Contract ITEA 2 - 07008

MIDAS

Multimodal Interfaces for Disabled and Ageing Society

Project number: ITEA 2 - 07008

ITEA Roadmap application domains:
Major: Home____
Minor: Nomadic______________________________________

ITEA Roadmap technology categories:
Major: Human-Computer Interface_________________________
Minor: Data and Content Representation____________________

WP N°: 7

Deliverable D7.6 b Use of Standard

Investigations on Continua Health Alliance and MIDAS DPWS middleware

Due date of deliverable: 30/09/2011
Actual submission date: 15/08/2011

Start date of project: 01/10/2008 Duration: 36 Months

Project coordinator name: Laure Chotard

Organisation name of lead contractor for this WP: Orange Labs
Task leader: CEA
Contributors: Orange (David Excoffier, Laure Chotard, Olivier Graille), TAS (Xavier Ladjointe), IIMS (Ismael Fuentes), CEA (Mariette Soury, Christophe Leroux), Geomobile (David Doise)

Editor: Christophe Leroux **Revision:** 01

TABLE OF CONTENTS

1. Introduction

1.1. Purpose of this document

It is expected that seniors can live far longer than usual in their familiar home environment with the support of intelligent assistive technologies. Intelligent Assistance technology as increasingly wireless sensor networks, which collect information about the spatial environment and the vital functions of the residents, makes possible to detect unusual or dangerous events for the senior citizen and if necessary alert emergency services.

The continuous development of new and always smarter sensors and actuators by always more different manufacturers creates a need to platform independent interoperability for these embedded devices. Wireless embedded sensor networks and mobile devices usually have limited resources such as energy, computing power and communication range.

Continua Health Alliance is working to make simple and easy networking devices (from sensors to IT) possible in health domain. While UPnP, DLNA (for multimedia) and related technologies are established in networked home and small office environments, DPWS tends to be used in automation industries at device level, and it is spreading into higher levels such as enterprises integration. This document investigates the possibility of using such a technology as a generic middleware to improve work and features proposed by Continua Health Alliance.

1.2. Change History

Date	Author	Update description	Doc. version
10/06/2010	David Excoffier (Author) Olivier Graille (Reviewer)	First public release	01
5/5/2011	Mariette Soury (CEA)	Sections on Communication protocol and web services, digital video and teleassistance	02
10/5/2011	Priscilla Vandenbrouck	Reshaping of the document and illustrations in MIDAS	02
11/5/2011	Xavier Ladjointe	Contribution on scenario 6 Compliance with Continua for Health monitoring	02
5/2011	Internal Review		
15/08/2011	Final Review		

2. Context management

In the first year of the project the analysis conducted during the state of the art (D1.4) lead to recommend the use of the Context Management System (CMS) developed during the FP6 AMIGO project. The main reasons were:

- Its availability
- The support it could provide to most of the functional requirements identified in MIDAS

AMIGO CMS was developed in C for .NET and in Java on OSGi. The idea of using AMIGO CMS was abandoned after a while since the.NET and OSGi CMS of Amigo could not interoperate. That is if the CMS was developed in OSGi, a .NET client could not access the CMS and reversely. A solution to this issue would have been to use a standard like DPWS to access the CMS but this idea was not followed in the project.

3. Knowledge representation

D1.4 on technological survey mentions widely the extensive use of OWL to describe ontologies in knowledge representation. With the help of “Protégé”, this language promoted by the W3C tends to become a standard for knowledge representation. The OWL language was at one stage chosen for data representation for all MIDAS project using the frame developed during AMIGO project. Lead by Telefonica, this action was finally left over after the departure of the Spanish consortium.

4. DEVICE PROFILE FOR WEB SERVICES (DPWS)

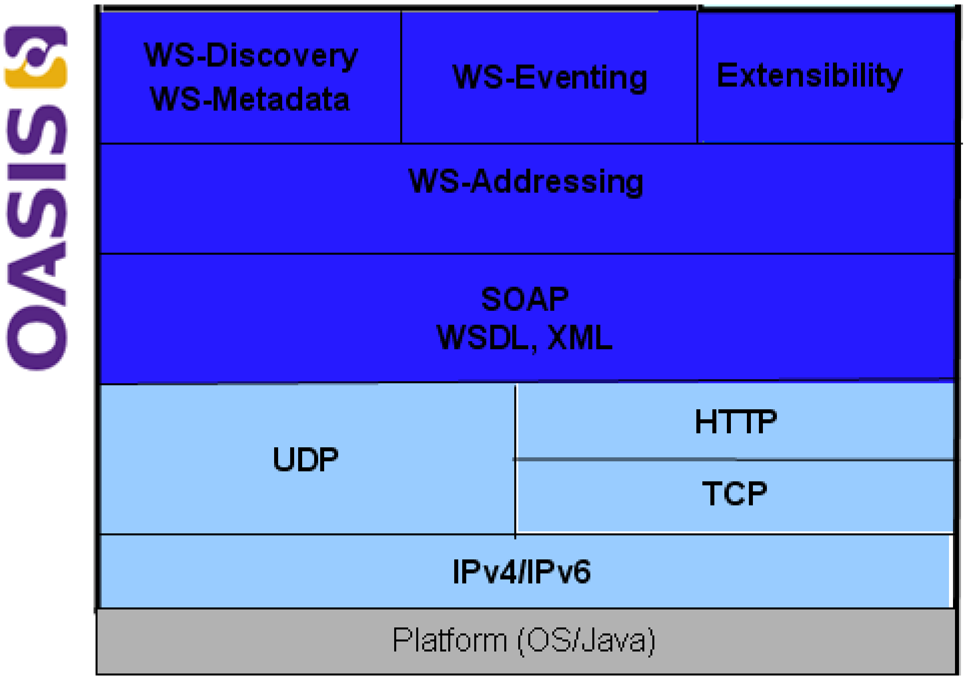
Service-oriented Architectures (SOA) are often used to improve flexibility and reusability of components in complex distributed applications, and achieved by creating functional blocks as independent and loose-coupled services. Web Services are widely used for implementing service-oriented architectures.

DPWS [1](“Device Profile for Web Services”) specification is developed by several major actors to enable Web Services capabilities on resource constrained devices. DPWS is currently widely used at device level in several fields of interest (automation industries, industrial devices monitoring, but also in health industries).

4.1. Main features of DPWS

DPWS is essentially a subset of Web Services specifications that a device must implement in order to ensure Web Service capability and compatibility while imposing only minimum requirements upon the device itself. DPWS was first introduced by Microsoft in 2004 and is now being developed jointly by individuals across the IT industry. The current version of DPWS is 1.1, which has been standardized and approved by OASIS in July 2009. This specification allows a full WS-I Basic Profile 1.1 interoperability.

[1] http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01



The subset of Web Services that DPWS defines (the dark blue parts in the previous schema) enables devices to communicate with each other over the network, dynamically discover other devices and subscribe to events.

The DPWS specification can also be easily composed with and extended by other specifications and technologies (the “Extensibility” blue part).

4.1.1. Basics

The main purpose of DPWS is to bring web services to small embedded devices. Devices implementing this technology can communicate with other DPWS enabled devices and applications using a unified and standardized protocol: No need of implementing a proprietary communication protocol thus enabling interoperability at minimum cost.

DPWS specification prescribes how to use the Web services features on devices to enable some networking functions:

- Dynamically discovering a Web service.
- Sending (secure) messages to and from a Web service.
- Describing a Web service.
- Interacting with a Web service on device by using Web Services Description Language (WSDL).
- Subscribing to, and receiving events from, a Web service.

Definitions
A *Service* can send and receive messages on a certain endpoint (usually an address).
A *Device* is a distinguished Service responsible for representation of the whole device and discovery and it hosts Hosted Services.
A *Hosted Service* provides desired functionality and each Device can host several Hosted Services.
Devices are communicating with each other by sending *messages*. Their structures are based on XML documents that contain a SOAP (1.2) Envelope. The messages are usually exchanged using HTTP over UDP.

Standards
DPWS is built on common standards for Web services:

- XML
- SOAP
- Hypertext Transport Protocol (HTTP)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Message Transmission Optimization Mechanism (MTOM)
- Secure Sockets Layer (SSL) and Transport Layer Security (TLS)–based security
- Web Service Description Language (WSDL)

4.1.2. Discovery

Based on WS-Discovery specification, discovery plays an important role when using DPWS.

Every DPWS-compliant device announces its presence on the network by sending a *Hello Message* and sends a *Bye Message* when it leaves the network. These messages are "*multicasted*" to the whole sub network so every DPWS-compliant device receives these messages.

A device can also search by itself on the network for any available DPWS-compliant devices, by *probing* the network for certain types (of services) or scopes. *Resolving* is available too, and it works in a similar fashion as probing, but is used when device knows the Endpoint (address) of the Device to reach.

A major advantage of DPWS is that due to this multicast service discovery with WS-Discovery, DPWS does not require any central service registry such as UDDI.

4.1.3. Description

Based on WS-Metadata specification, every DPWS device is able to describe itself and all the services it hosts.

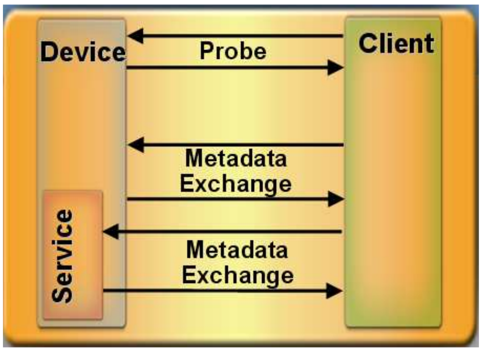
The device can respond to a WS-Transfer “Get” message by describing:

- The model of the device (Name, Manufacturer, etc.)
- The device itself (Firmware, Serial Number, etc.)
- Its hosted services (by listing their Endpoints, Types, etc.).

Once the Endpoint reference (EPR) of a hosted service is discovered, a device can also ask to describe itself too. That is done by a different Message: the GetMetadata message. The response contains the following information:

- The provided functionality described by WSDL (that basically contains all actions that can be invoked on the hosted service and the structure of arguments of those actions and any responses they might generate).
- The hosted service might however decide not to include its WSDL inline in the GetMetadata response and provide a reference (a link) to it instead.
- Any Policies the hosted service enforces (policies behave just like normal Web Services Policies but there are some amendments to their specification in the DPWS specification).

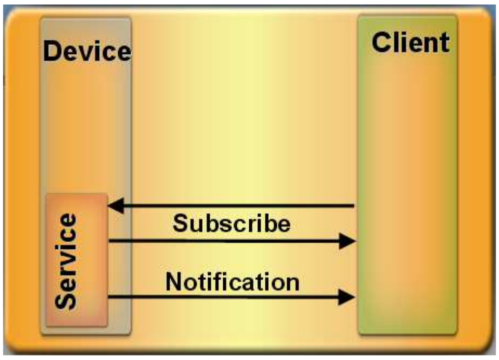
Here is a typical schema on a DPWS client discovering DPWS device interactions and retrieving metadata of this device (and its services):



4.1.4. Eventing

Based on WS-Eventing specification, a hosted service may provide events for clients to subscribe to. The client subscribes to all the events of the service, but may use a filter to specify only certain actions of interest. Events are subscribed for a certain duration of time (but this can be tricky as some resource-constraint devices might not have real-time clock) and can be renewed at any time.

Here is a typical client subscription to events coming from a device service and when a sends an event notification:



4.1.5. Security

Current DPWS specification only recommends using some kind of security, but no standard is required to be implemented. The specification recommends: A compact signature as a proof of authenticity when sending unencrypted Discovery Messages; Secure channel (TLS/SSL) for communication between individual Devices; and Certificates to verify the authenticity of a Device we are talking to.
In case of additional security needs, WS-Security and other WS-* specifications related to security (WS-Trust, WS-Reliable Messaging, WS-Federation, etc) can be implemented as additional plug in to DPWS. More details in the third part of the document.

4.1.6. Interoperability

The main advantage of DPWS is that it ensures you full interoperability. If your device is enabling DPWS as its communication platform, then you enable third-parties devices to connect to your devices straight away without you having to implement another API for them. It is a major advantage of DPWS.

A common example is a device Printer: A manufacturer X just need to implement the Printer Service on its Printer device and then any client that knows how to print on a Printer type will be able to find the “X” Printer on the network and print on it without the need of any drivers.
And vice-versa if you choose to support a Printer Type in your printing software, then your software will be able to dynamically discover any Printers (Devices hosting a Printer Service) on the network and print on any of them.
Thanks to the description capabilities of Services, it is possible to write a piece of software that enables users to control any DPWS device on the network. Given some “intelligent” behavior, your application can generate a nice GUI for any Device based upon its WSDL making it possible to control any device. This only makes sense for Devices that host Services that are meant to be controlled by humans in the first place, but if you consider let’s say a Home automation scenario it is perfectly reasonable. This way any vendor can create any remotely controlled appliance and your automation software enables the user to control it straight away again without any painful installation process or compatibility issues.

4.1.7. Resources, implementations and related links

There are several available implementations of DPWS specifications. Here are the most famous ones:

- Open source & platform independent implementations:
    - SOA4D DPWS stack: ANSI C, Java and OSGi DPWS stack, with a WS-Management plugin (for device management) if needed. This is the result of the SODA ITEA2 European Research Project. The Open source forge is active, and projects are still evolving
    - WS4D DPWS stack: A C (gSoap), Java (JMEDS) and Java (Axis2) stacks.
- Microsoft implementations:
    - Windows Vista (and now Windows 7) is embedding a DPWS stack called “WSDAPI” (C/C++ implementation). It is part of Windows Rally technologies.

- .NET Micro Framework (a reduced and light-weight .NET Framework for small resource-constraint devices) supports natively DPWS.

4.2. Using DPWS

Device manufacturers currently expend considerable resources developing proprietary protocols to support device implementations. Proprietary solutions typically entail higher costs and limited interoperability and deliver an inconsistent customer experience.
In contrast, DPWS enables device manufacturers to use an industry-standard contract model (WSDL) to decrease development costs and improve the return on investment.

The main advantages of using DPWS include:

- **Common standards**. DPWS protocols are built by using well-known Internet standards including XML and SOAP, HTTP, and TCP/IP. This allows developers to use a unified tool set that is transferable across an entire range of implementations.

- **Interoperability**. DPWS enables network-connected device implementations to share data and use capabilities from other applications. It does not matter how the applications are built, which operating system or platform is used, or what devices are used to access them.

- **Scalability**. DPWS is scaled down specifically to support resource-constrained devices. This support can be scaled out to support large organizations and diverse geographical locations.

- **Security**. DPWS supports channel-based security such as TLS with network-connected devices. It can also employ Web services message-based security such as WS-Security.

Communication protocols and web services – conclusion in MIDAS

5. CONTINUA HEALTH ALLIANCE

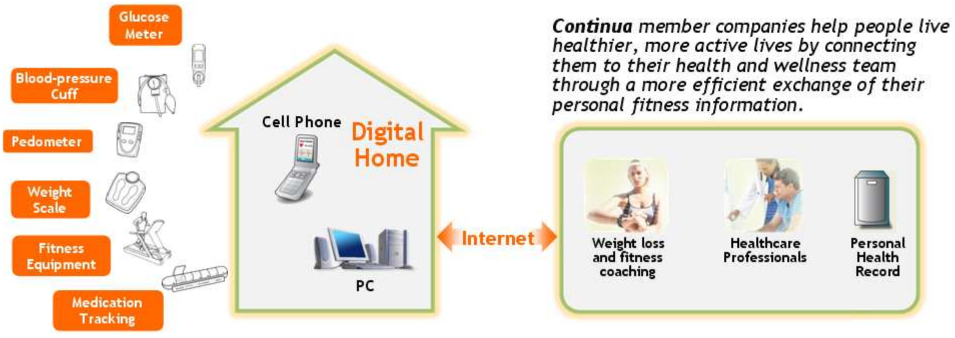
Continua Health Alliance is a non-profit, open international industry coalition (over 208 members companies), created to improve the quality of personal healthcare. Its goal is to establish a system of connected personal tele-health solutions to better manage health and wellness (while offering personalized health and wellness management solutions), by means of establishing a product certification program.

5.1. Focus on three main scenarios

Continua Health Alliance is focusing in three main scenarios related to health, and the main goal is to define sensor data transmissions, and data exchange.

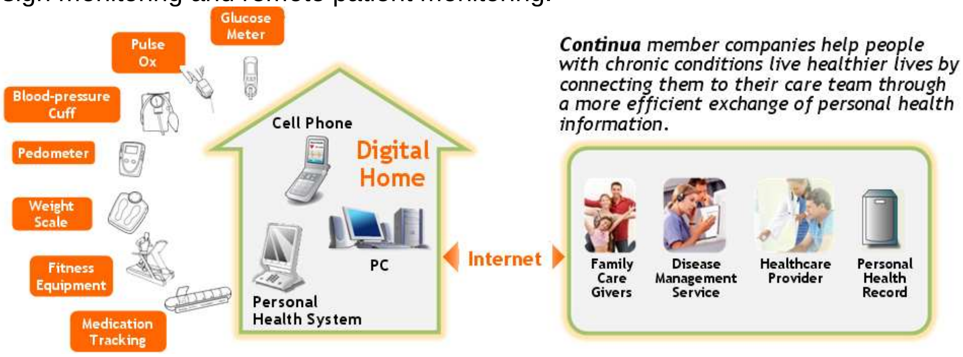
5.1.1. Health and Wellness

By allowing monitoring of vital signs, and in the future by extended the healthcare system into home environment.


Glucose Meter
Blood-pressure Cuff
Pedometer
Weight Scale
Fitness Equipment
Medication Tracking
Cell Phone
Digital Home
PC
Internet
Continua member companies help people live healthier, more active lives by connecting them to their health and wellness team through a more efficient exchange of their personal fitness information.
Weight loss and fitness coaching
Healthcare Professionals
Personal Health Record

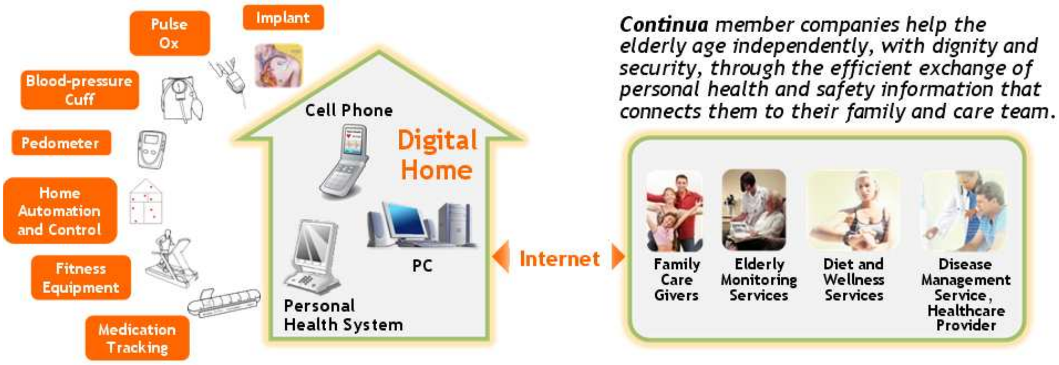
5.1.2. Disease Management

There are 860 million chronic disease patients worldwide (chronic disease, post-trauma, pre-op, etc). Allow vital sign monitoring and remote patient monitoring.


Glucose Meter
Pulse Ox
Blood-pressure Cuff
Pedometer
Weight Scale
Fitness Equipment
Medication Tracking
Cell Phone
Digital Home
PC
Personal Health System
Internet
Continua member companies help people with chronic conditions live healthier lives by connecting them to their care team through a more efficient exchange of personal health information.
Family Care Givers
Disease Management Service
Healthcare Provider
Personal Health Record

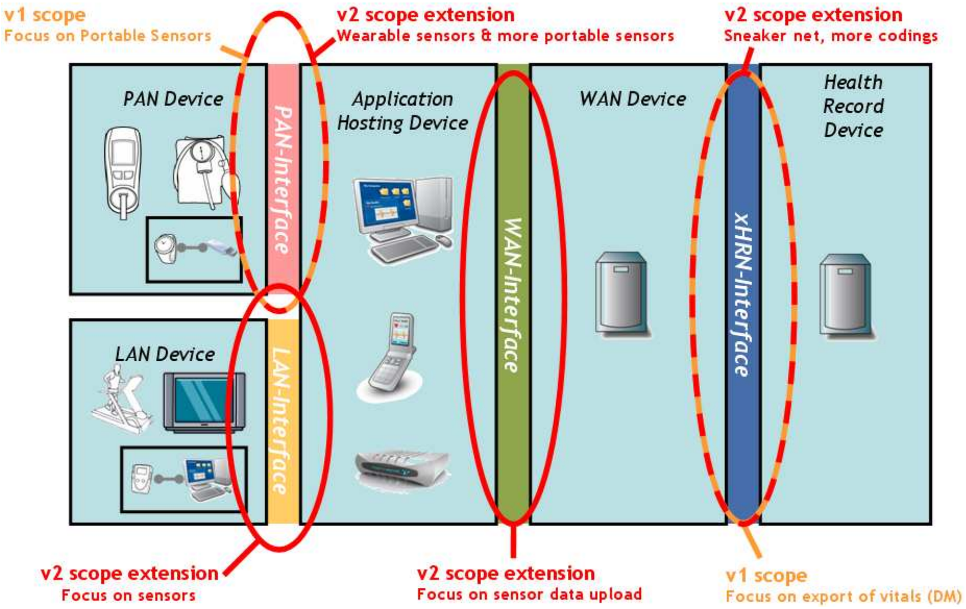
5.1.3. Aging independently

There are 600 million elderly individuals worldwide. Offer a basic life monitoring as appropriate (e.g. bed pressure, bathroom sensors, gas/water sensors, emergency sensors…).


Pulse Ox
Implant
Blood-pressure Cuff
Pedometer
Home Automation and Control
Fitness Equipment
Medication Tracking
Cell Phone
Digital Home
PC
Personal Health System
Internet
Continua member companies help the elderly age independently, with dignity and security, through the efficient exchange of personal health and safety information that connects them to their family and care team.
Family Care Givers
Elderly Monitoring Services
Diet and Wellness Services
Disease Management Service, Healthcare Provider

5.2. Continua E2E Reference topology

Continua Health Alliance provides a “framework” for member companies who want to be certified, called “**Continua E2E Reference topology**”.



5.2.1. Continua devices

5.2.1.1. PAN devices

PAN devices mean “Personal Access Network” devices. This class of devices is related to sensors and actuators devices which are in-body, on-body or wearable near body devices, dedicated to a single person usage (for specification v1 only, in v2, PAN devices will allow to be dedicated to several people). They collect raw data from one or more sensors or control one or more actuators, and for communication and interaction with application hosting devices, they have to adhere to the standards defined for the PAN interface (cf. §5.2.2.1).

5.2.1.2. LAN devices

LAN devices mean “Local Access Network” devices. This class of devices is related to sensors and actuators devices which are stationary or moveable, in a room, a house, a car, or at public space. They collect raw data from one or more sensors or control on or more actuators. They can have local processing from raw data and aggregate data from multiple sensors. For communication and interaction with application hosting devices, they have to adhere to the standards defined for the LAN interfaces (cf. §5.2.2.2).

5.2.1.3. Application hosting devices

This class of devices has four main functions:

- Communicate with the PAN and LAN devices to collect sensor data and to send commands to the actuators (by supporting PAN/LAN communication interfaces).
- Communicate with the WAN services to forward sensor data to the relevant back-end services and to process the response (by supporting the WAN interface).
- Optionally act as a local storage of data, aggregation of data. Reasoning/processing these data are not in the scope of current Continua specification. But if data processing is done, the device must comply with applicable standards related to the messages with respect to syntax (format and coding system), but also comply with applicable standards about ontology and vocabularies (WAN interfaces).
- Interact with user to present information or interact with him.

5.2.1.4. WAN Devices

WAN devices are called « PHR » (aka « *Personal Health Records* »). These devices are mainly fed by the patient himself.

5.2.1.5. Health Record Devices

These EHR devices (Electronic Health Records) are more specifically fed by health professional.

5.2.2. Continua Interfaces

5.2.2.1. PAN interface

The PAN interface is dedicated to realize the connectivity, communication and data exchange from PAN devices with one or more application hosting devices (or other devices on the LAN).
As this interface focuses on basic connectivity, communication protocols and also standardization messages, potential relevant standards are:
For Basic connectivity and communication: IEEE 802.15 (1.3a.4.6), Bluetooth, Zigbee, Z-wave, USB, UWB, RFID and NFC.

- For messages: USB device class for personal healthcare devices, Bluetooth health device profile.

For PAN interface version 2, Bluetooth, BT Low Energy, and USB are supported. In the use cases currently under discussion into Continua Consortium, NFC and Low Power Wi-Fi are studied. But neither UWB nor RFID are in the scope of Continua specification.

The network should be able to handle dynamic configurations so registration and pairing protocols are needed. Potential relevant standards are:

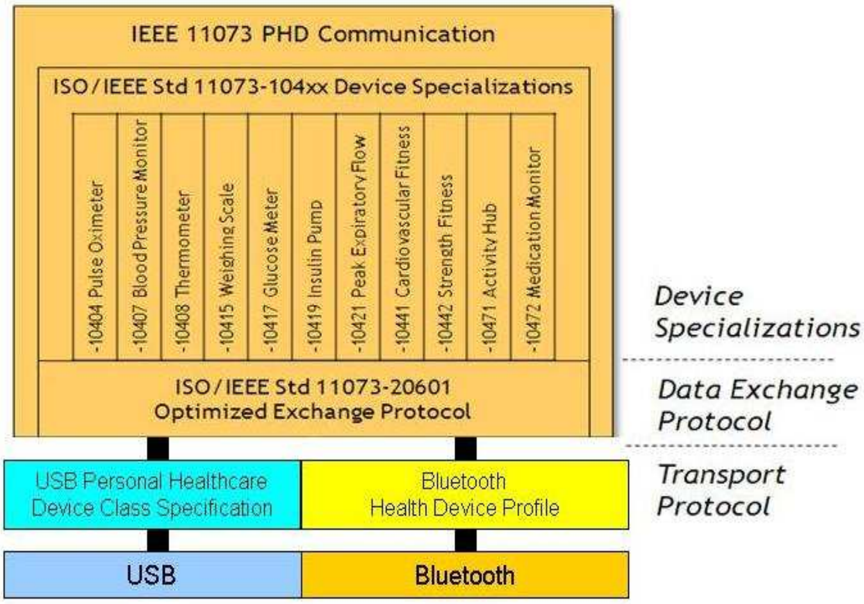
- Bluetooth, Zigbee, Z-Wave, but also:
- Device Profile for Web Services (DPWS).

Concerning the Security topic, due to healthcare domain, it is important to be able to prohibit unauthorized access to some devices, and secure transportation of sensitive health data. Main issues are: pairing procedures, certificate handling and encryption. Potential relevant standards are:

- USB Device class for personal healthcare devices
- Bluetooth health profile device
- Security for Zigbee (in preparation).
- WS-Security, WS-Trust, and other WS-* specifications, e.g. WS-Federation.

The reference topology v1 is focusing on portable sensors; the v2 will extend the scope to wearable sensors & more portable sensors. This is a unidirectional communication between the PAN devices to the application hosting device. Some works are in progress in Continua Consortium to define a bidirectional communication.

The PAN interface is structured in three distinct layers:

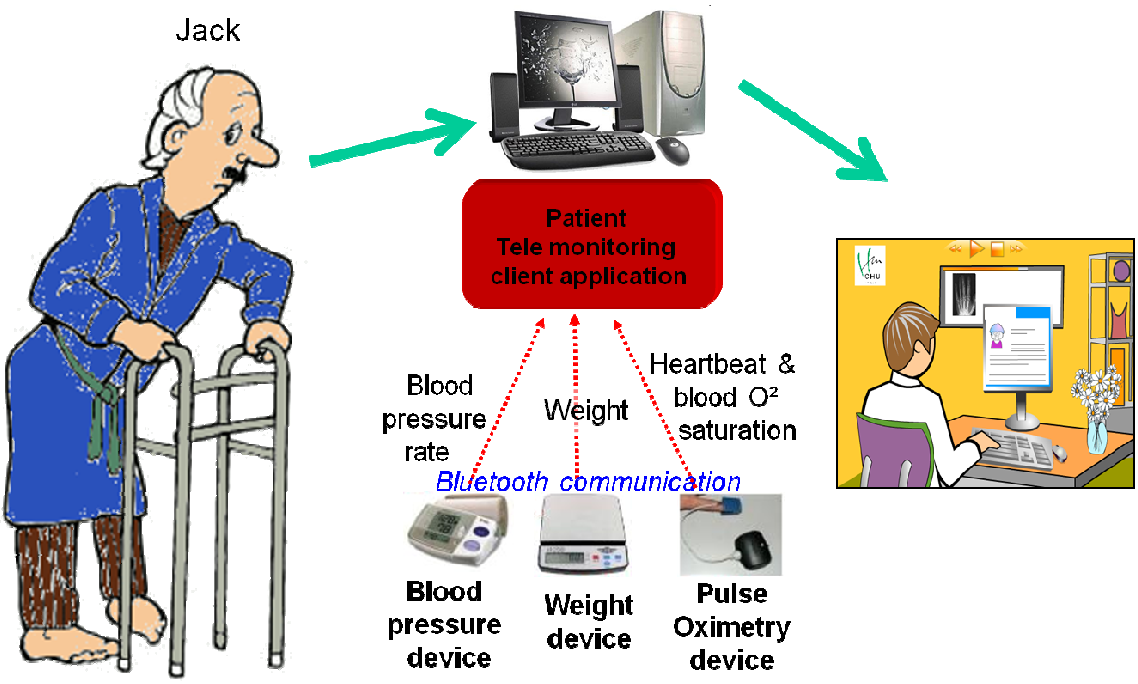

IEEE 11073 PHD Communication
ISO / IEEE Std 11073-104xx Device Specializations
-10404 Pulse Oximeter
-10407 Blood Pressure Monitor
-10408 Thermometer
-10415 Weighing Scale
-10417 Glucose Meter
-10419 Insulin Pump
-10421 Peak Expiratory Flow
-10441 Cardiovascular Fitness
-10442 Strength Fitness
-10471 Activity Hub
-10472 Medication Monitor
ISO / IEEE Std 11073-20601
Optimized Exchange Protocol
USB Personal Healthcare Device Class Specification
Bluetooth Health Device Profile
USB
Bluetooth
Device Specializations
Data Exchange Protocol
Transport Protocol

- The Continua E2E Reference Topology, defines several “profiles”, and a specific standard for each family of PAN devices: Standard is ISO/IEEE Std 11073-104xx, where xx is:
  - 04: Pulse Oxymeter.
  - 07: Blood Pressure Monitor.
  - 08: Thermometer.
  - 15: Weighing scale.
  - 17: Glucose meter.
  - 19: Insulin pump.
  - 21: Peak expiratory flow.
  - 41: Cardiovascular fitness.
  - 42: Strength fitness.
  - 71: Activity hub.
  - 72: Medication monitor.
- The data exchange protocol is based on ISO/IEEE Std 11073-20601.
- And on the same way as for device specialization, the Continua E2E Reference Topology defines also its transport mode. Depending on the version of the E2E Reference Topology, different transport modes are supported :
  - E2E version 1:
    - A wired interface: USB (Personal Healthcare Device Class).
    - A wireless interface: Bluetooth (Health Device Profile).
  - E2E version 1.5, same as version 1, plus:
    - Bluetooth LE (Low Energy).
  - E2E version 2, same as version 1.5, plus:
    - Wifi LP (Low Power).
    - NFC as well.

For the heath monitoring scenario 6, a software interface with the Bluetooth wireless 9560 Onyx II Oxymeter certified Continua Health Alliance product has been developed. This interface enables to acquire automatically health measurements into a software application that encapsulate them in a XML file.
The file is then requested and hosted in the MIDAS web portal with the rest of the health results.

The feedbacks on this development are:

- The interface is non-generic. A dedicated development shall be done for each manufacturer or each product family
- Each sensor has its own proprietary frames
- Each measurement is sent only once and not in a continue way by the sensor. This may lead to data lossin case of software manulfacturing



5.2.2.2. LAN interface

The LAN interface is dedicated to realize the connectivity, communication and data exchange from LAN devices with one or more application hosting devices. It is “an extension” to PAN interface for the Reference Topology for version 2.

As this interface focuses on basic connectivity, communication protocols and also standardization messages, potential relevant standards are:
For basic connectivity:

- IEEE 802.15, Bluetooth, Zigbee, Z-Wave, UWB, Homeplug (using in-home power line).
- Internet protocols and web based protocols (http, https)
- Specific protocols like X10, Lonworks, KNX or INSTEON.

For Continua LAN interface version 1.5, only Zigbee is currently supported.

Concerning data exchange, in the healthcare domain, a number of standards exist like ISO/IEEE P11073-104xx and P11073-20601, USB device class for personal healthcare devices, or Bluetooth health device profile, all used in Continua. The network should be able to manage devices and applications and to handle dynamic configurations. Potential standards relevant are:

- Bluetooth, Zigbee, Z-Wave
- Open Service Gateway Initiative (OSGi)
- Universal Plug and Play
- Device Profile for Web Services (DPWS).

In the security domain, potential data are:

- Bluetooth health device profile
- Security for Zigbee (in preparation)
- HTTPS for secure connections and transport.

5.2.2.3. WAN Interface

The goal of WAN interface is to move measurements from the Application Hosting Device (AHD) to a WAN device.

The main advantage is that the patient information can reside outside the Application Hosting Device (AHD), and support remote monitoring by professional services.

The definition of this WAN interface is currently in progress (a release 1.5 might be available at the end of S1 2010).

It is based on protocol HL7 v2.6, message using IHE PCD-01 (messages constrained to IEEE-20601 / 104zz nomenclature).

HL7 / Health-Level 7

HL7 provides a framework (and related standards) for the exchange, integration, sharing, and retrieval of electronic health information. v2.x of the standards, which support clinical practice and the management, delivery, and evaluation of health services, are the most commonly used in the world.

The name "Health Level-7" is a reference to the seventh "application" layer of the ISO OSI Reference model. The name indicates that HL7 focuses on application layer protocols for the health care domain, independent of lower layers. HL7 effectively considers all lower layers merely as tools.

5.2.2.4. xHRN Interface

This interface is dedicated to transfer patient information from a Continua WAN device (xHR Sender) to an electronic health record device (xHR Receiver).

Some guidelines exist for xHR interfaces, in order to establish the basic standards, rules, restrictions in the data, messages and transport protocols. Data transmitted are based on protocol HL7 v3 (HL7 CDA R2 standard). And concerning the transport layer, the IHE XDR profile uses current standard such as SOAP 1.2.

6. DPWS: A COMPLIANT MIDDLEWARE FOR CONTINUA DEVICES

6.1. Connectivity, protocols and data exchange

A lot of protocols exist in health area, where multiple products, applications and services have to collaborate and interoperate.

6.1.1. For PAN and LAN devices and eponymous interfaces

In its specification v1, Continua Health Alliance has chosen Bluetooth and USB for communication of PAN and LAN devices.

DPWS can be used with specifications specified by the Continua Health Alliance (Bluetooth, USB) and either be used on additional specifications such as Wifi for example. Some specific implementations of DPWS have been developed to run on different Operating Systems, and on different wired/wireless standards.

Wifi can be used with IPv4 or IPv6 protocol and USB with an Ethernet-over-USB driver or a low-cost adapter to allow Ethernet network running on USB, so it is trivial to use DPWS over it.

Considering Bluetooth, a DPWS-Bluetooth SDP (Service Discovery Protocol) bridge was developed by the University of Rostock to connect resource constrained devices to a European R&D project network[2]. Other solutions can be considered, such as using RFCOMM, a service based on RS-232 specifications, which allow to transport IP communication via Bluetooth (RFCOMM is used when data rates are lower than 360kbit/sec such as in mobile phones).

In addition, it is always possible for PAN and LAN devices to keep using their own proprietary (or not) technology (e.g. X10, Zigbee, RFID, etc) as long as an application hosting device is present and acts as a gateway, in order to communicate with their own protocol on one side and convert data flow into an unified DPWS-compliant communication on the other side.

6.1.2. For Application hosting devices and WAN interfaces

Concerning PAN and LAN devices, and interfaces with Application hosting devices, a DPWS stack can be embedded into Application Hosting Devices to communicate with DPWS-compliant PAN and LAN devices. Application Hosting Devices can also or act as a gateway to communicate with devices using their own protocol and converting data flow coming from PAN/LAN devices into a DPWS-compliant communication bound for other Application Hosting devices, or WAN devices.

Concerning transmission of data to WAN devices through WAN interface, the Continua Health Alliance has specified that data must be transmitted using HL7 v2.6 protocol, messages using IHE PCD-01 (messages constrained to IEEE-20601 / 104zz nomenclature).

In order to keep compliance with the Continua Health Alliance specification and devices, and also keep advantages of using WS-* specifications (network agnostic, TCP/IP protocol, open specifications), DPWS must be able to transmit data based on HL7 v2.6 protocol.

A first solution is based on the same way as previously defined with PAN and LAN devices: it is possible to create an Application Hosting Device acting as a gateway to convert PAN/LAN protocols into HL7 protocol to communicate with WAN devices.

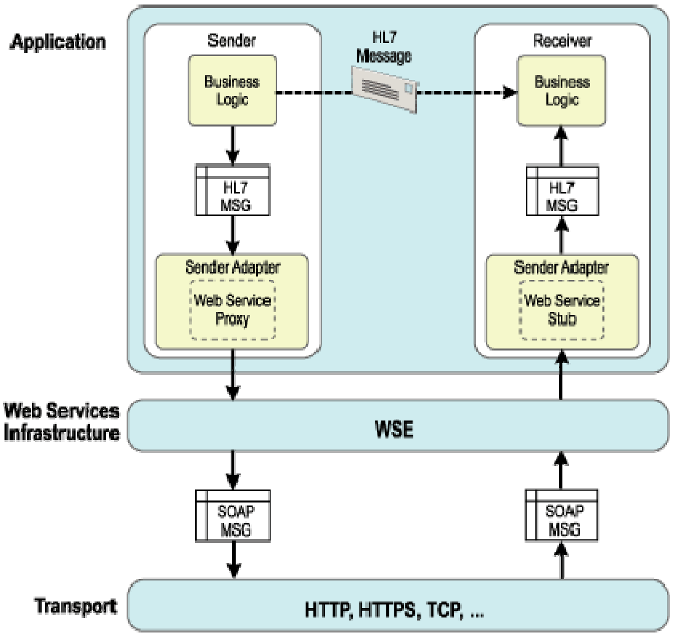
[2] H. Bohn, A. Bobek, and F. Golatowski. Bluetooth Device Manager Connecting a Large Number of Resource-Constraint Devices in a Service-Oriented Bluetooth Network. In *ICN'05*, St. Gilles Les Bains, La Reunion, 2005

Another solution is to keep using DPWS protocol between Application Hosting Device and WAN devices. Either as is, using DPWS protocol as main communication protocol to transmit data between devices, or by allowing DPWS to “understand” and communicate using HL7 protocol. In order to realize this, some implementations of DPWS stack has an embedded extensibility layer, which allow to extend DPWS features with news WS-* specifications/features, in the form of “plug ins”.

For example, ITEA2 European R&D SODA[3] project offers a management plug in to the DPWS stack, in order to manage DPWS devices, using a unified web service protocol specified by DMTF and named WS-Management[4]. This specification is on the way to become an ANSI and ISO standard[5].

Some can offer WS-Security implementations for security purposes, or EXI or XML binary specification plug in to reduce on-the-fly XML content. Some others are creating more specific plugs in to embed specific execution engine such as industrial Grafcet engine which is able to interpret “on the fly” the Grafcet language.

As DPWS is based on SOAP message, it is possible to represent applications which Send/Receive HL7 messages using DPWS stack (Web Service infrastructure) as following[6]:



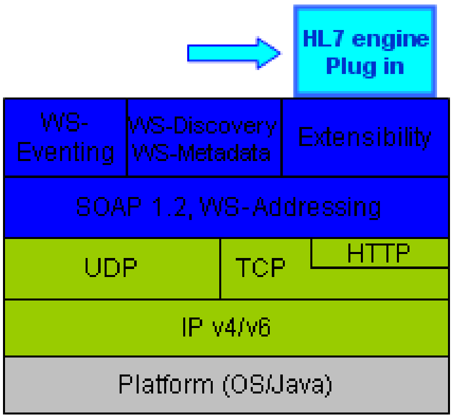
Using the DPWS extensibility layer, and on the same base as exposed previously for Grafcet engine, it is then conceivable to realize a HL7 v2.6 plug in engine for DPWS in order to manage HL7 protocol:

[3] http://www.soda-itea.org

[4] http://www.dmtf.org/standards/wsman

[5] http://www.dmtf.org/newsroom/newsletter/2010/05/page2#1

[6] http://msdn.microsoft.com/en-us/library/ms954603.aspx


HL7 engine
Plug in
WS-
Eventing
WS-Discovery
WS-Metadata
Extensibility
SOAP 1.2, WS-Addressing
UDP
TCP
HTTP
IP v4/v6
Platform (OS/Java)

6.1.3. For WAN devices, Health Record devices, and xHRN interfaces

Concerning transmission of data to Health Record devices through xHRN interface, coming from WAN devices, the Continua Health Alliance has specified that data must be transmitted using HL7 v3 protocol.

On the same basis as previous chapter, using a specific HL7 v3 engine plugged on DPWS stack, it is conceivable connect and offer bidirectional communication between WAN devices and Health Record devices using unified DPWS middleware.

6.2. Security

Security topic is an important issue in medical domain, because a lot of sensitive data and control commands have to be managed. In this medical field, demanded security services are:

- **Authorization**: This service must control access to device resources. When an entity sends a request to a device, the latest has to check whether such entity can access the appropriate resources.
- **Authentication**: This service must be able to validate that an entity identified in a message exchange is the one it claims to be.
- **Message Uniqueness**: This service must ensure that a specific message is not resubmitted for processing. An attacker could resend all or selective parts of a message causing undesirable side effects. Sending the same valid message is frequently used in many denial-of-service attacks.
- **Data Origin Authentication**: This service must guarantee the origin of the message. When an entity receives a message, it must be sure that the sender is not trying to pose as another entity.
- **Data Confidentiality**: This service must ensure that information is not made available or disclosed (view or eavesdrop) to unauthorized individuals, entities or processes.
- **Data Integrity**: This service must be able to detect any changes made in original messages.

Information security can be achieved at different levels: on transport layer or at message level.

6.2.1. Transport security layer

For transport layer security, a secure channel must be established for end-to-end communication. This secure channel is providing data integrity and confidentiality as long as both endpoints authorize each other at the beginning.

HTTPS is an example of transport layer security that uses the TLS protocol (Transport Layer Security protocol) which is the successor of SSL (Secure Socket Layer).

6.2.1.1. About Continua

At transport level, current Continua specification does not impose anything specific.

6.2.1.2. About DPWS

Current DPWS specification only recommends using some kind of security, but no standard is required to be implemented. The specification recommendations concerning transport security layer is to be able to secure channel (using TLS/SSL) for communication between individual devices.

Some current implementations of DPWS stacks are offering an adequate HTTPS implementation for securing connections and transport, as for example the latest ANSI C releases of SODA DPWS stack (since version 2.3.17).

6.2.2. Message security layer

For message security layer, each message is secured individually instead of sending unsecured messages through a secure channel. The benefit of message level security is that the messages can pass through non secured nodes. Furthermore, non-repudiation can be achieved on the message level by using digital signatures.

On the other side, the message level security is much more complex in contrast to the transport layer security, and needs more processing power as each exchanged message must be secured before serialization, making this solution more difficult to be integrated on small processing embedded systems.

As Continua and DPWS are both using SOAP messages, using message security layer on SOAP messages can infer other drawbacks, such as adding significant overhead to SOAP-messages processing due to the increased size of the message on the wire, XML and cryptographic processing, requiring faster CPUs and more memory and bandwidth, which might not be adapted to low CPU power of small embedded systems.

6.2.2.1. About Continua

Current discussion are in progress concerning securing of xHRN interface. Security for WAN-interfaces and xHRN-interfaces is a combination of WS-Trust, username tokens profile (OASIS) and SAML 2.0 token.

6.2.2.2. About DPWS

As seen previously in §6.2.1.2, current DPWS specification only recommends using some kind of security, and no standard is required to be implemented. The specification recommendations concerning message security layer are:

- Using compact signature as a proof of authenticity when sending unencrypted discovery messages.
- Use of certificates to verify the authenticity of a device we are talking to.

Due to DPWS extensibility layer, it is possible to implement additional security features at message level, by implementing full or partial specific security WS-* related specifications.

6.2.3. WS-* Security specifications

Here are as example three specifications which are part of the larger Web service Security framework. All of them can be implemented fully or partially, on a standalone way, or as a plug in to DPWS for example.

- **WS-Security:** This OASIS specification (version 1.1 released on February 17, 2006) is a flexible and feature-rich extension to SOAP to apply security to Web services. The protocol specifies how integrity and confidentiality can be enforced on messages and allows the communication of various security token formats, such as SAML, Kerberos, and X.509. Its main focus is the use of XML Signature and XML Encryption to provide end-to-end security.

[7] https://forge.soa4d.org/forum/forum.php?forum_id=195

- **WS-Trust:** This OASIS standard provides extensions to WS-Security, specifically dealing with the issuing, renewing, and validating of security tokens, as well as with ways to establish, assess the presence of, and broker trust relationships between participants in a secure message exchange.

- **WS-Federation:** This specification defines mechanisms for allowing disparate security realms to broker information on identities, identity attributes and authentication.

Other WS-* specifications related to security exist and can be implemented by user on demand (e.g. WS-SecureConversation, WS-Policy, etc.).

Based on previous list of needed security services, it is possible to cover all of them using WS-* Security specifications:

- **Authorization**: Normally, Authorization Servers are needed for validation. They store the policies established for each resource and recommend the device to deny or grant the access. In some light scenarios, this will have to be implemented by the own target device. This can be done by means of XACML together with SAML assertions.
- **Authentication**: Typically, an Authentication Server is required to provide this service, although it can also be achieved in a standalone basis.
- **Message Uniqueness**: The time stamping mechanisms are often recommended, but imply strong synchronization among entities or adding a new element in the scenario for centralized time checking, that cannot always be afforded. It could also be achieved by means of message ordering and reliable message delivery mechanisms together with XML-Encryption. Inserting in the message a ciphered hash built from the own message and the hash of the last sent message is also a preferred method.
- **Data Origin Authentication**: This service can be carried out by means of XML-Signature.
- **Data Confidentiality**: This service can be supported by the use of XML-Encryption.
- **Data Integrity**: The use of XML-Signature guarantees that the original message (or specific parts of the message) has been transmitted unaltered.

Concerning the performance of using WS-Security features, it is a fact that WS-Security adds significant overhead to SOAP-processing due to the increased size of the message on the wire, XML and cryptographic processing, requiring faster CPUs and more memory and bandwidth.

Some academic researches[8] are involved in order to identify opportunities for optimizing performances of Web Services Security.

6.2.3.1. In conclusion on the middleware

At the beginning of the project, OSGi was proposed to serve as a single frame for the development in MIDAS project. This idea was abandoned afterwards since it did not fit to the interest of all partners.

Participation to standard effort is one of MIDAS partners preoccupation. It did not give concrete result so far though. Studies are still conducted on possible usage and promotion of standards allowing interoperability like DPWS and data representation in ontologies.

[8] Performance Comparison of Web Services Security: Kerberos Token Profile against X509 Token Profile, 2008 conference IEEE “Congress on Services“ (http://www.netmode.ntua.gr/papers/amoral/PID383108.pdf)

7. Digital video

Regarding video standards, I&IMS is the first Spanish software developer company accepted as user member of ONVIF (Open Network Video Interface Forum, http://www.onvif.org/). The purpose of the forum is to facilitate the development and use of a global open standard for the interface of network video products. Although they are not making specific use of the ONVIF standard on this project, I&IMS uses compliant hardware able to provide the technical possibilities of this standard if needed.

The technology developed by IIMS is used in the scenario 7 Security and assistance which aims to monitor home thanks to different devices such as camera for an intelligent video surveillance.

Here is the IP camera used for the intelligent video surveillance:

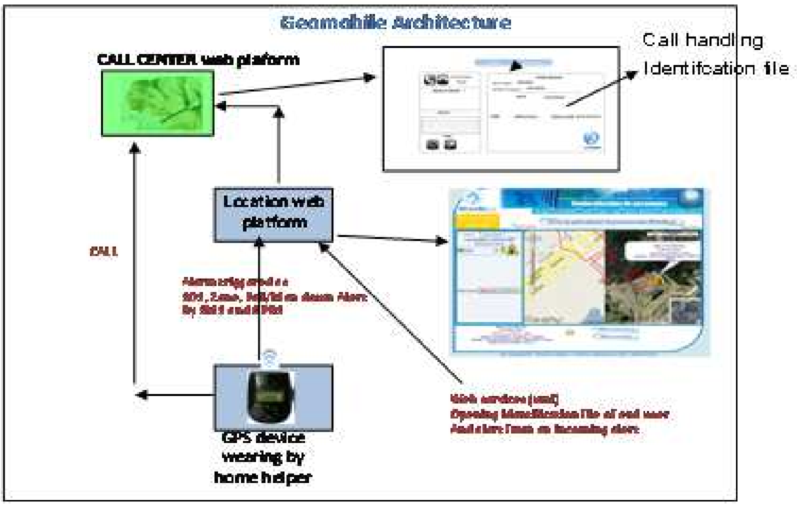
IP Camera

8. Teleassistance alert protocol

Geomobile choose to use protocol PFIG for its call centre. This protocol is used to send alerts to a call centre via an XML flow. Teleassistance centres, integrators or teleassistance device manufacturers make an extensive use of this protocol. PFIG is an open protocol for which a good amount of documentation is available. This protocol which can be considered as a de facto standard has not been standardized now though.

The technology developed by Geomobile is used in the scenario 7 Security and assistance which aims to monitor home thanks to different devices such as a GPS device able to make geolocalisation indoor and transfer information a call center in case of alerts.

Here is the Geomobile Architecture:



Geomobile Architecture

Call handling
Identifcation file

CALL CENTER web plaform
Location web platform
CALL
Alarms triggered as
SOS, Zone, Fall/Man down Alert
By SMS and GPRS
GPS device wearing by home helper
Web services (xml)
Opening Identification file of end user
And alert from an incoming alert

Geomobile platform can provide 4 types of alarms:

- SOS alert
- Zone alert
- Fall detection
- Alarm for future use

Each alarm is reported though voice communication channel. This is the common and easy way to identify the caller from his/her SIM phone number and open his/her identification file from the supervisor centre.

However voice call can fail sometimes due to trouble on GSM network (as GSM network congestion, lack of GSM coverage, low battery level of GPS device). Alarm has to be transmitted on other communication channels as GPRS and SMS channel.

Kompaï robot can be accessed using the WEB interface called Lokaria.

9. Conclusion

In the medical domain, the proposed DPWS-based architecture can act as a comprehensive middleware and connect from deeply embedded devices such as sensors or actuators to devices with richer resources.

It is designed to achieve full plug-and-play, bidirectional communication and asynchronous capabilities with devices from different vendors.

Even if some reservations have to be made concerning the performance of Web Services in general (in particular the applicability of Web Service technologies to reach real-time capabilities[9]), DPWS is designed to run perfectly on resource constrained devices as well as high-end computing systems, and Web Services in general are one of the most accepted building blocks for a Service-Oriented Architecture systems.

DPWS is a convenient middleware which offers full interoperability between devices, and can be enriched on several domains (security, management, XML compression) by implementing additional software plugs in. It is also possible to use hardware extensions in conjunction with DPWS middleware, if an enforced middleware is needed on some specific parts (encryption, certificate management...).

Continua Health Alliance is proposing a set of specifications dedicated to devices working in health domain. As seen in this document, marrying Continua specifications with an implementation of a middleware based on DPWS is an interesting solution in order to improve Continua concepts with dynamic and new features bring by DPWS.

Regarding the others points of the standardization effort in MIDAS project; we can say that they were seriously limited after the Spanish partner's renouncement one year before the end of the project. The restructuration implied for remaining partners to concentrate on the objective of the project, integration, testing and evaluations leaving on a second plan the standards to use for this objectives.

[9] http://www.imd.uni-rostock.de/veroeff/prueter-ApplicabilityWebServiceTechnologiesReachRealTimeCapabilities.pdf

Consortium

ORANGE LABS
France
www.orange.com

Project Coordinator :
Laure Chotard, Orange Labs
laure.chotard@orange-ftgroup.com

CEA LIST
France
www.cea.fr

CITIC
Spain
www.citic.es

CNRS
France
www.lifl.fr

ENERGY SISTEM SOYNTEC
Spain
www.energysistem.com

FICOTRIAD
Spain
www.ficosa.com

GEOMOBILE
France
www.whereru.eu

I&IMS
Spain
www.ims.es

INTUILAB
France
www.intuilab.com

KATRON
Turkey
www.katron.com.tr

KIT
South-Korea
http://www.kitvalley.com/

LI2G
France
www.chu-grenoble.fr

MORGAN CONSEIL
France
www.morganconseil.com

CREATIV IT
Spain
www.creativit.com

ROBOSOFT
France
www.robosoft.fr

ROBOTIKER-TECNALIA
Spain
www.robotiker.es

SIEL BLEU
France
www.sielbleu.org

TELEFONICA I+D
Spain
www.tid.es

THALES ALENIA SPACE
France
www.thalesaleniaspace.com