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

D4.1 Core Monitoring System Implementation

Stefan Asanin, Tobias Brodén, Matts Ahlsén CNET,
Jordi Rovira Simón TID & Axel Sikora SIG

Project start date: 1st April 2010 Duration: 30 months

Published by the inCASA Consortium 30-06-2011 – version1.0
Coordinating Partner: SANTER REPLY Spa

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

Dissemination Level: Public

Document file: D4-1_Core_Monitoring_System_Implementation_v1.0.docx

Work package: WP4

Tasks: Task 4.1, Task 4.2, Task 4.5 & Task 4.6

Document responsible: CNet Svenska AB

Document history:

Version	Author(s)	Date	Changes made
Draft01	Stefan Asanin	17/5 2011	Table of Contents
Draft02	Stefan Asanin	8/6 2011	Refined ToC
Draft03	Stefan Asanin	16/6 2011	n/a
Draft04	Stefan Asanin	17/6 2011	Refined ToC
Draft05	Jordi Rovira Simón	21/6 2011	Chapter 4.2
Draft06	Tobias Brodén	23/6 2011	Chapter 5.2
	Axel Sikora	23/6/2011	Chapters 4.1, 4.3 & 6.3
	Stefan Asanin	27/6 2011	Chapters 1-4, 4.4, 5, 5.2, 5.3 & 6
Draft07	Stefan Asanin	29/6 2011	Executive Summary & Chapters 7 & 8
Draft08	Matts Ahlsén	30/6 2011	Reference & document check
Draft09	Stefan Asanin	30/6 2011	Revised document

Peer review history:

Reviewed by	Date	Comments
Dirk Lill, SIG	30/6 2011	
Andrea Prestileo, REPLY	30/6 2011	

Index

List of Tables

List of Figures

Executive summary

This document represents the Core Monitoring System Implementation for the inCASA project. The content of that system comprises the inCASA gateway and the inCASA server envisioned through the implementation of the Hydra Middleware [1]. It deals with relevant subsystems connecting to the inCASA gateway, the communication between gateway and server side as well as the way the basic and intelligence layers of the inCASA solution architecture exposes underlying functionalities. The document also describes the features of each subsystem and component in order to give full insight in how the Core Monitoring System implementation is most properly conducted. A development and test plan is presented that will iteratively run over WP4 deliverables and for this deliverable a first set of test notations are given, analysed and reasoned upon.

1 Introduction

1.1 Background

The EU has outlined some particular dimensions in the 2006 e-inclusion targets (REF) that deals with the concern of growing healthcare demands and costs. It is related to an ageing population in Europe. This is one of the most important challenges faced by governments at local, regional, national and European levels. Demographic trends in Europe indicate that people age 65 or older already make up nearly one-fifth of the population in many European countries, and the proportion is rising. Low or falling birth rates and improved health and medical care that enable people to live longer are the main causes. Social services are limited with the support resources offered to the ageing population. Therefore, the key challenge is to promote healthy and active ageing for European citizens regardless available resources per member country. The outcome would imply that more years in good health equals a better quality of life, greater independence, and the possibility to remain active. But an ageing population in good health would also mean less strain on health systems and fewer people retiring from work due to ill-health. Suggestively this would have a positive impact on Europe's economic growth.

The market for products and services for ICT and ageing is still in its infancy and impose the first obstacle for realising a solution to the challenge. It plays a part as one way of solving the future problem of ageing independently. It is to enable a population to function in their natural and traditional environment, indoors and outdoors by the use of technology (*in order to maintain a better physical and mental condition when living in their own home and familiar surroundings*). The more and more growing percentage of elderly persons also justifies developing these kind of tools and to empower these individuals to contribute and play an active role in modern society. But low market awareness and visibility, lack of standards and interoperability, and uncertainty about the sustainability of business models constitute barriers for such an uptake. The inCASA project addresses these barriers and exploits the opportunities by placing the users at the centre. As such, the inCASA project will focus on the demographics and social development as of primary importance. The proposed solutions will be designed to make real difference and provide real impact when the service is deployed. Elderly people, in using the services to help their day-by-day living, may increase their independence and self-confidence (or prevent loss). When succeeded, the inCASA solution will meet the higher demands for healthcare by adapting its subsystems to the needs of an ageing population per region and country while keeping them sustainable in societies with a smaller workforce.

The objectives of the inCASA project will be reached by realising and testing in specific pilots efficient integrated care systems that combine innovative technological platforms for ubiquitous communication, advanced healthcare monitoring and state of the art domotic systems. This will be based by the inCASA subsystems that jointly will collect and manage data unobtrusively and non-invasively on behaviour, using wireless medical devices, detection of movement and activity. The architecture combines multiple types of sensors and will base their potentiation by integrating further input and/or resource data as to increase profiling accuracy and achieve the medical target. As such, the inCASA platform will be open for extensions and scalable.

1.2 Purpose and Content of this Deliverable

Work Package (WP) 4 will work on presenting the inCASA solution implementation. Moreover, it deals with the iterative implementation of all the parts of the inCASA system as well as the technical testing of its functionality. Its iterative processes involve a development plan that describes how the different software modules, subsystems and systems developed in inCASA will be integrated in order to easily interoperate while the test plan will be the actual basis for the testing process. This deliverable (D4.1) and the deliverables D4.2 and D4.3 will formalise a set of actions including any procedure, process, equipment, material, activity or system. These sets will help to understand whether the system performances meet the required specifications and quality attributes set in WP2.

The results will be validated according to certain validation criteria. It additionally determines needs for adjustments of the project user requirements specifications and design. The work is divided into the following implementation tasks (those estimated to be directly or in-directly relevant for D4.1 are presented in *italic* and with extended description in chapter 4). Each task focuses on a specific component in the inCASA framework.

- Task 4.1 – Hydra Customization and Sensor network Set up
- *Task 4.2 - Remote Monitoring gateway implementation*
- Task 4.3 - EPR and Application Customization
- Task 4.4 - Reasoning and Learning System
- *Task 4.5 – Service Delivery Platform*
- *Task 4.6 - Telehealth Applications*

The purpose of this deliverable is to describe the iterative implementation of inCASA's Core Monitoring System mainly consisting of the basic remote communication occurring between the inCASA Client and Server sides. It will also be based on those aspects that may affect the modifications necessary on the Hydra Middleware in order to provide sufficient adaptability in the inCASA Core Monitoring functionalities. This means that the Activity Hub and SARA client do both play a crucial role on how data is managed in the core of the inCASA platform. It will also present the first approach on development and testing plans which may be further modified to fit the technological scopes and requirements set by tasks in deliverables D4.2 and D4.3.

1.3 Outline of this Deliverable

This deliverable will begin describing the basics behind the inCASA Core Monitoring System in chapter 2. Here it will also relate those work tasks that have to do with the deliverable's intention and describe how these tasks contribute to an architectural solution for the inCASA platform. This architecture is presented as a draft where consortium partners and subsystems are mapped out. Chapter 3 goes through the development and test plans and tries to distinguish between work package vs. project specific plans in order to show how any work package related plan affects the general project development plan and what this is good for. The following chapter (4) lines up each subsystem's features and it shows how this possibly would interface the Core Monitoring System. Chapter 5 continues with the customization of the Hydra Middleware and what of its components are suitable for the inCASA Middleware parts. How to build an inCASA Core Monitoring System that manages all of the aspects discussed so far in the deliverable is treated in chapter 6. It describes how each subsystem is integrated in the Hydra Middleware and how Hydra exposes underlying subsystem functionalities to receiving SPP and others. Chapter 7 describes both the iterative tests conducted during integrating the Core Monitoring System but also the experiences gathered doing so. Finally, chapter 8 gives a summary and validation section that deals with the amount of success of this deliverable.

2 Description of the Core Monitoring System

The inCASA Core Monitoring System forms the basic steps of the platform, i.e. data collection. This very first step is to ensure that data enquired through sensors/inputs using Bluetooth, Wi-Fi, ZigBee and many more, is correctly gathered and the second step is to make this data available for service deployment on the remote end of the solution. The data is collected from the environment monitoring system (e.g. wireless motion/contact sensors to monitor the senior in their own home and smoke detector, natural gas detector, flooding sensors/water electro valves, and temperature) and pre-existing human monitoring system (e.g. the Piemonte Region frail people tele-monitoring system). Also, collected data related to pre existing monitoring system (e.g. domotic and health related) will be managed via the dedicated base station. Collected data from the monitoring system related to user behaviour will further be logged into the base station's data repository.

A key functionality of the inCASA platform is the synchronization between the base station and the carer headquarters. This is because the functionalities are predicted to allow the development of the behaviour model that will be the head feature of the platform. The data collected by the base station are periodically sent to the service provider (e.g. daily file transfer) and are used to create the behaviour model of the person (via the learning system) and to generate reports and/or alerts in case of anomalies.

The base station implementation is said to constitute one side of the inCASA Core Monitoring system. It will perform checks on collected data but also generate alerts (e.g. send SMS/Email/Web notifications) and/or take corrective actions (e.g. close water, close gas) based on the urgency of the issues (possibly set by the Telecare or Telehealth subsystems). In case of need the base station (initiated by the Telecare or Telehealth subsystems) should be able to generate an alert to the remote service provider for assistance, providing also all the data collected (file transfer of the data collected on the base station repository). In this way the headquarters may analyse it, and initiate the correct response. How much is implemented of these alert mechanisms by this deliverable depends on the distribution of work designated for the related WP4 tasks and if their time estimation span over the other WP4 deliverables. Furthermore, additional dependency is the actual need of alarm triggering set by the base station for the Smart Personal Platform (SPP) which will be described in D4.3. Still, the possibilities to generate alarms on the base station should be highly considered. For the development of the inCASA Core Monitoring System the use of the Hydra Middleware will be principal and consistent over the overall inCASA remote monitoring solution (please refer to the draft Core Monitoring Architecture in chapter 2.2 for more information).

In sum, the Core Monitoring System is based upon the base station functionalities that in turn are implemented by using the Hydra Middleware Client together with parts of its counter peer virtualization and representation on server side.

2.1 Task Related Work

For each WP4 deliverable there is a set of work tasks that more or less relate to each deliverable's content and intention. For D4.1 the major contributions that can be derived from the tasks are Task 4.1, Task 4.2, Task 4.5 and Task 4.6. Here comes first an high-level relation to the implementation work of D4.1 and then a task specific description:

- To install the Hydra Middleware on separate remote locations (Task 4.1 and Task 4.2).
    - Ensure communication between system settings.
    - Store collected data.
- To create automation and allow for distributed and remote control of domestic applications by integrating technologies (i.e. basic technology found in Task 4.5 and Task 4.6).

- Use the Hydra DDK and SDK [2] to configure how data from subsystems should be represented (i.e. basic technology of Task 4.5). This point requires subsystems to be platform cross-functional.

Task 4.1: Hydra Customization and Sensor network Set up (*Data survey Center implementation*); here data survey will better refer to automatic data collection, storage and its implementation using the Hydra Middleware. This task will cover the customization of the Hydra Middleware in order to manage the monitoring system. It will mainly take care of the following:

1. Customization of the Hydra middleware on the base station and its configuration to successfully collect the data from the monitoring system (health monitoring/ environment monitoring).

Task 4.2: Remote Monitoring gateway implementation. This task will cover the implementation of the base station applications to be able of the following:

1. Automatically collect data, via Bluetooth or Wi-Fi, from the health monitoring system and the environment monitoring system, and
2. Store collected data on the base station.

Task 4.5: As this task will cover the implementation of the operating central (i.e. remote healthcare provider platform services), the work for D4.1 will be based on simply provide the basis for such service exploitation throughout the inCASA solution on both ends of the inCASA solution.

Task 4.6: Telehealth applications involve the use of SARA, a telemedicine platform, provided by Telefonica, that allows carers to take contact and supervise their chronic disease patients remotely. The patients can do most of the common tasks related with their chronic conditions from their home, with the remote assistance of the health professionals. As such, with the use of the SARA client it conclusively proposes that the Core Monitoring System will need to be able to communicate and handle various input components and services (i.e. subsystems integrating with the Hydra Middleware, alias inCASA Client side).

2.2 Draft Core Monitoring Architecture

Creating a draft architecture of each of the topics in WP4 deliverables enables the consortium partners to ease the development when having a common model view of the inCASA solution (i.e. how subsystems are meant to co-work). The intention of such models is to give the inCASA platform the advantages of existing subsystems and products and integrate them in order to create an enhanced socio-medical platform. This platform should be able to monitor both user behaviour and clinical conditions as well as create a user behaviour model.

What this draft architecture aims to do is to present new services created for this platform and lead the way for the work done in D4.2 and D4.3. It particularly gives focus on creating user profiles that can be used in different regional and national healthcare information systems (HIS), creating a common working framework (i.e. the underlying part of the draft Core Monitoring architecture) and to finally provide integrative support for creating a scalable and open platform that easily allows further integration of additional services.

By benchmarking the software platform architecture described in the inCASA DoW[1], WP4 may postulate that the original view of the platform goes on as sustainable from the point that service specific requests in such solution would be end-to-end driven (e.g. from home - over Internet - to

[1] inCasa project Description of Work

service provider and vice versa). Looking at Figure 1 one can easily map the draft Core Monitoring architecture segments presented in Figure 2.

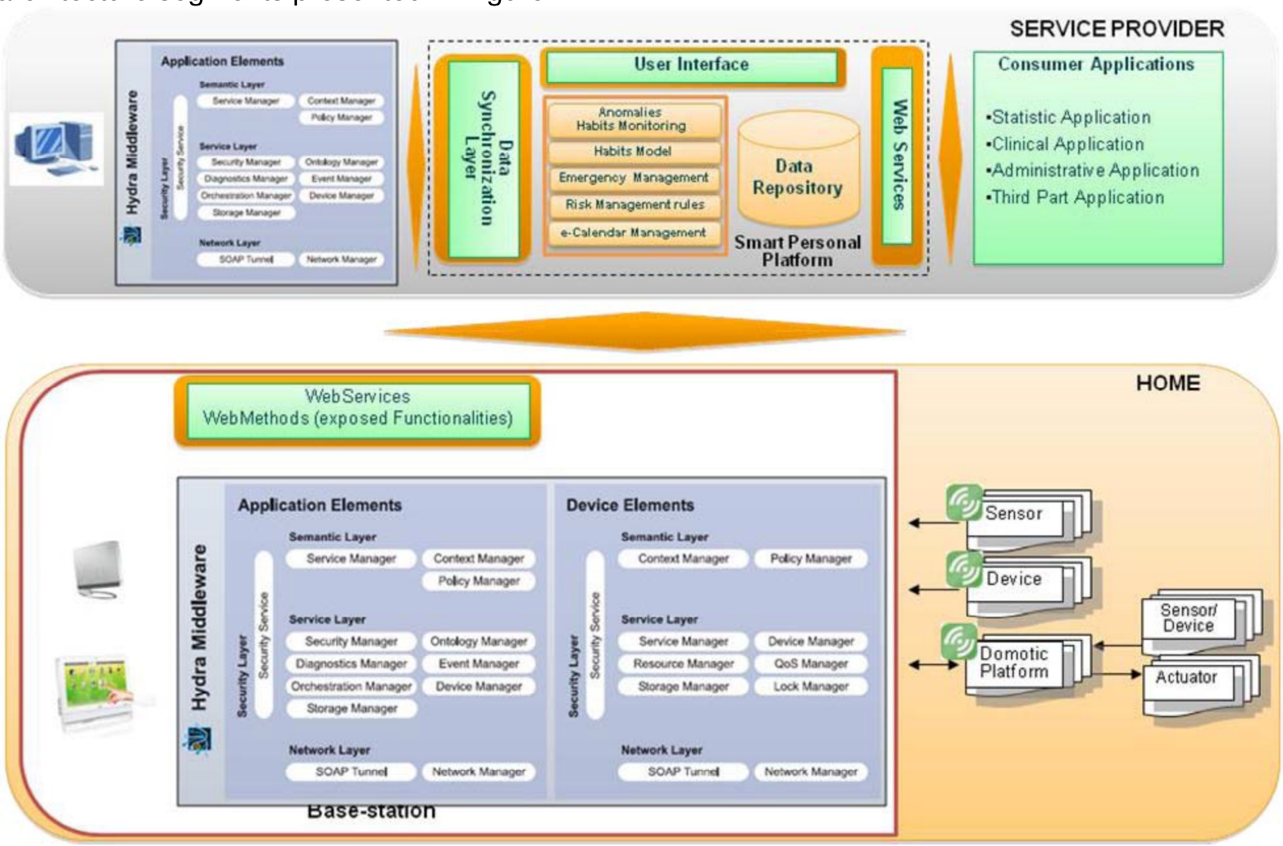

Figure 1 - Software Platform Architecture according to the inCASA DoW.

While Figure 1 refers to the original abstract view the consortium had on developing the inCASA platform the Figure 2 points at the current work of implementation for this deliverable, i.e. the core monitoring mechanisms. It mainly involves a two-sided Hydra Middleware implementation, one on client side (i.e. Patient Side Layer) and one on server side where the Middleware Layer forwards unmanaged as well as managed data to the Basic Services Layer, Intelligent Layer and Final Added Value Services Layer.

As a project, inCASA intentionally involves SMEs and academic institutes for their innovative approaches and resources of platform subsystems. Therefore, the Figure 2 includes notations on which SME or institute is owner of what subsystem. CNet contributes with the main part of the Core Monitoring Architecture and that is implicitly by the Hydra Layered Architecture, Hydra SOA approach and its middleware technology (read more about these in chapters 5.1, 5 and 6.2). To its conducive management of collected data there are the Telehealth (TID) and Telecare (SIG) partner subsystems. The Hydra Middleware exposes mediated data to next coming subsystems in the inCASA SPP module through Web Services. This eases interoperability between subsystems and opens up for a local scalability strived for within the project.

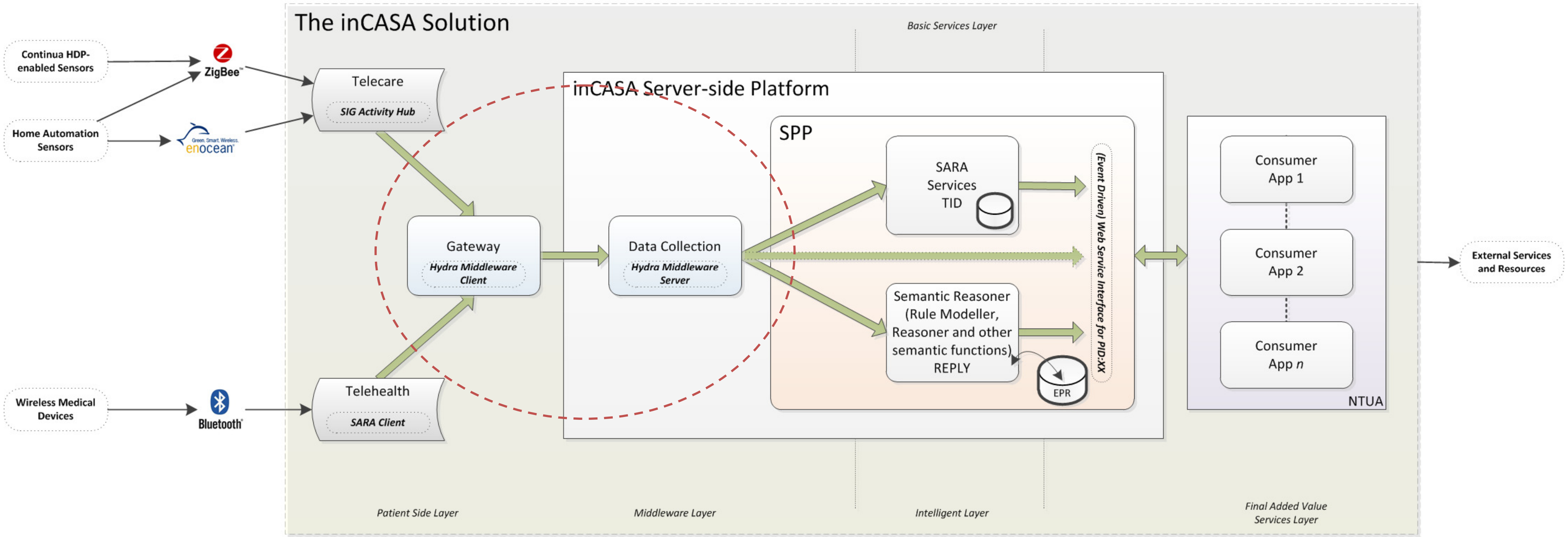

Figure 2 - Current draft Core Monitoring implementation where subsystem specific consortium partners involved are marked.

3 Project Development and Test Plans

Parts of the main objectives of WP4 are the creation of a development and test plan, testing of components functionality and usability and to ensure that these are performed by a good fashion monitored according to a quality management procedure. These project objectives shall be reached by realising and testing in specific pilots efficient integrated care systems that combine innovative technological platforms for ubiquitous communication, advanced healthcare monitoring and state of the art domotic systems. As such, WP4 does not only deal with the iterative implementation of the inCASA system parts but, as stated here, also with the technical testing of system functionality. The implementation and test planning process will take place in this work package with a development and test plan presented here.

The development plan will describe how the different software modules, subsystems and systems developed in the inCASA project will be integrated so that they can easily interoperate (see chapter 3.2). Subsequently, the test plan will become the basis for the testing process. The results presented in chapter 7 will by then have to be validated according to the set validation criteria. Further, any needs for adjustments of the project user requirements specifications and design specifications will be determined in the summary (chapter 8). The work here is divided into the implementation tasks presented in chapter 2.1. These will each try to focus on a specific component in the inCASA framework.

3.1 Work Package vs. Project Specific Plans

Generally, the inCASA work plan is an implementation of iterative methodology where the length of each cycle will be 6 months. This approach will ensure a tight cohesion and continuous communication between all the different phases of the process so as to minimize risks in the implementation of the project through the early and continuous monitoring of its results. For WP4, this mostly relates to the pilot execution performed in WP6 (after parallel implementation in WP5) and the feedback that WP6 gives back to WP3 (see Figure 3). During this process the pilots will carry out a wide validation of the integrated, innovative inCASA solution. The validation here involves use case activities and deployment plans. The data collected from the pilot monitoring, technical, objective and subjective data will be analysed and reported into a final evaluation of the applicability of inCASA solution, i.e. D6.6 Pilot Evaluation Report and inCASA platform validation and recommendation Report at month 30.

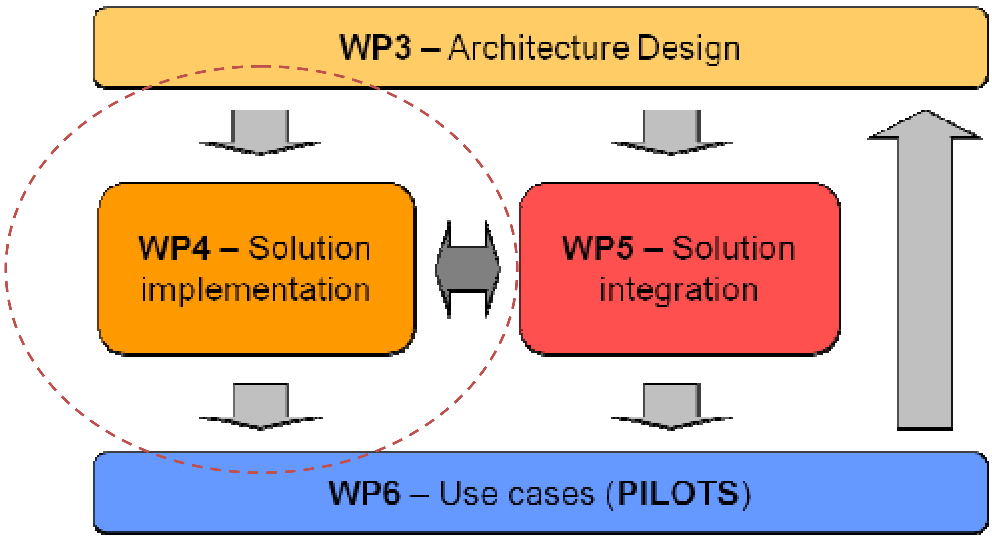

Figure 3 - Sectioned Pert diagram showing WP4 relevant WP dependency.

Testing the solution in a new environment (e.g. pre-pilots) will increase the customization that will be required to meet local requirements (e.g. pilots). Ultimately, the commercialisation and finalisation of the inCASA platform will commence immediately after the last pilot. It is anticipated that further pilot services (testing service concepts before deployment) will be implemented in addition to the five services already in operation within the first year of the project end. To its service, it gain knowledge and resources from each updated set of quality attributes and set of actions for subsystems and components available in D4.1, D4.2 and D4.3.

3.2 Quality attributes and set of actions for subsystems and components

The test plan for WP4 includes formalising a set of actions. These should involve any procedure, process, equipment, material, activity or system that will help consortium partners to understand whether the system performances do meet the required specifications and quality attributes. These set of actions are deliverable specific, i.e. they relate to the matter of topic (i.e. here the Core Monitoring System).

The Core Monitoring System comprises all communications that are expected by a structured network approach in inCASA. It affects the secondary communication, which allows the local monitoring of data and events to be transparent to the server side and tertiary communication between the gateway and the provider of the functionality (i.e. inCASA SPP and Web Service Interfaces).

Some quality attributes that can be reclaimed through D3.2 Reference Architecture Iteration 1 and matched against the cohesion of D4.1 are the following:

The Core Monitoring System should preferably have the following quality attributes;

1. a defined dedicated communication channel for timely alert generation,
2. asynchronous communication with provided reliability mechanisms instead of synchronous,
3. Continua [3] compliance according to IEEE 11073 Personal Health Device standards, and
4. Continua compliance according to IHE-PCD01 standards and WAN-IF.

The set of actions for subsystems and components are presented throughout this deliverable. They all contribute to fulfil the stated quality attributes above and are further anticipated to reveal separate subsystems' needs of counteractive actions to be set.

3.3 Deploying an iterative process for the plans

The actual execution of subsystem and component actions are performed by an iterative process throughout the WP4 deliverables. D4.1 starts off including those arguments posed in WP3 deliverables and especially the D3.2. The submission of this deliverable will consequently contain additional outcomes of the subsystem and component validation that may conduce to altered quality attributes to accomplish. It further leads to a set of actions to be executed in forthcoming WP4 deliverables. As such, the WP4 implementation declares an iterative process for the parted but still collective development and test plans, which will be separately and implicitly tested throughout the WP6 and WP5 as to generate reformed development plans.

4 inCASA Core Monitoring System Line Up and Features

The inCASA project will reuse the results of previous projects and existing organization resources and technologies as placeholders for WP4 deliverable specific implementation material. Particularly, the project will make use of efficient integrated care systems, combining innovative technological platforms for ubiquitous communication, most advanced sensing components useful in the healthcare domain and state of the art domotic systems.

The inCASA Core Monitoring System is as stated in chapter 2.2 mainly based on the Hydra Middleware connecting Telecare and Telehealth subsystems to basic and intelligent services. This chapter will focus on giving a heads up knowledge on each of these partners' technologies so that the implementation of the inCASA Core Monitoring System runs smoothly.

4.1 SIG Activity Hub

The activity hub [4] acts as a generic low-cost bidirectional gateway between HTTP-XML-traffic low-level short-range wireless connectivity. Thus, it interconnects the input from the wireless Telecare sensors to the Hydra middleware.

A radio interface is required for the dedicated WPAN protocol. The data is then encapsulated into IP-based traffic and sent to the hydra middleware. For the Internet communication, different physical interfaces are possible, including Ethernet (IEEE802.3), Wireless LAN (IEEE802.11), mobile communication (GPRS), or legacy Public Switched Telephone Network (PSTN) modems. TCP communication provides end-to-end reliability. During periods of missing Hydra connectivity, e.g. when the person is travelling or in case of communication failure, the SIG Activity Hub logs data and can for example to continue to post data from latest point in time. Travelling is supported as the SIG Activity Hub is equipped with a battery that keeps the functionality for more than a day before recharging. This feature mainly applies for body-worn sensors, the home automation sensors are foreseen in a fixed installation.

As the SIG activity hub interconnects two realms of the inCASA network, it has to provide two interfaces. One interface supports the communication with the different radio modules and is described in chapter 4.1.1. The second interface supports the external activity hub communication. It is described in chapter 6.2.4. Thus, the SIG activity can be applied not only in the Telecare, but also in the Telehealth pilot installations to connect devices that are not supported by the SARA platform.

4.1.1 Internal activity hub communication

The activity hub combines several radio modules and radio protocols. The internal communication uses a generic serial protocol, which unifies the communication from the activity hub's host controller to the various kinds of network interface modules.

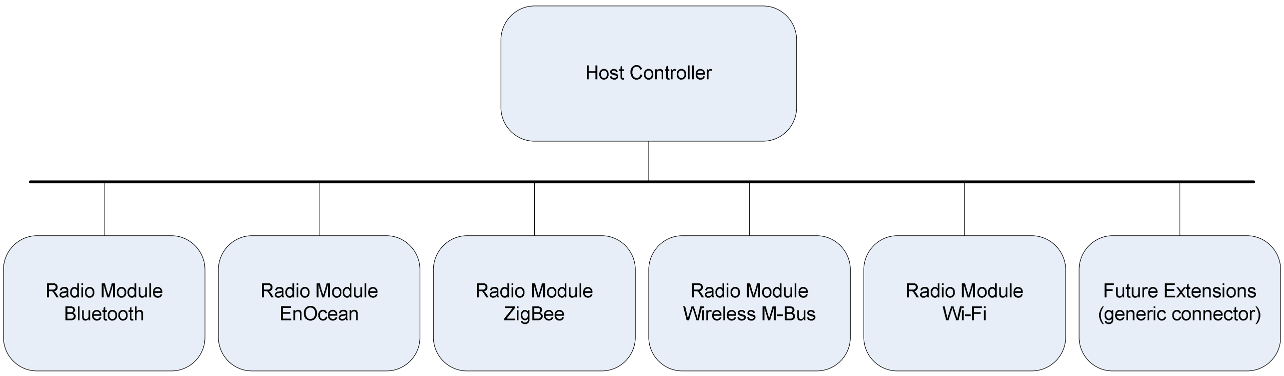

Figure 4 - Radio modules of the Activity Hub

4.1.2 Overall Frame Format

Each frame is split into a header, a data part and footer. The header starts with a Start of Frame delimiter (SFD, 0x5A) followed by 2 bytes length (MSB) which sums all bytes following the length field except the CRC. The type field describes the data following to the header. The footer contains a 16bit CRC (Cyclic Redundancy Check), which is calculated using the following polynomial:

$$x^{14} + x^{13} + x^{12} + x^{11} + x^{9} + x^{7} + x^{6} + x^{3} + 1$$

An example frame is shown in Figure 5.

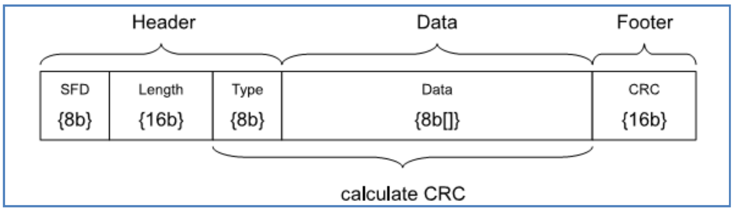

Figure 5 - Internal Activity Hub communication frame

The host controller supervises the connected modules using “Ping” commands that the modules have to answer in time. The other commands shown in Table 1 are used to send and receive data and to configure the devices.

Command	Description
00	Confirmation
01	RF_DATA
02	LOCAL_DATA
03	Request Rawdata
05	Set configuration
06	Get configuration
0A	Ping
0B	Power down on idle
0C	Set sleep mode
0D	Wake up
0F	Send binary
20	Status
30…8F	Reserved for Wireless M-Bus
90…AF	Reserved for ZigBee commands
FE	Manufacturer specific protocol
FF	Reset

Result Code	Explanation
00	OK
01	Application is busy
02	Action failed
03	No data
10	Internal error
11	Internal warning
12	Internal information
13	Unknown command
14	Too few parameters
15	Serial input buffer overflow
16	Command checksum invalid
20	Configuration type not supported
21	Mode not supported
22	Channel not supported
23	Device not supported
24	Encryption not supported

Table 1 - Internal Activity Hub commands and result codes

4.2 SARA Client

The SARA Client is the branch within the architecture that permits the inCASA solution to gather all information related to healthcare. It is a Health client that is able to read different incoming data rates and make the extrapolation of the information in order to serve useful information onto the platform. Communication through the platform (Hydra Client Side) is done using a conventional Internet broadband connection, i.e. cable, 3G, etc.

Final implementation of SARA client for inCASA has involved mainly the next decisions:

- Because we know what devices are going to be used during the pilots, parsing the datagrams coming from the sensors has been embedded into the client. Therefore, adding a new device would imply new programming.
- A layer of Web Services Clients has been implemented in order to send sensor data and alerts to the platform.
    - These WS Clients are integrated into the platform through the Hydra middleware.
    - These WS Clients also ultimately need to acquire a response by the SARA Server Services in order to initiate subsystem application services.

In order to publish the services ordered by the SARA Client, the SARA Services within the SPP module need to be able to exploit the Web Services while being integrated in the inCASA solution.

4.3 Actigraph

The Actigraph logs the patient's activity during the night. It transmits the sensor data using Bluetooth Low Energy technology to the SIG Activity Hub, where it is treated like any other sensor data, and forwarded to the Hydra platform. As the Actigraph is worn over night, a comfortable housing is crucial. The device is integrated in a wrist watch permitting valuable information about the patient's activity.

4.4 Building Blocks in Hydra

Initially the Hydra Middleware was presented in the DoW and deliverable D3.2 Reference Architecture Iteration 1. For the sake of relevance for this deliverable, a targeted description on those functionalities available through the Hydra Middleware will here be presented. On the other hand, all those issues related to the actual customization enabling various subsystems into the infrastructure of the inCASA platform by the use of Hydra will be introduced in chapter 5.1.

As reference to the programming abstraction and middleware functionalities provided by the collection of reusable core software components in Hydra, Figure 6 shows its elaborated assisting managers. These managers are enclosed by the physical communication layer and the application layer. Several network connection types (e.g. Bluetooth, ANT+, TCP.) are realized in the physical layer while the application layer contains any on top built applications. These applications can encapsulate various implementation models such as interfaces, business logic and configuration details. The physical and application layers are not part of Hydra but are instead seen as consumers of the middleware solution. As such, experienced (but also novice) programmers may use these software components and programming abstractions (i.e. managers) in order to build their own applications together with specifically defined network communication types without reducing the details of any underlying implementation approach.

A basic idea to keep in mind for Hydra developers is to differentiate between the physical devices and the application view of the device, in terms of so called Hydra Devices. Moreover, a Hydra Device is the software representation of a physical device or even an application. A Hydra Device is said to Hydra-enable a physical device. The software representation of physical devices can be implemented in two ways: 1) by a proxy running on a gateway device, or 2) by embedded Hydra managers on the actual device. For inCASA there will be a proxy on the client side dealing with both Telecare and Telehealth defined communication types. This should ensure a stable flexibility in the inCASA platform when switching between different Tele settings.

The inCASA developer will benefit from applications that are built by programming networked ambient intelligent devices. Regardless if the programmer wants to implement it as proxy or as embedded on the device, the method of choice will be transparent to the inCASA application developers, as they access all devices based on a pure service and event based programming model. The API of this programming model is manifested by the Hydra SDK, for application development (described in chapter 6.2).

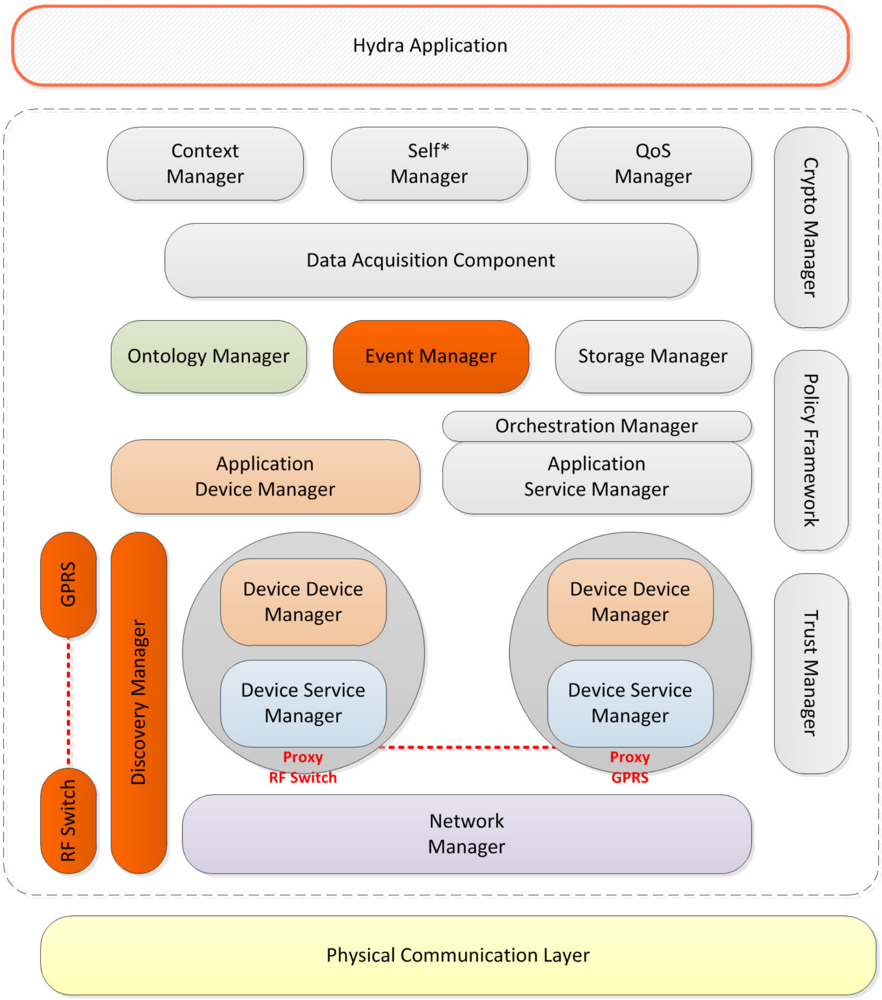

Figure 6 – Hydra Middleware layered architecture model.

If quickly returning to Figure 6 the layered architecture model presented reveals that the inCASA programmer may use the following relevant middleware components in Hydra for development or in the particular case of the inCASA project – Hydra-enabling and configuring both physical devices (i.e. Activity Hub) and software-based applications (i.e. SARA Client). The Hydra components relevant here are (*non-relevant are marked in grey but still available through Hydra*):

- The Network Manager which implements Web Services over JXTA as the P2P model for device-to-device communication.
- The bundled Device and Device Service Manager that implements a service interface for a physical device but will also handle several service requests and manage the responses.
- The Application Device and Application Service Manager, which provides programming interfaces and information for the different devices to the inCASA developers.
- The Discovery Manager, which automates and facilitates the discovery of devices in a inCASA network.
- The Event Manager, which provides a topic based publish-subscribe service available in Hydra.
- The Crypto, Trust and Policy Managers that will on the other hand take care of cryptographic operations, the evaluation of trust in different tokens including the enforcement of access control security policies.
- The Data Acquisition Component, which retrieves the data delivered by the sensors, and
- The Hydra Storage Architecture which realises the persistent storage of information in Hydra middleware.

5 Core Data Handling in Hydra

Core data handling for the Core Monitoring System can generally be recognised by the inCASA solution's centrally generalised and automated task management provider. It could also be as primed by the solution's work- or data flow affiliation. For this purpose the Hydra Middleware make the largest proportion of subsystems present in inCASA that vouch for these features.

Chapter 5 will deal with the underlying procedures of implementing the Core Monitoring System and data. It describes how the Hydra Middleware was customized to fit the inCASA platform solution and discusses how the WP4 partners may approach some developmental issues.

5.1 Hydra base station customization

Customizing the Hydra Middleware to fit the inCASA solution we are using a SOA-based architecture framework that combines web services and UPnP with P2P networking. It further deploys a technique called SOAP tunnelling (see more in chapter 6.2.1) making it possible to make web services calls between physical devices or software applications in two different networks.

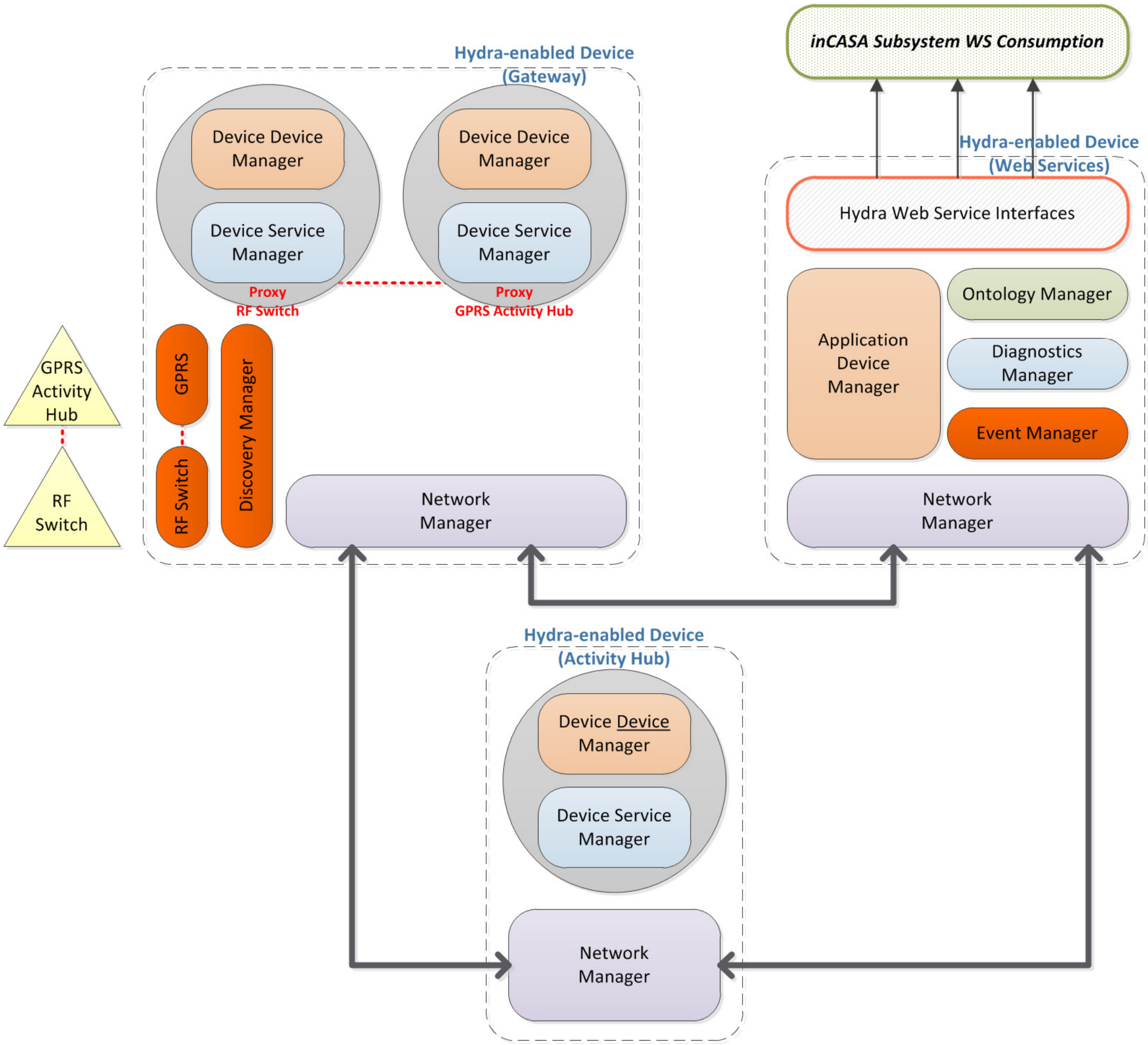

Figure 7 – Hydra Software components as solution for the inCASA Core Monitoring System.

Hydra takes support from its MDA approach by which design and execution of such applications is set to work by providing a set of models, transformations and component assemblies. The objectives of the MDA are to facilitate device programming through the Hydra SDK and Hydra enabling physical devices through the Hydra DDK.

Besides the SDK and DDK provided by Hydra it makes use of several inCASA relevant software components. These are seen as an integrative part of the Core Monitoring System presented in this deliverable and described here in relation to Figure 7.

First of all, the Discovery Manager implements the base class for all protocol specific discovery managers in Hydra. It is part of the Application Device Manager and keeps track of the devices it has discovered. As long as the devices are unresolved they keep being treated as Embedded devices of the Discovery Manager. A Discovery Manager runs locally on the inCASA gateway where it looks for remote devices such as the Activity Hub devices. The Discovery Manager has also direct access to the device objects it has created.

One of the key components in the Hydra middleware is the Device Ontology (managed by the Ontology Manager), where all meta-information and knowledge about devices and device types are stored. The purpose of the Application Ontology Manager is to provide an interface by using the Device Ontology. The Device Ontology also contains the other models in addition to devices such as device HW and SW capabilities, device services, device state-machines and device security capabilities.

The Device Service Manager implements a service interface for physical devices. Its main functions are to map services to physical device operations as well as to map (physical) device events to Hydra enabled events.

On the other hand, the Device Device Manager handles several service requests and manages the responses. It is a generic class that is sub-classed depending on device type (e.g. Bluetooth Device Manager, Basic Phone Device Manager, Basic Switch Device Manager). The main functions are to map requests to device services, generate responses, advertise Hydra device descriptions as well as to advertise device services.

The purpose of the Application Service Manager is to discover, create and execute Semantic (Web) Services on top of all the Hydra-enabled devices. The manager adds a semantic layer and complements the Application Device Manager with a service perspective which might map to several device functionalities.

To briefly describe the purpose of the Application Diagnostics Manager it can be said that it is to monitor the system conditions and state. This manager is responsible for error detection and logging of device events and thereby an important component in providing devices self describing properties of Hydra.

Finally, the Application Device Manager will manage all knowledge regarding devices or software applications that have been discovered and are active in the inCASA network. The Application Device Manager knows all about devices from a network perspective but will not handle the locations or context of the devices.

5.2 Hydra Continua Extensions

The Continua guidelines add support in Hydra to use predefined device specialization based on Continua ISO/IEEE Std. 11073. Using this standard allows interoperability between different components, system and subsystem to follow the same structure for different health systems.

Specializations that will be used in inCASA follows specialization ISO/IEEE Std. 11073-10471 Independent Living Activity Hub which has multiple support for different sensors, such as thermometer, humidity, door detector, etc. Each specialization describes information about the sensor being used (e.g. type, identification number, location, measurement, etc.).

More specialization can be implemented which is supported by Continua ISO/IEEE Std. 11073 for healthcare or homecare (e.g. blood pressure monitor, weighing scale monitor, glucose monitor etc.) and this supports a scalability point of view for the project.

5.3 Development issues

Being an integrative project inCASA involves several technical partners by several technical conditions. When developing the inCASA platform it is important to address some issues that may and should be elaborated into a common wisdom. This wisdom reduces flaws and redundancy in the integration work. These issues reasonable concern:

- While both the Hydra Middleware and the SPP (DoW) are based on SOA the inCASA platform intracommunication herein should be based on Web Services for granted interoperability.
- The Activity Hub currently posts its data through a GRPS or Ethernet connection to the inCASA gateway but communication protocol implementations such as ZigBee or WLAN need to be considered and estimated in a future solution of inCASA.
- The SARA Client uses Web Service to communicate with the SARA Services located at server side. This needs to be modelled and integrated in such way that the inCASA platform incorporates this subsystem’s technical prerequisites.
- Adopting parts of the Continua guidelines allows the consortium to evaluate the announced penetrating power that the alliance is said to have. But as the inCASA solution will incorporate different subsystems that all may not follow the Continua guidelines it is also important to provide flexibility and customized but fair development opportunities. Choosing a single set of IEEE specializations (i.e. 11073-10471) should further enable equal potentials in extending the inCASA platform to manage other specializations as well. This kind of support should preferably be executed near the client side of a solution so that architectural guidelines set by Continua are followed.

6 Implementation Build Up and Resources

One of the project's logical framework objectives is to validate the technical and operational implementation of a decision system, based on the data-collection framework developed in this deliverable. The aim is to further use behavioural models to generate alerts and escalate to preventive actions presented in D4.3. Also, the system architectures proposed and refined throughout the WP4 deliverables together with the expertise within the consortium, the inCASA project intend to define a "pilot implementation methodology" that will be shared between all the pilot site stakeholders in order to try defining a common European framework.

The stepwise approach for this is the outcome of WP3. This had a direct effect on the implementation in WP4 Solution Implementation by providing a high-level mapping of the key concepts, and the functional and non-functional requirements formulated in WP2. D4.1 is the first deliverable to contribute to this goal and this chapter describes how this has been performed.

6.1 SVN

According to the development plans an SVN should be set up. Such SVN has been accessible since the May 2011 for consortium partners after requesting individual accounts. The address is https://svn.cnet.se:8443/svn/InCasa/ and is hosted in the premises of CNet (Stockholm, Sweden). The SVN is used to share development codes and issues.

6.2 Hydra enabling and subsystem configurations

Task 4.1 deals with the Hydra customization on the base station and its configuration to successfully collect the data from the monitoring system (Telehealth and Telecare). It also works on the customization of the Hydra Middleware on the remote healthcare provider platform in form of publishing available resources of communication. As stated in chapter 4, each of the subsystems enabled in Hydra need their own specific configuration solution. For the SARA client and the SARA Services it involves providing a transparent way to forward the Web Service calls in between. For the Activity Hub the case would be to solve how post data is sent from the hub to the gateway in a reliable and resource saving manner.

6.2.1 SOAP Tunnelling over P2P Networks

The Hydra Network Manager is responsible for creating a P2P network and provides addressing schemes as well as the communication facilities. The Hydra intercommunication is done using SOAP messages that are tunnelled in the P2P network.

The Hydra Event Manager is deployed in an InCASA architecture as a service in close cooperation with other components (see Figure 7), among them the Network Manager, providing publish/subscribe functionality, i.e., the ability for publishers to send a notice to multiple subscribers while being decoupled from them (in terms of, e.g., not holding direct references to subscribers). The following sections will detail the Hydra network architecture including the addressing scheme as well as the event architecture.

6.2.2 SOAP Tunnelling Approach for inCASA

As the Hydra, and InCASA, architecture is service-oriented, where web-services (WS) is the technology used to implement it, the communication between applications running in different Hydra-enabled devices will also be based on SOAP messages. Usually, SOAP messages are forwarded through TCP connections to the destination. The destination address corresponds to the endpoint contained in the message.

Traditional WS architectures are based on client-server architectures, where the server is an always-on end system with a well known endpoint address, which should be known by clients beforehand (using either service descriptors or UDDI registries). The SOAP tunnelling approach proposes a way to replace this client-server architecture for a distributed one, using the Network Manager P2P platform. In this architecture, all the peers will act as clients and servers at the same time. Figure 8 shows an example of a client-server based architecture and the distributed approach. Furthermore, actual WS communications require direct connection between the client and the server, making it impossible to consume services across networks.

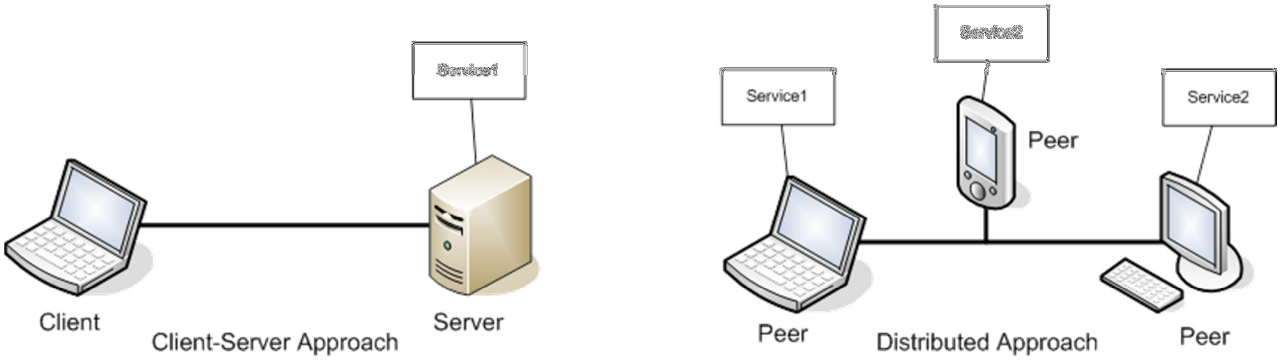

Figure 8 - Client-Server vs. Peer-peer

Moreover, in the Hydra middleware, devices are presented as UPnP devices by the Device Manager. But UPnP discovery information is usually restricted to Local Area Networks (LAN). Using the SOAP tunnelling the Device Manager will be able to exchange the UPnP information between different Discovery Managers in the Hydra Network manager infrastructure. Thus other Device Managers will be able to control UPnP devices located in remote networks using the SOAP technique presented in this section.

Therefore the main objective of the SOAP tunnelling approach for inCASA is to enable SOAP messages exchange across different networks, making it possible to consume services provided by SARA Client such as different Hydra Enabled devices/applications or controlling UPnP devices located in different LAN**.** The Figure 9 below shows a SARA example of the application of SOAP tunnelling. Due to the Network Manager and the SOAP tunnelling approach, HED2 (SARA Services) is able to discover UPnP devices located at home network (weight scale and thermometer) and consume the web services offered by the application running on HED1 (SARA Client).

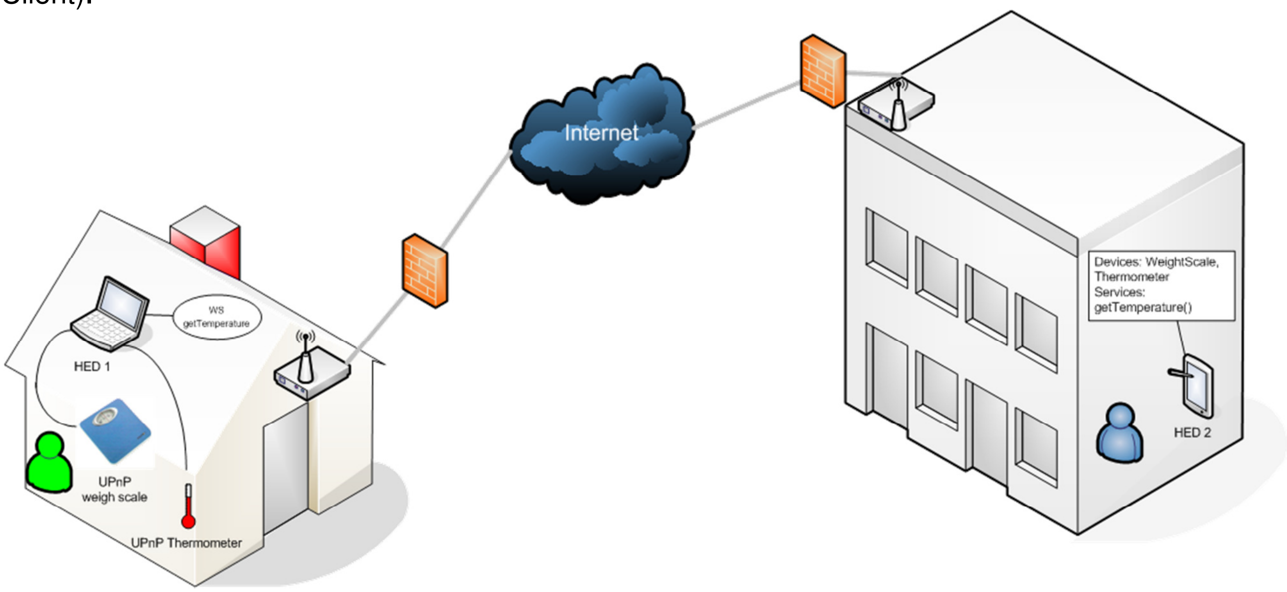

Figure 9 - SOAP tunnelling example.

6.2.3 SOAP Tunnelling for the SARA intercommunication

Thus, the Network Manager enables a way to communicate different Hydra Enabled devices transparently, building an overlay network in which resources (devices, services and contents) are identified by an Hydra-ID (HID). The main objective of the SOAP tunnelling communication proposed for Hydra is to provide SOAP messages exchange using the P2P transport schemes provided by the Network Manager.

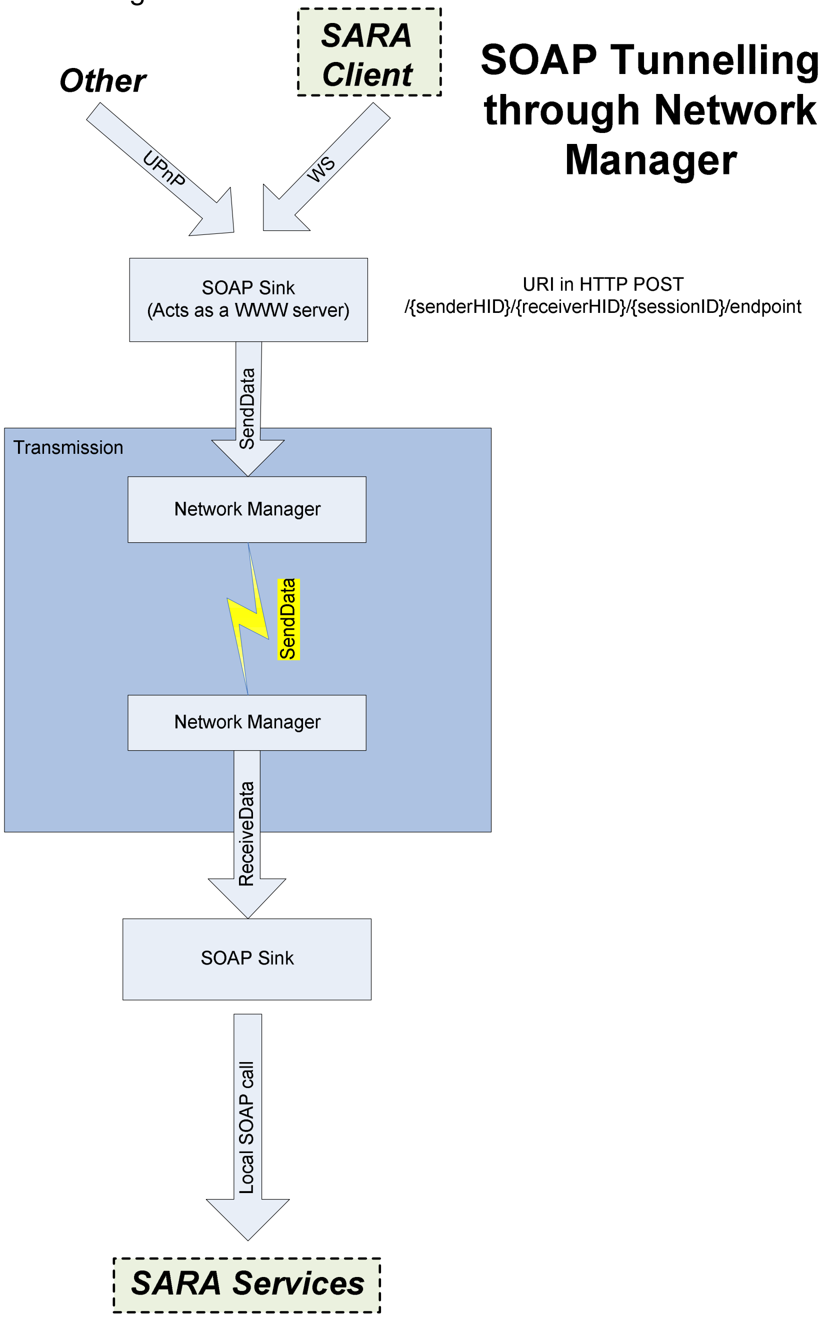

Figure 10 - SOAP tunnel solution for the SARA subsystems.

In order to use P2P networking/addressing/transport schemes together with web services and UPnP we need some kind of virtualisation of endpoints that allow us to use P2P networking. For this reason, all endpoints for UPnP and web service calls are grounded in a SOAP sink (ideally locally) which repackages the SOAP message containing the SARA Client WS and routes it through the Network Manager, as shown in Figure 10. The Network Manager is responsible of the message transmission and finally calls the SOAP sink that performs a local SOAP call to the intended SOAP endpoint.

The P2P networking with the SOAP tunnelling technique will facilitate event management, as well as SOA in general, for the InCASA architecture. Certain adaptation will however be needed, and these have been addressed in chapters 5.3 and 6.3.

6.2.4 Collecting the Activity Hub Post Data

The connectivity of the Activity Hubs uses local installation (LAN) in many cases. But especially for the ATC pilot, GPRS connectivity is crucial. As many Internet Service Providers only permit to access web content on mobile devices, and not the provision of web services, HTTP POST is used to transfer the data. Figure 11 shows an example packet sent to the backend.

```
POST /hub.php HTTP/1.1Host: 192.168.3.22User-Agent:
stzedn - emBetter http client 3.1.0Content-Type:
application/x-www-form-urlencodedContent-length:
31Connection: keep-alive0=174&1=0052C2FFFEAC9E12&2=12B7
HTTP/1.1 200 OK
```

Figure 11 - Data transmission from the Activity Hub

The parameters after "`keep-alive`" have following meaning while sensor values are interpreted as shown in Table 2:

- Parameter 0: timestamp when receiving the radio packet in the wireless sensor network,
- Parameter 1: unique address of the sensor, common for all radio protocols,
- Parameter 2: sensor value.

Sensor	Technology	Size	Min	Max	Comment
Temperature	ZigBee	16 bit	0x0000 →0.00°C	0xFFFF →655.35°C	0.01°C per step
Humidity	ZigBee	16 bit	0x0500 →5.00%	0x251C →95.00%	0.01% per step
Motion	ZigBee	8 bit	0x00 → no activity	0x01 → activity	Current configuration has 30s pause after each activation
Tampering	ZigBee	8 bit	0x01 → tampering alert		Contact when opening housing, alternative use
Flood	ZigBee	8 bit	0x30 → no alert, 0x31 → flood alert,		Can also be configured to cause alert when dry
Door	ZigBee	8 bit	0x30 → door closed, 0x31 → door open,		
Chair sensor	EnOcean	8 bit	0x00 → not occupied	0x10 → occupied	Binary information
Light switch	EnOcean	8 bit	0x00 → Button released	0x10 → Button pressed	Binary information, multiple information possible if necessary
Presence / ID	Wireless M-Bus	8 bit	0x00 → Person absent	0x01 → Person present	ID sensor in range is used.
Distance / Attenuation	Wireless M-Bus	8 bit	0x00 → weak signal	0xFF → strong signal	ID sensor RSSI is used.
Fall sensor	Wireless M-Bus	8 bit	0x01 → fall alert		
Activity sensor	Wireless M-Bus	8 bit	0x00 → no activity	0xFF → strong activity	Can be used to check if the ID sensor is worn

Table 2 - Sensor value interpretation

At the moment of writing, the Activity Hub posts data through a GRPS connection to the inCASA Gateway located somewhere in the home of the elderly. Future development of the inCASA platform will also support different communication types between the hub and client, e.g. wired by USB and ZigBee.

Collecting data that is posted by the Activity Hub to the inCASA Gateway is performed through an open web server that logs each data post. As SIG has the aim to implement the IEEE11073-10471 specialisation it is here important to enable such data management. In the first phase the Activity Hub posts its raw data to the client where some software needs to be capable interpreting it according its future IEEE specialisation. The Hydra Middleware has a Continua Manager component that can be used for this purpose. What this manager does is to retrieve the incoming message from its log and map it to the IEEE specialisation reference. By doing so it knows what its Domain Information Model (DIM) looks like and hereby may interpret the content of its message. The Continua Manager available in Hydra then fills in the empty information spaces (e.g. report header or other segments) for the Activity Hub before it lets a HL7 parser transform it to a ORU-R01 message (plain unsolicited observation results transmission). The change to the IEEE11073-10471 format will be performed pilot by pilot, and as each activity hub has its individual settings for data forwarding, the remaining pilots are not concerned when transferring the format for a specific pilot installation.

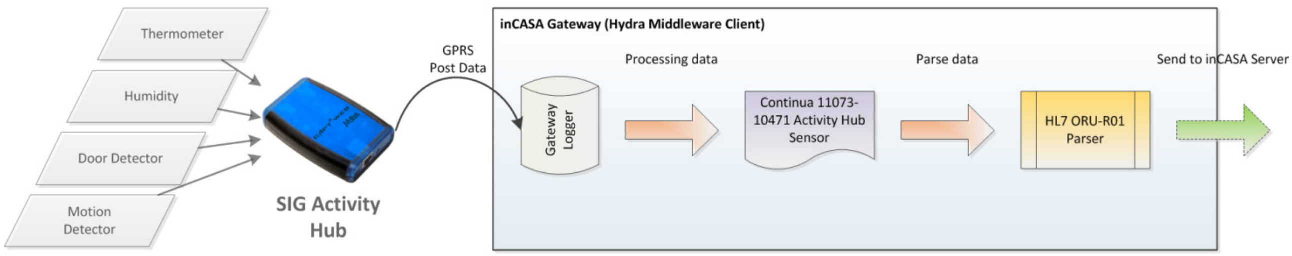

Figure 12 – 11073-10471 specialization handling and message format parsing in Hydra.

The ORU-R01 message is then sent to the inCASA Server side (i.e. Data Collection module) for further processing or as it is directly to recipient subsystem (e.g. SPP). At client side it is possible to convert this message to XML in order to ease data extraction hereafter. This allows the data to be interpreted on several layers in the inCASA platform. Figure 12 shows how the inCASA platform deals with the Activity Hub IEEE specialisation. Each sensor type sends its data to the Activity Hub which in turn posts the data to the inCASA Gateway. The inCASA Gateway logs this data where after Hydra's Continua Manager processes it, maps to a IEEE specialisation definitions library and parses the correct specialisation type to a HL7 ORU-R01 message which is delivered to eligible recipients post client side.

6.3 Composite Web Services at inCASA Server Side

Looking back to Figure 2 the Data Collection component at the inCASA server side will publish some type of composite services tunnelled by the inCASA client side. These services will be customized to fit each of the recipient subsystems in the SPP module (including proprietary service endpoints).

These subsystems form the main technologies of the inCASA solution. By an abstract definition these can be described as the Home Sensor Network (HSN) where the activity hub plays a main role and Human Monitoring Sensors (HMS) are governed by the SARA Client. These are connected to the Base Station (i.e. Hydra Middleware) which collects and propagates data to the SPP.

As stated in chapter 5.3 the SPP platform and Hydra are based on SOA. This should out of evolutionary implementation aspects be seen as guideline in the development of the inCASA solution. The SPP can store and manage single user social and health care data, with embedded elaboration capabilities in order to send messages to other applications. For this, it is arbitrary that Web Services from the Data Collection component would be the most suitable for this intercommunication.

Meanwhile, the SARA Client and Service parts require transparent communication of their Web Service calls. For this it is most suitable to provide this communication by the local SOAP call where native SARA Web Service WSDL manipulation may be conveyed. The Hydra SOAP sink consequently acts as indirect Web Service redistribution (see Figure 10).

As an alternative, the Hydra can allow for Web Service access to its Device Application Catalogue up to application development level so that inCASA developers may benefit from its AmI Web Services further establishing additional context inputs in the inCASA environments.

This composite Web Service provision is closely linked to the very first initiations of task 4.5 that deals with implementation of the operating central in terms of remote healthcare provider platform (SPP with the Semantic Reasoner) services finalization and task 4.6 that implements SARA as a telemedicine platform. In sum, the Data Collection component implemented with the Hydra Middleware at server side will become obtainable to adjacent subsystems in the inCASA SPP by a SOA-based Web Service communication method.

7 Test Notations

For this chapter it is important to respect that testing a solution in the real context of any application should provide a more reliable evaluation. The complexity of the real context will on the other hand stress the requirements based on some descriptive scenarios. The field trial in inCASA will highlight possible priorities and needs that would have an impact on the final inCASA service model.

By doing so, the project objectives will be reached by realising and testing in specific pilots efficient integrated care systems that combine innovative technological platforms for ubiquitous communication, advanced healthcare monitoring and state of the art domotic systems. This deliverable just plays a small but still important role in achieving this.

In chapters 1.2 and 3 we stated that the test plan will be the basis for the testing processes formalised in this deliverable. We also said that these will compound a set of actions including any procedure, process, equipment, material, activity or system that will help us to understand whether the system performances meet the required specifications and quality attributes. These results are presented in this chapter and later validated according to the validation criteria and needs for adjustments of the project user requirements specifications and design specifications in chapter 8.

7.1 Aspects of Data Flow

It is rather clear that the data flow in the inCASA platform is versatile and requires stable control of requested services. The flow of data is by parallel bidirectional communications forcing the Core Monitoring System to manage the combined complexity of service calls and sensor value forwarding. The Hydra Middleware uses SOAP tunnelling per sensor or application for any individual request between inCASA client and server side. This is estimated to provide sufficient underlying mechanisms to cover this issue as each SOAP tunnel is impenetrable for concurrent processes. Additionally, an upper Web Service interface layer can superintend each separate subsystem’s functionalities and hence ensure inCASA data flow distinction.

7.1.1 Tested Devices and Applications

This deliverable contributes to the Pilot Scheme Set Up phase 2a and 3. As the inCASA system is designed for monitoring and support of the elderly and will reuse existing infrastructure the inCASA solution needs to exploit existing wearable monitoring system and obtain other hardware that will be adapted for the specific needs in the pilots. So has been done, the testing of components throughout this deliverable show on their initial functionality and usability.

The tested devices and applications are:

- ***SARA Client*** – Provides Web Service addresses and WSDL for attachment in the Hydra SOAP tunnelling for later depletion in the SARA Service subsystem sited within the inCASA SPP. Testing show that this decentralised approach enables native exchange of structured information (i.e. WSDL and methods) to be returned through the Hydra Middleware and presented in a standardised machine-parseable format to the SARA Service subsystem.
- ***Activity Hub*** - Posts data to the inCASA client log (i.e. a SQL database) and this is done without any remarkable complications. The postdata visible in Figure 13 shows us certain sensor type connected to the Activity Hub together with either its state or certain value. Testing the Activity Hub logging is smoothly to be transferred from receiving emulator sensor data to real-time Activity Hub sensor data.

SQLQuery1.sql - bu....master (SA (61))*

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [id]
      ,[postdata]
      ,[URI]
      ,[created]
      ,[headers]
      ,[originalip]
      ,[method]
  FROM [ActivityHub].[dbo].[HttpCalls]
  ORDER BY [created] DESC
```

Results | Messages

	id	postdata	URI	created	headers	originalip	method
8	500	0=1305&1=00000050C2AC9E0B&2=30	/peeter.html	2011-05-30 10:06:57.700	Connection:close, Content-Length:30, Content-T...	217.86.190.116:1514	POST
9	499	0=1304&1=00000050C2AC9E0C&2=0B34	/peeter.html	2011-05-30 10:06:42.073	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1513	POST
10	498	0=1312&1=00000050C2AC9E0C&2=0984	/peeter.html	2011-05-30 09:54:13.650	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1542	POST
11	497	0=1312&1=00000050C2AC9E0C&2=0983	/peeter.html	2011-05-30 09:51:13.350	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1541	POST
12	496	0=1305&1=00000050C2AC9E0C&2=0988	/peeter.html	2011-05-30 09:48:17.497	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1540	POST
13	495	0=1310&1=00000050C2AC9E0C&2=098D	/peeter.html	2011-05-30 09:45:21.647	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1539	POST
14	494	0=1311&1=00000050C2AC9E0C&2=0986	/peeter.html	2011-05-30 09:42:25.800	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1538	POST
15	493	0=1307&1=00000050C2AC9E0C&2=0988	/peeter.html	2011-05-30 09:39:29.973	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1033	POST
16	492	0=1314&1=00000050C2AC9E0C&2=0984	/peeter.html	2011-05-30 09:36:34.127	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1536	POST
17	491	0=1307&1=00000050C2AC9E0C&2=0991	/peeter.html	2011-05-30 09:33:38.267	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1535	POST
18	490	0=1901&1=00000050C2AC9E0B&2=35	/peeter.html	2011-05-30 09:31:27.460	Connection:close, Content-Length:30, Content-T...	217.86.190.116:1534	POST
19	489	0=1325&1=00000050C2AC9E0B&2=34	/peeter.html	2011-05-30 09:31:25.540	Connection:close, Content-Length:30, Content-T...	217.86.190.116:1533	POST
20	488	0=1306&1=00000050C2AC9E0C&2=098F	/peeter.html	2011-05-30 09:30:42.407	Connection:close, Content-Length:32, Content-T...	217.86.190.116:1532	POST

Figure 13 – SQL query of the Activity Hub sensor data.

- ***Miscellaneous*** - Continua certified monitors and proprietary sensor devices that are directly coupled and enabled as Hydra devices. These may be used as backup if similar services are not available through the SARA Client or the Activity Hub. This is only for emergency solutions and not favoured as announced inCASA solution.

7.1.2 Communication and Formats

The GPRS (General Packet Radio Service) is for this deliverable a suitable solution for communicating the Activity Hub. But there are some factors that play a role for future implementations of inCASA:

- It bases its mobile data service on the 2G and 3G cellular communications. Usually this means that usage is charged based on data volume which could imply difficulties in deploying it to inCASA users.
- Its Quality of Service depends on a concurrent service usage. Many users at once might then imply that variable throughput and latency for the data sent.
- In larger installations, use of Ethernet connectivity is recommended to reduce cost. In the current Activity Hub implementation, both, Ethernet and GPRS can be enabled at the same time leading to GPRS as a fallback solution in case of LAN failure or to support travelling.

Web Services are generally seen as a communications method between two networked devices but when it comes to SOA concept implementations Web Services, instead of being an operation, corresponds to a communication message accomplishing the principles of any service-oriented system architecture design. As the inCASA platform is built up by different subsystems that use expedient communications solutions its strived service-orientation will thereby also satisfy the need of loose coupling of services. Deploying Web Services as a favoured communication inside the inCASA platform strengthens and gives reliability for the intercommunication between subsystems and components but will also require new additional ones connecting to the platform to at first hand comply with the Web Service standards.

HL7 and the message format ORU-R01 is part of the Continua guidelines and making this work enlightens the possibilities such format gives. It is a universal approach by Continua and proving that the inCASA platform is able to execute and run such standards by a recommended design it also shows the expandability the platform has, i.e. that developers easily can understand the structure of the platform and extend its functionalities to deal with additional IEEE specialisations and/or HL7 message types.

7.2 Integration Risk Analysis

The analysis of potential integration risks is practically based on all the pros described in the chapters so far.

We can see that the cross functionalities provided by the different subsystems and components also may give integration complications as different technical approaches may not always be compatible (even if this more concerns future platform extensions).

When giving the basic and intelligence layers of the inCASA solution architecture Web Service interfaces as means to publish underlying subsystem or component functionalities also imposes a risk that actual intelligence moves higher up in the inCASA architecture solution to the application layer (i.e. Consumer Application level). With this in mind careful advises are given to expose all possible intelligence by mechanisms as close as possible to the subsystems or components.

7.3 Re-designed Specification

A re-designed solution implementation specification is regarded as not necessary.

8 Summary and Validation of Subsystems and Components in inCASA

This deliverable's intention has been to describe the implementation of a Core Monitoring System based on a structured network approach and a SOA model. A number of significant driving requirements (in this document seen as quality attributes) from WP3 has been considered and established with respect to the contents of this document. The main quality attributes have been realised through single subsystem features. These are:

- an asynchronous communication with provided reliability mechanisms instead of synchronous including a defined dedicated communication channel for timely alert generation through Hydra event management and SOAP tunnelling.
- Continua compliance according to IEEE 11073 Personal Health Device standards, and Continua compliance according to IHE-PCD01 standards and WAN-IF through Hydra Continua Management and HL7 handling by REPLY's Semantic Reasoner and SPP.

The referred work tasks for WP4 are considered to coordinate with the content of this deliverable. D4.1 has managed to incorporate Task 4.1 and Task 4.2 where data is automatically collected and stored by the Hydra Middleware as well as at the Activity Hub level. Task 4.5 provides for the Core Monitoring System the basis for remote healthcare provider platform service exploitation throughout the inCASA solution on both ends of the inCASA solution. Finally, Task 4.6 by using the SARA client the Core Monitoring System is be able to communicate and handle various input components and services (i.e. subsystems integrating with the Hydra Middleware, alias inCASA Client side).

The test conducted throughout the Core Monitoring System implementation will hereafter feed knowledge and implemented technology for the Core Monitoring System to WP5 and WP6. This will give expected return in terms of refined or re-designed implementation specification.

9 Glossary

WP	Work Package
ICT	Information Communication Technologies
SPP	Smart Personal Platform
HIS	Healthcare Information System
WS	Web Service
P2P	Peer-To-Peer
LAN	Local Area Network
HID	Hydra ID
HSN	Home Sensor Network
HMS	Human Monitoring Sensors
DIM	Domain Information Model
WLAN	Wireless Local Area Network
SME	Small Medium Enterprise
GPRS	General Packet Radio Service
HL7	Health Level Seven
IHE	Integrating the Healthcare Enterprise

10 References

[1] M. Eisenhauer, P. Rosengren, and P.Antolin, "A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems," presented at the Proceedings of the 6:th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009), 2009.

[2] M. Ahlsen, P. Kool, P. Kostelnik, and P. Rosengren, "D6.10 SOA Middleware components for the IDE prototype," IST project 2005-034891, Hydra Project Deliverable D6.10, 2010.

[3] Continua, "Continua Design Guidelines," Continua Health Alliance, 2010.

[4] A. Sikora, "An Embedded Web2.0 Monitoring and Gateway Platform for Spatially Distributed Wireless Networks," presented at the IEEE 5th International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2009), Rende, Cosenza, Italy., 2009.