, the Pedometer data are read and transmitted in a HL7 formatted message by LinkSmart to the SPP, Smart Personal Platform. The transmitted data records are also showed on the screen as a windows pop-up message.

4. **Blood oxygen and blood glucose reading**

The measurements are transmitted in wireless way to the PC. The measured data are read and transmitted in a HL7 formatted message by LinkSmart to the SPP, Smart Personal Platform. The transmitted data record is also showed on the screen as a windows pop-up message.

5. **Weight and Blood pressure data**

These data are collected by LinkSmart and transmitted on the PC to the LIVA Patient application. The LIVA Patient application takes controls of the MMI and guide the patient through the process in reading and managing the weight and blood pressure measurement. When the reading has been successfully completed, the patient can decide if this measurement should be transmitted to the SPP, Smart Personal Platform. If the transmission is decided, the LIVA Patient application activates LinkSmart to transmit the measurements in a HL7 formatted message to the SPP, Smart Personal Platform.

The patient browser is a standard Windows IE browser, in which the patient can access presentation of own data. This includes both current data and historical data. The data presentation can either present data from a single measuring source, or combine data from different measuring sources.

3.5.2.5 Skive Consumer Application Extension

The deployed Skive Consumer Application offers the unique feature that a patient can also access their data via the Web Portal. In the inCASA Pilots, access to the Consumer Application was given only to the Professional Users. In Skive transferability test though, it was included in the requirements to allow patients to enter the Portal and be able to view only *their own data and nothing else*. Another Role-Based Access Mechanism was therefore developed in the Skive CAs using JavaScript, in which when the user logs in with credentials associated with a patient, they can only view their data and there is nothing rendered from any other patient.

Apart from the above extension, new unique visualizations were added to the Skive CAs due to the addition of new monitored parameters like smoke or activity.

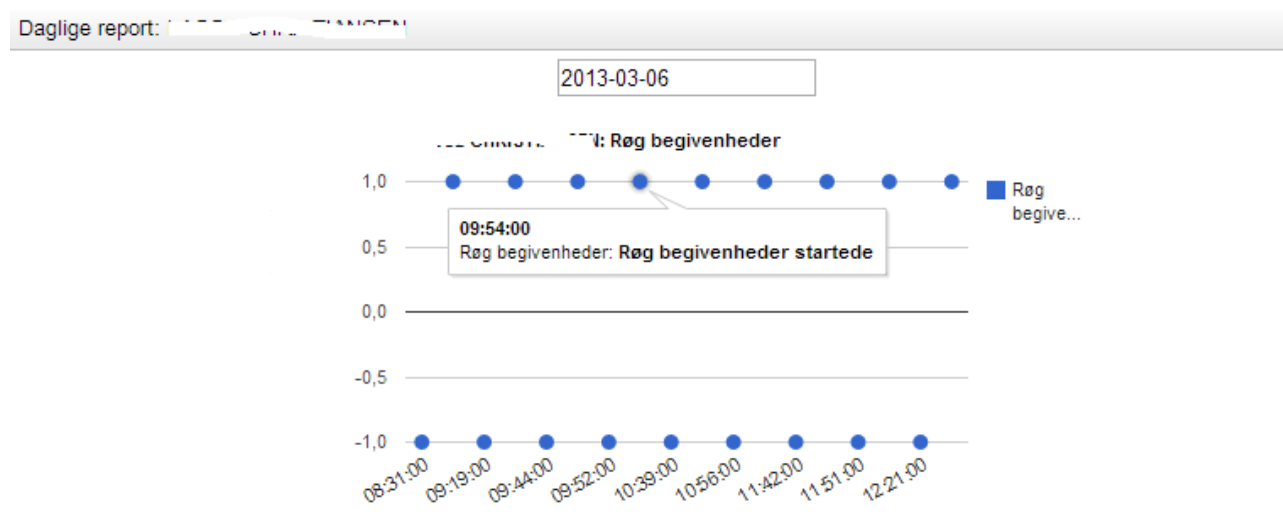


Figure 86: Daily smoke report (1 is smoke ON, -1 is smoke OFF) with hidden patient name

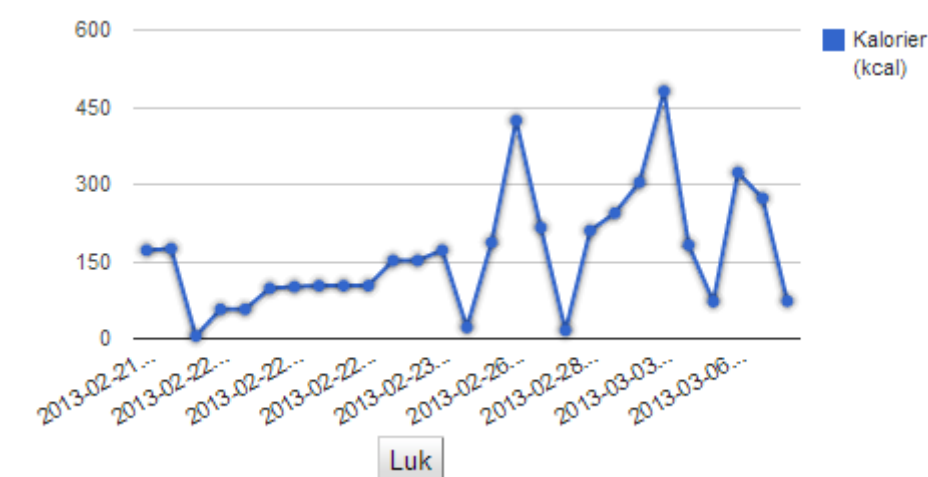


Figure 87: Calories burned graph (sub-measurement of Activity monitoring graph)

3.6 CHC pilot

3.6.1 Chorleywood (UK) procedures

The general approach to commissioning has been described. However, we have chosen to pre-pair all devices with the gateway in advance of installation with the patient to ensure there are no problems and time in the home is minimised. All pairing is performed at Brunel, and a complete “kit” of requested devices for each patient is provided to Chorleywood for installation. The devices are tested to ensure they are working.

Devices must be allocated to a patient. This is performed through the clinician portal. We do not include any patient identifiable information in messages sent from the gateway to overcome issues of privacy and security. Instead, all devices have a unique identity (EUI-64) and this is used to map incoming data to the correct patient record.

3.6.2 Experiences

Installation of the gateway is simple but includes finding an appropriate site for the gateway as mobile signal is poor in several areas. We have obtained a GSM signal strength meter to assist. There have also been issues around finding available power sockets for devices, such as the bed sensor, as sockets in the bedroom are overcrowded, and often in difficult to reach places (e.g.

behind the bed). Care is also exercised to ensure devices in the home are within operating range of the gateway. This is checked by inserting the batteries in the device and ensuring that the green LED shows solid.

That the gateway is small and unobtrusive is proving advantageous, as elderly people wish to avoid the stigmatisation of technology in open view. Furthermore it removes the perceived (or actual) need for devices to be close to a station, and so allows devices to be placed discretely in the home at the preference of the patient.

The gateway is self-contained and requires no configuration by the user.

3.6.3 Solution extensions

The UK solution is designed to operate as a single platform. It is easy to install and devices are designed to be easy to use. There are no user interfaces to further simplify use and enable patients to place devices for convenience and to fit with routine and lifestyle. As we develop the ZigBee radio modules, we may extend the range of devices as required.

3.6.3.1 Solution overview

The UK solution has been purposely designed for simplicity in installation and use. A single, simple, gateway for all devices reduces system cost significantly. The system is capable of being self-installed, but for inCASA we have chosen to install as patients are frail and elderly. Removing installation cost will have advantages in some applications and will allow the solution to scale.

The wide range of devices covering telehealth and independent living on same the platform is attractive, as it supports multiple co-morbidities and complications, along with managing the frailty. Analysis of data may show correlation between data sets and power in their combination.

4 Conclusion

The inCASA integration architecture is an instance of a complex Event-Driven Architecture. This integration architecture pattern has been selected with the goal of managing large flows of events (messages, measurements, alerts) and the consequent reactions to these, enabling both operators and systems to respond as much effectively and timely as possible.

The widespread use of the HL7 interchange format, according to the Continua guidelines, allows the interoperability among the several subsystems and between the inCASA system and the outside world, e.g. the addition of new devices, or the integration with Consumer Applications.

Integration and installation made in any inCASA pilot site proceeds from the Base Station up to the Service Provider platform and to the Consumer Applications. Key for the integration at each level is a middleware component that enables the integration by means of standard, HL7-based interfaces or by other extension of customised pilot solution.

At the Base Station level, the Activity Hub and the SARA Client ensure the integration of, respectively, telecare and telehealth sensors for the collection of events generating in the patient's house. In the UK pilot there is an alternative solution using ZigBee gateway and devices.

The LinkSmart middleware provides the integration towards the Service Provider. Its two parts glue the Base Station and the Service Provider together, handling the continuous flow of measurements coming from the possible several houses in the inCASA network.

At the Service Provider level, the Mediator component integrates the Reasoner and EPR core logics in order to recognize and to store the multiple events and to both propagate the necessary alerts to the Consumer Applications where users are allowed to execute data requests.

As a summary, all pilot installations (with or without extensions) went well and customisable considerations were met with great enthusiasm and progressed with only minor changes. As such, any care giver will with this deliverable have a good base when adopting the inCASA solution in their national or regional environment.

5 Glossary

API	Application Programming Interface
CEP	Complex Event Processing
EDA	Event-Driven Architecture
EPR	Electronic Patient Record
EUI-64	An 8-byte hexadecimal Extended Unique Identifier number defined by the IEEE
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HBS	Home Base Station
HDP	Health Device Profile
HL7	Health Level 7
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IHE	Integrating the Healthcare Enterprise
IP	Internet Protocol
LE	Low Energy
PAN	Personal Area Network
PCD	Patient Care Device
PIR	Passive Infrared (Motion Sensor)
SOAP	Simple Object Access Protocol
SPP	Smart Personal Platform
USB	Universal Serial Bus
WAN	Wide Area Network
WP	Work Package

6 References

- [1] IHE Patient Care Device (PCD) Technical Framework Volume 1 Revision 1.0 Final Text August 12, 2011 (IHE_PCD_TF_Vol1_FT_2011-08-12.pdf)
- [2] G. Lamprinakos, *D3.1 System and Functional Specifications*, inCASA Project – 250505, Report 18-04-2011 2011
- [3] J. Rovira (TID) , other co-Authors (REPLY, SIG, NTUA, CNET), *D3.3 Reference Architecture iteration 2*, inCASA Project – 250505, Report 07-03-2012
- [4] S. Asanin (CNET), other co-Authors (REPLY, SIG, NTUA, TID), *D4.3 Advanced Monitoring System Implementation*, inCASA Project – 250505, Report 15-07-2011
- [5] C. Barbero, P. Dal Zovo, B. Gobbi, A flexible context aware reasoning approach for IoT applications, 12th International Conference on Mobile Data Management, 2011 12th IEEE International Conference on Mobile Data Management
- [6] Event-driven Architecture, http://en.wikipedia.org/wiki/Event-driven_architecture
- [7] B. M. Michelson, Event-Driven Architecture Overview, February 2, 2006
- [8] Department of Health. (2012-03-14). *Guidance for NHS trusts on the NHS friends and family test*. Available: <http://www.dh.gov.uk/health/2012/10/guidance-nhs-fft/>