socialize

service oriented assisted living infrastructure

WP 2– Development of SOCIALIZE system components

Document: D2.1 Access Devices

D2.6 Elderly Monitoring Service

Date: <22/09/2016>

Confidentiality: ☐ Public ☑ Restricted ☐

Version: ☐ Draft ☐ Final ☑

AAL PROGRAMME

Document information

Acronym of lead partner for the deliverable	ISTI-CNR
Work package	WP 2 - Development of SOCIALIZE system components
Contractual date of delivery	July 2016
Date of delivery	September 2016
Nature	Report
Dissemination level	Public

Document Responsible	ISTI–CNR (IT)
Author(s)	D. Russo, V. Miori, L.Pillitteri

Release	1
Version	3
Date	22/09/2016

REVISION HISTORY

Date	Version	Author	Paragraph added/modified	Description
15/10/2015	1.1	ISTI-CNR	Added Par. 1,2,3,4,5,6	First draft
25/04/2016	1.2	ISTI-CNR	Added Par. 7,8,9	Second draft
22/09/2016	1.3	ISTI-CNR	Modified Par. 6,7,8,9	Final version

Table of Contents

1 Introduction

The aim of the *SOCIALIZE* project is to create a complete technological solution consisting of a system able to integrate and bring together in one place a multiplicity of services addressed to both the senior population and caregivers ("secondary" end users). Such integration helps to create a space tailored to facilitate participation of the elderly in social life. The included services aim to build and strengthen social relations among people who share similar interests, activities, backgrounds or real-life connections and to monitor the elderly's daily activities and share health information with their network of acquaintances.

The document presents the study and development of the *Elderly Monitoring* service, fully integrated into the *SOCIALIZE* platform. The function of the service is to monitor the activities of the home occupant, collect and analyze personal and environmental parameters from various, technologically heterogeneous devices and enable them to cooperate and share data. The results of the monitoring can be accessed by doctors, relatives and/or caregivers as deemed appropriate with regard, for example, to the cognitive, health status and wishes of the person in question.

In this regard we deploy the development of the *SmartSMILE* (*Socialize MIddLeware Elderly)* framework. *SmartSMILE* is an interoperability software architecture based on *Web Services* and *XML* technologies to create intelligent domestic atmospheres that concur at the integration and interoperability of the offered services.

2 The Socialize services

The services supplied by the project (figure 1) can be divided in two categories: *socialization services*, which aim to improve the social interactions of the elderly, and the *cross services*, vertical applications common to all socialization services.

The socialization services are:

(a) communications, which allows data transfer between members of the *SOCIALIZE* community through an email-like system and an audio / video chat for real time communications;

(b) socialization, which enables cultural and social exchanges between *SOCIALIZE* members; members can post and view the activities of friends or of all users;

(c) the forum, which provides users with the means to actively participate in the issues relating to their needs and requirements; this service utilizes voting (or 'like' mechanism) to simplify use of the service and bring news immediately to the user's attention;

(d) the services catalogue, which contains information about services (description, features, type, etc.) and organizations that provide these services (description, contacts, etc..);

(e) a photo book, which allows members of the *SOCIALIZE* community to store and organize their photos and videos;

(f) elderly monitoring, which offers an interface showing the results of the monitoring system, as well as analyses of user activities and personal and environmental parameters. The results of this monitoring can be viewed by doctors, relatives and / or caregivers as is deemed appropriate with regard, for example, to the cognitive, health status and wishes of the elderly;

(g) distance care, which can follow the elderly's movements and monitor their daily activities, and even warn against any threats

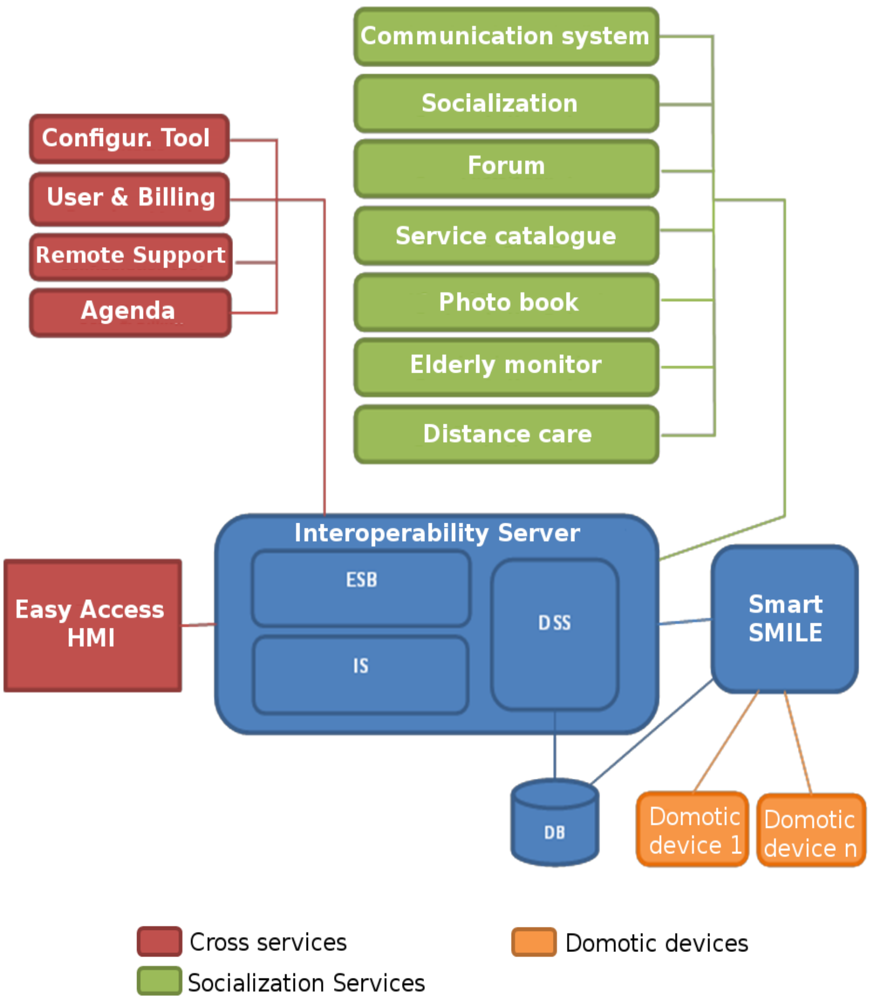

Fig. 1 SOCIALIZE architecture

The implemented cross services are:

(a) the interoperability server, which is the *SOCIALIZE* platform component that allows the different services to communicate by sharing data and functionalities;

(b) the configuration tool, which provides for the registration to, and configuration of, the system and services;

(c) user & billing, which manages *SOCIALIZE* platform users and their access credentials;

(d) remote support, which provides help on using the *SOCIALIZE* platform or, more generally, computers or the device used. This takes the form of video / audio communication with an operator who can help users find a solution to their problem;

(e) easy access, which is a web application with public and private content that acts as the entry point to the *SOCIALIZE* platform;

(f) the agenda, which collects all the events and date/ time-related activities for a given user.

The Interoperability Server is composed of an Enterprise Service Bus (*ESB*), an Information Service (*IS*), a Data Distribution Service (*DDS*) and a Database (*DB*).

3 The domotics

The slang of new technologies has gone enriching of new terms as *Home Automation*, *Building Automation*, *Smart Home*. They are all synonyms diverted by the domotique neologism. The domotique neologism was coined in France and it is composed by the fusion of two terms: *domus* and *informatique*. The domotique neologism represents the technology applications of the information and communication in the domestic world, to improve the comfort and the control of domestic plants.

Domotics parnorama is still in development. The industry and the market have played (and they still play) a role of primary importance for the definitive take-off of the domotic technologies.

Different industry coalitions have promoted numerous middlewares and standards always more sophisticated and always too poorly interoperable each other. This has made the domotics an unexploded bomb. The world of research and industry are waiting for that “boom” that is not still happened.

The real employment of the domotic technology is still hindered by the attempt of each industry to impose its standard on the others.

The presence of a so vast number of domotic standards announces that hardly will be the definitive consecration of one of them. It is as much unlikely that all the coalitions gather about to realize a unique system that represents the standard de facto for the domotics.

Even though many implementations have started to appear in several contexts, few applications have been made available for the home environment and the general public.

The *domotics* constitutes all the technologies that are dedicated at the integration of electronic devices, appliances, systems of communication and control placed inside a domestic environment. The purposes that the domotic applications establish are:

- the increase of the level of comfort: domotic applications must offer a more pleasant space in the house facilitating its management especially for the elderly and diseased consumers (for example, controlling audio/video, monitoring and activating remote appliances, shine, blinds, gates, etc.);
- the attainment of a suitable safety level: the domotic applications have to guarantee the safety of the people to front situations of emergency (for example, anti-intrusion techniques, the arise of fires, the escape of gas, etc.);
- the search of techniques for the energetic saving: the employment of the domotics has to allow a more accurate and efficient management on the level of the energetic consumptions (for example, advanced tools to get information that allow a more equitable distribution of the energetic loads).

It’s sure that the *domotics* proposes futuristic and positive impact sceneries on the quality of our life. Unfortunately there are numerous technologies and different domotic standards that are poorly interoperable among them.

This incompatibility represents an obstacle to integrate distinct systems and so to the expansion of the market. From the point of view of the final consumer is very complicated and perhaps even little comprehensible perceive the necessity to buy domotic technologies instead of traditional devices. In fact, potential beneficiaries who could buy these products would not be able to justify a greater economic effort.

To understand the interoperability issue, let's suppose to have a coffee pot that is able to turn on itself through a radio alarm to a pre-established hour. If the radio alarm and the coffee pot "speak the same language" (have the same standard), every morning you will be able to have a coffee ready as soon as you wake up. If one of the two appliances has to be replaced, it is necessary to replace the device with another one of the same standard.

4 AMBIENT INTELLIGENCE

The *Ambient Intelligence* (*AmI*) vision described by *Weiser* [1] is still far from becoming a reality, though many steps forward have been taken over the last few years. Research activities are continuously proposing new services and algorithms able to provide ever more powerful and sophisticated solutions.

The *Ubiquitous Computing*, *Ambient Intelligence* [2], and *Context-Aware* paradigms paint a picture of future society as one in which humans will be surrounded everywhere and at all times by intelligent interfaces embedded in everyday objects, such as furniture, clothes, vehicles, roads and smart materials. This vision foresees high-capacity connectivity, by which people and objects are able to interact with each other and with the environment. One prerequisite for realizing such a vision is that the environment, in turn, be able to identify and locate users.

According to this vision, the most advanced technologies are those that disappear: at maturity, computer technology should become invisible. All the objects surrounding us must possess sufficient computing capacity to interact with users, the surroundings and each other. The entire physical environment in which users are immersed should thus be a hidden computer system equipped with the appropriate software in order to exhibit intelligent behaviour.

The spaces of the future will surely be a more technologically rich environments, offering tens or even hundreds of pervasive network-based services ranging from simple video entertainment streams to complex home energy-management packages. Some of these services will be provided by physical appliances within the living environments and others by external service providers.

Such services enable anyone to operate a wide range of domestic appliances and control other vital home functions. In recent years attempts to enhance the autonomy of the elderly, the sick and the disabled have begun to exploit easily available technologies. Environmental controls can be considered effective aids to enhance many users' functional capabilities. People with disabilities may be completely unable to physically manipulate objects in their environment. Thus, a domotic environment which may provide simply useful services to most, can, to a disabled user, become a necessity for overcoming personal physical limitations. The environmental system may be the only way for such people to control the world around them. The typical items that can be controlled are: lighting, windows and blinds, doors, loudspeakers, heating, ventilation, air-conditioning, home appliances (refrigerator, washing machine, cooker, etc.), audio-video equipment, burglar and fire alarms, telecommunications.

Many operations control sequences could be automated to save the user the effort of adjusting the system themselves. Thus, a great improvement to such control would come from adding a degree of "intelligence" to the system by enabling it to recognize recurrent activities, unusual or dangerous situations in order to anticipate health problems or special home user needs. Such problems may be addressed by monitoring users' habitual activities, in order to capture and formalize their normal behavior. However, many people have privacy concerns about having their activities monitored, the rules for monitoring implementation and what information is transmitted, and to whom. Such concerns must obviously be addressed through appropriate privacy agreements and safeguards.

One existing solution for modeling the complex interactions occurring between independent but interoperable entities within an intelligent environment is to implement the so-called *Intelligent Agent paradigm* [3], whose features make it particularly appropriate for intelligent management of autonomous, proactive devices and entities able to communicate, interact and coordinate with each other [4].

Given the heterogeneity of these services, it will be up to suitable support middleware to make these services discoverable by and accessible to residential environments.

Focusing on the residential environment, *Ambient Intelligence* may be seen as the layer on top of the domotics, per se. Its aim is to progress from the mere programming of isolated devices, to integrating them in order to achieve global, unified goals. Home networks, that is, the specific Local Area Networks for home environments, include both networks to the living environment (access networks) and networks throughout the living environment.

Following this vision, the domestic network must connect all household appliances and interact with the personal area network and the 'body network' of each person in the environment. Within homes, local networks with potentially different underlying implementations can be combined, or at least managed, as one logical network. The need to support interoperability services such as bridges, gateways or adapters between different networks in the backbone, as well as in access and local networks is an essential prerequisite for all *AmI* applications.

The emerging field of domotics as yet suffers from a number of shortcomings, first amongst which is a lack of definition of application requirements. It has thus far offered a large number of only ad hoc solutions, which unfortunately are often limited in scope and difficult to integrate. In order to make possible the advent of genuine *AmI* applications, it is crucial that standard interoperability protocols be defined and implemented.

5 Evolution of Home Automation Systems

In early domotic systems, the component devices were mainly independent of each other, or sometimes grouped into small independent subsystems. Essentially, the hardware and the software were tightly linked and any changes in one system component would likely require updates to the whole system.

The two main challenges facing designers and developers of such early domotic systems were to:

(i) physically connect devices and subsystems;

(ii) achieve fully modular software systems. Physical connection of domotics devices is an important issue, one major challenge being to build an efficient network while reducing as much as possible the number of devices and cables that have to be added to an existing home or building.

In order to achieve a working domotic system, all the sensors and actuators must be able to 'talk' to each other. Moreover, continuous information exchange must be maintained between the sensors and the system, and any action (opening a door or turning on lights) must be initiated by sensor data, rather than merely having an electric charge activate them.

A number of standard protocols have been created in recent years to enable communication between network devices; a list is shown in Table 1. One major trend in today's domotic systems is the adoption of wireless communication technologies [5].

Standard	Media	Description
Bluetooth	RF	Technology specification for small form factor, low-cost, short range radio links between mobile PCs, mobile phones and other portable devices.
CEBus	All	Developed by the Electronic Industries Association (EIA) to support the interconnection and interoperation of consumer products in a home.
EIA 570A	Twisted	Residential Telecommunications Cabling Standard
HAVI (Home Audio Visual Interope-rability)	IEEE	Consumer Electronics (CE) industry standard that will ensure interoperability between digital audio and video devices from different vendors.
HBS (Home Bus Sy-stem)	Coax Twisted Pair	A consortium of Japanese companies, supported by government agencies and trade associations, encompasses links among appliances, telephones, and audio-video.
Home Plug and Play	All	Provides interoperability among products with multiple transport protocols. Overseen by the CEBus Industry Coun-cil.
HomePlug Al-liance	Power Line	The alliance's objective is to enable and promote rapid availability and adoption of cost effective,interoperable and specification-based home power line networks and products.
IEEE 802.15.4	Wireless	A low data rate solution with multi-month to multi-year battery life and very low complexity. It is intended to operate in an unlicensed, international frequency band.
JINI	All	It provides simple mechanisms which enable devices to plug together to form an impromptu community without any planning, installation, or human intervention.
KNX	All	Link sensors and actuators to building systems that control HVAC, security, access, and life safety. The common asso-ciation of EIB, BatiBUS and HES.
LonMark Interoperability Association	All	The LonMark Association's mission is to enable the easy integration of multi-vendor systems based on LonWorks networks using standard toolsand components.

OSGI (Open Service Gateway Initiative)	All	To create an open standard for a service gateway that is inserted between the external network and the internal network.
UPnP (Universal Plug and Play)	All	Industry group of companies promoting Universal Plug and Play networking protocols and device interoperability standards.
ZigBee	Wireless	Association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard.

Table 1. Main Domotic Alliances and Working Groups.

However, domotic communication standards vary greatly one from the other and in most cases are designed for specific purposes. For example, *ZigBee* and *Z-Wave* are suitable for the sensor systems monitoring a person's health (measuring heart rate, pressure, etc.), but they do not support management of audio and video, as do *HAVI* or *UPnP* (Table 1).

Moreover, some standards have been proposed with the intention of creating home networks, others with the sole purpose of managing them, so existing domotic network protocols can be classified according to the *ISO/OSI* model, which divides systems components according to the particular networking level at which they operate (physical, data link, network, transport and application). Thus, the *KNX* standard, which covers all network levels, is present in each level, while low-level standards, such as *Bluetooth*, are to the contrary found only in the lower levels. An example of such a breakdown is shown in Figure 2.

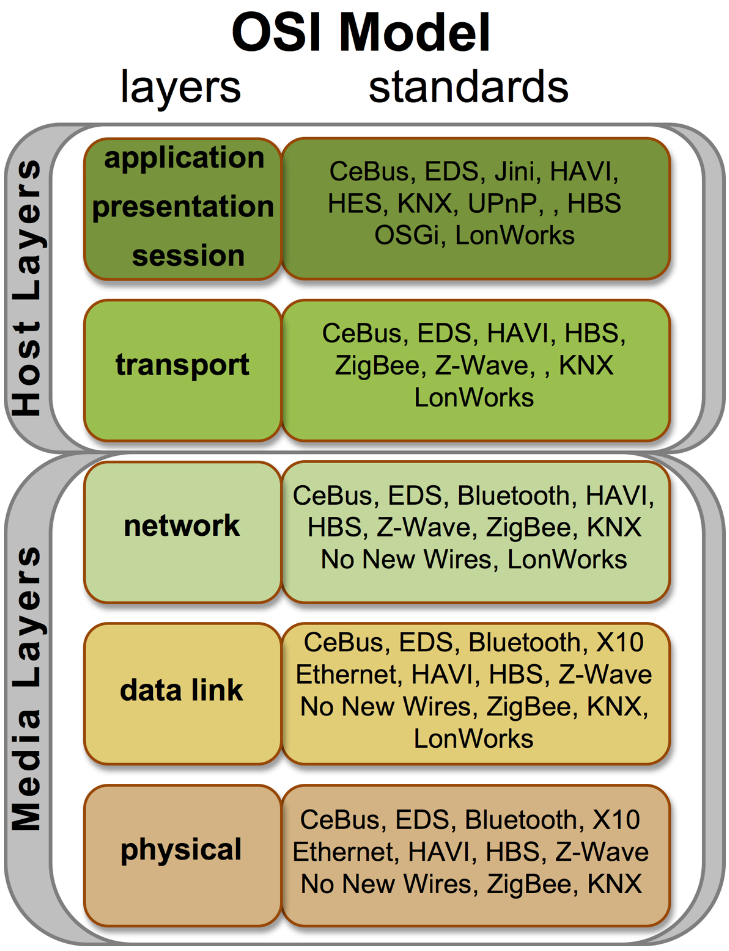

Fig. 2. Classification of domotic standard according to the ISO/OSI architecture.

Today's most advanced domotic systems have been implemented using platforms whose functionalities are provided as '*services*', and the interactions between these services occurs via '*messages*'. The *SmartSMILE* framework, discussed bottom in details, has been designed following this approach.

6 ELDERLY MONITORING SERVICE

The *Elderly Monitoring Service* is a module of the *SOCIALIZE* platform. The core of this service is a framework called *SmartSMILE* (*Socialize MIddLeware for the Elderly*) that aims to fulfill a twofold functionality:

a) to acquire and collect data from sensors placed in environments occupied by the elderly (e.g. homes) as well as from their body area networks (e.g. smartwatches) for analysis and presentation to medical/caregiver staff;

b) to execute complex tasks that enable the elderly to live and work in environments where they feel "protected" and "safe" through awareness of the automatic activation of alarms to alert operators, doctors or family in the event of any emergencies.

One of the main objective of the work is to study and to implement an architecture to face the issue of the cooperation among heterogeneous devices from the technological point of view.

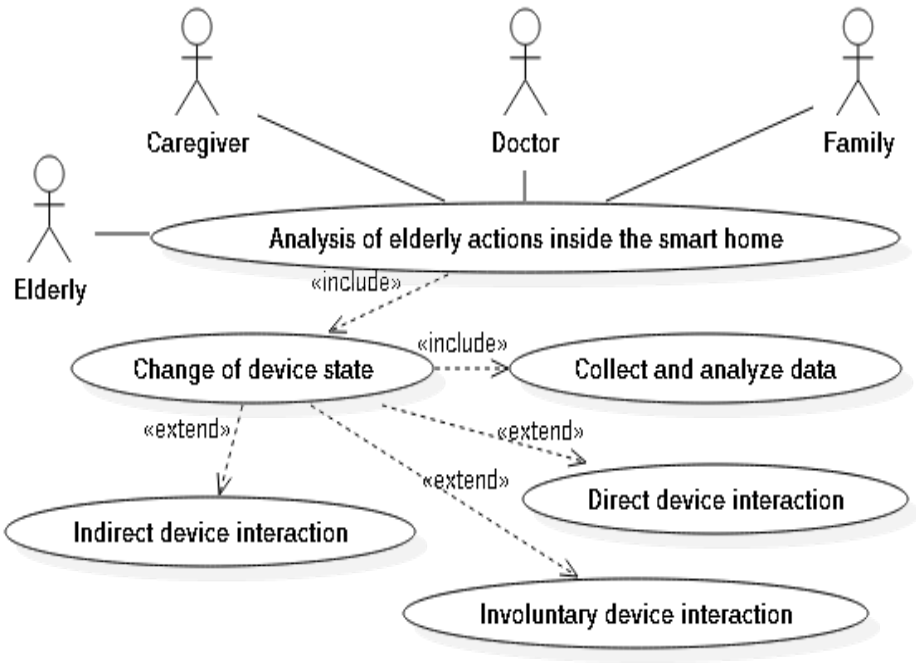

Fig. 3 Elderly monitoring service

As shown in Figure 3, the service analyzes the occupants' habits inside their home environments, which are equipped with smart devices. The occupants need only carry on with their lives at home, conducting their daily affairs as usual. Interactions of the user with the environment can occur in three different ways:

(i) *indirect interactions*, when the user intentionally intervenes in the system using a tablet, smartphone or any web interface;

(ii) *involuntary device interaction*, when the user's behavior activates devices without any explicit intention (e.g., through a PIR - Passive InfraRed sensor device);

(iii) *direct device interaction*, when the occupant acts directly on the system by touching some device (e.g., flipping a switch). The collected data is processed using specialized algorithms to detect 'abnormal' health situations.

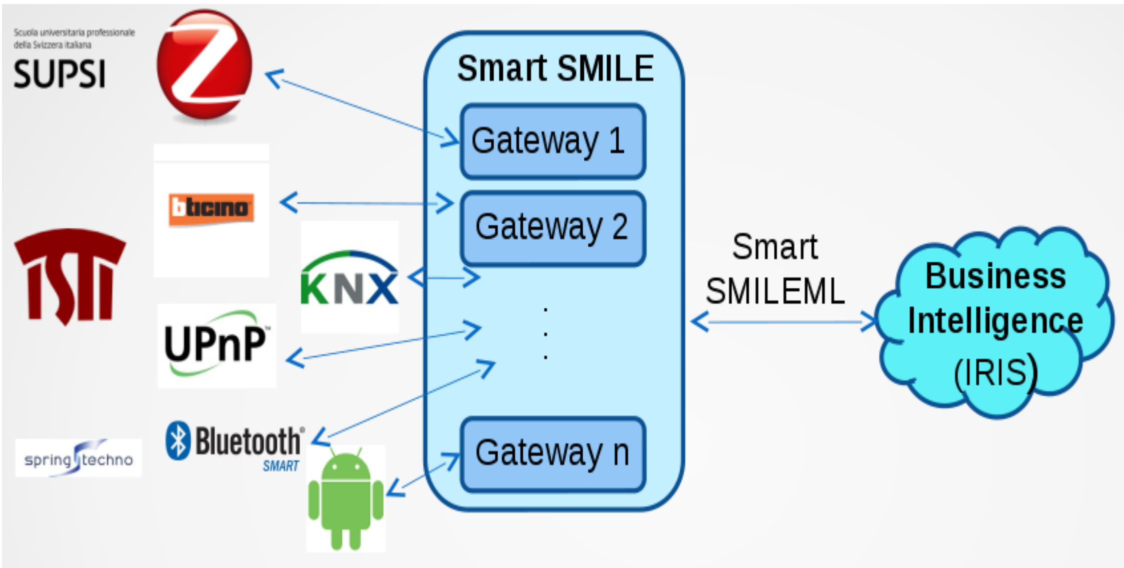

Fig. 4 *SmartSMILE* architecture

SmartSMILE (Figure 4) is made up of a set of sub-modules that work as gateways to interface with specific domotic systems purposely installed inside the living environmets or other service devices. Each gateway serves the function of managing its own set of devices, integrating and interfacing them to the platform.

7 *SMARTSMILE* IN SOCIALIZE

The *Elderly monitoring service* receives the data from the domotic devices collected by the *SmartSMILE* framework. Collected data are stored in the *Interoperability server* database and can be accessed and displayed using a web interface.

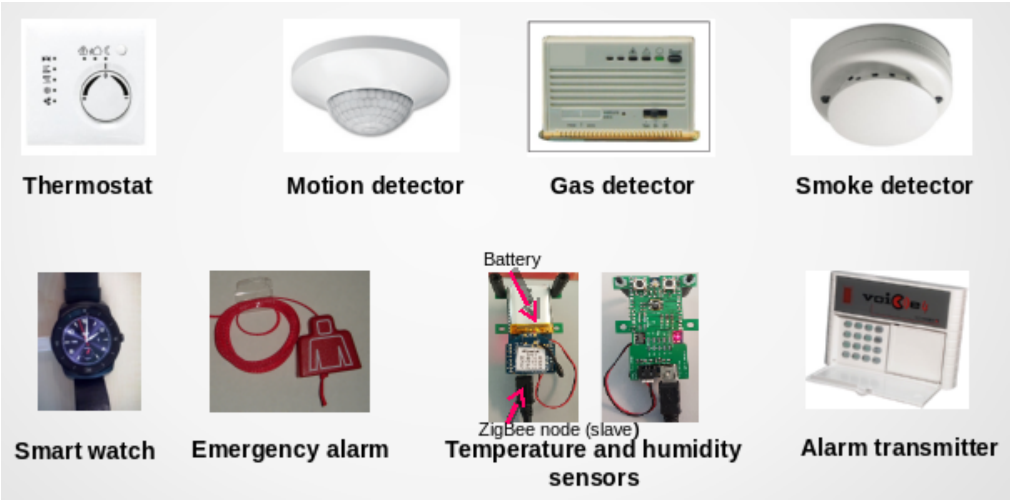

Fig. 5. Some devices integrated into the Elderly Monitoring Service

Some of the devices used in the project are shown in Figure 5. In particular, the SUPSI TTHF partner developed three *ZigBee* sensors to measure current temperature, humidity, and the state of the device's battery.

The project partner Spring Techno GmbH implemented an *Android* app to communicate the data from a *Bluetooth* smartwatch. The smartwatch is able to monitor heart rate, the number steps and distance walked daily, and the duration of sleep during rest times.

ISTI-CNR interfaced *KNX* devices such as a thermostat, a motion detector, gas, water and smoke detectors, an automatic phone dialer, and an emergency alarm.

Data coming from the sensors are converted by their respective *SmartSMILE* gateway into *SmartSMILE* Messages and sent to the *Business Intelligence*, hosted by the IRIS Consortium, for analysis and storage in a database.

7.1 THE SMART SMILE ARCHITECTURE

This is possible creating a level of abstraction that allows to characterize the domotic devices from a behavioural and physical point of view. The abstraction layer aims to furnish a homogeneous description of all devices independently from the belonging technologies.

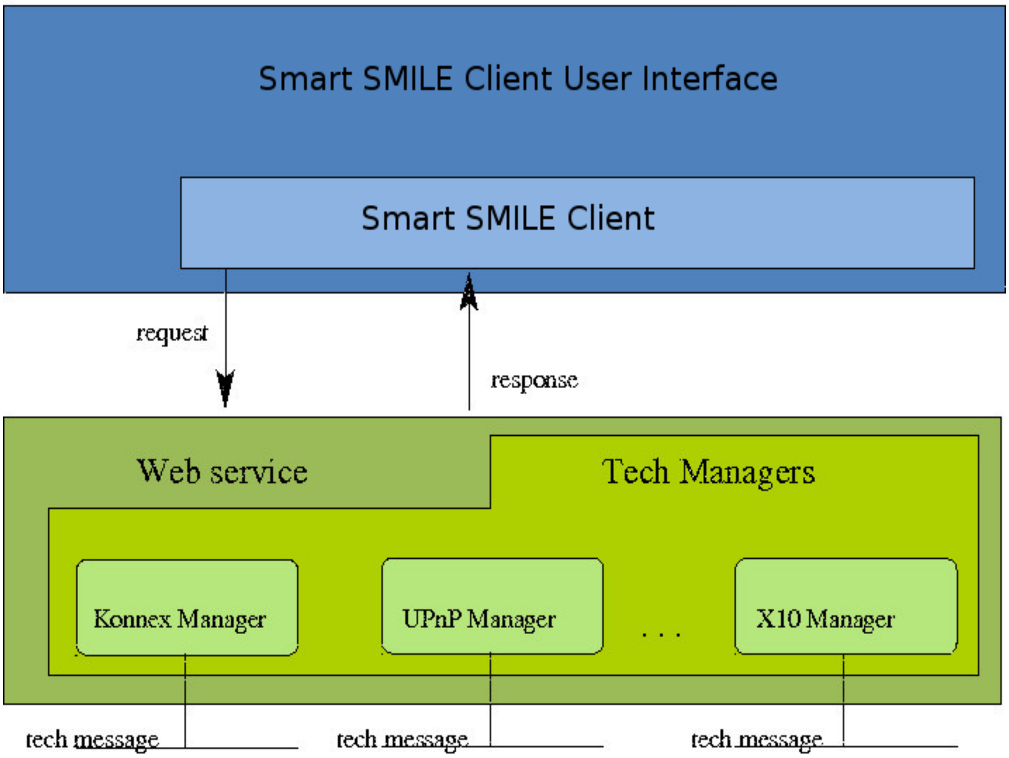

Fig. 6. *SmartSMILE* functioning

SmartSMILE exploits *W3C-recommended standard Web technologies*. It exposes services using *Web Services*, *SOA* and *XML* technologies. The advantage to using *W3C* standard solutions is that they ensure that the developed applications are fully compatible with other standards-based software and are not tied to any particular software system, programming language or computer architecture.

Moreover, *SmartSMILE* permits exposing its devices externally. Exploiting the results of the work described in [6], every device within the system is reachable using an *IPv6* address, and socialization can be achieved between devices belonging to different *SmartSMILE* instances.

The *Smart SMILE architecture* is composed by a server part that implements the management of domotic devices like the abstraction layer and the cooperation feature, and by one or more clients able to send requests to the server in order to activate domotic actuators and to query domotic sensors receiving a response from them. Client functioning can range from a simple remote control to a more sophisticated tool able to take decisions by evaluating contextual and environmental parameters to implement intelligent and specialized services (e.g. related to Ambient Intelligence, e-health, energy saving).

In Figure 6, the client is composed by two elements:

1) the *SmartSMILE Client* that implements the business logic and the methods for interactions with the sever part, integrated with the *SmartSMILE Client User Interface* that implements the user interface suitable for a Web or desktop application or as an app for smartphone;.

2) the *SmartSMILE Server* that is composed by the communication interfaces and business logic (*webService)* and by one or more *techManagers*. *techManagers* are special gateways able to interface domotic technologies (like *KNX*, *UPnP*, *X10, ZigBee* and so on) and to integrate them in the middleware.

7.2 DICTIONARY

- *device*: domotic physical device belonging to a specific technology;
- *real address*: the address of the device that is used inside the technology of the domotic device;
- *domoAddress*: a couple composed by the *URL* of the *web service* that manges the *domoDevice* and an *id* that identify the *domoDevice* inside the middleware;
- *domoDevice*: a *SmartSMILE* device described using *SmartSmileML* language;
- *service*: a service (or function) offered by a *device* and described using *SmartSmileML* language;
- *domoMessage*: an interaction described using *SmartSmileML* formalism;
- *SmartSmileML*: high level *XML* based language used to describe and to abstract devices, functions and interactions inside the *SmartSMILE* middleware;
- *techLanguage*: each domotic technology is represented and interfaced using an appropriate *techManager*. *techLanguage* is the language that devices belonging to a specific domotic technology use and that is used in the *techManager* to communicate with its belonging *device*s.

7.3 SMARTSMILEML

SmartSmileML is a subset of the *XML* language and it used as *lingua franca* abstracting the underlying technologies of devices in a way to level out them. In this way devices, functions and interactions inside the *SmartSMILE* middleware are represented using a unique representation. It's the standardized language by and towards which translate *techLanguages* for the proper functioning of the middleware. Infact, all the logical parts of the middeware use *SmartSmileML* and only the physical iterations between real devices and *techManagers* are in the single *techLanguage*.

By exploiting *SmartSMILEML*, *SmartSMILE* is able to achieve a suitable environment for putting into action:

(i) the ability of devices to navigate throughout the network: even devices with no networking capabilities are able to communicate using Internet protocols, thereby permitting full interconnection and interaction amongst them;

(ii) service-to-service communications: the service provided via a device can share its input and output with the services of other devices to reach a common objective;

(iii) scalability: in order to reach a certain goal, devices can collaborate with each other; moreover, when changes occur (new devices become available or some are missing), tasks can be re-distributed.

SmartSmileML is composed by two sub-languages:

1) *domoDevice*: describes and creates the abstraction of the devices;

2) *domoMessage*: describes iterations from and to devices.

7.4 DOMODEVICE

The *domoDevice* sub-language has the purpose to create a compact, effective, simple and complete way to represent devices abstracting them from the underlining domotic technology.

The *domoDevice* can be represented through a wide and poorly deeper tree. The tree can have maximum four levels:

device: opens the *domoDevice* description and gives general informations about the device as follow:

- *description*: a natural language description for the device;
- *id*: identifies the *domoDevice* inside the *Web Service*;
- *manufacturer*: the manufacturer of the device;
- *positionDescription*: a natural language description of the location of

the device.

- *serialNumber*: the serial number (it can be composed by letters too) of the device.
- *tech*: the technology device represented;
- *type*: the typology of the device;
- *URL*: identifies the *web service* that exposes the *domoDevice*.

All this fields are optional except for the *id*, *URL* and *tech* attributes because they permit to identify and use the device.

service: describes a service or function offered of the *domoDevice*. This information is used to create a *domoMessage* and to converting it to a *techMessage*. For each *domoDevice* there can be more than one *service* tags. The attributes for this tag are:

- *description*: a natural language description for the *service*;
- *name*: an identifier that can be useful when a *domoMessage* the will be generated and translated to *techMessage*;
- *output*: the datatype expected as return value when the service is invoked. This attribute must not be placed if no return value is provided.
- *outputDescription*: a natural language description for the *output* attribute. If no *output* attribute is provided this attribute must not be placed;
- *prettyName*: a label for the service using natural language;

two possible tags for this level (*service*):

input: describes an input for the service. This tag is optional and can be repeated as many times as needed. The attributes for this tag are:

- *name*: identifies the input;
- *description*: a description of the input;
- *type*: the datatype of the expected input;

linkedService: describes an interoperability action. This tag permits to invoke a *service* of another *domoDevice* sharing states. This tag is optional. Its attributes are:

- *URL*: the url of the *web service* that manages the involved *domoDevice that owns the* service *to call*;
- *id*: the identifier inside the specified *web service* of the involved *domoDevice that owns the* service *to call*;
- *service*: the name of the *service* to call;
- *ifInput*: condition to be verified by the input of the service described. If the input name has value

as specified in *hasValue* attribute, the *service* of the involved *domoDevice* is called;

- *hasValue*: used with *ifInput* attribute. It specifies the value to be matched in order to call the *service* of the *linkedService;*

two possible tags for this level (*linkedService)*:

- *allowed*: if the previous tag was *input*. It represents a possible value that can assume the input. This is an optional parameter and each *input* may have more than one *allowed* tags. The only attribute for this tag is:
  - *value*: the allowed value.

- *linkedInput*: if the previous tag was *linkedService*. It permits to share the value of an input of the described service with an input of the service to call. Attributes for this tag are:
  - *from*: the name of the input of the described service from which take the value;
  - *to*: the name of the input of the service to call;
  - *value*: used alternatively to the *from* attribute, it specifies the value to be set for the input named in the *to* attribute.

An example of a *domoDevice* for a lamp in shown in Figure 7.

In this example is described a *domoDevice* lamp produced by the brand “pholips” and that it is located on the table beside the bed. It is belonging of the *KNX* technology and is of type *lamp*.

The lamp is managed by the *web service*: [*http://www.thiswebservice.it/service*](http://www.thiswebservice.it/service).

The lamp offers two services: to get and to set its state using a boolean value. The boolean value can be *TRUE* or *FALSE*.

When is invoked the service to set status, it is also called the service named *setPower* of the *domoDevice* managed from the *web service: http://www.otherwebservice.it/service* with *id: 3*.

The value used as input for the described service (*status*) is also used as input value for the service to call (*power*). This means that if it is invoked the service *setStatus* of the *domoDevice* with *url*: *http://www.thiswebservice.it/service* and with *id: 3,* using as input the value *TRUE*, it is also called the service *setPower* of the *domoDevice* with *url*: *http://www.otherwebservice.it/service*, and *id*: *3* using as input the same value.

```
<device description="energetic saving lamp" id="0" manufacturer="pholips"
 postionDescription="on the bedside table beside the bed"
 serialNumber="xxxxxxxxx" tech="KNX" type="lamp"
 url="http://www.thiswebservice.it/service">
 <service description="Get the status" output="BOOLEAN"
  outputDescription="The value" name="GET_STATUS" prettyName="Get status" />
 <service description="Set the status" name="SET_STATUS"
  prettyName="Set status">
  <input description="The value" name="status" type="BOOLEAN">
   <allowed value="TRUE" />
   <allowed value="FALSE" />
  </input>
  <linkedService id="3" service="setPower"
   url="http://www.otherwebservice.it/service">
   <linkedInput from="status" to="power" />
  </linkedService>
 </service>
</device>
```

Fig. 7. Example of a description for a lamp

7.5 DOMOMESSAGE

The *domoMessage* sub-language as the role to create a simple, compact and effective mechanism to represent the interactions between the *domoDevice* without taking care of the underlying technologies. *domoMessage* is also represented through a wide and little deep tree with maximum 2 levels:

1. *message*: it opens the tag that it describes the *domoMessage* and it contains the following attributes:
   - *message*: the body of the message, typically the name of the service to call;
   - *messageType*: the type of message. It can be:
     - *COMMAND*: if it is a service to execute;
     - *SUCCESS*: it is used as response of a *COMMAND* message. It reports

the success of the call. In this case the attribute *message* can contain value of the answer of the execution of the service;
  - *FAILURE*: it is used as response of a *COMMAND* message. It reports the failure of the call. In this case the attribute *message* can contain error code;
  - *UPDATE*: it's a notification of the change of the state of a value of an *input* of a *service* of a *domoDevice*.
- *receiverId*: the identifier of the receiving *domoDevice* of the message;
- *receiverURL*: the *URL* of the *web service* that contains the *domoDevice* requested;
- *senderId*: the identifier of the sender *domoDevice* of the message;
- *senderURL*: the *URL* of the web service that contains the *domoDevice* that sends the message;

2. *input*: if message type is *COMMAND*, it represents a value to be sent as input of the service to call. Otherwise, if message type is *UPDATE*, it represents a new value of the state of a device. This tag is optional and every tag *message* can have more *input* tag. The attributes for this tag are:
   - *name*: the name of the *input*;
   - *type*: the data type of the *input*;
   - *value*: the value of the *input*;

An example of *domoMessage* in order to turn on a lamp is in Figure 8.

```
<message message="SET_STATUS" messageType="COMMAND"
 receiverId="1" receiverURL="http://www.otherwebservice.it/service"
 senderId="0" senderURL="http://www.thiswebservice.it/service">
 <input name="status" type="BOOLEAN" value="TRUE" />
</message>
```

Fig. 8. Example of message of type COMMAND

This example shows how to execute the service *SET_STATUS* on the *domoDevice* with *web service:* http://www.thiswebservice.it/service and *id*: *1.* The service has an input of type *boolean* with value *TRUE*. After the execution of the called service, if no errors was found, the middleware creates a message of type *SUCCESS* (Figure 9):

```
<message message="TRUE" messageType="SUCCESS"
 senderId="1" senderURL="http://www.otherwebservice.it/service"
 receiverId="0" receiverURL="http://www.thiswebservice.it/service" />
```

Fig. 9. Example of message of type SUCCESS

8 Server Side

8.1 Smart SMILEWS

At startup time, the server (Figure 10) sets up the *techManagers* that are in the *techManagerList* data structure and it requests them a list of the *domoDevices* that are currently avaiable on them. In fact, each *techManager* works as gateway for a specific domotic technology. *TechManager* has the duty to manage and interact physically with the domotic technology and with its own devices. When a *techManager* recognizes a device of its competence, it creates the corresponding *domoDevice.* In this way, the device is described in terms of functionalities and characteristics and notified in Smart*SMILEML* formalism to *SmartSMILEWS. SmartSMILEWS* then includes the *DomoDevice* description in the *DomoDeviceList* structure assigning to it a *domoAddress*.

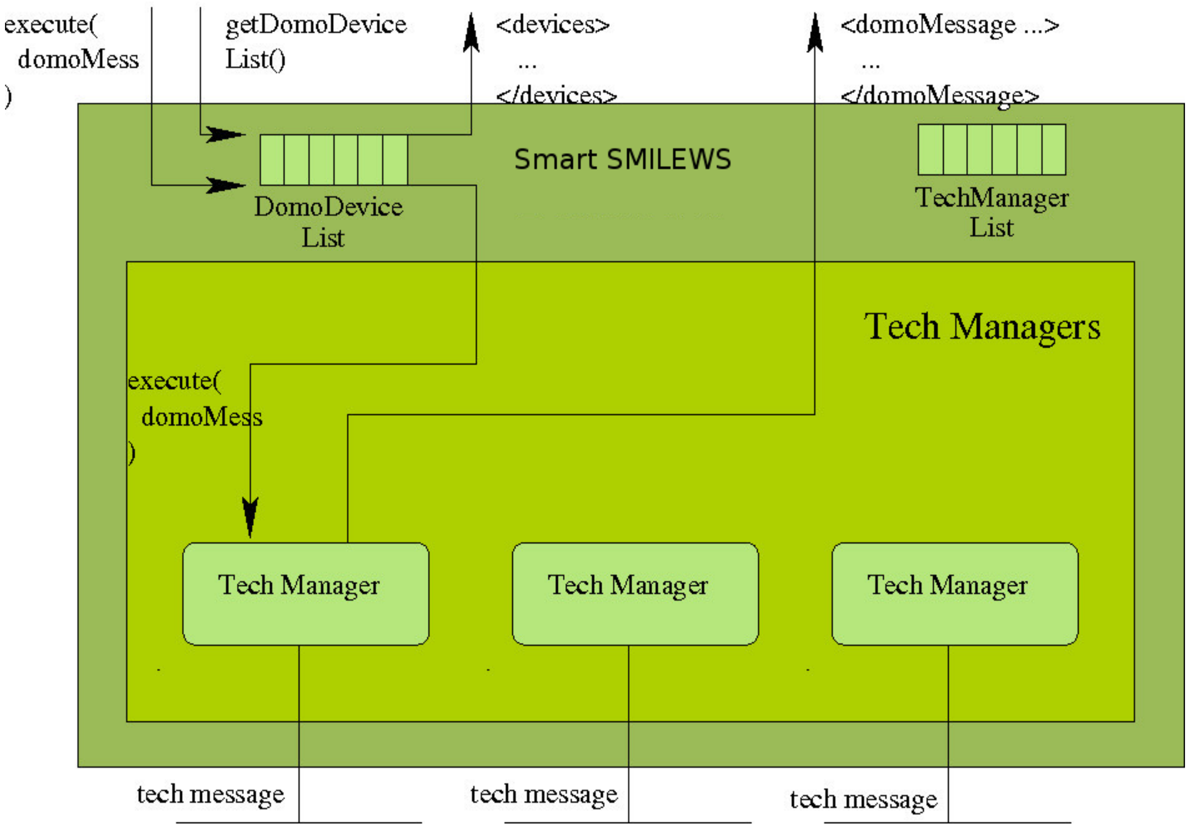

Fig. 10. SmartSMILEWS (server side) architecture

8.1.1 THE EXECUTION OF A DOMOMESSAGE FOR THE SMART SMILEWS

The request to execute a *domoMessage* can be originated by a *Smart SMILE client* (Figure 6) or by another instance of *SmartSMILEWS* placed somewhere on Internet (Figure 12).

When *SmartSMILEWS* receives a request to execute a *domoMessage* (Figure 10), it identifies the *domoDevice* involved in the operation. The identification of a *domoDevice* is realized throught its *DomoDeviceId* that is obtained combining the *receiverURL* and *receiverId* attributes of the *domoMessage*.

Using the *domoDevice* description, the server finds the *techManager* that manages the corresponding *device* of the *domoDevice* using the attribute named *tech*.

The *domoMessage* is then tranfered to the found *techManager* to be translated in *techMessage* to provide to its execution.

Once the *domoMessage* is executed, the *techManager* prepares a new *domoMessage* of type *failure* or *success*, in the outcome of the transaction.

8.2 TECHMANAGER

A *techManager* is a gateway used to physically interface the system with a specific domotic technology such as *KNX*, *UPnP*, *X10, ZigBee, etc*. To do that, the *techManager* interfaces directly the specific wired or wireless domotic bus of the technology to integrate, to capture and to send data packets to domotic devices.

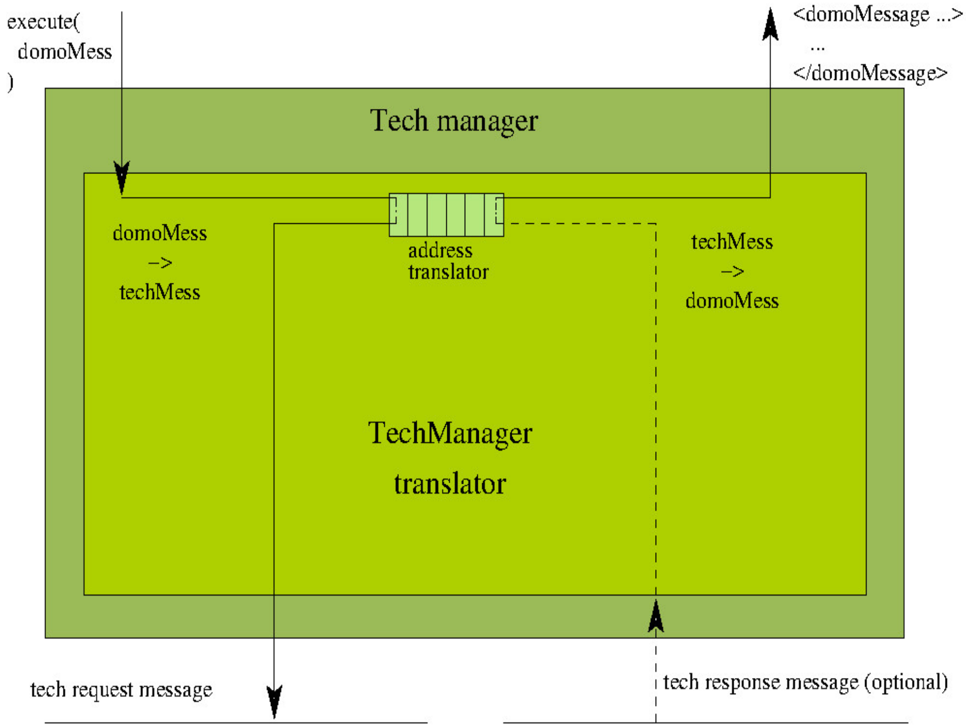

Fig. 11. techManager architecture

techManager main functionalities are:

1) to create *domoDevices* for the *devices* that are available in the domotic bus of competence of the *techManager*;
2) to translate a *domoAddress* in a *real address* and vice versa;
3) to translate a *domoMessage* in a *techMessage* and vice versa;
4) to write and read data to and from the domotic devices using, when available, the domotic bus.

8.2.1 Executing a domoMessage

When a *techManager* receives a *domoMessage* to be executed (Figure 11), it must find the *realAddress* of the *device* and the *domoDevice* that are involved in the communication.

To do that, the *techManager* exploits the field *receiverId* of the received *domoMessage* twice: once as input parameter for the *address translator* functionality defined in the *techManager* in order to find the *real address* and of the *device,* and the other used as *domoDeviceid* to get the corresponding *domoDevice*.

For each *domoDeviceId* it's possible to associate more than one *real address* (that is useful for some domotic technology such as *KNX*) but for each *real address* it's possible to associate only one *domoDeviceId*.

The *message* field of the received *domoMessage* contains the name of the service of the *domoDevice* to invoke. Moreover, in the *domoMessage* are eventually described input and output parameters with the corresponding datatypes.

According with the characteristics of the domotic system managed by the *techManager*, all these information contained in the *domoMessage* are translated into a *techMessage* and sent to the *device* through the domotic bus.

If the *techMessage* sent to the device requires a response from it, the reply will be captued from the domotic bus and translated to *domoMessage*. The *domoMessage* containing the response will be forwarded to who requested the information. In any case, a *domoMessage* of type *SUCCESS* or *FAILURE* is emitted by *techManager* to inform about the outcome of the operation.

8.3 THE COOPERATION BETWEEN TECHMANAGERS FOR THE DOMOTIC INTEROPERABILITY

To reach the interoperability objective, the middleware exploits the cooperation between *techManager*. Involved *techManager*s can belong to the same (Figure 10) or to different *SmartSMILEWS*s (Figure 12). In fact, every *SmartSMILEWS* can import devices belonging to different *SmartSMILEWSs*. Imported devices are recognized using the *url* field of their *domoDevice* descriptions that identifies their belonging *SmartSMILEWS*.

Each *techManager* is able on the one hand to "talk" the domotic technology language and the other the *SmartSMILEML* formalism. Being *techManager*s able to translate from one language to the other language, to act cooperation it is sufficient to exploit *SmartSMILEML* facilities.

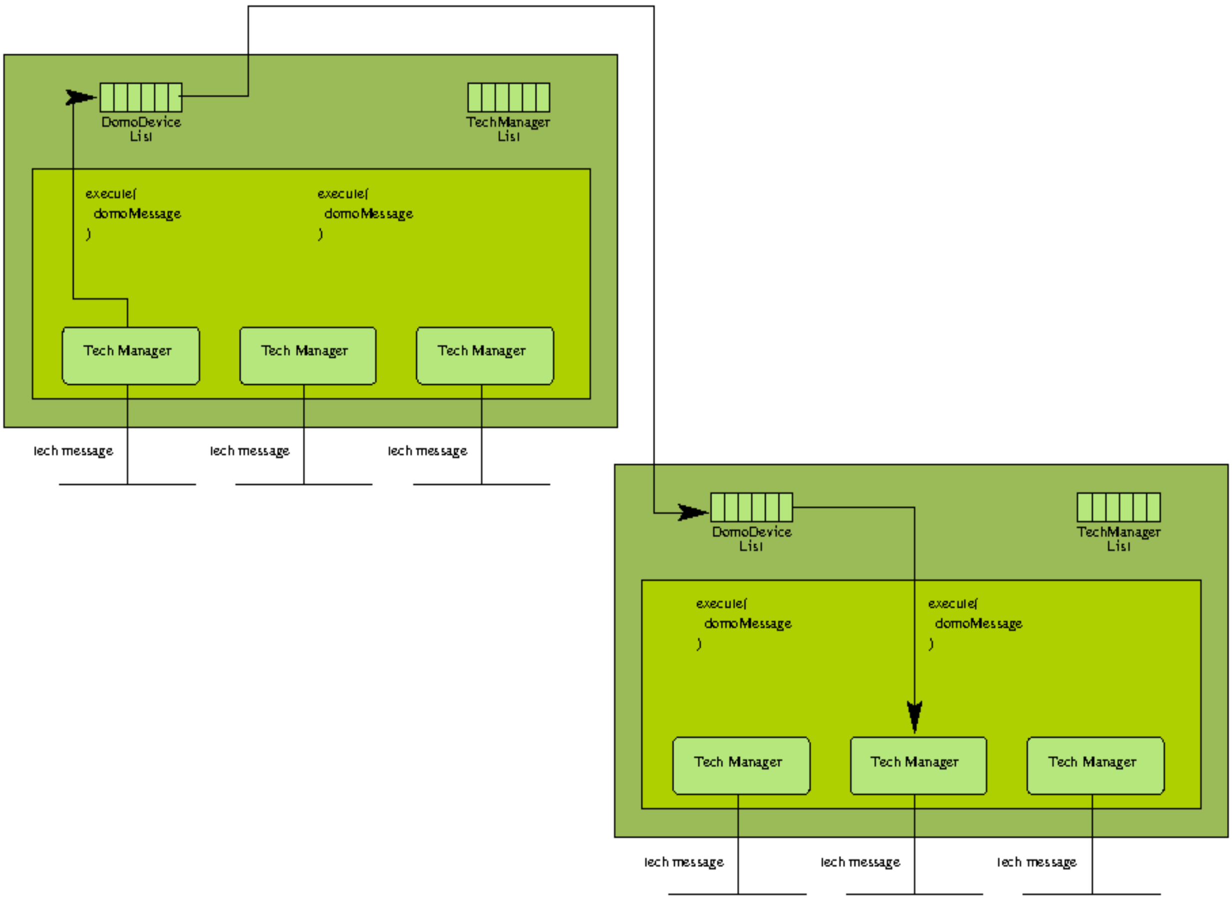

Fig. 12. Two distinct instances of SmartSMILEWS interactions

In Figure 10 and in Figure 12, a *techManager* receives from the *SmartSMILEWS* a request to execute a *domoMessage.*

Analyzing the *domoMessage*, the *techManager* is able to find the correct *device* and *service* that are related with the request.

With the acquired information, the *techManager* is able to translate the *domoMessage* in *techMessage* and to send it in the domotic bus for its execution.

After the execution of the *techMessage* and, after the eventually response by the involved device, the *techManager* sends the result to *SmartSMILEWS* that verifies the existence of the *linkedService* tag in the description of the invoked service in the *domoMessage*.

If *SmartSMILEWS* finds at least a *linkedService* tag, it generates and sends new *domoMessages* to the corresponding *techManager*s for their exectution.

9 CLIENT SIDE

The *SmartSMILE* Client side is composed by two elements: the *SmartSMILE Client* and the *SmartSMILE Client User Interface.*

Following the *MVC* (*Model View Controller*) paradigm, *SmartSMILE Client* implements the Model and Controller part whereas *SmartSMILE Client User Interface* implements the View part.

The *SmartSMILE C*lient is an application to interact with one or more *SmartSmileWSs*. It implements the necessary functions to manage and to get data to implement services to control devices.

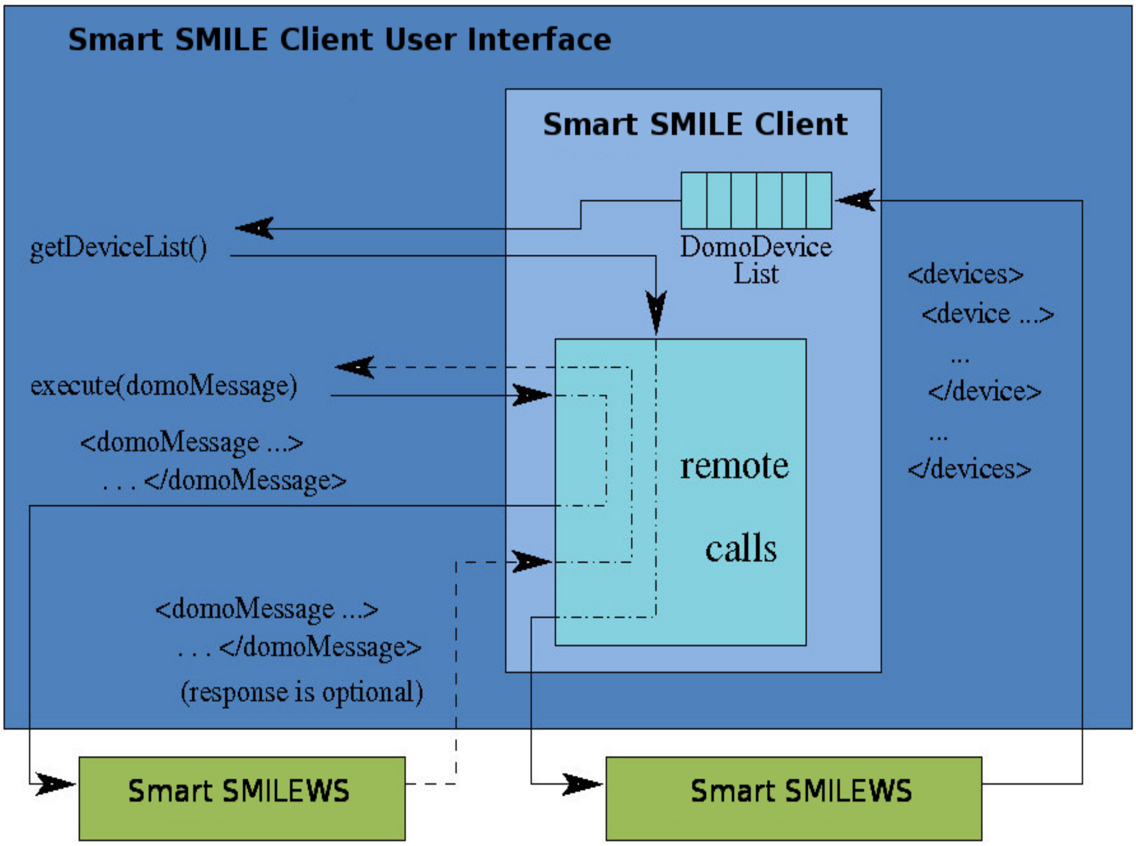

Fig. 13. SmartSMILE Client Architecture

*SmartSMILE C*lient (Figure 13) can get the list of *domoDevice*s that are available in the *SmartSmileWSs* invoking the *getDeviceList* function. *DomoDevice*s are stored in the

DomoDeviceList structure. *SmartSMILE Client* is able to create and to send *domoMessage* commands exploiting the descriptions provided in *domoDevice*s. To do that, *SmartSMILE Client* uses the *execute* function.

When the *output* field of the *domoMessage* is not empty, *SmartSMILE Client* waits for a response form the *Smart SMILEWS* after having submitted the request.

If needed, *SmartSMILE Client* is able to receive notifications about the change of the state of sensors and actuators that occur in the *SmartSMILEWS*.

Instead, *SmartSMILEClient User Interface* can be a web or desktop application, or an app for smartphone or tablet. Its function is to offer to users an interface to interact easily with the *SmartSMILE Client* application.

9.1 BIBLIOGRAPHY

1. Weiser, M. The computer for the twenty-first century. Sci. Am. 1991, 3, 94-104.
2. José R.; Rodrigues H.; Otero N. Ambient Intelligence: Beyond the Inspiring Vision. J of Univers. Comput Sci 2010, 16, 1480-1499.
3. Isern, D.; Sánchez, D; Moreno, A. Agents Applied in Health Care: A Review. Int. J. Med. Inf. 2010, 75(3), 145-166.
4. Ali, R.; Abdel-Naby, S.; Maña, A. Muñoz, A.; Giorgini, P. Agent Oriented AmI Engineering, Springer Paris, 2008.
5. I Research's Home Automation and Monitoring study. http://www.abiresearch.com/research/1005784-Home+Automation+and+Monitoring (accessed on 18 April 2012).
6. Miori V., Russo D. Domotic evolution towards the IoT. In: WAINA 2014 - 28th IEEE International Conference on Advanced Information Networking and Applications (Victoria, Canada, 13-16 May 2014). Proceedings, pp. 809 - 814. IEEE, 2014.