AMBIENT ASSISTED LIVING, AAL

JOINT PROGRAMME

ICT-BASED SOLUTIONS FOR ADVANCEMENT OF OLDER PERSONS' INDEPENDENCE AND PARTICIPATION IN THE "SELF-SERVE SOCIETY"

D4.1 Integration Plan

Project acronym:	**ProMe**
Project full title:	**ProMe – Professional Intergenerational Cooperation and Mentoring**
Contract no.:	**AAL-2013-6-026**
Author:	**SIVECO, GLUK**
Delivery date:	**30.06.2017**
Dissemination:	**Public**

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

This deliverable provides a short outline of the plan for the implementation and integration of the ProMe components, which is critical for the next steps.

The structure of the deliverable is following the development phases, in the sense that the key components, on which others are based on, are developed first.

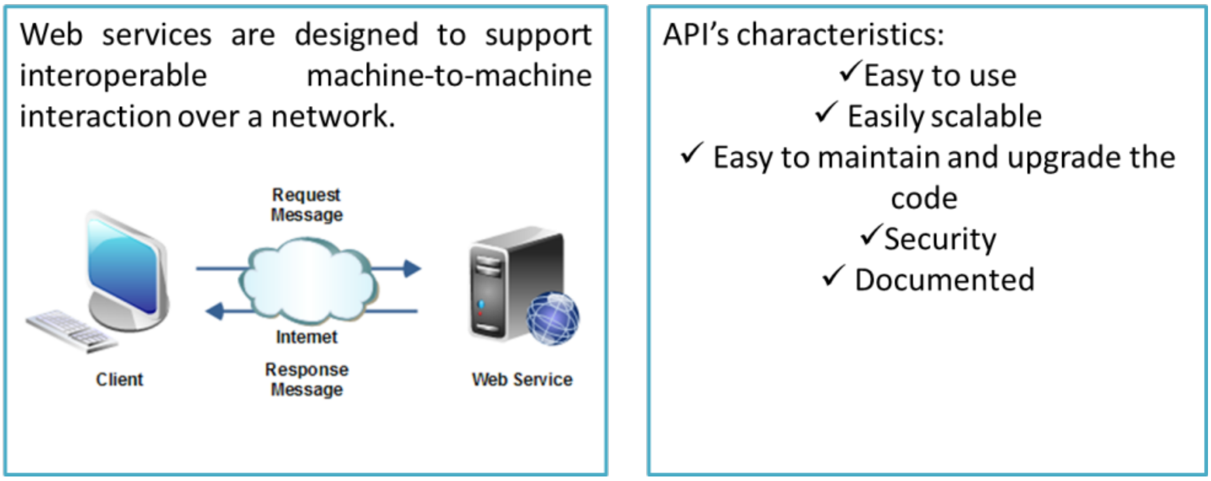
1.1 State of the art

The aim of this task is to reach a workable solution for pilot experimentations on the basis of the system requirements (WP3) and user requirements (WP2) and to integrate all functionalities.

D4.1 Integration Plan (M33): The integration plan will be developed in order to allow interoperability of the system with components already available and to facilitate the development of new functionalities.

2. INTEROPERABILITY – METHODOLOGICAL APPROACH

The Interface Processing approach is selected for the integration in which we use well-defined interfaces to focus on the integration of both packaged (Drupal) and custom applications (Recommendation System). The proposed Interface Processing externalizes information out of packaged applications through a well-defined API. The API will make data available through web services.



Web services are designed to support interoperable machine-to-machine interaction over a network.

API's characteristics:

- ✓ Easy to use
- ✓ Easily scalable
- ✓ Easy to maintain and upgrade the code
- ✓ Security
- ✓ Documented

Figure 1: Proposed interface processing process for ProMe platform interoperability

The communication between platforms/applications will be assured through web services protocols like SOAP, XML-RPC or architectural styles like REST.

Table 1. Assessment of communication technology

ASSESSMENT OF COMMUNICATION TECHNOLOGY			
	SOAP	XML-RPC	REST
Transfer protocol	HHTP	HTTP	HTTP
Data packaging	XML	XML	ANY (i.e. XML, JSON, BINARY)
Data package size	LARGE	LARGE	SMALL
HTTP Verbs	NO	NO	YES
Simple point-to-point communication	NO	NO	YES
Method exposure	YES	NO	NO
Singe point for multiple actions (i.e. CRUD)	NO	NO	YES
Uses Web Caching mechanisms	NO	NO	YES

Due to its many benefits and flexibility, the REST architectural style has been used to implement the API. As for data packaging, JSON has been used in order to limit the size of the messages, which will result in less data consumption for clients and less processing power to interpret the information (resulting in less energy consumption).

REST is a style of architecture defined by a set of principles that describe how data is defined and addressed. RESTful (or sometimes referred as REST-style) architecture consists of clients and servers. Clients are the ones that initiate data requests to the servers, which process these requests and return a response. Fundamentally in REST each resource is first identified using a URL and a new resource for every service required is created. The data returned by the service must be linked to the other data, hence making it in to a network of information unlike the Object-Oriented design, which encourages the encapsulation of information. REST architectural style consists on the principles:

a. Uniform Interface

Data is individually identified and manipulated using URLs and verbs. The information has multiple forms of representation (XML, JSON, HTML) when sent to the client. Clients can manipulate the information through these representations.

b. Stateless Interactions

All data needed to make a request is contained in the URL, query parameters, body or header.

c. Cacheable

Clients can cache server responses; the responses must contain the information if it can be cached or not in order to avoid sending inappropriate data in further requests.

d. Client-Server

Servers and clients are separated from each other, meaning that the client does not have direct access to the data storage and cannot interrogate information in this way.

e. Layered System

Clients do not need and don't know if they are connected directly to the web service server or to an intermediate. This gives the capability of multiple layers of systems to act in the same way.

f. Code on Demand

An option that extends a basic service, giving a temporary possibility for the server to execute code, sent by the client. This leads to a security risk and is hardly used.

REST is based on URLs (or also named URIs) and request methods (GET, POST, PUT and DELETE), which will be used to perform corresponding requests.

3. IMPLEMENTATION AND INTEGRATION PLAN

3.1 Refined architecture

The ProMe platform architecture has not changed substantially since the design phase. Its core functionalities are summarized below.

3.1.1 ProMe platform

ProMe provides users with a unique way to communicate and exchange learning materials. The following figure illustrates the architecture based on the logical levels of ProMe platform.

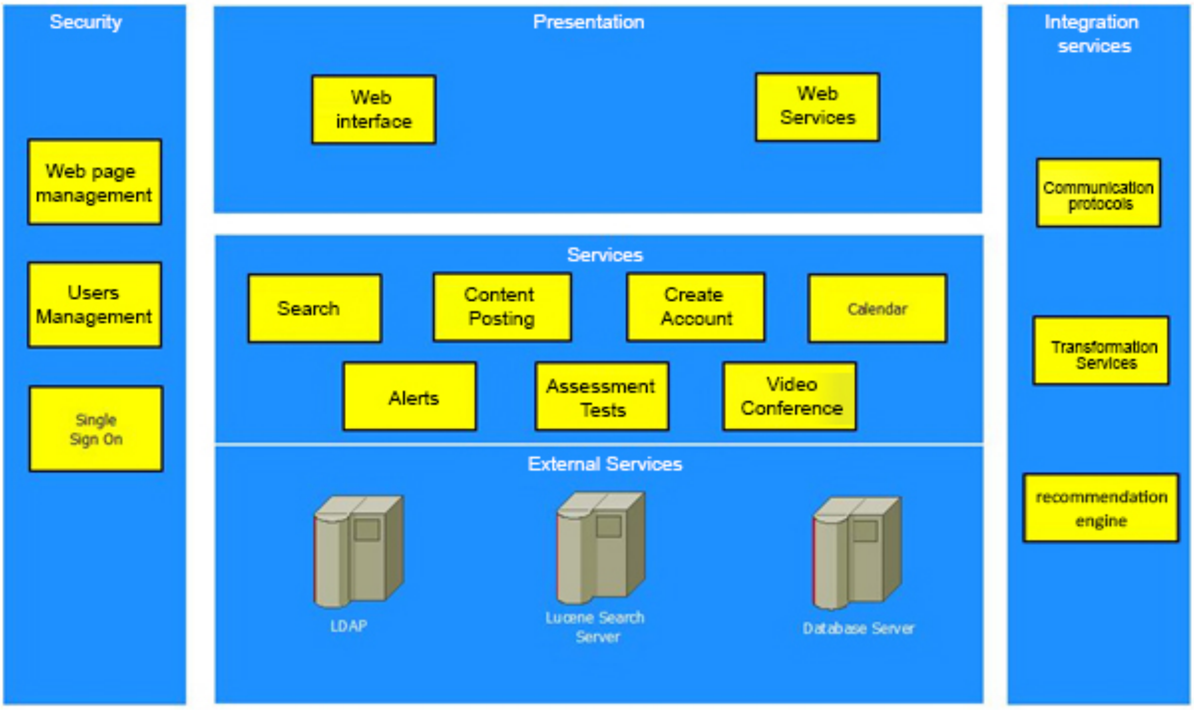


Figure 2. Levels of architecture, local approach

The presentation level reveals the components needed in the ProMe platform in order to provide services to users. The two components represent the graphic interface used by human users to access the portal, and portal information that allows external systems to access the ProMe services.

The graphic interface is based on new technologies such as HTML5 or CSS3. The inheritance elements are also provided for compatibility with browsers not supporting HTML5 or CSS3.

Due to real time features implemented that are in need of Web Sockets the browser compatibility list has changed to the following:

- Internet Explorer version 11.0 and subsequent.
- Firefox version 45.0 and subsequent.

- Safari version 9.1 and subsequent.
- Chrome 49.0 and subsequent.
- Opera 42.0 and subsequent.

For mobile devices, the ProMe platform is available to be used by any browser with significant market share:

- Safari Mobile iOS 4.2 and subsequent.
- Android Browser 4.0 and subsequent.
- Internet Explorer 9.0 for mobile on Windows Phone 10 and subsequent.
- Chrome 29.0 and subsequent.

The service level includes services for implementation of the business flow, in accordance with correspondent specifications. These services are used by the integration and presentation services to provide the functionalities corresponding to end-users.

3.1.2 Interoperability

The system is designed to facilitate communication between the ProMe web platform (SIVECO) and the Matching/Recommendation System (GLUK). For the communication between the two platforms REST web services with JSON message wrappers are implemented and used.

There have been identified three situations when the services will be called:

1. **Sending / updating profile information from the web platform to the Matching / Recommendation System**
2. **Search of matches in the Matching / Recommendation System by the web platform with multiple parameters**
3. **Search of matches in the Matching / Recommendation System by the web platform with only the profile id (recommendations)**

3.1.3 Recommendation System in ProMe

In order to achieve integration between the two major components of the ProMe platform we use the Microsoft Azure Cloud Services. We used the cloud system to host a stable version of the recommendation system composed from both the interface and the geo-replicated database. The integration service has been developed using Microsoft technologies. The application locally uses the ASP.NET MVC, version 5.2.3. Furthermore, it uses the MS Entity Framework, version 6.1.3 in order to manage the database layer.

The interface implements three basic functions:

- Profile process → Imports in the database new users’ data and/or modifies the existing
- Multiple Search → This is the main search function, implementing multiple search criteria according to the UI

- Search Profile → Additional function for search. This function is responsible for the inter-communication of the interface and the recommendation system. It retrieves all the recommendations according to user's personal preferences. Our software offers reliable recommendations by handling the information overload on large-scale user-based communities, and protects from spamming, malicious users and misreporting. This is achieved by using in-house implemented algorithms for machine learning, Internet data crawling, Internet data analysis, user behavior analysis and algorithms for recommendations/predictions.

In order to deploy the total interface service, we use the Microsoft Internet Information Services (IIS) web server (version 8.5) and the database version we use is Microsoft SQL Server 2012.

3.2 Overview of the integration plan

The integration plan is organized in six phases (i.e., phase 0 to 5) in which the development and integration follow a stepwise plan to allow both individual development for the technical partners and also to ensure a smooth integration of the components.

The start for the integration is the detailed use case, which facilitates the design of the database and around it the APIs to allow the integration of the various components. Each phase is planned to exploit the developments of the previous phase and integrate further components as they are also developed following the now detailed use cases, visual mockups and database design. The phases are summarized here:

Table 2. Integration plan

Phase	**Name**	**Related Deliverable**	**M16** **July -15**	**M18** **Sep-15**	**M30** **Sep-16**	**M37** **Apr-17**
Phase 0	Preparation for the development[1]	D3.2	**xxx**			
Phase 1	First development phase	D2.3	**xxx**			
Phase 2	First integration (Basic ProMe)	D3.3		**xxx**		
Phase 3	First integrated environment	D3.4, D3.5			**xxx**	
Phase 4	1st user testing in lab for 1st integrated prototype	D2.4			**xxx**	
Phase 5	Large scale pilot testing of fully integrated platform	D4.1, D4.2				**xxx**

[1] Use case iterations, mockups and clickable pdfs with proposed design

3.3 Detailed planning

3.3.1 Phase 0: Preparation for the development. (GUI Design)

The use case analysis was a necessary step for the design of the system in terms of the visual prototype, given the complexity of the system and the expected flexibility in the interactions of the modules.

Designing the ProMe platform was done taking into account the iterative rounds of feedback received from users and expert (heuristic) evaluations. Therefore, some delays occurred and at some point, the interface design overlapped the development phase.

3.3.2 Phase 1: 1st Development phase (July 2015)

The first phase of development is done

- The development of the integration environment:

- The development of the front-end mockups

- The development of individual components

3.3.3 Phase 2: First integration (Basic ProMe – September 2015)

The project integration follows the Interface Processing approach and uses well-defined interfaces to focus on the interoperability of both packaged (Drupal) and custom applications (Recommendation System). The proposed interface processing externalizes information out of packaged applications through a well-defined API.

Furthermore, during this phase the modules were further elaborated independently (not yet integrated). Also, some new modules started to be developed, since the database API clarified.

During the first attempt of the integration the interoperability of the ProMe portal including a videoconference solution (Big Blue Button) was not ready. Thus, the testing with users was done using Skype (via a call to action link on the platform)

3.3.4 Phase 3: First integrated environment – Full Scale ProMe – September 2016

This phase of the integration fills in the ready components in the integration environment and provides the first integrated prototype where the ready components of the previous phase will be integrated.

The system is designed to facilitate communication between the ProMe web platform (SIVECO) and the Matching/Recommendation System (GLUK). For the communication between the two platforms REST web services with JSON message wrappers should be implemented and used.

There have been identified three situations when the services will be called:

1. **Sending / updating profile information from the web platform to the Matching / Recommendation System**

Information about accounts will be sent by the web platform to the Matching / Recommendation System using a POST method in a URL provided by GLUK. An example of the message can be seen bellow:

JSON REPRESENTATION:

```
{
"first_name":"Klaus",
"last_name":"Klaus",
"email":"claudiu.burtea@siveco.ro",
"birth_date":"1977-02-01",
"country":"AL",
"gender":"male",
"user_type":"",
"time_to_invest":{"3":{"value":"3","name":"Donnerstag","periods":[{"did":"581","hid":"731","hour_from":"02:
00","hour_to":"14:00"},{"did":"581","hid":"732","hour_from":"02:00","hour_to":"14:00"},{"did":"581","hid":"7
33","hour_from":"02:00","hour_to":"14:00"}]}},
"interests":[],
"expertise":[],
"general_languages":[{"name":"assamese","reading":"fluent","speaking":"intermediate"},
{"name":"afrikaans","reading":"native","speaking":"basic"},
{"name":"","reading":"basic","speaking":"intermediate"}],
"field_flexible_availability":null,
"pid":"65"
}
```

EXPECTED RESPONSE IN JSON FORMAT:

```
{
"status":true, // the status of the process in the Matching / Recommendation System
"pids":1 // unique profile identifier sent in the request
“Message”: “” // error message if any
}
```

2. Search of matches in the Matching / Recommendation System by the web platform with multiple parameters

This function will be used for searching in the “Search” section of the web platform. Information will be requested by the web platform from the Matching / Recommendation System using a POST method in a URL provided by GLUK. An example of the message can be seen below:

JSON REPRESENTATION:

```
{
"user_type":"mentee", // string (only two possible values – mentor or mentee) ATT: this information is not
attached any more to the profile, you should not try to search for it just use it to know what you are going to
receive, interests or expertise
"interests":"it", // string (will be sent if a user searches for a mentee)
"expertise":"it", //string (will be sent if a user searches for a mentor)
"availability":{"1":"1","2":"2","3":"3","4":"4","5":"5","6":0, "7":"7"}, // array of days of the week where 1 is Monday as keys (integers) and the same in integer as value if it has been selected, 0 means it has not been selected
"country":"RO", // string, country two letter code
"name":"Dragos", // string
"field_flexible_availability":1, // integer, 0 if checked, 1 if not
"age_from":"20", // integer, age
"age_to":"55" // integer, age
}
```

EXPECTED RESPONSE IN JSON FORMAT:

```
{"pids":[1,5,14,23]}  // array of pids for profiles which match the search
```

3. **Search of matches in the Matching / Recommendation System by the web platform with only the profile id (recommendations)**

This function will be used for user suggestions in the “Search” section of the web platform. Information will be requested by the web platform from the Matching / Recommendation System using a POST method in a URL provided by GLUK. An example of the message can be seen bellow:

JSON REPRESENTATION:

```
{
"pid":1 // unique profile identifier for who suggestions should be displayed
}
```

EXPECTED RESPONSE IN JSON FORMAT:

```
{"pids":[1,5,14,23]}  // array of pids for profiles which match the search
```

3.3.5 Phase 4: 1st User testing in lab environment of 1st integrated prototype (August/September 2016)

Phase 4 will overlap partially with phase 3 in order to start testing the results of Phase 3 prior to testing the integrated prototype.

At the end of phase 4 the first integrated prototype is tested with the large scale scenarios in lab environment by users in order to collect feedback and refine the system. The feedback will be reported in D2.4.

3.3.6 Phase 5: Large scale pilot testing of fully integrated platform (April 2017)

During the 2nd user evaluation the technical partners continuously monitored the field trials and collected feedback to be implemented as soon as possible. The feedback from the end users is reported in D4.2.

4. CONCLUSION

This deliverable presents the implementation and integration plan of the ProMe components, organized in six phases, in which the development and integration follow a stepwise plan. Each phase exploits the developments of the previous phase and integrates further components as they are also developed, in order not to delay the next integration steps.

These phases follow the revised project time-plan and the delivery of related outcomes. Regarding the project integration plan the interface processing approach was selected allowing to test the integration specifications and to verify that the interfaces are all defined.

Taking into account the large-scale pilots and the complexity of the overall system, the compiled integration plan has ensured a proper user testing of the platform. The integration efforts were consistently high and more recommendations will be made in deliverable D4.2.

At the same time evaluation and validation of functional specifications will assist the commercial exploitation.