



PersonAAL



Deliverable 1.5.a

Integrated Platform

Responsible Unit: Reply

Contributors: CNR, FCID, PLUX, USI

Document Technical Details:

Document Number	D1.5.a
Document Title	Integrated Platform
Version	1.0
Status	Final
Work Package	WP1
Deliverable Type	Prototype
Contractual Date of delivery	30/9/2017
Actual Date of Delivery	9/10/2017
Responsible Unit	REPLY
Contributors	CNR, FCID, PLUX, USI
Reviewers	Roy Mitchley (IBM)
Dissemination Level	Public

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Document Change Log:

Version	Date	Status	Author	Description
0.1	1/5/2017	Draft	C.Chesta, REPLY	Initial Structure
0.2	10/5/2017	Draft	M.Manca, CNR	Working document on Context REST services
0.3	31/5/2017	Draft	M.Manca, CNR	Working document on Communication Protocol between web application and Adaptation Engine
0.4	10/8/2017	Draft	C.Azevedo, PLUX	Input for the Sensor Chestband and Context Delegate sections
0.5	29/8/2017	Draft	C.Duarte, FCID	Input for the Persuasion Module section
0.6	12/9/2017	Draft	J.Coelho, FCID	Input for the Social Module section
0.7	13/9/2017	Draft	I.Elhart, USI	Input for the Authentication Server section
0.8	22/9/2017	Draft	C.Chesta, REPLY	Draft ready for review
0.9	28/9/2017	Draft	R.Mitchley, IBM	Review
1.0	9/10/2017	Final	C.Chesta, REPLY	Final version

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Contents

1	INTRODUCTION	5
2	PLATFORM OVERVIEW	6
3	COMPONENTS AND INTERFACES	7
3.1	PERSONALIZATION RULE EDITOR	7
3.1.1	Authoring Tool REST Services.....	7
3.2	ADAPTATION ENGINE.....	8
3.2.1	Application Subscription.....	8
3.2.2	Event Subscription	8
3.2.3	Event Notification	9
3.2.4	Application Adaptation.....	9
3.3	CONTEXT MANAGER	9
3.3.1	Environment	9
3.3.2	Physiological attributes.....	10
3.3.3	Devices	11
3.3.4	Physical Objects	12
3.3.5	Personal Data.....	14
3.3.6	Weather	14
3.3.7	Motivation.....	15
3.3.8	Medication Planned and Medication Occurred.....	16
3.4	BEHAVIOR ANALYSIS MODULE	17
3.5	AUTHENTICATION SERVER.....	17
3.5.1	Authentication API: signup.....	18
3.5.2	Authentication API: change_password	18
3.5.3	Authentication API: login	19
3.5.4	Authentication API: logout	19
3.5.5	Authentication API: get_access_token	19
3.5.6	Authentication API: refresh_access_token.....	20
3.5.7	Authentication API: revoke_access_token	20
3.5.8	Authentication API: get_user_info.....	20
3.6	SENSOR CHESTBAND.....	21
3.7	CONTEXT DELEGATE.....	24
3.8	PERSUASION MODULE.....	25
3.9	SOCIAL MODULE.....	26
4	INTEGRATION PLAN.....	27
4.1	INTEGRATED SCENARIO REALIZED AT M18	27
4.2	INTEGRATED SCENARIO PLANNED FOR M36	28
5	CONCLUSIONS	31
6	REFERENCES	32

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



1 INTRODUCTION

The Work Package 1 focuses on design, development and integration of the functional components of the platform.

This deliverable contributes to the Task 1.5, is an accompanying document to the initial Integrated Platform, and describes the design of the interfaces for the system components and the integration tasks.

The document is organized as follows:

- Section 2 provides an overview of the platform and of communication between modules.
- Section 3 includes a description of the functionalities provided by each component and the interfaces between components.
- Section 4 reports the integration scenario used for the M18 demonstrator and details the envisaged future steps.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

2 PLATFORM OVERVIEW

Figure 1 shows the architecture of the personalization platform along with the communications between its modules. More details are provided in [1].

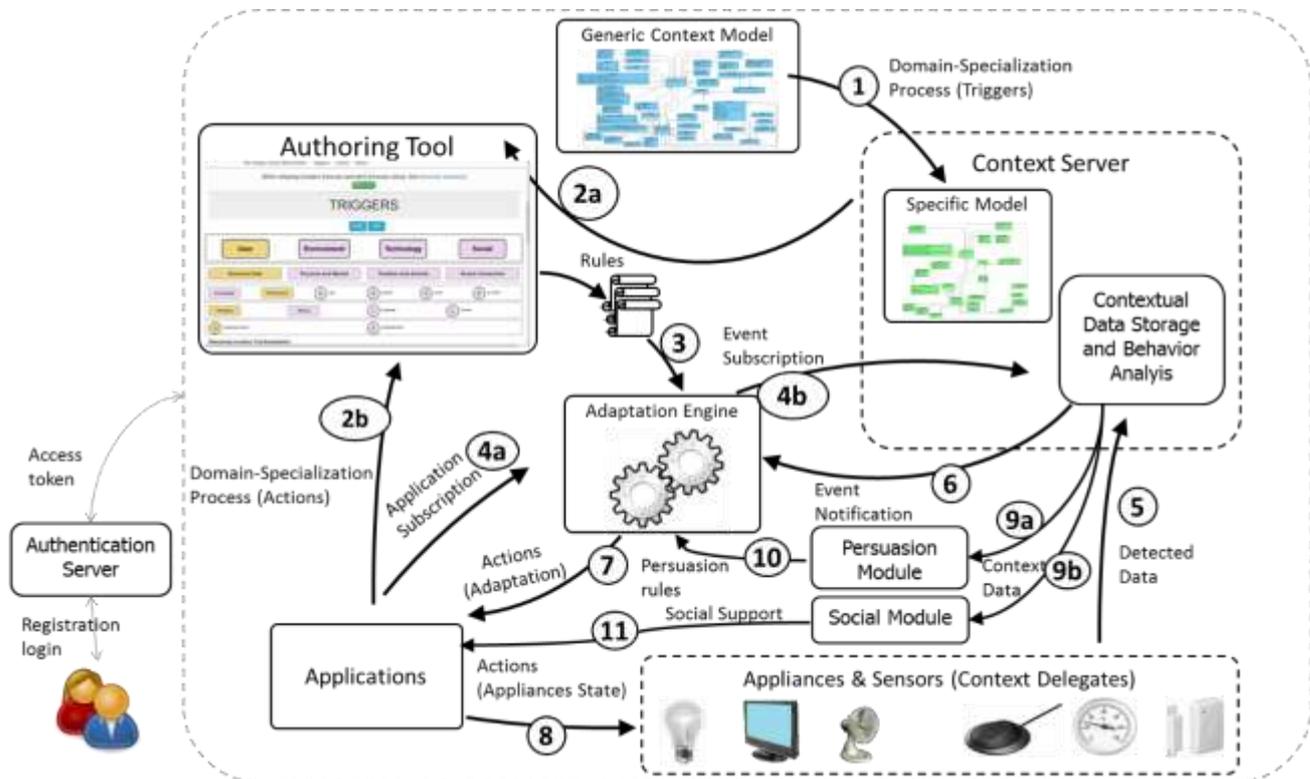
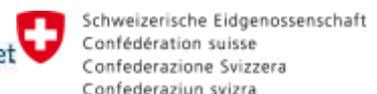


Figure 1 - PersonAAL platform architecture

In the following for each component, a brief description of the functionalities and of the interfaces exposed is provided.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



3 COMPONENTS AND INTERFACES

3.1 Personalization Rule Editor

To enable the personalization of both applications and persuasion mechanisms, we provide caregivers (and technological expert elderly) with an intuitive Web authoring environment where they can define and refine rules (which will be provided to the Adaptation module to manage the adaptations in the platform).

To create and refine rules, users can start either from triggers or actions. Regarding the former, selection is performed by navigating in the hierarchy of concepts associated with each contextual dimension until a basic element is reached. When this element is chosen, the tool shows the possible attributes and relevant values to build the trigger of the concerned rule. In a similar way, when selecting an action, the tool shows the corresponding supported options. Triggers refer to elements identified in the contextual domain-specific model and at the highest level consider: i) user characteristics, ii) environment aspects, iii) technology, and iv) social aspects. Actions involve appliances, UI modifications, UI distribution, functionalities, alarms and reminders.

The rules are expressed through an ECA-based (Event, Condition, Action) format; where events are changes of context state, conditions are Boolean predicates referring to context state (they are optional) and actions are changes in the interactive application, in the state of the appliance or they may activate some functionalities.

The language used to describe rules is presented in [1], together with a detailed description of the Authoring Tool.

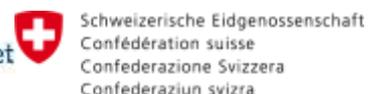
3.1.1 Authoring Tool REST Services

The Authoring Tool offers REST Services to save and delete rules. When a rule is saved, its XML description is sent via HTTP(S) POST to the Adaptation Engine (3).

Save Rule

Method	POST
Endpoint	<code>http://giove.isti.cnr.it:8880/NewAdaptationEngine/rest/saveRule/appName/{appName}/userName/{userName}</code>
Parameters	<p>appName =[alphanumeric]: the application name associated to the rule</p> <p>userName =[alphanumeric]: the user name associated to the rule</p>
Response	<p>JSON object with two fields:</p> <ul style="list-style-type: none"> • status (SAVED, NOT_SAVED) • msg

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Delete Rule

Method	GET
Endpoint	http://giove.isti.cnr.it:8880/NewAdaptationEngine/rest/deleteRule/appName/{appName}/userName/{userName}/ruleName/{ruleName}
Parameters	<p>appName =[alphanumeric]: the application name associated to the rule</p> <p>userName =[alphanumeric]: the user name associated to the rule</p>
Response	<p>JSON object with two fields:</p> <ul style="list-style-type: none"> status (DELETED, NOT_DELETED, ERROR) msg

3.2 Adaptation Engine

The Adaptation Engine enables Web applications (and the system) to have adaptive behavior (changing in accordance to relevant events occurring in the elderly's context, needs, requirements, (dis)abilities, etc.). It is responsible for deciding the best combination of modalities to render messages to the user. It receives rules specified by both previous modules and the caregivers' (through the personalization rule editor) and communicates with the applications and the context manager.

The application on the user device, when loaded, subscribes for updates by sending a request to the Adaptation Engine via HTTP(S) POST. This is done by the scripts previously injected into the HTML. The Adaptation Engine extracts the rules content and subscribes to the Context Manager for the specified events, conditions.

3.2.1 Application Subscription

The application, when is loaded, subscribes for updates by sending a request to the Adaptation Engine (4a) via HTTP(S) POST; the subscription parameters are: application name, user name and action format (JSON or XML). This can be done in two ways: Web Socket communication or REST communication. More details on the communication mechanisms are reported in [1].

Method	POST
Endpoint	http://giove.isti.cnr.it:8880/NewAdaptationEngine/rest/subscribe
Parameters	<ul style="list-style-type: none"> userName appName sessionID actionFormat (JSON or XML).
Response	<p>JSON or XML including:</p> <ul style="list-style-type: none"> errorMessage subscribedRules array

3.2.2 Event Subscription

The Adaptation Engine retrieves the rules associated to that user and application and subscribes to the Context Manager for the specified events and conditions (4b).

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



3.2.3 Event Notification

When one or more events are verified and the rule conditions are met, the Context Manager notifies the Adaptation Engine (6) via HTTP(S) POST.

3.2.4 Application Adaptation

The Adaptation Engine forwards the list of actions to be applied to the UI (7). This is done via Web Socket or via a REST service defined in the target application. The client scripts, used to subscribe to the Adaptation Engine, will receive the action part of the rules and should be able to interpret them in order to apply the adaptation.

When the application receives the actions list from the Adaptation Engine, a function called `applyRules` is invoked; this function is defined in `adaptation-script.js` file and for each action, depending on the action type (e.g. update, create, delete, etc.), invokes the corresponding function which will apply the customization. At the moment, in the file `adaptation.script` there are a few examples of this kind of function, but each application should implement the behaviour of the function.

3.3 Context Manager

The Context Manager is the module that gathers and manages contextual data. It is composed of a server and several delegates installed in various devices (e.g., a smartphone can host software detecting environment noise through the device's microphone).

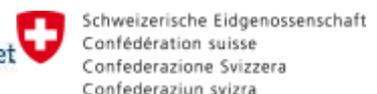
These delegates collect data and pass them to the server. Data is gathered from sensors (physical activity, temperature, noise, light, etc.) or external services (e.g. weather forecast).

The Context Manager offer REST Services to update the context sending a value sensed from a sensor and to get the current context values. The services currently available are listed below.

3.3.1 Environment

Method	GET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/environment/{attributeName}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>attributeName</code> =[alphanumeric]: the desired environment attribute. This is the list of the possible environment attributes:</p> <ul style="list-style-type: none"> • temperature; • gas-sensor; • humidity; • light_level; • motion; • noise_level; • smoke-sensor;
Response	JSON object with fields:

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



	<ul style="list-style-type: none"> • status (ERROR or OK) [a typical error is the wrong environment id]; • message; • value
--	--

Method	SET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/environment/{attributeName}/{attribute value}
Parameters	<p>userId =[alphanumeric]: the user identifier attributeName =[alphanumeric]: the desired environment attribute. This is the list of the possible environment attributes:</p> <ul style="list-style-type: none"> • temperature; • gas-sensor; • humidity; • light_level; • motion; • noise_level; • smoke-sensor; <p>attribute value =[alphanumeric]: the new value</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • entityName; • value; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • msg

3.3.2 Physiological attributes

The physiological attributes include: heart rate, body position, respiration rate, number of daily steps and body temperature, detected using BITalino chestband.

It is possible to get and update the value of each single attribute, or to get all these attributes through a single invocation calling the `physiological` attribute.

Method	GET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{user_id}/{attributeName}
Parameters	<p>userId =[alphanumeric]: the user identifier attributeName =[alphanumeric]: the desired physiological attribute. This is the list of the possible physiological attributes:</p> <ul style="list-style-type: none"> • heartRate • body_position • respirationRate • steps • bodyTemperature

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Response	<ul style="list-style-type: none"> • physiological <p>JSON object with fields:</p> <ul style="list-style-type: none"> • status (ERROR or OK) [a typical error is the wrong environment id]; • message; • value
-----------------	--

Method	SET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{user_id}/{attributeName}/{attribute value}
Parameters	<p>userId =[alphanumeric]: the user identifier</p> <p>attributeName =[alphanumeric]: the desired physiological attribute. This is the list of the possible physiological attributes:</p> <ul style="list-style-type: none"> • heartRate • body_position • respirationRate • steps • bodyTemperature • physiological <p>attribute value =[alphanumeric]: the new value</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • entityName; • value; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • msg

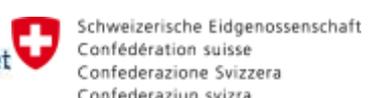
3.3.3 Devices

In order to update an attribute related to a device you should first register the device to the context and then update a device attribute.

Method	Add the device to a user context
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/addDevice/{deviceName}
Parameters	<p>userId =[alphanumeric]: the user identifier</p> <p>deviceName =[alphanumeric]: the device name</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • deviceId; • status [ADDED, NOT_ADDED]; • msg;

Method	Add the device to an environment
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/environment/{environmentId}/addDevice/{deviceName}
Parameters	environmentId =[alphanumeric]: the environment identifier

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Response	<pre>deviceName =[alphanumeric]: the device name</pre> <p>JSON object with fields:</p> <ul style="list-style-type: none"> • deviceId; • status [ADDED, NOT_ADDED]; • msg;
-----------------	--

Method	GET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/devices/{deviceId}/attribute/{attributeName}</code>
Parameters	<pre>deviceId =[alphanumeric]: the device identifier attributeName =[alphanumeric]: the desired device attribute. This are some of the possible device attributes:</pre> <ul style="list-style-type: none"> • name • room • state • color • ...
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • status (ERROR or OK) [a typical error is the wrong device id]; • msg (error message); • value

Method	SET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/devices/{deviceId}/attribute/{attributeName}/{attributeValue}</code>
Parameters	<pre>deviceId =[alphanumeric]: the device identifier attributeName =[alphanumeric]: the desired device attribute. This are some of the possible device attributes:</pre> <ul style="list-style-type: none"> • name • room • state • color • ... <pre>attribute value =[alphanumeric]: the new value</pre>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • entityName; • value; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED); • msg (error message)

3.3.4 Physical Objects

Physical Objects are very similar to devices: before updating the value of an attribute you have first to register the physical object to the context.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



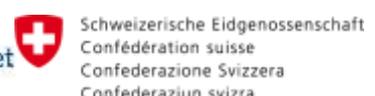
Method	Add the physical object to a user context
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/addPhysicalObject/{objectName}
Parameters	userId =[alphanumeric]: the user identifier objectName =[alphanumeric]: the physical object name
Response	JSON object with fields: <ul style="list-style-type: none"> • physicalObjectId; • status [ADDED, NOT_ADDED]; • msg;

Method	Add the physical object to an environment
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/environment/{environmentId}/addPhysicalObject/{objectName}
Parameters	environmentId =[alphanumeric]: the environment identifier objectName =[alphanumeric]: the physical object name
Response	JSON object with fields: <ul style="list-style-type: none"> • physicalObjectId; • status [ADDED, NOT_ADDED]; • msg;

Method	GET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/physicalObject/{physicalObjectId}/attribute/{attributeName}
Parameters	physicalObjectId =[alphanumeric]: the physical object identifier attributeName =[alphanumeric]: state
Response	JSON object with fields: <ul style="list-style-type: none"> • status (ERROR or OK)[a typical error is the wrong physical object id]; • msg (error message); • value

Method	SET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/physicalObject/{physicalObjectId}/attribute/{attributeName}/{attribute_value}
Parameters	physicalObjectId =[alphanumeric]: the physical object identifier attributeName =[alphanumeric]: state attribute_value =[alphanumeric]: the new value (on or off)
Response	JSON object with fields: <ul style="list-style-type: none"> • entityName; • value; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED); • msg (error message)

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



3.3.5 Personal Data

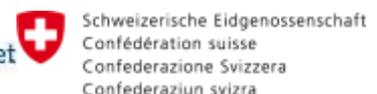
Method	GET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/personalData/{attributeName}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>attributeName</code> =[alphanumeric]: the desired personal data attribute. This is the list of the possible personal data attributes:</p> <ul style="list-style-type: none"> • age; • gender; • name; • surname; • education
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • status (ERROR or OK); • message; • value

Method	SET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/personalData/{attributeName}/{attribute value}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>attributeName</code> =[alphanumeric]: the desired personal data attribute. This is the list of the possible personal data attributes:</p> <ul style="list-style-type: none"> • age; • gender; • name; • surname; • education <p><code>attribute value</code> =[alphanumeric]: the new value</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • <code>entityName</code>; • <code>value</code>; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • <code>msg</code>

3.3.6 Weather

Method	GET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/weather/{attributeName}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>attributeName</code> =[alphanumeric]: the desired weather attribute.</p>

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



	<p>This is the list of the possible weather attributes:</p> <ul style="list-style-type: none"> • temperature; • humidity; • wind;
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • status (ERROR or OK); • message; • value

Method	SET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/weather/{attributeName}/{attribute value}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>attributeName</code> =[alphanumeric]: the desired weather attribute. This is the list of the possible weather attributes:</p> <ul style="list-style-type: none"> • temperature; • humidity; • wind; <p><code>attribute value</code> =[alphanumeric]: the new value</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • <code>entityName</code>; • <code>value</code>; • <code>status</code> (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • <code>msg</code>

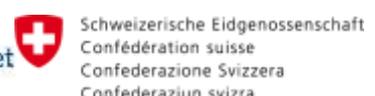
3.3.7 Motivation

The motivation parameter is used by the Remote Assistant application to

Method	GET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{user_id}/motivation/</code>
Parameters	<code>userId</code> =[alphanumeric]: the user identifier
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • status (ERROR or OK); • message; • value

Method	SET
Endpoint	<code>http://giove.isti.cnr.it:8880/cm/rest/user/{user_id}/motivation/{motivation value}</code>
Parameters	<p><code>userId</code> =[alphanumeric]: the user identifier <code>motivation_value</code> =[alphanumeric]: the new value. The possible values are: "wellness", "health", "social" e "fitness"</p>
Response	<p>JSON object with fields:</p> <ul style="list-style-type: none"> • <code>entityName</code>;

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



	<ul style="list-style-type: none"> • value; • status (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • msg
--	--

3.3.8 Medication Planned and Medication Occurred

These parameters are used to model the medication data used by the IBM Medication monitoring application. In the final version we also plan to add Activity Planned and Activity occurred parameters for use in the Physical Rehabilitation application.

Method	GET
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/medication_planned http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/medication_occured
Parameters	userId =[alphanumeric]: the user identifier
Response	JSON object with fields: <ul style="list-style-type: none"> • status (ERROR or OK); • message; • value

Method	POST
Endpoint	http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/medication_planned http://giove.isti.cnr.it:8880/cm/rest/user/{userId}/medication_occured
Parameters	userId =[alphanumeric]: the user identifier data sent to the context manager should be a JSON object having the following format: <pre> for medication_planned attributes: { "notification_timestamp": "Sat, 28 Apr 2017 10:00:00 GMT", "notification_time": "10:00", "medication": "EPCLUSA", "dosage": "600MG" } for medication_occured attributes: { "registration_timestamp": "Sat, 28 Apr 2017 10:05:00 GMT", "registration_time": "10:05", "medication": "EPCLUSA", "dosage": "600MG" } </pre>

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Response	JSON object with fields: <ul style="list-style-type: none"> • <code>entityName;</code> • <code>value;</code> • <code>status</code> (ENTITY_NOT_UPDATED or ENTITY_UPDATED) • <code>msg</code>
----------	--

3.4 Behavior Analysis Module

The Behavior Analysis Module is expected to analyze the data collected in the context manager, model the elderly's behavior (activity levels, social interaction, etc.) and detect deviations from standard behavior showing that the individual behavior is deteriorating, or situations of no progress towards elderly's goals. The output of this analysis is then passed to: i) the Persuasion module, to identify what and how necessary persuasions are going to be applied, and ii) the Adaptation module, to adapt the outcome of user behavior analysis before being delivered to the elderly.

3.5 Authentication Server

The PersonAAL authentication server provides the main security and data protection mechanisms to all platform components and associated web applications. The server safeguards information collected within the platform against unauthorized access and ensures data confidentiality and data integrity. The server allows only authenticated users, web applications, and platform components to upload, access, and process collected and stored data within the platform. For example, it allows only authenticated sensors to upload user data to the platform eliminating any possibility of an intentional injection of false data through malicious sensors and components. Also, the server provides a single sign-on mechanism allowing a single user account, i.e., a single set of log-in credentials to be used across all applications and platform components. This distributed authentication architecture allows for storing personally identifiable information on the secure authentication server (e.g., user name, email, password, or user id) and separating it from actual personal user data (e.g., activity data or health information) collected and stored within different elements of the platform or web applications. The collected personal user data is then identifiable by pseudonymous user identifier or authentication token issued by the authentication server after a successful authentication procedure.

The authentication server is based on the state-of-art "OAuth 2.0" protocol implementation that can authenticate all components within the system. The server is hosted on a professional and secure hosting platform that provides uninterrupted and continuous operation and access. The server provides a set of API calls available to the web applications and platform components. The API calls are encapsulated into classes and objects that can be directly integrated into the application or platform code. Within the project, we provide a set of authentication components implemented using popular development technologies such as Java, JavaScript, PHP, NodeJS, and Android. For example, the JavaScript authentication component can be directly integrated into HTML file as a DOM object providing a custom login page that handles all communication to the authentication server. After a successful authentication, the authentication component receives a unique access token that identifies personal user data and can be used for subsequent

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



calls to the platform components. The set of API calls allows users and components to signup, change password, login, logout, get access token, refresh access token, revoke access token, and get user account information. The API calls are described below.

The integration of all platform components with the Security Server is ongoing and will be completed before starting the Field Trial with end users.

3.5.1 Authentication API: *signup*

The *signup* API call allows users to create online accounts through the web applications and platform components. The call requires "client_id", i.e., the id of the web application or platform component through which the user wants to create an account. The users account contain "email" and "password" information. Additional user information is contained within "user_meta" field. The authentication server automatically sends a verification email before allowing users to log in.

```
POST https://personaal.eu.auth0.com/dbconnections/signup
Content-Type: 'application/json'
{
  "client_id": "meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ",
  "email": "EMAIL",
  "password": "PASSWORD",
  "connection": "CONNECTION",
  "user_metadata": "{\"name: 'Jon', colour: 'blue'}"}
```

An example response to the *signup* API call:

```
{"_id": "58457fe6b27...", "email_verified": true, "email":
"test.account@signup.com"}
```

3.5.2 Authentication API: *change_password*

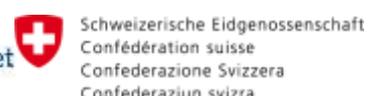
After creating the account users can change the password using *change_password* API call. After the API call, the server sends an email with instructions and a separate link where users can change the password.

```
POST https://personaal.eu.auth0.com/dbconnections/change_password
Content-Type: 'application/json'
{
  "client_id": "meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ",
  "email": "EMAIL",
  "password": "",
  "connection": "CONNECTION",}
```

An example response to the *change_password* API call:

```
{"We've just sent you an email to reset your password."}
```

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



3.5.3 Authentication API: *login*

Users that have created the account can login into the server to obtain access or refresh token as well as to retrieve user account information.

```
GET https://personaal.eu.auth0.com/authorize?
  response_type=code|token&
  client_id=meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ&
  connection=CONNECTION&
  redirect_uri=https://uc-dev.inf.usi.ch/personaal&
  state=STATE&
  additional-parameter=ADDITIONAL_PARAMETERS
```

3.5.4 Authentication API: *logout*

The logout API call allows users to logout from the authentication server.

```
GET https://personaal.eu.auth0.com/v2/logout?
  client_id=meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ&
  returnTo=LOGOUT_URL
```

3.5.5 Authentication API: *get_access_token*

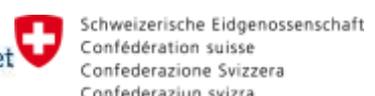
Users and platform components can obtain the access and refresh tokens. The access token is used to authenticate users and platform components without using user credentials within the system. In subsequent calls to other web applications or platform components that provide PersonAAL services, only a valid access token is needed.

```
POST https://personaal.eu.auth0.com/oauth/token
Content-Type: 'application/json'{
  "grant_type": "authorization_code",
  "client_id": "meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ",
  "client_secret": "zIpBb2ROMTq2-zVqik_d3-wV2Zh-
  KG1YkA6JHRTagnCqH841tbp5K01WgDmnoQJs",
  "code": "AUTHORIZATION_CODE",
  "redirect_uri": https://uc-dev.inf.usi.ch/personaal}
```

An example response to the *get_access_token* API call:

```
HTTP/1.1 200 OK
Content-Type: application/json{
  "access_token":"eyJz93a...k4laUWw",
  "refresh_token":"GEbRxBN...edjnXbL",
  "id_token":"eyJ0XAi...4faeEoQ",
  "token_type":"Bearer",
  "expires_in":86400}
```

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



3.5.6 Authentication API: *refresh_access_token*

The access tokens expire after a predefined time interval (e.g., 3600 seconds). After the timeout interval, the users or components do not need to login again. They can simply refresh the access token using the token API call with the "grant_type" field set to "refresh_token".

```
POST https://personaal.eu.auth0.com/oauth/token
Content-Type: 'application/json'
{
  "grant_type": "refresh_token",
  "client_id": "meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ",
  "client_secret": "zIpBb2ROMTq2-zVqik_d3-wV2Zh-
KG1YkA6JHRTagnCqH841tbp5K0lWgDmnoQJs",
  "refresh_token": "YOUR_REFRESH_TOKEN"}
```

An example response to the *refresh_access_token* API call:

```
HTTP/1.1 200 OK
Content-Type: application/json {
  "access_token": "eyJ...MoQ",
  "expires_in": 86400,
  "scope": "openid offline_access",
  "id_token": "eyJ...0NE",
  "token_type": "Bearer"}
```

3.5.7 Authentication API: *revoke_access_token*

In a case of a suspicious activity, the user or platform component can revoke access token and prevent any further use of the token within the system.

```
POST https://personaal.eu.auth0.com/oauth/revoke
Content-Type: 'application/json'
{
  "client_id": "meVGye3Pdf6fy7zAmc1JpNJ4TSr34xQZ",
  "client_secret": "zIpBb2ROMTq2-zVqik_d3-wV2Zh-
KG1YkA6JHRTagnCqH841tbp5K0lWgDmnoQJs",
  "token": "YOUR_REFRESH_TOKEN"}
```

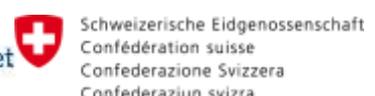
An example response to the *revoke_access_token* API call

```
HTTP/1.1 200 OK
(empty-response-body)
```

3.5.8 Authentication API: *get_user_info*

Authenticated and login users can access personal login credentials and information stored on the authentication server using *user_info* API call.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



GET <https://personaal.eu.auth0.com/userinfo>
 Authorization: 'Bearer {ACCESS_TOKEN}'

An example response to the `get_user_info` API:

```
{
  "email_verified": false,
  "email": "test.account@userinfo.com",
  "clientID": "q2hnj2iu...",
  "updated_at": "2016-12-05T15:15:40.545Z",
  "name": "test.account@userinfo.com",
  "picture": "https://s.gravatar.com/avatar/dummy.png",
  "user_id": "auth0|58454...",
  "nickname": "test.account",
  "created_at": "2016-12-05T11:16:59.640Z",
  "sub": "auth0|58454..."
}
```

3.6 Sensor Chestband

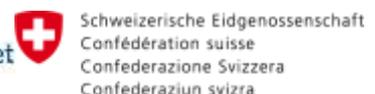
The chestband is comprised by two elements that attach to each other: a **wearable textile band** and a **hub** to acquire and gather measurements from the different sensors. The textile band has embedded a temperature transducer, two ECG electrodes and a displacement transducer (that is used to measure the respiration) as it can be seen in the figure below.



Figure 2 - Chestband embedded sensors

These transducers are connected to an acquisition unit that converts analogue values into digital ones and then streams them via Bluetooth to the Chestband context delegate. The acquisition unit connects to the textile band through snaps and the both together form the chestband that has the following sensors:

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



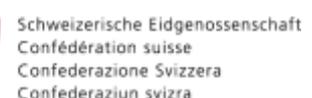
- Respiration: this sensor measures the thorax displacement and it allows to extract not only the breath rate but also the volume of the chest.
- ECG: this is a two leaded ECG placed in the V3 and V4 position that allows the bipolar measurement between these two points. The signal is equivalent to a lead I measurement.
- Accelerometer: Triaxial, with 3.6g range accelerometer, used to obtain the body position and the motion. The accelerometer is included in the acquisition unit pcb and is not embedded in the textile band as the other transducers.
- Temperature: 0-50°C body temperature sensor, positioned in the textile band such as it stays under the armpit, for a more accurate reading.



Figure 3 - Snap connectors that allow the textile band to attach to the acquisition unit

This architecture can be better understood with the help of the visual diagram below.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



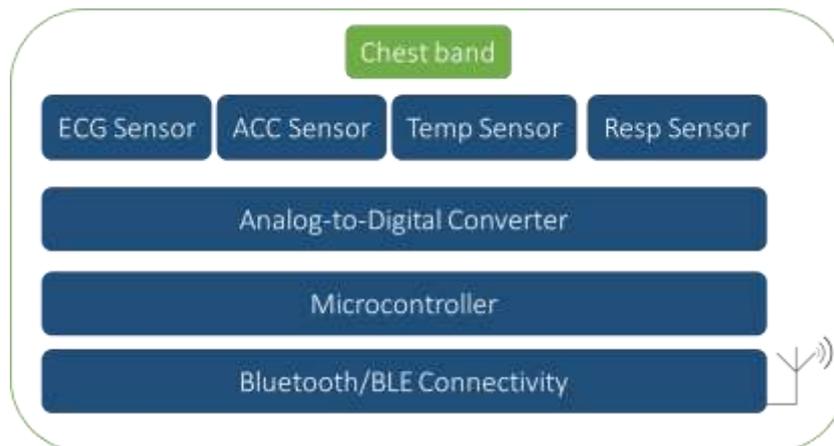


Figure 4 - Chestband high level architecture diagram

The communication protocol used to transmit the sensor signals of the chestband to the context delegate is described in full detail in the next section.

The chestband has a rechargeable battery that lasts up to 18 hours continuously streaming and is powered at 3.3V. The system is then rechargeable via micro-USB, and it takes 1h30 until be fully charged. The system has three LEDs:

- Status LED: fades once per second if the system is in standby and blinks twice per second if the system is in acquisition mode.
- Low Battery LED: turns on solid orange if the battery is below 20%.
- Charging LED: is solid orange if the system is correctly charging; is solid purple if the system is fully charged; is solid red if the system is not charging either because the device is on or the battery has problems.

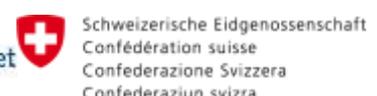
Since the sensors of the chestband are directly connected to the body, for safety reasons, the charging module is galvanically separated from the rest of the system, only allowing to charge if the system is manually turned off.

Under the acquisition unit was placed the MAC address of the BTH module to allow the easy assignation of this number to the patient profile.

Alongside the features described above, the chestband also has the following characteristics:

- Includes a shoulder strap, to prevent the band from slipping;
- Has lycra electrodes to ensure the maximum comfort, without compromising the ECG measurement;
- Is washable and reusable;
- Features a system on the back that allows the band to be adjusted to various chest sizes;

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



- Elastic band that allows the accurate measurement of the respiration while ensuring maximum comfort;

3.7 Context Delegate

The context delegate is the module responsible for the all the exchange of queries and data between the chestband and the mobile device, as it is also responsible for the extraction of feature of the raw data and send them to the Context Manager. This module is a standalone module and it can be described with three workflows: start service workflow, connection workflow and acquisition workflow.

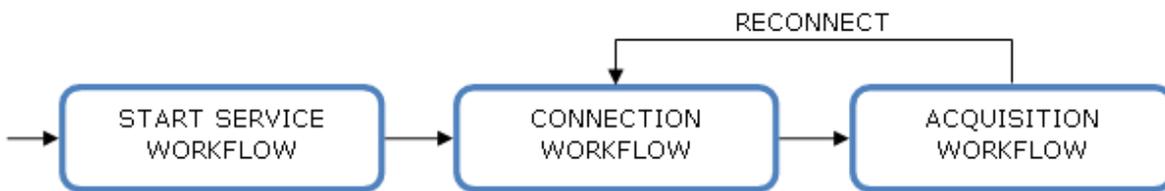


Figure 5 - Context delegate workflow

The Start Service Workflow is the implementation that allows our service to be launched when the device's boot is completed. This behaviour guarantees that our Context Delegate Service is always running in background, when the mobile device is turned on. When this procedure is completed, the Connection Workflow starts.

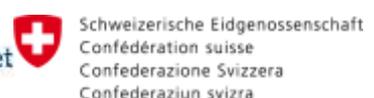
The Connection Workflow is the segment responsible for the scan and connection to the chestband. The scan job will always be running until a suitable device is found. As soon as the device is found and it is available to establish a connection, the scan job stops and the connection procedure starts. After a successful connection between the devices, the Acquisition Workflow starts, otherwise we keep trying to establish the connection. If for any unknown reason the connection is lost after it is established, the Context Delegate will return to this stage and will keep trying to reconnect.

The data acquisition is set for the chestband's six channels with a sampling frequency of 100 Hz. Once this is set, the context delegate will receive the data asynchronously. As data becomes available, it is run through the processing algorithms in order to extract the features of interest. The features extracted from each sensor are:

- Respiration – after filtering and smoothing the signal, the respiration rate is extracted;
- Electrocardiogram – after filtering the signal, the heart rate is extracted;
- Accelerometer – the body position is extracted. For this feature there are 5 possible positions: UP, SUPINE, PRONE, RIGHT and LEFT. From this sensor we also run a pedometer algorithm that enables the estimation of the number of steps taken;
- Temperature – in this case, a transfer function is directly used to obtain the temperature value in degrees celsius.

To avoid big data issues, all the extracted features are summarized into 5 minutes object data by: the average value of the respiration rate, heart rate and temperature during the period of time, the number of steps taken during the period of time and all the positions during this time.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



To enable position tracking, this value is associated with the timestamp whenever the position changes.

After getting this package of data, it needs to be sent to the Context Manager. To avoid any loss of data, a simple local database was structured and implemented to save all data until it is sent to the Context Manager. Once we receive the information that this data was sent with success, it is deleted from the database. To send data to the Context Manager a job scheduler was implemented, running a task every 5 minutes, if a wi-fi connection is available, accessing our database and sending the information to the server.

3.8 Persuasion Module

The PersonAAL Persuasion Module aims to motivate older adults to adopt behaviours that improve their well-being and, ultimately, increase the time they can live independently at home. The Persuasion Module uses the data collected by the PersonAAL framework to reason about the current behaviour of the user and decide about the motivation techniques that can be applied to try to reinforce or change the behaviour.

The concepts grounding the Persuasion Module can be found in Deliverable D1.3a. In that deliverable, the implementation details of the five modules comprising the Persuasion Module are also described. *Figure 6* presents the architecture of the Persuasion Module, showing the five modules that process the data and generate the persuasive events, the other PersonAAL components that feed and receive data from the Persuasion Module, and an alternative data source for specific activity sensing software or hardware (which can be used when the required information is not available from the Context Server).

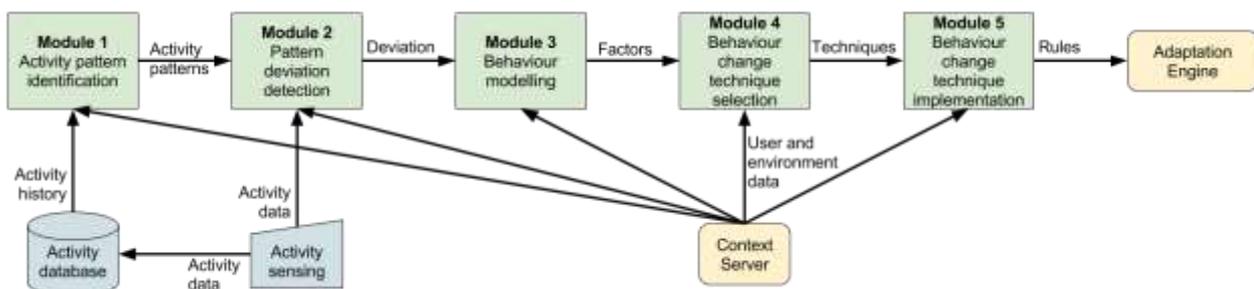
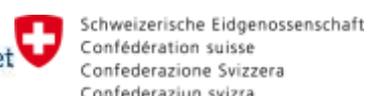


Figure 6 – Architecture of the Persuasion Module

The Persuasion Module primarily exchanges information with two other components of the PersonAAL framework: the Context Server and the Adaptation Engine. The Context Server provides the data this is required for the Persuasion Module operation. This data is required to identify the user’s behavioural patterns, detect deviations from the patterns and implement behavioural change techniques. The Persuasion Module stores the context variables that are required to run the behavioural models stored and to generate the messages or events associated with behavioural change techniques. Requests to the Context Server are made to collect the most recent values of these context variables. These requests are made through the REST API made available by the Context Server.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



The outcome of the Persuasion Module is a message to be presented to the user. To have these messages reach the user in a manner that is independent from existing applications, the Persuasion Module creates rules directly in the Adaptation Engine. Given the different types of behaviours and variables at play, the Persuasion Module executes its modules with a specified periodicity. When this process leads to the creation of a persuasive message, the Persuasion Module creates a rule that triggers at a specified time and executes an action that corresponds to the presentation of a reminder (whose text was created in the Persuasion Module) to the user. For example, the Persuasion Module will run at 16:00 to check the physical activity levels of a user in a given day (as observed in the user's step count for that day). If the module finds a deviation from the user's activity pattern it will generate a motivational message (to try to get the user to walk more if the deviation was negative, or to congratulate the user if the deviation was positive). This will lead to the creation of a rule in the Adaptation Engine to be triggered in the next minutes and the generated message to be shown to the user. These rules are described in JSON and sent to the Adaptation Engine through the available REST API. The Persuasion Module is also responsible for deleting the created rules from the Adaptation Engine, to prevent them from, incorrectly, triggering in the next day. This is also done through the Adaptation Engine's REST API.

3.9 Social Module

The PersonAAL Social Module will be responsible for providing a set of functionalities or techniques focused on social support. It will focus on contexts like sharing experience/knowledge with others, facilitating information transfer and communication and getting in contact (both online and offline) with other people. Information regarding the concepts grounding the Social Module can be found in Deliverable D1.4a., including the definition of each use case related with every technique to be employed by the Social Module.

The Social Module will both provide opportunities for social activities to be activated explicitly and implicitly. For this to happen several PersonAAL components are involved. Initiation can be defined by users (such as caregivers) or by the Persuasion Module through the creation of rules and correspondent rule activation. Every time a trigger is activated the rule is fired and a related user interface procedure is activated (through the adaptation module) with the respective social opportunity being initiated by the social module. Additionally, the Social Module might also be responsible for identifying social opportunities or lack of social activities through a set of applicational sensors (social listeners) in the users' smartphone (number of calls, number of contacts, contact information, etc.) or from online social network services (number of interactions, type of interactions, etc.). This social information is then sent to the Context Manager, where both the applications and the Social Module can pick it up for both identifying social behaviour deviations or the aforementioned social persuasion ends.

The Social Module is mainly set to be used through the Remote Assistant Application. This application will be integrated with the social module in two ways: 1) will offer to the users the ability to initiate social activities when planning activities; and 2) will be the main platform mediating the interventions initiated by the rule-editor and the persuasion module providing ways of combining these interventions with social opportunities managed by the social module. The Rehabilitation Application will also offer the possibility of combining exercises with the ability of sharing knowledge about these with others, using for that purpose techniques offered by the social module.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



4 INTEGRATION PLAN

In this section we report the narrative of the scenario realized for the M18 demonstrator, and the planned scenario for the final demonstrator at M36, in order to highlight the integration aspects involved and the plan for the next steps.

In the M18 demonstrator the components involved were:

- Personalization Rule Editor
- Context Manager
- Adaptation Engine
- Context Delegates
- Sensor Chestband
- Persuasion Module
- Applications

In the final scenario we will show full integration also of:

- Authentication Server
- Social Module
- Behaviour analysis module

The timeline of the integration plan for the next months is on line with the field trial plan and is outlined in the picture below:

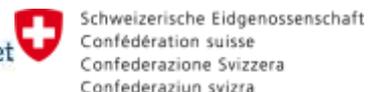
	Ott '17	Nov '17	Dec '17	Jan '18	Feb '18	Mar '18	Apr '18	May '18	Jun '18	Jul '18	Aug '18	Set '18	
Field Test Plan	User Acceptance and Comfort level		Rework		User Activity			Rework	Social Activity		Analysis of Results and Rework		
Platform version	2.1 – Authentication Server				2.2 – Behaviour Analysis				2.3 – Social Module				3.0

Figure 7 - Timeline for Integration Plan

4.1 Integrated Scenario realized at M18

Scenario description	Component Involved
User accesses the Personalization Rules Editor. User shows the context dimensions that compose the triggers and the action structure	Personalization Rules Editor
User defines the following rules: <ol style="list-style-type: none"> 1. WHEN motivation is wellness AND time is 16:00 AND daily steps are less than 1000, DO show Step Goal element, show weather element, hide all the others elements, send a reminder with the text: “Your daily steps are below the goal, please go out for a walk”; 2. WHEN respiration rate (or heart rate) is more than 80 breaths/min (or 180 bpm) AND body temperature is 	Personalization Rules Editor Adaptation Engine

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.

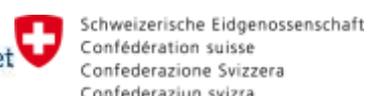


<p>more than 38°, DO send an ALARM with the text: “Your body temperature is too high”.</p> <p>3. WHEN time is 10:00 AND the body posture is laying down, DO turn on the bed room light and set the green colour;</p> <p>The defined rules are sent to the adaptation engine.</p>	
<p>User accesses the Remote Assistance Application and shows all the implemented functionalities (health, plan, profiles, contacts). In the health section show the ECG and temperature values received live from the sensor chestband.</p>	<p>Remote Assistant application Context Manager Context Delegates Sensor Chestband</p>
<p>Trigger of rule 1: Modify manually through the context manager REST Service the value of steps and time, modify the motivation through the application and highlight the effects of the actions on the application</p> <p>Trigger of rule 2: Modify manually the values of respiration rate and body temperature and highlight the effects of the actions on the application;</p> <p>Trigger of rule 3: the user change its position to laying down and manually another partner change the time through the REST service and the lights turn on.</p>	<p>Context Manager Adaptation Engine Remote Assistant application</p>
<p>User accesses the IBM Medication application and show the implemented functionalities</p>	<p>Medication application</p>
<p>User shows that the medication information entered through the Medication Application are sent to the Context Server.</p>	<p>Medication application Context Manager</p>
<p>Persuasion Module generates a rule and then send it to the adaptation engine.</p> <p>Load the adaptation engine page that shows all the stored rules (http://giove.isti.cnr.it:8880/NewAdaptationEngine)</p>	<p>Persuasion Module Adaptation Engine</p>

4.2 Integrated Scenario Planned for M36

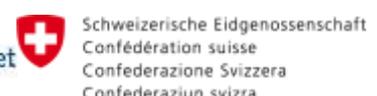
Scenario description	Component Involved
User register to the Authentication Server	Authentication Server
User log-in and accesses the Personalization Rules Editor. User shows the context dimensions that compose the triggers and the action structure	Authentication Server Personalization Rules Editor
User defines the following rules: <ol style="list-style-type: none"> WHEN motivation is wellness AND time is 16:00 AND daily steps are less than 1000, DO show Step Goal element, show weather element, hide all the others 	Personalization Rules Editor Adaptation Engine

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



<p>elements, send a reminder with the text: “Your daily steps are below the goal, please go out for a walk”;</p> <p>2. WHEN respiration rate (or heart rate) is more than 80 breaths/min (or 180 bpm) AND body temperature is more than 38°, DO send an ALARM with the text: “Your body temperature is too high”.</p> <p>3. WHEN time is 10:00 AND the body posture is laying down, DO turn on the bed room light and set the green colour;</p> <p>The defined rules are sent to the adaptation engine.</p>	
<p>User accesses the Remote Assistance Application and shows all the implemented functionalities (health, plan, profiles, contacts).</p> <p>In the health section show the ECG and temperature values received live from the sensor chestband.</p>	<p>Authentication Server Remote Assistant application Context Manager Context Delegates Sensor Chestband</p>
<p>The user add to the plan a “walk” activity and he is given the possibility to invite a contact.</p>	<p>Social Module Remote Assistant application</p>
<p>Trigger of rule 1: Modify manually through the context manager REST Service the value of steps and time, modify the motivation through the application and highlight the effects of the actions on the application</p> <p>Trigger of rule 2: Modify manually the values of respiration rate and body temperature and highlight the effects of the actions on the application;</p> <p>Trigger of rule 3: the user change its position to laying down and manually another partner change the time through the REST service and the lights turn on.</p>	<p>Context Manager Adaptation Engine Remote Assistant application</p>
<p>User accesses the IBM Medication application and show the implemented functionalities</p>	<p>Authentication Server Medication application</p>
<p>User shows that the medication information entered through the Medication Application are sent to the Context Server.</p>	<p>Medication application Context Manager</p>
<p>If user has not complied to medication as planned over time a persuasion rule is activated to help persuade the user to take their medication. This rule sends messages to the Notification app on smart phone / tablet.</p>	<p>Medication application Persuation module Adaptation engine</p>
<p>User accesses the IBM Physical application and shows the implemented functionalities. Creating an activity plan, being notified about planned activity, record completing activity. Receiving persuasion messages to the Notification app when not performing the planned activities.</p>	<p>Physical Rehabilitation Context Manager Adaptation Engine</p>

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



<p>If a deviation from the expected behavior is detected (medications taken in the wrong order) an alert is sent to the user.</p>	<p>Behaviour Analysis</p>
<p>Persuasion Module generates a rule and then send it to the adaptation engine. Load the adaptation engine page that shows all the stored rules (http://giove.isti.cnr.it:8880/NewAdaptationEngine)</p>	<p>Persuasion Module Adaptation Engine</p>

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



5 CONCLUSIONS

The document presented the integrated PersonAAL platform, as it is at M24, detailing the interfaces exposed to the applications and the future steps towards final platform at M36.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



6 REFERENCES

- [1] D1.1b - Architecture Specification (Final), PersonAAL 2017.

The project PersonAAL is cofunded by the AAL Joint Programme (AAL-2014) and the following National Authorities and R&D programs in Italy, Portugal, Norway and Switzerland.



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra