



Project FP7-216923

SMARTMUSEUM

Cultural Heritage Knowledge Exchange Platform

Deliverable D1.1

Smartmuseum architecture and use scenarios

VERSION 4: based on the Florence meeting, May 2008

ELIKO, Tanel Tammet

| Version | 4 | Date | May 30, 2008 | Classification | Restricted | Status | Final |
|---|---|-------------|--------------|-----------------------|------------|---------------|-------|
| Abstract | | | | | | | |
| The current document presents the main principles of the Smartmuseum software architecture and describes fundamental ideas of communication and context presentation. | | | | | | | |
| Revision history | | | | | | | |
| Rev. 4. Minor modifications of Florence meeting incorporated, T. Tammet, 30.05.2008. Release of Deliverable D1.1 | | | | | | | |
| Rev. 3. Smartmuseum architecture prefinal, T. Tammet, 06.05.2008. | | | | | | | |
| Rev. 2. Smartmuseum architecture draft, T. Tammet, 14.04.2008. | | | | | | | |
| Rev. 1. Smartmuseum architecture draft, T. Tammet, 31.03.2008. | | | | | | | |
| Rev. 0. Initial architecture presentation prepared, T. Tammet, 17.01.2008. | | | | | | | |
| Authors | SMARTMUSEUM Consortium (www.smartmuseum.eu) Responsible Editors: Tanel Tammet, ELIKO | | | | | | |



- 1. INTRODUCTION..... 3
- 2. SCENARIOS..... 4
 - 2.1 Inside scenario..... 4
 - 2.2 Outside scenario..... 7
 - 2.3 Museum admin scenario..... 8
- 3. ARCHITECTURAL PRINCIPLES AND COMPONENTS : BIG PICTURE..... 11
 - 3.1 Architectural principles: functionality, technology, complexities..... 11
 - 3.2 Systems created in the project..... 15
 - 3.3 External systems used by the project..... 16
- 4. USER PDA/SMARTPHONE..... 18
 - 4.1 Location recognition..... 18
 - 4.2 Reading and presenting object information 20
 - 4.3 Contacting the museum and the recommendation servers..... 22
 - 4.4 User profile management..... 23
 - 4.5 SmartContent provisioning and finding people on location for chatting (using Bluetooth)..... 24
- 5. CENTRAL PROFILE ENGINE/SUGGESTION CALCULATOR..... 26
 - 5.1 Storing profiles..... 26
 - 5.2 Fetching and storing object information from (museum) web servers..... 28
 - 5.3 Calculating recommendations..... 31
- 6. USER PC APPLICATION..... 33
- 7. GEOLOCATION SERVER..... 34
- 8. OBJECT URL AND IDENTITY MANAGEMENT..... 36
- 9. ANNOTATION FORMAT FOR HTML PAGES ABOUT OBJECTS, ROOMS, MUSEUMS..... 37
 - 9.1 VRA Core 4, other approaches..... 38
 - 9.2 RDFa..... 40
 - 9.3 RDFa and VRA core usage example and principles..... 43
 - 9.4 Wiki and semantic wiki..... 47
- 10. DATA FORMAT ON RFID TAG..... 47
- 11. USER PROFILE FORMAT, SENDING PROFILE TO SERVER..... 49
 - 11.1 Personal details..... 50
 - 11.2 General preferences..... 50
 - 11.3 Concrete objects liked/disliked..... 51
- 12. CONCLUSIONS..... 52

1. Introduction

The current document presents the main points of the Smartmuseum software architecture. It is based on earlier drafts (public draft 1, internal draft 2,3) and the following discussions and feedback, in particular the Florence meeting.

The major conceptual changes from the draft 1 are:

- User comments added to the user profile, can be added via PDA or profile editing web app.
- User PC application dropped in preference to web app for manual profile editing: this also entails user name and password usage, high-level picture correspondingly changed.
- New communication details/principles: use https for contacting server (name/password used), delete uploaded data from PDA (profile accessible via web anyway).
- Additional PDA functionalities: SmartContent provisioning and finding matching same-time visiting users for direct chatting on location added, using bluetooth or some other advertising method.

The major detail additions to the draft 1:

- Workpackage references added.
- Profile storage schema description improved and extended, more details added.
- Details and examples about VRA Core and RDFa usage and annotated text given.
- Museum/object information fetching and storage description detailed.
- Geolocation server details added.
- Weather server and museum opening times fetching notes added.
- Research and development task list added and ordered by complexity.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

2. Scenarios

We use two principal scenarios as a motivation for the architecture:

- **The inside scenario:** user visits a museum.
- **The outside scenario:** user walks around the city, looking for interesting places/buildings to visit/look at. The outside scenario focuses on cultural heritage, museums etc, although it is conceptually not limited to cultural sites.

In both scenarios the user has a PDA as a main device for both locating user and presenting information. Both scenarios have to:

- Locate the user. We will use GPS outside and RFID reading or manual numeric input inside.
- Send the user profile from PDA to the recommendations server and calculate the probable interests of the user.
- Present the suggested places to visit.
- Present detailed information at interesting places on users request.
- Store user interests/feedback in the profile.
- Allow the user to see and modify the profile by hand.
- Offer administrative tools both for museum and city places-of-interest administrators.

Both scenarios assume that the user has created a user name and password on his PDA. This user name will be used for identifying the user profile. User can create new user names if she so wishes, but the PDA should not manage more than just one profile at a time.

We will additionally present a *Museum Admin* scenario, writing data on RFID tags and putting information on web servers (specific, wiki etc.).

For following scenarios we will use hypothetical actor called John, who represents visitors in museum; Tanja, who represents visitors in outside scenario and Peter as a person who is responsible for administration of content (*Museum Admin*).

2.1 Inside scenario

John plans the museum visit at home:



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

- In case John wants to plan the museum visit very carefully (not obligatory!) he can use the web application for getting visit recommendations before the visit. In most cases we assume the visit is not planned very carefully, hence he will get recommendations during the visit.

John enters the museum.

- He must tell the PDA which museum is he in. Museum will be identified by URL.
- Hence he reads the museum RFID tag with museum uri at the entrance or alternatively types in the museum URL.
- PDA sends John the user profile to the profile server along with museum URL.
- Profile server sends John back information about the suggested rooms / areas to visit.

John walks around the museum. He can get information – when he explicitly wants – for rooms and separate objects:

- Each object is represented by a URL.
- When entering an interesting room he either scans the RFID tag at the door or types in room number near the RFID tag.
- When looking at an interesting object, he either scans the RFID tag near the object or types in object number near the RFID tag.
- PDA fetches corresponding information:
 - reads object URL from RFID tag
 - reads basic data from RFID tag directly if available
- PDA reads extra information from web at indicated URL from semantic wiki, using directly the URL on RFID tag or combining museum URL with nr.(does this mean, that it actually opens specified web page in browser component of the application?)PDA starts showing / speaking initial brief information about the room/object.
- PDA stores all RFID readings/number entries (actually – it stores URLs?) into user profile with an “interesting” mark.
- PDA presents further choices: read/listen more, etc.
- If more information is requested for object, PDA updates object mark to “very interesting”.
- John can – if he so wishes – mark the object in PDA as “I like it” or “I do not like it”: PDA updates object mark correspondingly.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- John can also type in short comments for the object, possibly including links to his blog where more remarks are given. This information is also stored in user profile.
- PDA may also decide to recalculate recommendations during visit and present new recommendations.

During the visit – in the museum rooms or while drinking coffee - John may be interested in talking to people interested in the same issues, like certain paintings or schools of paintings.

- To do so, John will indicate this interest in the PDA application.
- The application will fetch information from server about people with the same interest (indicating an interest to talk) and a matching profile, who are currently also in the same museum. The server knows who is in the museum based on the recommendation asking timestamp: recent visitors are probably yet here.
- The server gives John a list of matching profiles (potential privacy issue – must be thought out beforehand). John selects some, indicates his current whereabouts and signs for finding (man with a red hat) and the PDA starts sending out the corresponding message (probably via bluetooth (only up to 10m, so probably something else), but alternative possibilities should be investigated as well), also starting to show John the messages of other users on the spot.

When exiting the museum:

- John can – if he so wishes – read the museum RFID tag at the exit or select “exit museum” from PDA. After that the PDA sends updated profile to profile server and – possibly – fetches extra message for John from the museum.
- John can – if he so wishes – also send up the profile even without reading the museum exit RFID tag (for example, John does not find that tag), possibly for sending out his comments and scores to other visitors as quickly as possible.
- In case John does not mark the exit from the museum, PDA will not send updated profile information to server. However, whenever John uses the PDA again for finding interesting items in the future, the PDA first sends the current profile to server.

When at home:

- John can have a look at his profile at his PC using a browser: he has to enter the user name and password in the museum server web application. Observe that the user name does not give us any ways to identify the real person, unless John prefers to enter his personal details as well (fully possible to do).

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Since profile records visit times and object URL-s, John can have a more detailed look at interesting items using ordinary web browser called by the profile software.
- John can – if he so wishes – also edit his profile in the web application, possibly
 - removing items or
 - changing scores or
 - adding comments to visited objects or
 - giving general interest information about himself.
- He can then sync his PDA scores to the web server scores from the PDA so that the PDA will contain information just updated in the server.
- John can also get recommendations for the museum visit BEFORE the actual visit.

2.2 Outside scenario

The outside scenario is adapted from the scenario description by TKK.

Tanja, a 25 years old woman from Finland is on a vacation in Firenze (Florence, Italy) for one week. Tanja has a novel GPS-enabled phone with good maps and a browser. It is Friday, Tanja has just arrived in Firenze and decides to take a walk in the center of Firenze with a help of her GPS-phone. The weather is sunny.

Starting a walk around the city:

- Tanja puts herself on a map with a help of her mobile phone. SmartMuseum starts to show interesting locations based on her profile.
- Tanja walks a while and arrives to a house that looks interesting. The system informs Tanja: You are next to house where Galileo Galilei used to live. Technically this is achieved by sending the gps-obtained location from Tanjas' phone to the system server, see the geolocation chapter for details.

Wanting to get more information about a particular place/house:

- Tanja wants to know more about Galileo Galilei who used to live in the house. The system informs Tanja: Here is a portrait of Galileo Galilei. Recommendation: If you want to see this painting and items created by Galileo Galilei, go to Institute of the Museum of the History of Science. The forecast says it will start to rain soon, so a visit to museum is strongly recommended (again, the technical details of obtaining the forecast etc are given in forthcoming chapters).

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Tanja decides to go to the IMSS because she is interested in Galilei and because it will start to rain. The system informs Tanja: The route to the Institute of the Museum of the History of Science is shown on a map (proposed technology is Google maps track finding).
- Tanja walks to the recommended location of IMSS, buys a ticket and goes in.

At the museum and when at home:

- Actions covered in the previous “inside” scenario for John.

Planning next day:

- Tanja wants to plan her next day and asks SmartMuseum to recommend a nice route in Firenze based on her learned profile and weather forecast. Tanja further edits the route in her browser and saves it for sending. The system informs Tanja: A recommended route for tomorrow has been sent to your mobile phone.
- Tanja opens her mobile phone and sees she has received a recommended route for her Saturday in Firenze. Tanja gets an SMS from SmartMuseum based on her profile: This is a recommended route for next day (Saturday) in Firenze.
- On Saturday Tanja continues to walk in sunny Firenze with the route adapting as she goes along.

When at home:

- After one week Tanja goes back to Finland. Tanja gets an SMS from SmartMuseum based on her profile: You were interested in Galileo Galilei at IMSS in Firenze. Related events: A Space Exhibition in Finland until 2008-02-28

2.3 Museum admin scenario

The museum admin Peter is in charge of the museum information and its presentation to smartmuseum users. He has two main tasks:

- Updating room and object information on the web.
 - Whenever a new object is added to museum/display about which Peter wants the system to know and inform, he will create a new web page dedicated to that object.
 - System works better if there are also web pages indicating which objects are in which room and which rooms are there in the museum: Peter takes care of that as well.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform





- Sometimes Peter adds new information to pages of already existing objects.
- On the web page there will be ordinary, varied human-readable information with some parts of the text semantically annotated via RDF: this makes it possible for software to understand relevant parts of the otherwise human-oriented text.
- Peter can write/copy all this information either manually or use an internal database with information (if available) for automatically creating the web page using custom software.
- For manual entry Peter prefers wiki.
- Peter may even use universal, standard wikis like wikipedia instead of setting up a special wiki for the museum.
- For easier semantic annotation entry there is a special semantic wiki version of wikipedia, set up by the project: Peter may choose to use that wiki for museum pages.
- Information about objects may lie on different servers and point to more material on different servers: the only things that matter for the system is that the web page should be semantically annotated for better matching and better information presentation on pda.
- The absolutely basic functionality – simple matching and some data presentation for visitors - will work even without semantic annotations, though.
- Updating general and outdoors-oriented information about the museum.
 - Peter has to enter the museum main page URL and rooms URLs (if available) to the project database of participating museums. This can be done by simply mailing the information to the project general administrator in charge of the central matching server.
 - Peter also maintains the main pages of the museum. For easier finding of the museum outdoors – and getting general information – he puts semantically annotated information about the museum coordinates on the museum pages: these coordinates do not necessary have to be visible on the page.
 - Peter helps the community by adding similar coordinate and base information about the interesting buildings and locations nearby.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform





- Writing necessary information to RFID tags near doors and near objects. This activity assumes that information about the objects is already put to the web, as explained in the last section.
 - Peter walks around with spare RFID tags and a web-connected PDA equipped with RFID writer.
 - Whenever a new object is added or some RFID tag near an object is broken, Peter has to add a new RFID tag.
 - He will find the object description on the web, using the PDA software (which, in turn, will probably use a browser on the PDA)
 - He will choose “write this information” from the PDA.
 - PDA will write the found object's URL along with basic data (selected from semantic wiki) from the web to the RFID tag.
 - If the object is new and has no visible number attached, Peter can assign one automatically. One possible way to implement this is the following: Peter will choose “find new number” from the PDA. PDA will connect the server, find an unused number, associate it with the URL on the wiki and tell it to Peter. Peter will write the visible number near the RFID tag.



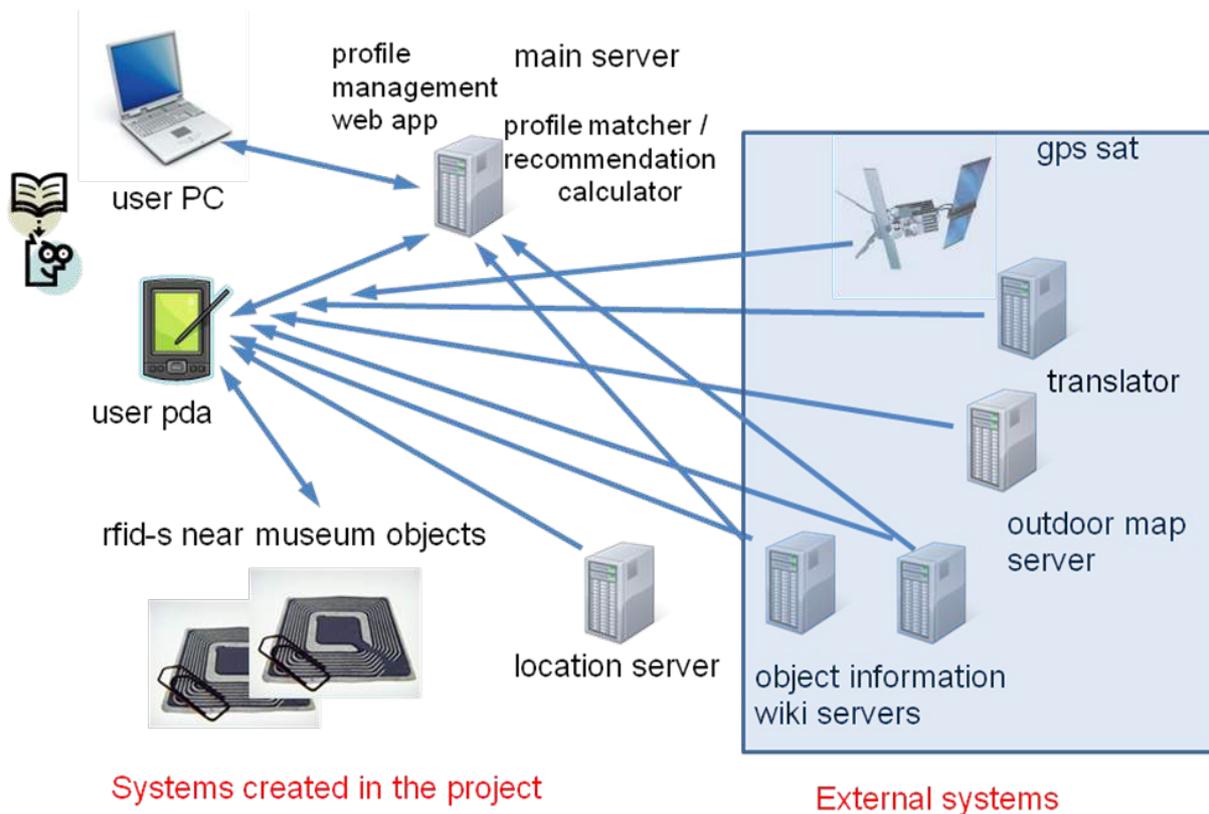
Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

3. Architectural principles and components : big picture

The high-level picture of the system components presents these main subsystems:



3.1 Architectural principles: functionality, technology, complexities

The main functional goals we are striving to achieve:

- Give the user as good suggestions for visit (outside or inside) as possible, basing the suggestions on (a) user profiles of a visitor and previous visitors, (b) machine-processable semantic information about objects.
- Give the user as good/suitable information about an object the user wished to know about as possible: initial brief audio clips/brief textual information on PDA, with easy possibility to

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



navigate further, direct reading of basic information from RFID tag if possible (otherwise over wireless), more detailed information on PC.

- Allow even basic, standard hardware (PDA without RFID reader, ordinary PC at home) to be used by the system, providing better experience and more functionality with better hardware (RFID, wifi, gprs/edge, gps etc).
- Allow active end user participation: adding object information via wikis, giving bad/good/super marks to objects, adding comments to visited objects.
- Allow users interested in communicating with other users find matching profiles (visiting same location, same interests, ...)
- Do not require active end user participation: most users are passive users, will not add information on wikis, will not give bad/good/super marks to objects.
- Require as little as possible from the museum: in particular, a participating museum has no need to put up special software or servers. Museum may even lack computers.
- Preserve privacy of users: tracking identified users is not OK unless the user explicitly makes it possible, user should be able to see, understand and change his/her own profile.

Throughout the architecture we make the following technological preferences/choices/assumptions:

- Assume totally distributed storage and widely varying structure of object information (picture author, basic description, context, audio clips, etc etc): different museums may keep that in different ways, in different servers.
- Assume that object information is primarily created for human consumption: no special-format files of web services with purely machine-oriented data will be required.
- Use semantic annotations based on the RDFa standard for organizing human-oriented web pages into machine-processable form.
- Assume that every object - picture, room, museum, interesting building - has a globally unique id (in other words, URI: Universal Resource Identifier) - which is in almost all cases an URL referring to an actual web page.
- Create and use ontologies based on the OWL standard for allowing generalizations to be made from object and profile data, with the goal of better matching of profiles for better recommendations.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Base the user profile on a machine-processable, yet human-understandable controlled-English grammar and semantics, such as Attempto Controlled English.
- Follow a sensible compromise on the user privacy versus better matching issue. In particular, do not save user identities on the profile server. However, user visit profiles are OK to be stored, assuming we do not store exact geographic locations (GPS coordinates) and exact times. Inexact visit times (like a year of a visit) are the borderline case.
- Use existing infrastructure (google maps, translation services, wikipedia, etc) whenever possible.
- Use existing technologies (semantic wikis, database engines, etc) and existing standards (RDFa, OWL, Attempto Controlled English etc) whenever possible.

The main expected research tasks for building the system are following, with the profile matching algorithms creation being the hardest, three earlier being of roughly similar complexity:

- Determining the exact RDFa and VRA Core 4 usage restrictions and extensions/modifications to the VRA Core 4 usage, in particular for object description methods and museum room/object listing methods. Handled by:
 - WP 4.2: Developing semantic and ontology components for web services.
- Creating initial, yet practically useful ontologies for museum and outdoors objects. Handled by:
 - WP 4.4: Initial profiling existing digital content and validation of external information sources.
- Creating methods for fetching and storing object/museum information from the web. Handled by:
 - WP 4.1: Developing architecture for dedicated „heritage content search services“.
- SmartContent provisioning and methods/principles for finding other matching people for chatting on location via Bluetooth. Handled by:
 - WP 3.2: Development of low level communication software for mobile device(s) (b) WAN and short range PAN communication („memory bubble“)
- Creating ontologies useful for profile matching based on the general preferences of the user and the RDF contents of the museum/object information web pages. Handled by:

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform





- WP 2.3: Developing theoretical solution for dynamic user profile creation and modification
- Working out the profile matching algorithms merging the statistical approach and ontologies-based approach working on general user preferences. Handled by:
 - WP 2.2: Developing resource efficient user profile representation format for distributed databases.
 - WP 2.3: Developing theoretical solution for dynamic user profile creation and modification.
 - WP 4.2: Developing semantic and ontology components for web services.

The most complex software components which have to be built, starting from simpler ones:

- User web application for profile management. Handled by:
 - WP 4.3: Developing user front end for web services and setting up voice synthesis software.
- Automatically reading and caching museum object list/object information from the RDFa-annotated web pages of museums/objects. Handled by:
 - WP 4.2: Developing semantic and ontology components for web services. TTK will be responsible for this component.
- Geo-location server. Handled by:
 - WP 4.2: Developing semantic and ontology components for web services.
- User PDA application, both for the end user and the museum administrator. Handled by:
 - WP 3.1 responsible for specification: Development of detailed software specification for user hand-held device and selection of toolset and existing software components to be used.
 - WP 3.2 responsible for the main implementation: Development of low level communication software for mobile device(s).
 - WP 3.3 responsible for the user front end: Development of user application software for mobile device(s).
- Profile matcher/recommendation calculator, containing the ontologies/general preferences handling part. The calculator software is the most complex part of the Smartmuseum system, consisting of several nontrivial subsystems. Handled by:

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- WP 4.2: Developing semantic and ontology components for web services.

3.2 Systems created in the project

For workpackage allocation of responsibilities, see the previous chapter.

- User PDA software, containing:
 - Temporary profile storage (before sending to server).
 - Route presenter: map-based for outdoors, textual for indoors scenario.
 - Place-of-interest presenter: locating is different for outdoors/indoors versions, presentation technology is shared.
 - RFID tag writer for museum administrators: write information to RFID tags near objects/rooms.
 - If possible: automatic text reader / audio synthesis.
 - If possible: locating nearby users interested in communication (interested in same museum objects, similar profiles etc) via Bluetooth.
- Central profile engine/suggestion calculator:
 - Stores general user profile information (username, personal details (if made public by a user)), general interests of the user.
 - Stores all detailed object visit score/comment information uploaded from user PDA-s in a fairly large database.
 - Automatically fetches museum object lists and object information from museum web pages/object web pages, filters out RDFa information and caches this information in an RDF-schema database.
 - Contains semantic profile matcher / recommendation calculator.
 - Contains basic statistical matcher for finding similar profiles / measuring profile similarity.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Contains web reader and RDFa parser for automatically fetching semantic information about objects on the profile and objects in the museum and caching them in the database.
- Contains restricted-english parser for parsing user general preferences text.
- Contains OWL-based semantic concept information about art style taxonomies etc.
- Contains semantic derivation engine, capable of deriving generalizations, used for better matching.
- Called from either (both used):
 - For outside use/museum visit: PDA, sends text/maps to PDA.
 - For home preparations: browser of the user, as an ordinary web system for recommendations.
- User profile management web application enabling interested users to:
 - View the profile.
 - Edit the profile: add/remove like/dislike marks, add/change general preferences like “I like expressionism”, add/change comments.
 - Contains the same restricted-English parser for parsing user profile text as the central profile engine.
- Possibly a simple location server for finding nearby outside objects, from which the profile engine/suggestion calculator filters necessary information. Alternatively, we might be able to use google maps services for the same purpose: needs further investigation.

3.3 External systems used by the project

- Object information wiki servers:
 - Containing varied information (texts, pictures, audio) for museum objects and outdoors objects.
 - May optionally contain audio streams for presenting prepared audio (not common in wikis).

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform





smartmuseum

- May be custom-built web pages of museums or (both can be used)
- Wiki systems, based on either (both can be used simultaneously):
 - conventional Wikimedia software (ordinary Wikipedia can be used directly) with RDFa annotations can be added to standard wiki text (although this is a bit cumbersome).
 - existing semantically enhanced Wikimedia software for content management: no custom software built in the process required.
- Outdoor map server:
 - Should be capable of presenting a required map area with selectable points of interests nearby.
 - We will use Google maps (both the PC version and PDA version) and create mashups on these using google javascript API-s.
- Translation server:
 - Should be capable of translating short texts between various languages automatically
 - We will use Google online translation services.
- Gps worldwide system:
 - PDA should be GPS-enabled for outdoors use.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

4. User PDA/smartphone

User PDA/smartphone is one of the two main subsystems: the other one is the profile matching / recommendation server.

We will initially use a Windows-CE smartphone. We will also write parts of the software on Symbian smartphones, with the assumption that parts of the software can be ported to the Windows-CE implementation and the principles of user interface can be re-used in the Windows-CE implementation. In particular, a Java implementation is possibly a good path to go.

We also note that the smartphone implementation is not required to perform any computationally complex tasks, except an optional component of speech synthesis, which – if realistic – should be based on a ready-made software component and not be built in the scope of the current project.

Two different applications will be written for the smartphone:

- **End-user application for use outside and inside.** The outside and inside uses are fairly different in many aspects, hence the outside and inside applications could be considered to be almost two separate applications. However, several principles are shared (both applications give suggestions and information based on the user profile and user location) and several software components (like information presentation, choice of further information etc) are shared.
- **Museum administrator application.** This is used strictly by the museum workers writing information to RFID tags near the objects (pictures, rooms) and writing visual object numbers next to the objects to complement or replace RFID tags (user can type in object number and fetch information over web from the calculated URL).

Both applications should require user name/password to be entered. For the museum application the user name has to be registered in the central server, in order to allow data entry. The ordinary end-user has to invent a username and a password before starting to use the system. This username/password will be stored in the PDA, and there is no need to enter it again before each use. The user can also take a new username/password whenever she wishes. The PDA will store preferences for only one username at a time, but the old profile will be kept in the server as well, unless the user determines to delete it.

4.1 Location recognition



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

Location recognition outside is normally performed using gps. For the outside scenario we hence assume that

- the user PDA has an inbuilt GPS,
- is connected to a separate GPS module,
- or the user is manually typing in gps coordinates (obtained, say, from a hand GPS not connected to the PDA or obtained from the map application after user finds her location on the map).

Hence two separate location recognition ways have to be built: the continuous GPS-based location recognition and human-input recognition-when-ever-user-prefers.

The google maps “my location” service for phones should be considered as a possible third location recognition alternative outside: the “my location” phone service is based on GSM antenna location and may be available for users in some places. From the practical standpoint this is a least likely solution to be used and should probably not be implemented.

The actual object location – i.e. “I am now in front of the house where Galileo lived” – is also performed using the same location methods: whenever user wishes to get more information about a place, the PDA contacts the servers to find out whether she is in fact in the immediate neighbourhood of some concrete object about which she wants to get more information.

For the inside scenario we assume that the user either

- Puts her RFID-reader-enabled PDA near the RFID tag on the object, reading the contents of the tag, or
- Types in a short number visible near the object as a local object identifier.

None of these two ways typically gives an actual location of the user: both are used for recognizing the identity of the object and fetching information about the object (from RFID tag, from web or both).

In particular, we note that we are not trying to use any of these alternative approaches to location recognition:

- Using wifi by recognizing hotspots or performing triangulation-like calculations. We are using wifi in the system for web access whenever possible, but not for location finding.
- Using GSM-antenna-based-location-finding (either based on one antenna or doing triangulation), except when this method is made openly available by a global infrastructure system like google maps on the phone. The problem with GSM-antenna-based methods lies in major complexities and restrictions in the interests of privacy.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



For the inside scenario it must be also possible to indicate which museum we are in an whether we are just entering or leaving the museum.

These three tasks should be doable by both rfid-reading and manual input. Concretely:

- Upon entrance the user can read the “entrance RFID tag” which contain the general museum URL (museum identifier) and the conceptual indicator that we are entering the museum, hence recommendations should be fetched. Alternatively the user should be able to manually type in the full museum URL and select the “entrance” option from the menu.
- Upon exit the the user can read the “exit RFID tag” which contains conceptual indicator that we are exiting museum, hence updated profile should be sent. Alternatively the user should be able to select the “exit” option from the menu.

4.2 Reading and presenting object information

There are two ways for reading object information:

- Reading the object web page using the object URL. Basically this assumes using completely standard http and ordinary human-readable html pages. Obviously the PDA has to be able to reach web, either by wifi or gprs or edge etc.

The separate question here is how to find out the object URL. We implement this in two ways

- Reading object URL directly from the RFID tag (see the RFID data format chapter).
- Composing the object URL from the museum URL (already entered to PDA during entrance) and the object number written next to the object (local object id in the museum). The composed URL may be either the main URL of the object or just reference the main URL via indirection (see the object URL and identity management chapter for the composition and indirection details).
- Reading the basic information about an object directly from the rfid (see the rfid data format chapter). Obviously only a small amount of data (ca 1 Kbyte) can be put on the rfid, but this amount could be often sufficient. Whenever possible, the basic information should be first read from the rfid: this requires no web access and is thus probably faster and may be cheaper (if no free wifi is available).

In case the end user would have a full PC at hand while looking at the objects, we would simply use a common browser for showing the object web pages directly as they are (indeed, this is a proper way to surf the visited or interesting objects from home).

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



However, the PDA/smartphone screen is small, user has little time for each object, may prefer audio, may prefer information in a language different from what is given. Moreover, we want to update the user profile based on what information/for how long does she look at for an object.

Hence the object information presentation subsystem on the PDA is a specialized application, built for the project. The information presentation subsystem uses the common browser on the PDA for presenting selected chunks of information.

There are three main kinds of information that should be presented:

- **Outside map** with recommended outside objects. The application should use google maps for phone. The recommended objects should be calculated by the profile server, possibly using the special object location server for finding nearby objects. See the geolocation chapter for details.
- **Museum map** (best case) **or list** (normal case) of recommended rooms. In the best case the museum has put an annotated map of rooms in the server. The map location URL should be given with a suitable semantic annotation (see the html annotation format chapter) on the main museum web page. In case no map is available, a list of museum rooms should be given (museum room pages contain both general information and the list of room object URL-s).
- **Concrete object** information (painting, building, museum room, ...). Concrete object information is read from the object URL as html text, possibly containing links to additional material. Both the relevant html text snippets and relevant URL-s should be annotated (see the html annotation format chapter). Additionally, it should be possible to see comments entered from other users for this object, sorted by better matching profiles.

The PDA initially finds short basic audio clip and short basic textual information about the object and presents this information (text is preferably presented in the browser) along with buttons to select further information: longer audio, full descriptions, additional links etc. The order of preferences – audio preferred or not, etc - should be configurable by the user.

It is possible – or likely – that most objects do not have an associated audio clip on the server. In that case the PDA should – if technically possible - perform automatic speech synthesis from the relevant, filtered-out text. During the creation of the PDA software the option of finding and using the ready-made text-to-speech software component should be investigated for several PDA platforms.

It is also possible that the user has a clear preference (configured in the PDA software) for textual information in a language different from the language used on the web page: the PDA should determine the language of the page and use an external translation service – our recommendation is google translation service – for translating the text before presenting to the user.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



4.3 Contacting the museum and the recommendation servers

Protocol-wise the PDA uses standard http for fetching information from the museum server. In normal cases the text returned from museum server is ordinary – though annotated - html.

The recommendation server is contacted by the PDA for two purposes:

- Uploading the user profile file during or after visit for storage. See the user profile format chapter for the specification.
- Fetching recommendations before visit. To do this, the PDA has to:
 - First upload the user profile at the beginning of the visit for finding recommendations (profile is not stored at the server, unless the user PDA failed to upload the profile after the last visit: PDA has to keep track of this). Again, see the user profile format chapter.

The same request type can be used for both profile storage, outside recommendations (map or list of places to go) or inside recommendations (rooms to visit): input parameters and expected format of the result will be somewhat different.

We will use the standard encrypted https POST operation for user profile file upload. The POST request will pass these cgi parameters to the profile server:

- **Username:** name selected by the user in the PDA user interface.
- **Password:** password selected by the user in the PDA user interface.
- **resulttype:** expected result: value is either the word **storage** for pure profile storage (no recommendations) or **outdoors** or **indoors** for one of the two scenarios.
- **locationurl:** Museum (or some other place) URL to be visited: expected information is a list of recommended rooms in the museum. Parameter may be missing for outdoors use. However, it could be also possible to look for interesting nearby outdoor objects as if gps were given, in case the location server is capable of localising the place by the URL.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- **locationlatitude:** gps latitude as used in google maps for location. Used outdoors. Parameter missing if resulttype is not outdoors.
- **locationlongitude:** gps longitude as used in google maps for location. Used outdoors. Parameter missing if resulttype is not outdoors.
- **userprofile:** full user object visit profile file in the gzipped format.
 - As a result of uploading the profile, fetch the recommendations as a special annotated html file. This html file can be either shown by the PDA using a browser or further processed, should that be necessary. See the annotation format chapter for details.

4.4 User profile management

The main reasons for the user profiles in the system are:

- Giving recommendations based on the profile.
- Storing preference information for other people: recommendations are partially based on what people with a similar profile liked.
- Storing additional free-text comments/links for concrete objects for other people. Again, the order of comments while shown is based on what people with a similar profile liked.

Especially the latter two goals require storing user liking/disliking visited objects. There are several levels of user liking/disliking an object, hence the user liking/disliking score has to be presented as a numeric value, which can have both positive and negative values. Conceptually we will use fixpoint numbers with values from -1 to 1.

There are several ways to find out if a user liked or disliked a visited object:

- User presses like/dislike buttons (clear indication, but it is not likely most users will give such numbers).
- User fetches information about an object. This clearly indicates some – likely positive – interest.
- User fetches additional/large amounts of information about an object or spends a lot of time investigating an object (long period of time between interactions for that object). This indicates serious positive interest.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

- User visits a room, likes some objects in the room, fetches no information about other objects in the room. This could be interpreted as a very slight dislike of these other objects present in the room.

The exact meaning of the like/dislike number is a target of further investigations, but the relevant levels are on the following lines:

- 1: strong like, manually input.
- 0.7: user fetches a lot information about the object.
- 0.3: user fetches basic information about the object.
- -0.01: user visits room with the object, shows no interest.
- -1: strong dislike, manual input.

The profile should also keep track of an approximate time of storing like/dislike information about the object, for the reason that the preference becomes less trustworthy over time. For privacy reasons it would be not suitable to send very detailed information to server, unless user explicitly ok-s that. The timestamp granularity (none, year, year/month, date, date/time) should be configurable in the PDA.

The user object visits profile is unlikely to grow huge, especially since the the PDA only stores the part of the object visit information which has not yet been uploaded to the server. For better transparency of the system (privacy issues) and easier reading/modification of the profile by other applications we will store the profile as a plain text file. See the user profile format chapter for the specification.

The profile can be seen and modified by the user: we will use a separate web application for this. Note that the full profile is kept in the server, no updating of the profile part on PDA (when profile on server is changed) should be done.

4.5 SmartContent provisioning and finding people on location for chatting (using Bluetooth)

Two kinds of additional functionalities will be added/related to the PDA connectivity:

- SmartContent provisioning
- Finding other people



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

By SmartContent provisioning we assume both context sensing and efficient delivery over WLAN.

As indicated in the indoor scenario for John, the PDA application should make it possible to find people on location who are interested in direct chat on commonly interesting subjects.

Finding other people should probably function over bluetooth, but other possibilities should be explored as well.

It should be possible to turn on advertising in the PDA. As a result, the PDA should fetch a list of users who are currently likely to be on location (based on last recommendation asking for the same spot) and who match the criteria for chatting (common interests / similar general preferences / suitable personal profiles etc) and who have indicated similar interest.

The profile/recommendations server should find suitable matches and order them based on the goodness of the match.

The user should be able to indicate his/her whereabouts (drinking coffee on the second floor, man with a red hat) in free text which will be advertised to other users on the same location.

The user PDA should start getting advertisements of other people as well as sending out its own advertisement.

The concrete methods, profile matching principles and general technology should be worked out in a suitable workpackage.

5. Central profile engine/suggestion calculator

The central profile engine/suggestion calculator is the main, crucial component of the system. Finding effective and efficient means for suggestion calculation is one of the main scientific tasks for the project.

It should offer the following interfaces:

- Input of profile for storage (for the reason of storing preferences), see the PDA chapter and profile chapter for transmission and profile formats.
- Input of profile along with location identification for recommendation calculation, resulting with the recommendations, see the PDA chapter and profile chapter for transmission and profile formats.
- Computing and giving statistics for museums (password required, museum administrators must be registered).
- Administration possibilities for the central system administrator.

The central profile server has to also conduct identity management for the cases when the base URL of some items is changed: see the object URL and identity management chapter.

5.1 Storing profiles

The engine should be able to keep all the uploaded profiles.

The suggested way to this is to keep the profile information in an ordinary database with the following schema, which contains one huge **score table** for all object visit/like/dislike score events, a smaller **profile table** for simply connecting the one-profile-upload fields in the score table, **user personal information table** and a **user preferences** table containing general user-entered preferences entered via the web application.

For example, suppose Martin uploads an object visit profile (with a username *museumshark*) containing only two items:

| Object URL | score | comments | time |
|------------|-------|----------|------|
|------------|-------|----------|------|

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



http://en.wikipedia.org/wiki/Mona_Lisa 0.5 2007

http://en.wikipedia.org/wiki/The_Raft_of_the_Medusa 0.8 “cool! See <http://myblog.com>” 2007

This will create two rows in the score table, both with a new row id number, one containing the Mona Lisa URL along with like/dislike score 0.5 and year of seeing 2007, another containing the Medusa URL along with the like/dislike score number 0.8, the comment string and the year of seeing 2007.

In case no existing *museumshark* profile is found (user has not used the system before or has has created a new profile), a new internal profile number N along with a new profile table row is generated, username and password hash are added to this row, both beforementioned score table rows will have the profile id field set to N. In case an existing profile is found for *museumshark*, this profile number is used as N instead.

- **Profile table:** one record per one uploaded profile, with fields
 - Internal profile id number, each profile gets a new id number
 - Username
 - Password hash
 - Storage timestamp: when was profile stored/last updated
- **Score table:** one record per each item, with fields:
 - Internal score row id number (each row has a new id number)
 - Corresponding profile id from the profile table for associating score rows of one profile
 - Internal string resources table id for object URL (URI or string id for the actual painting, building etc), see the object information table from the next chapter
 - Like/dislike score number for the object URL stored in the same row
 - Free text string with the user comment, must be of a restricted length, normally missing
 - Date obtained from profile (normally just year, may be missing), indicating seeing time

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- **User personal information table:** one record per each item, un-encoded RDF-style with fields:
 - Internal personal information table row id number (each row has a new id number)
 - Corresponding profile id from the profile table.
 - User personal information property name (first name, second name, sex, birthday (year), preferred language, other languages, country, email, web page etc: entered only in case the user explicitly enters this information. User may fake this information at will.
 - User personal information property value (John, Smith, male, john@gmail.com, <http://john.com>,)
 - Property datatype as xml schema string (string, URL, date, integer, ...)
- **User preferences table:** one record per each item, with fields:
 - Internal preference table row id number (each row has a new id number)
 - Corresponding profile id from the profile table for associating score rows of one profile
 - OWL expression for the preference.

Note that the user preferences table encoding does not necessarily have to be the OWL expression: we could use a specially processed form for speed reasons. The exact way to store/process/use the OWL expressions has to be handled in a corresponding work package as a research task.

5.2 Fetching and storing object information from (museum) web servers

The main reasons for fetching object information from museum/object web servers to the central server are:

- We have to know a list of rooms/objects in the museum in order to give recommendations to the server.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

- We have to know object information in order to use the general preferences of the user (and to some extent, to improve statistical matching based on visit scores, although this aspect is less important).

This information could, in principle, be fetched before the recommendations calculation for each museum visit for each user, but this would be very time-consuming and unnecessarily repetitive. Instead, the central server should contain a caching table for museum rooms/objects listings and object information.

The big picture of information fetching is as follows:

- Whenever information is required for some museum/object, try to find it from the corresponding central server tables and check whether the last update time is sufficiently recent (update time should be configurable, recommended value one month).
- If up-to-date information found in the database, use it.
- If up-to-date information is not found, read the museum URL and object URLs, filter out rdfa-encoded rdf data, store them in the cache along with the current timestamp.
- If new data should be stored but the size of the table exceeds preconfigured limits, throw away a preconfigured percentage (default, say, 20%) of the oldest object information table rows.

The museum room/object listing and object information data should be stored in the rdf-style database tables:

- **Object information table:** one record per each item, RDF-style with fields:
 - Internal object information table row id number (each row has a new id number)
 - Internal object URL string id number in the resources table.
 - Internal property name string id number in the resources table.
 - Internal value string id number in the resources table.
 - Internal value type (as XML schema expression) string id number in the resources table.
 - Timestamp of last addition/modification.
- **Resources table:** one record per string, for storing strings uniquely:
 - Internal string id number, automatically increased for each string

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Actual string

The motivation for using a separate resources table in the museum/object information database stems from:

- the need to find matching objects quickly (integer index better than string), especially when matching user profiles.
- Large amount of repetition of strings in the object information table: integer indexes consume less space than actual strings.

The string table is expected to grow relatively big, yet we expect that a special garbage collection/reindexing procedure for the string table (and tables using the string table) is not required: the user personal details and user preferences - which vary a lot – are not kept in the string table, whereas the object URL-s, property names and values vary less. It should be possible to keep all of them during the entire lifecycle of the system.

The forthcoming “Object URL and identity management” chapter contains further information object URL and identity management. In particular, it recommends using a special URL identity table for quick identities check. Here we suggest the schema for URL identity management:

- **URL identity table** contains museum-managed information for identifying the situations where several URL-s point to the same object.
 - Internal URL identity table information table row id number (each row has a new id number)
 - Internal proper object URL string id number in the resources table (should prefer that for new object additions)
 - Internal identical URL string id number in the resources table (should map to proper URL).
 - Museum main URL: indicates the museum who has a right to manage the information
 - Timestamp of last addition/modification.

Observe that that the proper URL and identical URL are really symmetric for matching purposes. Also note that the identity classes created by the URL identity table can be arbitrarily large, although we expect the URL identity table to be relatively small for practical purposes. In principle it would be possible to encode the URL identities in the common object information table, but this would be somewhat inefficient and harder to manage.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



In addition to the URL identity table we will also need management information tables for people with a right to modify this table for specific museums and setting up main museum pages, etc. Such administrative structures should be determined and built during the actual design phase of the main server system.

Designing/selecting the actual RDF property names and principles for storing museum room/object listings and object information is a task which should be solved in a corresponding work package, together with the annotation details for the museum room/object web pages.

5.3 Calculating recommendations

The main idea of recommendation calculations is the following: combine purely statistical search for similar profiles and their likes/dislikes with derived and general preferences. For outside use, take into account the weather and museum opening times, if available.

For example, the simplest statistical search method will find the profiles with the best matches for likes/dislikes of common items (somebody has been to the museum I am entering and she has also been in some of the museums I have and liked/disliked similar objects). It will then base the recommendation on the concrete item likes/dislikes of the matching profiles.

A well-known project encouraging competition between different algorithms for a similar task is a Netflix prize competition <http://www.netflixprize.com/> where the algorithms have to give movie recommendations based on the previous preferences of the user and the preference profiles of other users:

Upon registration, Participants may access the Contest training data and qualifying test sets. The training data set consists of more than 100 million ratings from over 480 thousand randomly-chosen, anonymous customers on nearly 18 thousand movie titles. The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received by Netflix during this period. The ratings are on a scale from 1 to 5 (integral) stars. To protect customer privacy, all personal information identifying individual customers has been removed and all customer ids have been replaced by randomly-assigned ids. The date of each rating and the title and year of release for each movie are provided. No other customer or movie information is provided. No other data were employed to compute Cinematch's accuracy values used in this Contest.

On the other end of the spectrum, no such matches might be found (say, the user profile of visits is empty) but the user has indicated some general preferences ("I like impressionism". "I dislike expressionism"). Based on the RDFa-marked semantic information fetched from the concrete item web pages, suggestions can be made.

In the middle of these two ends of the spectrum lie automatic generalisations and cluster analysis.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

Since both pure general-preference method and possible improvements for automatic generalisations need categorisation information about concrete items in the museum to be visited, the server should read large amounts of categorisation meta-information from the rdfa-annotated web pages of objects. This was explained in the previous section about fetching and storing object information in the central server.

The recommendation server has to also contain ontologies and concept definitions. All these should be written either in OWL or controlled English sentences translatable to OWL. Suggested language for the latter is ACE: Attempto Controlled English. The server software should employ existing ACE-to-OWL translator and OWL-to-ACE translators (see the ATTEMPTO web pages).

The exact details of OWL and ACE usage in the project should be a task for the ontologies workpackage.

For outside use we should take into account the external sources for

- Current weather and weather forecast. If possible, the geolocation server should give the recommendations calculator information about weather, read from some of the publicly available weather servers/web pages for the current location.
- Museum opening times. If possible, the preferences server should use museum opening times meta-information possibly available from suitably annotated museum web pages. This information should be kept in the cached object information table with a selected property name.

Both the weather forecast fetching methods (dependent on the selected weather server) and museum opening times annotation guide (using rdfa) for the museum web page should be given during the actual design of the recommendation application.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

6. User PC application

As indicated before, the user PC application should enable the user to:

- Log into the system using her username/password.
- See and understand the profile file.
- Browse the profile file and visit the web pages of objects.
- Modify preferences of objects, if the user so wishes.
- Enter, understand and modify general preferences (I like impressionism. I dislike expressionism. I like 20-th century art) if the user so wishes.
- Fetch the recommendations for any museum/location exactly as it is possible from the PDA.
- Enable to find users willing to personally communicate on the spot.

Recall that the user profile in the web server is obtained from the PDA by the profile upload from the PDA.

The user profile should be presented in an easy-to-understand, sortable, searchable manner. It must be possible to click on an object and

- See its ordinary web page in the browser (no special handling of the web page is necessary, differently from the PDA).
- Change the like/dislike score value of the object.
- Add or change free-text comments to the object (possibly containing links)
- Delete the object from the profile, delete the range of objects from the profile.
- Remove exact date information (keeping only years or even deleting this information).

In order to make it possible to enter and modify general preferences, the application has to be able to parse the preferences part of the profile (again, see the profile chapter) .

7. Geolocation server

The geolocation server has to perform several tasks:

- Find the outdoors object URL corresponding to the current location given by gps.
- Find nearby outdoors object URLs (interesting houses, museums, etc) in the object information table.
- Create a google map using page for the PDA with the interesting nearby objects suitably marked up, using the google maps api.
- Fetch weather information for the current location.

All these tasks should be performed as separate web services, preferrably using a simple CGI get format with gps latitude and longitude parameters as input and a suitable text format output for the preferences server: either as a list of URLs, a web page string or suitably encoded weather information.

Since the geolocation server must have access to the object URL-gps coordinates mapping table, it could be physically implemented inside the central recommendation server machine. However, since it is likely that the set of interesting outside objects could be larger than the information contained in the SmartMuseum system (say, locations of restaurants, possibly not present in the SmartMuseum system), it should be implemented as autonomously as possible.

It is also possible that during implementation a suitable existing system is found which can provide the list of URLs for interesting nearby objects out-of-the-box. In such case it is recommended to use the existing system. At the time of writing no such good universal system is known to the authors of the architecture document, hence the suggestion to build the geolocation services specially for the project.

We will give some proposals for implementing the main task, nearby object finding. Finding the URL located at the current gps position can be seen as a specialised subtask, where the search radius is very small, say (100 meters), and fetching the closest matching URL.

As said, the web service should accept a http get with the gps latitude and longitude parameters. The exact format of the coordinates has to be determined during implementation (the amount of existing coordinate presentation systems is very large). Two approaches are possible:

- If we would recommend to use the Google maps location format, which means that the PDA application should access the GPS location automatically from a GPS-enabled PDA (or it should accept manual input) and it should convert this location to the Google maps format.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



- Alternatively, the service could accept several formats and should guess from the syntactic form of the format the type of conversion necessary for Google maps format. It should then use this conversion.

Next, the server must have access to interesting location URLs (used as their id-s in the SmartMuseum system) and their Google maps coordinates. It should certainly have direct access to the recommendation server database object information table to use information there (replication would be complex and unnecessary). The recommended location format in the object information database is simply two triplets, like in this example:

Location_url, http://www.w3.org/2003/01/geo/wgs84_pos#long, -87.655788

Location_url, http://www.w3.org/2003/01/geo/wgs84_pos#lat, 41.947372

Notice that the VRA Core 4 does not provide markup for geographic coordinates, hence we have to use our own property names. We recommend to use the namespace http://www.w3.org/2003/01/geo/wgs84_pos# and property names geo:long and geo:lat, using the Google maps coordinates as values.

In case it is possible to access external object information in addition to the object information database in the SmartMuseum system, this information should be either

- cached in the geolocation server, using an rdf-type database with the same schema and the same caching behaviour as the object information database in the system.
- or kept in the same object system information database, together with the internally managed information.

The next step for the geolocation server is to find a list of nearby object URLs in the radius of X meters and present this information to the recommendation server as a list of N URLs together with coordinates and distance from the current location in meters, preferably sorted by distance:

Location1, 87.6566, 41.947344, 500

Location1, 87.6579, 41.947234, 1200

... etc ...

The maximal amount of objects and the maximal search radius X should be configurable in the geolocation server.

Finding objects in a given radius can be performed by scanning the object database for objects inside a certain pre-determined latitude/longitude rectangular window and then using trigonometry for calculating the distances, finally using the calculated distances for final filtering and sorting.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



Since several databases present special fast geographical search functions for finding and sorting objects based on their distance, using such special methods is an option. However, in that case it may occur that a special database schema is needed (for example, keeping both latitude and longitude on the same row) which can be achieved by either building a special geolocation cache table or special views. These operations may require additional overhead, however. Hence the initial suggestion is to write a simple search based on the existing database and then create the special structures if the initial approach appears to be too slow.

Looking for usable external geolocation data sources and services (if any), continued by the actual design of the geolocation server is a task for a suitable workpackage.

8. Object URL and identity management

As noted before, all objects will be identified by global identifiers, i.e. URI-s. Since object information is fetched from the server, the uri in our system is always an URL: the web page of the object.

Observe that even in case object information is fetched by the user directly from the rfid, it should be available on the server first so that the museum administrator PDA can fetch the information for processing and writing the relevant snippets to the rfid.

The object web page can be stored on the museum server, generally used wikipedia, centrally set up semantic wiki for the project, or some location: the actual server where the web page is located makes no difference for the system.

The user PDA can get the object identification in three ways:

- Reading the full URL from the RFID tag. Data is then fetched directly from this URL.
- Typing in a short number, which is a local id of the object in one particular museum. This local id must then be mapped to the full URL. We will use the following schema for the mapping:
 - By default, the full URL is obtained by concatenating museum main URL, string “/items/” and the short number. For example: museum URL is <http://www.museum.com>. Short number is 1234, the full URL is <http://www.museum.com/items/1234>
 - In case the museum main page contains an rdfa-given special value with the name **itembaseurl**, the full URL is obtained by concatenating this special value and the short number.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

Observe that the museum can set up indirection web pages for the items: the page <http://www.museum.com/items/1234> may refer to, say, a different page on, say, ordinary wikipedia. Hence the PDA application must be capable of resolving standard page redirection methods (http headers or the indirection meta field in the html header), just like ordinary web browsers do. However, the system-wide identifier for the object (i.e. the URL kept in the profile) remains the original URL, not the URL referred to.

- Computing the URL from the gps location outside: this is a task for the geolocation subsystem, see the previous chapter.

It is possible that over time the base part URL of the object changes: for example, the main museum URL like <http://www.museum.com> is not available to the museum any more, hence it is also not possible to set up a relevant redirection link. The museum has, say, changed its base address to <http://www.newmuseum.com>. In that case it must be possible to indicate that all items previously under <http://www.museum.com> are now equivalent to the same items under <http://www.newmuseum.com>. This equivalence is necessary for using old item URLs in profile matching.

In order to set up the equivalence the central profile server must contain a special equivalence table, creating correspondence between several base URLs. Administratively the equivalence table gets a new equivalence only in case the museum administrator tells the central server administrator about the new base URL. See the server databases chapter for a suggested schema.

When such an equivalence is entered, the profile table is modified by changing all matching old item URLs to start with the newest base URL for this equivalence.

When a profile is obtained by the server, it is first normalized: the matching check is performed for all URLs in the profile and for any matches the corresponding replacement is performed.

9. Annotation format for HTML pages about objects, rooms, museums

The HTML annotation format lies on three pillars:

- **the RDFa standard** of W3C for putting semantic markup into ordinary html (or xhtml or xml) text. Processing marked up text gives us standard rdf triples, essentially property/value pairs for the objects.
- **The VRA Core 4** elements as property names.



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

- **A number of cultural concepts and ontologies** for these concepts, worked out in the project. The concepts are meaningful words which can be used in the markup: software will not understand/use the markup which has no ontologies defined. The ontologies themselves are not really a part of markup, however: they are used in the server for profile matching and recommendation calculation.

We will normally assume that only a subset of RDFa and VRA Core 4 is used in our markup. However, software should be capable of handling full RDFa.

9.1 VRA Core 4, other approaches

VRA stands for “Visual Resources Association”. The VRA Core is a data standard for the cultural heritage community. It consists of a metadata element set (units of information such as title, location, date, etc.) which is directly used in the SmartMuseum project, as well as an initial blueprint for how those elements can be hierarchically structured, which is only indirectly used in the SmartMuseum system (we use the RDFa markup of html instead of plain XML hierarchies in the VRA Core).

Historically, VRA is specialization of Dublin Core for visual resources. For example, VRA properties “material.medium” and “material.support” are specializations of Dublin Core property “format”.

In other words, the VRA Core gives for the SmartMuseum project a concrete list of words for semantic markup, along with human-readable definitions: what each word should mean in a concrete markup. The SmartMuseum project uses VRA Core in two ways:

- As concrete words in the RDFa annotated-html: see the next chapters for an example.
- As concrete words in the (OWL) rules created in the SmartMuseum project, establishing some relations between the concepts marked by these words.

Since data in The SmartMuseum system is marked following the RDFa standard, it does not directly use the XML format suggested in the VRA Core 4. However, as RDFa is also a hierarchical markup language, the XML hierarchies suggested in the VRA Core 4 can be mapped to the RDFa hierarchies. Since such hierarchies can be somewhat complex for the annotator, we suggest to use as few hierarchical annotations as possible. In normal cases flat annotations should be used, which can be mapped to VRA Core 4 in the software using ontologies, should the need arise.

Notice that despite VRA Core defines several sub-element relations, it is itself not a full ontology system with rules and taxonomies. In particular, there is no immediate, simple match between the VRA approach and RDF/OWL approach for structuring complex data. Hence this mapping and

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



concrete usage principles have to be worked out in the project. One attempt towards this direction is available as <http://www.w3.org/2001/sw/BestPractices/MM/vra-conversion.html>. For other case studies, see http://www.edlproject.eu/conference/downloads/EDLconf_Schreiber.pdf

In order to illustrate different approaches to markup, we will bring up the CIDOC CRM model, which will not be directly used in the SmartMuseum project.

An important standard providing more ontology-friendly structuring of cultural heritage information is CIDOC Conceptual Reference Model (CRM). The CIDOC CRM provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation.

The CIDOC CRM is intended to promote a shared understanding of cultural heritage information by providing a common and extensible semantic framework that any cultural heritage information can be mapped to. It is intended to be a common language for domain experts and implementers to formulate requirements for information systems and to serve as a guide for good practice of conceptual modelling. In this way, it can provide the "semantic glue" needed to mediate between different sources of cultural heritage information, such as that published by museums, libraries and archives.

The CIDOC CRM is the culmination of over 10 years work by the [CIDOC Documentation Standards Working Group](#) and [CIDOC CRM SIG](#) which are working groups of [CIDOC](#). Since 9/12/2006 it is official standard [ISO 21127:2006](#).

For a detailed presentation of fairly complex capabilities of CIDOC CR see http://cidoc.ics.forth.gr/docs/crm_for_ER2007.ppt as well as the set of CIDOC publications http://cidoc.ics.forth.gr/official_release_cidoc.html

A good overview of different cultural heritage description standard drafts is given in <http://cidoc.ics.forth.gr/docs/fin-paper.pdf>. It also proposes its own standard draft CCO, with the following example:



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

See <http://www.w3.org/TR/xhtml-rdfa-primer/> for the introduction and <http://www.w3.org/TR/2007/WD-rdfa-syntax-20071018/> for further details.

These property/values can be easily converted to formal RDF triples which are normally stored in the database. In case no object id is marked which has these property/value pairs added, the URL of the web page is automatically considered the object. This is a standard case for object web pages: the web page is the object id and property/value pairs go for this object.

See the following example from the rdfa primer on W3C site, giving clear, formal rdf information about an object <http://example.org/staff/jo>

- Fn is Jo Smith
- Title is Web hacker
- Org is <http://example.org>
- Email is <mailto:jo@example.org>

Observe the red text: this is rdfa semantic markup entered into an ordinary html:

...

```
<p class="contactinfo" about="http://example.org/staff/jo">
```

```
<span property="contact:fn">Jo Smith</span>.
```

```
<span property="contact:title">Web hacker</span>
```

at

```
<a rel="contact:org" href="http://example.org">
```

```
Example.org
```

```
</a>.
```

You can contact me



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

```
<a rel="contact:email" href="mailto:jo@example.org">
```

```
  via email
```

```
</a>.
```

```
</p>
```

...

See also a full example from the same primer:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/Markup/DTD/xhtml-rdfa-1.dtd">

<html xmlns:cal="http://www.w3.org/2002/12/cal/ical#"
      xmlns:contact="http://www.w3.org/2001/vcard-rdf/3.0#">

  <head>

    <title>Jo's Friends and Family Blog</title>

  </head>

  <body>

...

  <p instanceof="cal:Vevent">

    I'm holding

    <span property="cal:summary">

      one last summer Barbecue,

    </span>

    on

    <span property="cal:dtstart" content="20070916T1600-0500">

      September 16th at 4pm.

    </span>
```

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



...

```
<p class="contactinfo" about="http://example.org/staff/jo">
  <span property="contact:fn">Jo Smith</span>.
  <span property="contact:title">Web hacker</span>
  at
  <a rel="contact:org" href="http://example.org">
    Example.org
  </a>.
  You can contact me
  <a rel="contact:email" href="mailto:jo@example.org">
    via email
  </a>.
</p>
...
</body>
```

Again: for detailed specification we refer to the W3C RDFa pages.

9.3 RDFa and VRA core usage example and principles

As a case study we will present a simple version of adding RDFa descriptions to Mona Lisa entry in the wikipedia.

The main idea of markup principles is to make it easy for markup writer, i.e. orienting the markupp content for human use, not so much for further machine processing and derivations. Markup property names and object id-s (typically URL-s) on the rother hand are formal and machine-processable.

The wikipedia entry http://en.wikipedia.org/wiki/Mona_Lisa starts with a brief description:



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

Mona Lisa (also known as **La Gioconda**) is a 16th century [portrait](#) painted in [oil](#) on a [poplar panel](#) by [Leonardo Da Vinci](#) during the [Italian Renaissance](#). The work is owned by the [French government](#) and hangs in the [Musée du Louvre](#) in [Paris, France](#) with the title **Portrait of Lisa Gherardini, wife of Francesco del Giocondo**.
[\[1\]](#)

The painting is a half-length portrait and depicts a woman whose expression is often described as enigmatic. [\[2\]\[3\]](#) The ambiguity of the sitter's expression, the monumentality of the half-figure composition, and the subtle modeling of forms and atmospheric illusionism were novel qualities that have contributed to the painting's continuing fascination. [\[4\]](#) Few other works of art have been subject to as much scrutiny, study, mythologizing and parody. [\[5\]](#)

This description is later followed by a longer description and a wealth of other materials. We will take the short part above, drop some of the less relevant markup and add new RDFa markup directly to the wikipedia article source. The added information is given in **bold red**.

First, the global id of Mona Lisa will be assumed to the same wikipedia URL http://en.wikipedia.org/wiki/Mona_Lisa, hence all the properties without the “above” context are assumed to be attached directly to this URL.

Second, observe that the local references in the source are mostly given to the wikipedia URL root, for example the Leonardo link `Leonardo da Vinci` points to http://en.wikipedia.org/wiki/Leonardo_da_Vinci which we will again consider to be the global id for Leonardo.

Third, we will use the `smm` prefix for smartmuseum namespace, defined in our example in the body as `xmlns:smm="http://www.smartmuseum.eu/"`. The VRA core names are a subset of `smm` names. Most of the `mms` /VRA names in the example are also Dublin Core names.

Fourth – perhaps most importantly – we will take an approach towards VRA Core 4 where we avoid hierarchical markup i.e. subelements (for example, giving measurements `s` as a tuple of horizontal/vertical length, where each length contains is again a tuple of a number and measure indicator). Instead, we will annotate the whole human-friendly measurement string “77 × 53 cm, 30 × 21 in”. It would be hard to split this human-friendly string into components by the software, but in the context of this project such conversions are not needed. Same goes for date: instead of using an upper-level date property with two subproperties for `earliestDate` and `latestDate`, we will simply use the latter two directly. Should we have a single date available, we would use that. We will also not care much about the date format (months, days, etc), assuming this is normally not available anyway.

Fifth, observe that whenever possible we use both a link to the object id (example: Mona Lisa author is pointed to as http://en.wikipedia.org/wiki/Leonardo_da_Vinci) and its name (Leonardo da Vinci). The latter (name in our example) could be fetched from the link contents, but this is time-consuming,

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



hence it is better to give this redundant information directly in the text. To accomplish this, we add both a rel attribute using the href contents as a value and a property attribute using the text string inside tag: `Leonardo da Vinci`

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body xmlns:smm="http://www.smartmuseum.eu/">
    <h1 property="smm:title" class="firstHeading">Mona Lisa</h1>
    <span rel="smm:image">
      
    </span>
    <tr>
      <td><i><b>Mona Lisa</b></i><br />
      (Italian: <i><b>La Gioconda</b></i>, French:<i><b>La Joconde</b></i>)</td>
    </tr>
    <tr>
      <td><b><a rel="smm:agent" property="smm:agent" href="/wiki/Leonardo_da_Vinci"
      title="Leonardo da Vinci">Leonardo da Vinci</a></b>, circa <span
      property="smm:earliestDate">1503</span>-<span
      property="smm:latestDate">1506</span></td>
    </tr>
    <tr>
      <td><a rel="smm:technique" href="/wiki/Oil_painting" title="Oil painting">Oil on
      poplar</a></td>
    </tr>
    <tr>
```

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

Page 45



```

<td property="smm:measurements">77 × 53 cm, 30 × 21 in</td>

</tr>

<tr>

<td><a rel="smm:location" property="smm:location" href="/wiki/Mus%C3%A9e_du_Louvre"
class="mw-redirect" title="Musée du Louvre">Musée du Louvre</a>, <a
href="/wiki/Paris" title="Paris">Paris</a></td>

</tr>

</table>

<p property="smm:description">

<i><b>Mona Lisa</b></i> (also known as <i><b>La Gioconda</b></i>) is a 16th century
<a href="/wiki/Portrait_painting" title="Portrait painting">portrait</a> painted in
<a href="/wiki/Oil_painting" title="Oil painting">oil</a> on a <a
href="/wiki/Poplar" class="mw-redirect" title="Poplar">poplar</a> <a
href="/wiki/Panel_painting" title="Panel painting">panel</a> by <a
href="/wiki/Leonardo_Da_Vinci" class="mw-redirect" title="Leonardo Da
Vinci">Leonardo Da Vinci</a> during the <a href="/wiki/Italian_Renaissance"
property="smm:stylePeriod" title="Italian Renaissance">Italian Renaissance</a>. The
work is owned by the <a href="/wiki/Government_of_France" title="Government of
France">French government</a> and hangs in the <a href="/wiki/Louvre"
title="Louvre">Musée du Louvre</a> in <a href="/wiki/Paris" title="Paris">Paris</
a>, <a href="/wiki/France" title="France">France</a> with the title <i><b>Portrait
of Lisa Gherardini, wife of Francesco del Giocondo</b></i>.<sup id="cite_ref-
Louvre.fr_0-0" class="reference"><a href="#cite_note-Louvre.fr-0" title="">[1]</
a></sup></p>

<p>The painting is a half-length portrait and depicts a woman whose expression is
often described as enigmatic.<sup id="cite_ref-paris1_1-0" class="reference"><a
href="#cite_note-paris1-1" title="">[2]</a></sup><sup id="cite_ref-ns1_2-0"
class="reference"><a href="#cite_note-ns1-2" title="">[3]</a></sup> The ambiguity
of the sitter's expression, the monumentality of the half-figure composition, and
the subtle modeling of forms and atmospheric illusionism were novel qualities that
have contributed to the painting's continuing fascination.<sup id="cite_ref-3"
class="reference"><a href="#cite_note-3" title="">[4]</a></sup> Few other works of
art have been subject to as much scrutiny, study, mythologizing and parody.<sup
id="cite_ref-4" class="reference"><a href="#cite_note-4" title="">[5]</a></sup>

</p>

</body>

</html>

```

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



This annotation will give us the following RDFa triplets (value in <...> marks object id, normally a href link):

```
<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:image
<http://upload.wikimedia.org/wikipedia/commons/thumb/8/85/Mona\_Lisa.jpeg>

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:title "Mona Lisa"

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:agent
<http://en.wikipedia.org/wiki/Leonardo\_da\_Vinci>

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:agent "Leonardo da Vinci"

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:earliestDate "1503"

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:stylePeriod "Italian Renaissance"

... etc ...

<http://en.wikipedia.org/wiki/Mona\_Lisa> smm:description " <i><b>Mona Lisa</b></i> (also ...
the whole text until end ..."
```

9.4 Wiki and semantic wiki

In case the museum does not have its own item information server already, we suggest to use mediawiki software for item web pages. The html has to be enriched by RDFa annotations as described before. It is even possible to use generally available wikipedia and not set up own server at all!

Adding RDFa to mediawiki pages is possible, but cumbersome: html source must be entered and several restrictions will apply. Hence we suggest setting up a special version of mediawiki: semantic mediawiki, which makes applying RDFa markup easier. The concrete choice of the semantic mediawiki software and the description/manual of its use (and the use of ordinary mediawiki with all the restrictions) has to be given.

10. Data format on RFID tag



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

We assume the RFID tag will contain about 1Kbyte of data. We will use this data for two kinds of information:

- Object URL (always)
- Object basic data (author, name, ...) and very short description/information.

We will not use a special format for RFID data: instead, we will use RDFa-annotated html just like the web pages. Since space must be conserved, we will use special simplified conventions when writing RDF-a marked up html to rfid.

First, we must add the object URL to html. We assume that whenever of the following two forms of markup are present in html., they represent the URL of the object:

```
<base href="http://example.org/john-d/" />
<span xml:base=" http://example.org/john-d/">
```

Second, we make the space-conserving simplification: we allow to skip all namespaces from the property names and will automatically assume that the namespace of any non-namespaced property name is the standard namespace of project concepts (to be determined). For example, instead of a definition of namespace for the abbreviation contact and the full form

```
<span property="contact:title">Web hacker</span>
```

we will use just

```
<span property="title">Web hacker</span>
```

The following is an example text adapted from the earlier annotated wikipedia text for Mona Lisa. The length of the snippet is below 1024 bytes, hence it can be put on the rfid even without gzipping first.

Notice that the user PDA should be able to present the full link to the article given in the xml:base attribute and the special audio links given at the end of the text. Both these and the internal, ordinary links in the body should be presented to the user either as an ordinary clickable text or under “additional information” selection, depending on the UI implementation.

```
<body xml:base="http://en.wikipedia.org/wiki/Mona_Lisa">
<b property="title">Mona Lisa</b> (also known as <b>La Gioconda</b>) is a 16th
century portrait painted in oil on a poplar panel by <a
href="/wiki/Leonardo_Da_Vinci" rel="agent" property="agent">Leonardo Da Vinci</a>
```

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



circa 1503–1506 during the [Italian Renaissance](/wiki/Italian_Renaissance).

The painting is a half-length portrait and depicts a woman whose expression is often described as enigmatic. The ambiguity of the sitter's expression, the monumentality of the half-figure composition, and the subtle modeling of forms and atmospheric illusionism were novel qualities that have contributed to the painting's continuing fascination. Few other works of art have been subject to as much scrutiny, study, mythologizing and parody

</body>

Third, we require the text on the rfid to be written gzipped. This will typically give ca 50%-80% of space reduction for 1Kbyte RFID.

As noted before the example, the user PDA software has to recognize a number of property names in the markup for intelligent presentation (title, year, longer text URL, audio clip URL etc). The full list of these names has to be worked out in conjunction with writing the presentation software for the PDA.

11. User profile format, sending profile to server

We will use a simple plaintext file for user profile kept in the PDA. The PDA keeps the last entered username and password separately from the profile file. The plaintext file contains concrete objects liked/disliked. This data is automatically obtained from museum or location visits and possibly extended by the user by

- adding/changing like/dislike scores manually
- adding free-text comments, possibly containing links (could be also just a link).

User may – if she so wishes – give personal details in the profile. No guarantee exists for their presence or correctness. If available, this information can be used for enhancing profile matching and possibly for finding parties interested in direct chat on the location.

The general preferences of the user - what he/she likes or dislikes, not mentioning concrete objects – is kept only on the web server and managed by the user via the web application. Most users will probably never enter this data.

Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform



Whenever the profile is sent to the server from the PDA, the server will either add new object rows to the existing profile or (if the username is not found among profiles) add a new profile before the objects are stored.

When the server successfully stores the profile and the PDA gets back a success identifier, the PDA will empty the user profile file in the PDA: from now on these profile entries are only kept on the server.

11.1 Personal details

The user should be able to add her personal details to the profile and indicate whether these are made public or not. These include name, address, sex, birthdate, country, email, web etc.

Some users may be willing to make these details public and searchable. The default should be “not public”. Even in the latter case the details can be used for enhancing profile matching.

We assume that the general preferences can be entered only via the web application, not the PDA application. Of course, in case the user PDA is capable of running the web application as well in the built-in browser, the user could enter/edit the preferences also in the PDA. It is also conceivable that the general preferences part can be built into the PDA application as well, although this may require a significant amount of additional work.

The suggested schema for storing personal details was given in the central server chapter.

11.2 General preferences

Similarly to the personal details, we assume that the general preferences are entered via the web application.

Conceptually the simplest form of general preferences is a list of keywords liked and disliked (I like impressionism, Kandinsky and Klee. I dislike expressionism).

The preferences could contain exceptions (I like impressionism except Monet).

The preferences could contain concept intersections (I like 20-th century american art).

We will not allow concept definitions or taxonomy rules to be entered in the user profile: such definitions and rules are general, initially generated in the project and maintained by the recommendation server administrator, preferably in OWL or controlled English sentences translatable



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

to OWL (suggested language is ACE: Attempto Controlled English). See the recommendation server chapter and the ACE web site: <http://attempto.ifi.uzh.ch/site/>.

The language of general preferences in the user profile will have the following simple syntax:

- I like/dislike concept1,, conceptN [except conceptE1,, conceptEM] where the [...] part is optional. The word “and” can be used instead any comma in a concept list.
- I prefer concept1 to concept2.

Any concept can be a list of concepts, interpreted as a an intersection after merging/removing phrases, like in “20-th century american art”.

The number or like/dislike sentences is arbitrary.

The interpretation of preferences using ontologies is performed by the recommendation server: this is not a task of the PDA or user web application. However, the user web application should be able to parse and check the syntactic correctness of the sentences, plus it should be able to check whether the corresponding concept exists in the server database (otherwise the concept would not be understood by the server).

The exact details for the general preferences human-readable syntax and their translation to owl should be taken up in the workpackage handling ontologies.

11.3 Concrete objects liked/disliked

The information about the concrete objects is given in an (almost) standard CSV format with each row having a form:

Object_URL, score, time_of_entry, comments

Where:

- Object_URL is an XML schema anyURI: ordinary full object URL
- Score is a like/dislike number as an XML schema decimal: a positive or negative floating-point number without an exponential part (for example, 1, 0.5, -0.01)
- Time_of_entry may be missing. If available, it is either:
 - Full year as XML schema gYear (like 2008)



Grant Agreement Number: FP7-216923

Acronym: SMARTMUSEUM

Project title: Cultural Heritage Knowledge Exchange Platform

- Date as XML schema date YYYY-MM-DD (like 2008-10-30)
- Date-time up to seconds either as XML schema date TIME or a simpler form of the latter without the time zone information YYYY-MM-DDTHH:MM:SS (like 2008-10-30T14:12:23)
- Comments is a free text string: either as plain text or HTML text, where end-of-line and carriage-return are encoded so as to not mix up with the CSV end-of-line characters. Comments may contain CSV separators (commas): since the comments entry is the last element of the line, the CSV parser should ignore them.

12. Conclusions

The document described Smartmuseum architecture and user scenarios to be implemented. The main idea is that the solution will be built around user profile storage server. The content shall be organized Wikipedia-like way with dedicated RDFa compatible annotation tags. The user profiling solution will contain two methods: user clustering by interests and direct visited item matching (“similar purchase”). The main idea of privacy preserving is centralized profile storage minimizing profile information retranslation. Detailed specifications of web services and user device software will be described in D4.1 and D3.1.