



D2.3.2.2

Semantic Portable Profiles Prototyping (SP3)

Requirements for a mobile proactive recommendation information service

Confidentiality: Consortium

Date and status: 31.3.2012 – V1.0

This work was supported by TEKES as part of the next Media programme of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT)

Version history:

Version	Date	State (draft/ /update/ final)	Author(s) OR Editor/Contributors	Remarks
0.1	15.3.2012	First draft	Janne Saarela	Initial structure created
0.2	16.3.2012	Second draft	Juho Autio	Added features from D2.3.2.1 document + some other features and additions to the existing ones.
0.3	16.3.2012	Third draft	Project group	
0.4	29.3.2012		Project group	Individual contributions from group members
0.5	30.3.2012	5 th draft	Janne Saarela	Consolidated comments to a single master document
1.0	31.3.2012	Final	Janne Saarela	Incorporated internal review comments

Participants	Name	Organisation
	Janne Saarela	Profium
	Juho Autio	Profium
	Timo Pastila	Kii
	Sari Vainikainen	VTT
	Heikki Väänänen	Osumus
	Jukka-Pekka Salmenkaita	Osumus
	Jenni Niemiaho	Forum Virium Helsinki

Next Media
www.nextmedia.fi
www.tivit.fi

Executive Summary

This document is a requirement specification for a mobile service where end-users can get recommendations when they access the service. These recommendations are based on both explicit and implicit profiles about the user and can thus provide the user with relevant content when the user wants. The system is also capable of providing proactive information services by matching new incoming content with users' interest profiles.

The requirements of this document originate from earlier work with Semantic Portable Profile Prototyping. User feedback for some of the service concepts was sought with a test group and this work was reported in another deliverable D2.3.2.1 Use cases, concept definition and evaluation.

Yhteenveto

Tässä dokumentissa kuvataan vaatimukset mobiilille palvelulle missä loppukäyttäjät saavat suosituksia sisällöstä. Nämä suositukset perustuvat sekä eksplisiittiseen että implisiittiseen profiilitietoon käyttäjistä ja tarjoavat relevanttia sisältöä silloin kun käyttäjä haluaa sitä saada. Palvelu pystyy myös tuottamaan proaktiivisia informaatiopalveluja täsmäyttämällä uusia sisältöjä käyttäjien profiilitietoja vasten ja lähettämällä tiedot käyttäjälle ilman että käyttäjän tarvitsee aktiivisesti itse käyttää palvelua.

Tämän dokumentin vaatimukset on johdettu aikaisemmasta työstä semanttisten siirrettävien profiilien prototyyppityössä. Käyttäjiltä on ennen kehitystyön alkamista haettu palautetta palvelun konsepteista ja tämä palaute on luettavissa dokumentissa D2.3.2.1 Use cases, concept definition and evaluation.

Table of Contents

Executive Summary.....	2
1 List of Acronyms and Abbreviations	5
2 Overall description.....	6
2.1 Roadmap	6
3 Business requirements	6
4 Use Cases	7
4.1 UC01 User logs into the Service	7
4.2 UC02 User views and modifies her profile	8
4.3 UC03 User controls notifications.....	10
4.4 UC04 User deletes account	11
4.5 UC05 User receives a notification to his/her mobile phone.....	11
4.6 UC06 User views recommendations.....	12
4.7 UC07 User gives feedback	13
4.8 UC08 User shares content to his/her network.....	14
4.9 UC09 User views her favourites.....	15
4.10 UC10 User views favourites of her social network	16
4.11 UC11 User searches content.....	16
4.12 UC12 Content partner submits new content	17
5 Functional requirements	18
5.1 FR-MOBILE-1010 Protocol for server notifications	18
5.2 FR-MOBILE-1020 HTML5 representation of server notifications	18
5.3 FR-MOBILE-1030 Position information from mobile app to the Service	19
5.4 FR-MOBILE-1040 Disable/re-enable location information	20
5.5 FR-MOBILE-1110 User authentication	20
5.6 FR-MOBILE-1112 Updating user's location information	21
5.7 FR-MOBILE-1130 Modular access restrictions to profile content.....	21
5.8 FR-MOBILE-1140 Connecting friends to profile.....	21
5.9 FR-MOBILE-1150 Meaningful places: map view	22
5.10 FR-MOBILE-1155 Meaningful places: current position	22
5.11 FR-MOBILE-1210 Blocking notifications: current notification type	22
5.12 FR-MOBILE-1230 Rating notifications	23
5.13 FR-MOBILE-1220 Blocking notifications: filtering	23
5.14 FR-PROFILE-100 Profile Storage.....	23
5.15 FR-PROFILE-200 Create profile	24
5.16 FR-PROFILE-300 Read profile	25
5.17 FR-PROFILE-400 Update profile	25
5.18 FR-PROFILE-500 Delete profile	26
5.19 FR-ANALYSER-100 Semantic analyser	27
5.20 FR-ANALYSER-110 Semantic categorization.....	27
5.21 FR-ANALYSER-200 Facebook analyser	27

5.22 FR-ANALYZER-300 Web Analyser.....	28
5.23 FR-IMPORT-100 Facebook importer	28
5.24 FR-IMPORT-200 TV-program importer.....	29
5.25 FR-PENGINE-100 Implicit user profiler (engine).....	29
5.26 FR-PENGINE-200 Implicit Recommendation engine.....	29
5.27 FR-CONVERT-100 Profile converter (engine->user).....	30
5.28 FR-STORAGE-100 User Interaction Storage.....	30
5.29 FR-STORAGE-200 Implicit Profile	31
5.30 FR-REC-100 Fetch Recommendations	31
5.31 FR-PENGINE-300 Semantic Recommendation engine	31
5.32 FR-METADATA-100 Metadata storage	32
5.33 FR-TRIGGER-100 Triggered evaluation.....	32
5.34 FR-NOTIFY-100 Personalised notifications	33
5.35 FR-NOTIFY-200 Notification server	33
5.36 FR-MOBILE-1250 Browsing recommended content.....	34
5.37 FR-SEARCH-100 Searching content.....	34
5.38 FR-SEARCH-200 Persistent search	34
5.39 FR-SAVE-100 Saving content as favourite	35
5.40 FR-SHARE-100 Social discovery and sharing of content	35
5.41 FR-INPUT-100 Content submission API	35
5.42 FR-INPUT-900 Skimm.tv input interface	35
5.43 FR-INPUT-901 stadi.tv input interface	36
5.44 FR-INPUT-902 HelMet new videos input interface	36
6 Non-functional requirements	36
6.1 NFR-MOBILE-100 PhoneGap implementation	36

1 List of Acronyms and Abbreviations

API	Application Programming Interface
OGP	Open Graph Protocol
LOD	Linked Open Data

2 Overall description

This document lays out requirements for an information service which the end-users access via smart mobile handsets. The Service is capable finding a match between user's explicit and implicit interest profiles and the digital content made available to the service by different types of content partners. The service is also capable of proactively notifying the end-user about relevant information as soon as it becomes available in the Service.

The structure of this document is split into business requirements which are followed by the main uses cases of the Service. Functional and non-functional requirements are then laid out with tractability to map them back to the use cases and explicit dependencies to other requirements.

This document is not a design document but such a document will be developed in parallel and which will address architectural and technological aspects of the overall software system.

2.1 Roadmap

The Service in its software elements will be developed iteratively and these requirements will become detailed in that process. The 1st spring contents are to be defined at the of writing this version of the document.

3 Business requirements

This sections enumerates some high-level requirements that the planned software system must meet with from the business perspective.

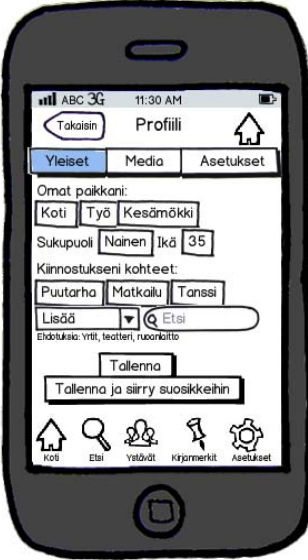

- BR-001: User can fetch content which matches with his/her interests. User can express their interests implicitly by indicating whether something he/she viewed was of interest or not. User can also express their interests explicitly by defining an interest profile
- BR-002: User can be proactively notified of valuable information items or events which are not annoying.
- BR-003: User can be provided the service on smart mobile platforms

4 Use Cases


4.1 UC01 User logs into the Service	
Actors	Service, User
Description	<p>User is able to login system with Facebook Connect.</p> <ol style="list-style-type: none"> 1. The User logs into the service by giving his Facebook user id and password. In case he is logged in to Facebook, log in is made automatically when the User presses Connect button. 2. When login the first time to the Service: <ol style="list-style-type: none"> a) The Service asks permission to use profile information (Facebook). b) The Service checks if the User already has “The portable profile” c) If no, The Service creates account to the Profile Storage. d) If yes, The Service asks the User’s permission to access his portable profile. e) The Service imports the User’s Facebook profile information. 3. Once the Service has been given permissions and started creating account and FB profile importation, the mobile client asks the person to give information about his/her important locations and interest areas. 4. Once the user has created the profile and clicked “go to recommendations”, the service asks recommendations for the User based on the FB profile and the information on locations and interests user has given. 5. The User is shown the main page with the recommended content. <p>The attached UI mock-up images show the FB login screen and the home screen (list of recommended content) but UI mock-ups for the steps in-between are still missing (defined in the context of detailed implementation of the FB Connect).</p>


Pre condition	
Post condition	User gets to the main page that shows recommended content.
Priority	
Business benefits	
Goal	
Exceptions	Incorrect user id and password (included to Facebook Connect functionality). No user profile information, user gets general recommendations and guidance to tell more about herself. If a User does not give permission to use his Facebook information, he is not able to login to the Service.
Impact(s)	
Open issue(s)	

4.2 UC02 User views and modifies her profile	
Actors	Service, User
Description	<p>User can view her profile information and modify it.</p> <ol style="list-style-type: none"> 1. User clicks link to her profile page 2. User can view profile information 3. User can add new interest or delete old ones 4. User can add age and other demographic information 5. User can add place information on the map 6. User can link profile with social media account

	 <p>7. User can add intentions. This is something that should be made easy for the users and hence added a link to this on the home page and created a separate view for that so that this would be made as easy as possible to express to update the recommended content based on the new intended location.</p> 
<p>Pre condition</p>	<p>User has logged in.</p>
<p>Post condition</p>	<p>User profile and settings are stored. User is shown the main page and updated recommendations.</p>
<p>Priority</p>	
<p>Business benefits</p>	

Goal	
Exceptions	
Impact(s)	
Open issue(s)	

4.3 UC03 User controls notifications	
Actors	Service, User
Description	<p>User can define for what kind of content gets notifications and when.</p> <p>The attached mock-up image is a placeholder for settings related to notifications. This will be refined during the next phase.</p> 
Pre condition	User has logged in.
Post condition	Notifications settings are stored in the Service.
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	

4.4 UC04 User deletes account	
Actors	Service, User
Description	<p>A User deletes his account and all his profile information on the service.</p> 
Pre condition	User has logged in.
Post condition	User account is deleted.
User	
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	


4.5 UC05 User receives a notification to his/her mobile phone	
Actors	Service, User
Description	<p>The User receives a visual, audio, vibration or a combination of these in the mobile phone via a message sent by the Service.</p> <ol style="list-style-type: none"> 1. User observes the incoming notification 2. User clicks on the notification to view the full details

	3. User closes the application where the notification was viewed
Pre condition	Service has knowledge about the User’s mobile application which is ready to receive notifications. Service has knowledge about the User’s preferences as to what type of notifications the User is willing to accept.
Post condition	Service has knowledge that the notification was successfully sent and will not send it again.
Priority	
Business benefits	User can be proactively notified about relevant content without requiring the user to be the active information seeker.
Goal	
Exceptions	If the notification was deemed irrelevant by the user, the user can easily block notifications altogether or indicate that the received notification was not relevant and similar notifications should not be sent to the User in the future.
Impact(s)	
Open issue(s)	

4.6 UC06 User views recommendations	
Actors	Service, User
Description	<p>User is able to browse recommendations .</p> <ol style="list-style-type: none"> 1. User is shown the list of recommended items (based on current location, permanent interests, intentions, favourites of friends etc) 2. User clicks on the recommendation to view the full details. 3. The service may allow the user to view content from one channel only (details TBD later). 4. The content can be viewed as a list or on a map based on the location data of each content item.


Pre condition	User has logged in. Service has knowledge about the user’s profile and preferences.
Post condition	
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	Need to define in the UI how to allow the user to select individual channels only.

4.7 UC07 User gives feedback	
Actors	Service, User
Description	User is able to give feedback about content she/he is viewing <ol style="list-style-type: none"> 1. User can select thumb up or down if wants to give feedback on the item.

	
Pre condition	User has logged in and is viewing notifications or recommendations.
Post condition	User profile will be updated based on this information. Content that gets thumb up is stored to the user's favourites.
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	


4.8 UC08 User shares content to his/her network	
Actors	Service, User
Description	User is able to share content or notifications she/he is viewing to his/her social network. (Share in Facebook ?) 1. User clicks share button
Pre condition	User has logged in and is viewing notifications or recommendations. Service has knowledge about the user's social network.
Post condition	
Priority	
Business benefits	
Goal	
Exceptions	

Impact(s)	
Open issue(s)	

4.9 UC09 User views her favourites	
Actors	Service, User
Description	<p>User can view of list of items that have been added in the favourites by giving it thumbs up.</p> <p>User can delete items on the favourites list.</p>
	
Pre condition	User has logged in.
Post condition	
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	

4.10 UC10 User views favourites of her social network	
Actors	Service, User
Description	User can view the favourites of his/her friends as a list can pick up items for further viewing.
Pre condition	User has logged in and has connected with friends via the Service.
Post condition	
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	User could invoke this feature either from the Friends view or the Favourites view – or both. The use case needs further discussion and definition work during the next phase of the project. The key issue is that would there be just a user option to allow access to all Facebook friends to view your favourites or should one be able to restrict this access more accurately e.g. only for certain friends or even fine-tune this sharing per friend and interest area. This could be the area that would allow improvements to the privacy policy of Facebook and hence could provide some true differentiation. However, all this is also a significant piece of work.

4.11 UC11 User searches content	
Actors	Service, User
Description	The User submits a query to the system and expects relevant results.

	
Pre condition	User has logged in.
Post condition	User receives results.
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	<p>Needs to define how the search is implemented but the assumption is a well defined search API to the metadata and content servers available for the mobile client.</p> <p>Need to also decide whether search is provided at all and if is provided, would it be simple search only or provide also advanced search features (e.g. search only within a certain media type and within a certain radius from here etc) and would it provide to the user some simple shortcut type of search options, e.g. show me all content in the Service that has a location stamp within a radius of 500 m (or 1 km) or all nearby content published within the last week/day or such.</p>

4.12 UC12 Content partner submits new content	
Actors	Content Partner, Service
Description	Content partner has new relevant content which is wishes to make available for the Service user. It proceeds by preparing an information package which includes descriptive metadata about

	<p>the content and content itself (can be of various types supported by mobile handsets).</p> <p>Service provides an API for automated acknowledgement whether the submitted content package passed validation checks. For eventual errors the API will provide a description back to the Content Partner as to what was wrong in the package.</p>
Pre condition	Service is running
Post condition	Service is running, new content is persisted in the Service.
Priority	
Business benefits	
Goal	
Exceptions	
Impact(s)	
Open issue(s)	Protocol and payload need to be defined for this content submission phase. Additionally one could refer to the actual content via a URL instead of sending the content to the Service. This would allow more dynamic content to be provided in the service (that may change as a function of time and not remain as such).

5 Functional requirements

5.1 FR-MOBILE-1010 Protocol for server notifications	
Description	The mobile application can create a socket or similar persisted connection to a Service which can then make use of this communication channel to push notifications to the user.
Dependencies	
Source	UC05
Notes	
Priority	

5.2 FR-MOBILE-1020 HTML5 representation of server notifications	
Description	The notifications sent by the Service to the mobile app can be encoded in HTML5 and the mobile app can view these notifications with a local Web browser widget.

Dependencies	FR-MOBILE-1010
Source	UC05
Notes	
Priority	

5.3 FR-MOBILE-1030 Position information from mobile app to the Service	
Description	<p>The mobile application can be configured to send the User's current position to the Service every X seconds where User gets to configure X. The configured value will be persisted locally on the mobile device so that it survives application and/or phone restart.</p> <p>The position will be sent to the Service using a HTTPS encoded protocol channel using a REST API and Basic authentication. The HTTPS payload will carry</p> <ul style="list-style-type: none"> • coordinates (latitude, longitude, altitude) of the user using WGS84 reference coordinate system. • accuracy of the coordinates • direction of movement • speed (m/s) • timestamp for when the position information was recorded <p>Position data is transmitted to the endpoint using a JSON based representation. The URI endpoint is hardcoded in the application and cannot be changed by the user.</p> <p>The client will indicate with two symbols the status of the Location indicators to the Service: One symbol which is presented whenever the Service has returned HTTP 200 OK and another symbol which is presented whenever the Service has not been reached or whenever it has returned an error in the 4XX or 5XX series.</p> <p>The client must implement an HTTP timeout of 2 seconds for the location calls.</p>
Dependencies	FR-MOBILE-1010
Source	UC05
Notes	
Priority	

5.4 FR-MOBILE-1040 Disable/re-enable location information	
Description	The user must be able to use a single touch UI element to disable and again re-enable the submission of the User's location to the Service. Current status is indicated with different symbols each with different color (not just one symbol with green and red versions).
Dependencies	FR-MOBILE-1030
Source	UC05
Notes	
Priority	

5.5 FR-MOBILE-1110 User authentication	
Description	<p>User authentication supports Facebook login. If a user already has a profile the Service can access the user's profile with the user's consent. The approval process follows the OAuth protocol, version 1.0:</p> <ol style="list-style-type: none"> 1. The Service initiates the approval process by asking for the profile (Facebook id) from the Profile Service. 2. The Profile Service asks permission from the User by presenting question "The site example.com wants to access your profile, is this ok?" 3. When the User confirms the request, the Service is issued a grant to access the User's profile information. 4. The User can later revoke this grant whenever he wants, and can also see logs on when his profile information were accessed by the Service. (this is done on the account page in the profile service (current PC user interface))
Dependencies	FR-PROFILE-100
Source	UC01
Notes	More information about OAuth and its usage at the Profile Service can be found at http://profile.vtt.fi/oauth.html
Priority	

5.6 FR-MOBILE-1112 Updating user's location information	
Description	Location tracker updates information about a user's current location to the profile storage (user_id, geo co-ordinates, datetime).
Dependencies	FR-MOBILE-1030
Source	UC05
Notes	
Priority	

5.7 FR-MOBILE-1130 Modular access restrictions to profile content	
Description	<p>The user is able to share only part of his or her profile to a service using the portable profile. The profile needs to be modular in a way that is easy for users to understand, use and manage: Demographics, interests, intentions, real identity, ...).</p> <p>In practice, when granting permissions for an application the user can select which permissions she is willing to allow.</p> <p>User must be able to manage all the existing permissions in a centralized view.</p>
Dependencies	
Source	
Notes	
Priority	

5.8 FR-MOBILE-1140 Connecting friends to profile	
Description	A user is able to connect his or her profile with those of the friends and to get recommendations based on this combined information.
Dependencies	
Source	UC06
Notes	

Priority	
-----------------	--

5.9 FR-MOBILE-1155 Meaningful places: map view	
Description	The user is able to indicate meaningful places of his or her life on a map (e.g. home, work place, summer cottage, hobbies).
Dependencies	
Source	UC02
Notes	
Priority	

5.10 FR-MOBILE-1155 Meaningful places: current position	
Description	The user is able to add a meaningful place of his or her life using the current position. User can optionally add a title and description to the place and categorize the place to one of pre-configured categories (work, home, restaurant, pub, ...)
Dependencies	
Source	UC02
Notes	
Priority	

5.11 FR-MOBILE-1210 Blocking notifications: current notification type	
Description	When displaying a notification in the mobile application, user must be able to indicate that she does not want to receive this type of notifications any more.
Dependencies	
Source	UC05
Notes	
Priority	

5.12 FR-MOBILE-1230 Rating notifications	
Description	When displaying a notification in the mobile application, user must be able to rate the notification by submitting a thumbs up or down rating.
Dependencies	
Source	UC07
Notes	This information is used for calculating relevancies for future notifications, but this doesn't necessarily mean that the user couldn't be notified by a similar notification again, even if she rated it not relevant.
Priority	

5.13 FR-MOBILE-1220 Blocking notifications: filtering	
Description	Based on topic or context. Also able to select how often.
Dependencies	
Source	UC05
Notes	
Priority	

5.14 FR-PROFILE-100 Profile Storage	
Description	Profile storage functionality includes both profile management and storage. The functionality is based on VTT profile service prototype. The profile service has a REST API for creating, reading, updating and deleting a profile. Current REST API is described at http://profile.vtt.fi/api/doc/profile/ . OAuth protocol version 1.0 is used to create, read, update and delete profiles. Read
Dependencies	
Source	UC02

Notes	Additional information can be found http://virtual.vtt.fi/virtual/nextmedia/Deliverables-2011/NM_WP3_%20generic%20research_D3.3.1.2_v1.0.pdf API will be updated based on requirements.
Priority	

5.15 FR-PROFILE-200 Create profile	
Description	<p>A User profile is created to Profile storage. Create profile: http://profile.vtt.fi/api/profile/ POST - oAuthconsumerKey (*) - oAuthconsumerSecret (*)</p> <p>Creates a profile.</p> <p>To let the user see their profile on the server, link the user to <a href="http://profile.vtt.fi/?uuid=<profile.id>">http://profile.vtt.fi/?uuid=<profile.id> where they should "create an account". In reality they only create a login mechanism and the created account is the one that already exists.</p> <p>@return 201 Created - on success * the Location header is set to the profile location * the profile data is returned in the document content (json or xml) * profile.oauthToken contains the OAuth token to be used for profile access/modification * profile.oauthTokenSecret contains the OAuth token secret to be used for profile access/modification @return 400 Bad request - if any required parameters are missing @return 401 Unauthorized - if the credentials are incorrect</p>
Dependencies	FR-PROFILE-100
Source	UC02
Notes	Additional information: http://profile.vtt.fi/api/doc/profile/
Priority	

5.16 FR-PROFILE-300 Read profile	
Description	<p>User profile information is read from Profile storage.</p> <p><a href="http://profile.vtt.fi/api/profile/<profile_id>">http://profile.vtt.fi/api/profile/<profile_id> GET - format (optional) "NORMAL" or "FULL" specifying how much semantic info to include</p> <p>A) If profile id given on patch, read the corresponding profile. B) No profile id was given on path - use OAuth to determine profile :</p> <p>oauthConsumerKey: oAuth consumer key for the Service oauthConsumerSecret: oAuth consumer key for the Service oauthToken: oauthToken access token for the profile to read oauthTokenSecret: oauthToken access token secret for the profile to read</p> <p>@return 200 OK, or 204 No content - on success * the profile data is included in the response (json or xml) @return 404 Not found - when the profile wasn't found</p>
Dependencies	FR-PROFILE-100, FR-REC-100, FR-PENGINE-300, FR-PENGINE-200
Source	UC02, UC06, UC05
Notes	<p>Additional information: http://profile.vtt.fi/api/doc/profile/</p> <p>Format of the output can be modified based on requirements.</p>
Priority	

5.17 FR-PROFILE-400 Update profile	
Description	<p>Updating profile information to Profile Storage</p> <p>http://profile.vtt.fi/api/profile/ PUT - label (*); tag label; e.g. rock - lang ISO 639-2 language code of the label - uri (optional) semantic LOD uri of the label - description (optional) - goodness (optional) Tag goodness; how important the user finds the tag (0-2, 1 is normal) - coLabels (optional): Label used together with this tag - format (optional) "NORMAL" or "FULL" specifying how</p>

	<p>much semantic info to include</p> <p>Use OAuth to determine profile.</p> <p>oauthConsumerKey: oAuth consumer key for the Service</p> <p>oauthConsumerSecret: oAuth consumer key for the Service</p> <p>oauthToken: oauthToken access token for the profile to update</p> <p>oauthTokenSecret: oauthToken access token secret for the profile to update</p> <p>@return 200 OK - on success</p> <ul style="list-style-type: none"> * the profile data is returned in the document content (json or xml) * site profiles can't be created/updated/deleted by api calls <p>@return 404 Not Found - when profile doesn't exist</p> <p>@return 400 Bad request - if any required parameters are missing/invalid</p>
Dependencies	FR-PROFILE-100, FR-CONVERT-100, Explicit Profile Management
Source	UC02
Notes	Additional information: http://profile.vtt.fi/api/doc/profile/
Priority	

5.18 FR-PROFILE-500 Delete profile	
Description	<p>User profile is deleted from Profile storage.</p> <p>http://profile.vtt.fi/api/profile/DELETE</p> <p>Use OAuth to determine profile:</p> <p>oauthConsumerKey: oAuth consumer key for the Service</p> <p>oauthConsumerSecret: oAuth consumer key for the Service</p> <p>oauthToken: oauthToken access token for the profile to delete</p> <p>oauthTokenSecret: oauthToken access token secret for the profile to delete</p> <p>@return 204 No Content - on success</p> <p>@return 401 Unauthorized - if the credentials are incorrect</p> <p>@return 403 Forbidden - if the user has taken control of his profile. can't delete it through the api anymore</p>
Dependencies	FR-PROFILE-100, Explicit Profile Management

Source	UC02
Notes	Additional information: http://profile.vtt.fi/api/doc/profile/
Priority	

5.19 FR-ANALYSER-100 Semantic analyser	
Description	Support semantic analysis of keywords and short text descriptions. Analysis includes keyword extraction and semantic enrichment of keywords. Enrichment utilizes Linked Data sources. Analyzer can be used both for enriching user and content metadata. Content metadata is stored to Metadata Storage. User metadata to Profile Storage.
Dependencies	FR-ANALYSER-200, FR-ANALYSER-300
Source	
Notes	Uses VTT algorithms: Algorithm for Informativeness-based Keyword Extraction Semantic analysis
Priority	

5.20 FR-ANALYSER-110 Semantic categorization	
Description	Semantic categorization match user's interests to categories (such as sport, nature, persons, places etc.). This categorization is used for visualising the profile to the user. Categorization is used also for showing recommendation result to end user's.
Dependencies	
Source	UC02
Notes	
Priority	

5.21 FR-ANALYSER-200 Facebook analyser	
Description	Input from the user's Facebook account based on Facebook Opengraph API (profile, likes, social network...)

	Definition of which information is relevant and how it should be further analysed. Facebook analyser utilises Semantic analyser, but it might include additional Facebook specific features (e.g. pre-categorisation of the most liked pages). The result is stored to Profile Storage. Development of API so that service utilizing Facebook Connect can store the result into their systems. User is able to see result on his profile page.
Dependencies	FR-ANALYSER-100, FR-ANALYSER-300, FR-IMPORT-100
Source	UC01
Notes	It should be checked (either here, or most probably in the calling routine) if the FOG data is already analysed and if it needs to be reanalysed.
Priority	

5.22 FR-ANALYZER-300 Web Analyser	
Description	Collects relevant textual information from Web pages to Facebook Analyzer and to Semantic Analyzer
Dependencies	FR-ANALYZER-200
Source	
Notes	<ul style="list-style-type: none"> Analyser could focus on some specific set of Web pages and on some specific set of HTML tags in there. It could / should also work with schema.org metadata.
Priority	HV_3

5.23 FR-IMPORT-100 Facebook importer	
Description	Collects user information (age, sex, address, social network ...) and user profile metadata (keywords, tags, text, Facebook likes) from user's Facebook account based on Facebook OpenGraph. Facebook likes are further processed by <i>Facebook OpenGraph analyser</i> and other textual metadata directly by <i>semantic analyser</i> . User information is returned and user profile is stored to the <i>Profile Storage</i> .
Dependencies	
Source	

Notes	
Priority	HV_1

5.24 FR-IMPORT-200 TV-program importer	
Description	Returns TV-program data (content items), which includes name, date, time, channel and synopsis for all the TV-programs from preselected list of channels.
Dependencies	
Source	
Notes	<ul style="list-style-type: none"> Data is retrieved from Osumus Recommendations TV-program database, which is Background IPR in the project.
Priority	HV_1

5.25 FR-PENGINE-100 Implicit user profiler (engine)	
Description	Creates and updates implicit user profile based on user reactions (likes and dislikes) and initial user profile (not needed necessarily). Initial user profile (metadata schema) is created from user information (from Facebook Open Graph data or given by user explicitly) by <i>semantic analyser</i> . User profile is updated based on user reactions to the content items (both service content items and Facebook likes). Returned user profiles are processed profile converter for user readable (and editable) format of the profile.
Dependencies	FR-ANALYSE-200, FR-PENGINE-200, FR-CONVERT-100
Source	
Notes	<ul style="list-style-type: none"> Uses the Osumus recommendation engine, which is Background IPR to the project. API (internet socket?) and the feature space (meta-data / profiles) are implemented in the project.
Priority	HV_1

5.26 FR-PENGINE-200 Implicit Recommendation engine	
Description	Sorts list of profiled content items based on their recommendation (OsumusRank) value to the user. A “general recommendation value” can be defined for unprofiled (unregistered) users. Engine is called by semantic recommendation engine, which provides list of content item ids and user id. Recommendations are created based on user reactions, user profiles and

	content items metadata (profile). User reactions (likes and dislikes) are added to the engine when user gives them (and initial values for content item profiles are when it is given (UC11 User gives feedback). Initial values for content item profiles (metadata schema) are created and given by (Web analyser and Facebook analyser). There don't have to be metadata for all the content items to be recommended (content item profiles are then created just based on user reactions to the content items), but all the content items have to be imported (at least the id) to the engine before recommendations are asked for them.
Dependencies	FR-ANALYZER-200, FR-ANALYZER-100, FR-PENGINE-300
Source	
Notes	<ul style="list-style-type: none"> • Uses the Osumus recommendation engine, which is Background IPR to the project. API (internet socket?) and the feature space (meta-data / profiles) are implemented in the project.
Priority	HV_1

5.27 FR-CONVERT-100 Profile converter (engine->user)	
Description	Creates user readable / editable profile (list / combination of words) based on float valued profile and confidence values in implicit profile (initial value for <i>implicit recommendation engine</i> and <i>implicit user profiler</i>).
Dependencies	FR-ANALYZER-200, FR-PENGINE-300
Source	
Notes	<ul style="list-style-type: none"> • User readable / editable profile consists on predefined set of words (metadata schema) and can include also word combinations (e.g. "Finnish (AND) reality"). Conversion / world selection is based on implicit profile and confidence values and user statistics.
Priority	HV_1

5.28 FR-STORAGE-100 User Interaction Storage	
Description	The User Interaction Storage stores the user's interaction such as thumbs up/down, search and browsing history. It gets information from Implicit Profile.
Dependencies	FR-PENGINE-100, FR-PENGINE-200
Source	UC06, UC07, UC11
Notes	

Priority	
-----------------	--

5.29 FR-STORAGE-200 Implicit Profile	
Description	Implicit Profile stores user's interactions to the User Interaction Storage. Thumbs up/down; userID, contentID, value, date Search history; userID, search term, date Browsing history; userID, contentID, page, date
Dependencies	FR-STORAGE-100
Source	UC07
Notes	
Priority	In the first phase thumbs up/down will be implemented.

5.30 FR-REC-100 Fetch Recommendations	
Description	Fetch recommendations functionality calls Semantic recommendation engine by giving user id (facebook identifier) and parameters relating to filtering (e.g. media type, location, time)
Dependencies	FR-PENGINE-300
Source	UC06
Notes	
Priority	

5.31 FR-PENGINE-300 Semantic Recommendation engine	
Description	Semantic Recommendation engine creates ranked list of content items based on user's profile and context. Semantic Recommendation engine reads user profile information from Profile storage and content metadata from Metadata storage. Recommendations are made matching user profile information with content metadata and context by using semantic relations and/or Solr based recommendations. The result is ranked based on relevance, location and time.

	<p>Semantic Recommendation Engine supports recommendations created based on stereotypes (e.g. age and gender). This is used when the system has only little information about the User.</p> <p>The result (user_id, List of Recommendations (content id, weight)) is forwarded to Implicit Recommendation engine, which rank the result based on user's implicit profile.</p> <p>Semantic recommendation engine has REST API. Semantic Recommendation engine is called by Fetch Recommendations or Trigger.</p>
Dependencies	FR-TRIGGER-100, FR-NOTIFY-100, FR-PENGINE-200, FR-PROFILE-100, FR-METADATA-100
Source	UC06
Notes	Semantic Recommendation interface is part of the VTT Profile Service
Priority	

5.32 FR-METADATA-100 Metadata storage	
Description	The Service must be capable of storing metadata which results from the analysis of incoming content objects. This metadata is made available for triggering to match the union of the incoming metadata and previously seen metadata with user interests.
Dependencies	FR-PENGINE-300, FR-TRIGGER-100, FR-NOTIFY-200
Source	UC05
Notes	
Priority	

5.33 FR-TRIGGER-100 Triggered evaluation	
Description	Whenever new metadata becomes available in the Service from input process, it gets evaluated against user interests. If there is a match in his process, a notification will be sent proactive to the

	corresponding user.
Dependencies	FR-NOTIFY-200
Source	UC05
Notes	
Priority	

5.34 FR-NOTIFY-100 Personalised notifications	
Description	<p>User profile, preferences and context information is used to make personalised alerts based on user's needs. Personalised notifications functionality is part of the Semantic Recommendation engine.</p> <p>Personalised notifications is called by Trigger. Input for Create notifications is content_id, location, date&time.</p> <p>Personalised notifications searches users who are interested in to get this type of notifications (topic, location, data&time). Result is sent to Notification Server (user_id, content_id, date&time) which delivers notifications to users.</p>
Dependencies	FR-PENGINE-300, FR-TRIGGER-100, FR-NOTIFY-200
Source	UC05
Notes	
Priority	

5.35 FR-NOTIFY-200 Notification server	
Description	Notification server provides an API for other components to send a single notification at a time to a single recipient. The server implements the specified push protocol and supports the specified payload. The API is provided as a RESTful HTTP service.
Dependencies	FR-MOBILE-1010, FR-MOBILE-1020
Source	UC05

Notes	
Priority	

5.36 FR-MOBILE-1250 Browsing recommended content	
Description	Browse recommended content (pull).
Dependencies	
Source	UC06
Notes	
Priority	

5.37 FR-SEARCH-100 Searching content	
Description	Search content (this provides the user an alternative way of finding material and information).
Dependencies	
Source	UC11
Notes	Full-text or metadata or a mix of these two is the currently planned approach to implementing this.
Priority	

5.38 FR-SEARCH-200 Persistent search	
Description	User can save a search to be notified about new information that matches the search terms.
Dependencies	FR-MOBILE-1260
Source	UC11
Notes	
Priority	

5.39 FR-SAVE-100 Saving content as favourite	
Description	User can indicate content being viewed as being favourite to have direct access to it later.
Dependencies	
Source	UC09
Notes	
Priority	

5.40 FR-SHARE-100 Social discovery and sharing of content	
Description	User can browse which recommendations and content his or her friends have liked, and share recommendations and notifications to his or her social network. (support alternative way of finding content).
Dependencies	
Source	UC08
Notes	
Priority	

5.41 FR-INPUT-100 Content submission API	
Description	Content partners can submit new content via an API call. The API call assumes both content description and content to be present. The API will perform validity check for the content description and on the compatibility of the content format before persisting them in the Service.
Dependencies	
Source	UC12
Notes	
Priority	

5.42 FR-INPUT-900 Skimm.tv input interface	
Description	Module to submit content from skimm.tv will be implemented to support generic input API

Dependencies	FR-INPUT-100
Source	UC12
Notes	
Priority	

5.43 FR-INPUT-901 stadi.tv input interface	
Description	Module to submit content from stadi.tv will be implemented to support generic input API
Dependencies	FR-INPUT-100
Source	UC12
Notes	
Priority	

5.44 FR-INPUT-902 HelMet new videos input interface	
Description	Module to submit content from www.helmet.fi will be implemented to support generic input API
Dependencies	FR-INPUT-100
Source	UC12
Notes	
Priority	

6 Non-functional requirements

6.1 NFR-MOBILE-100 PhoneGap implementation	
Description	The mobile app must be built with PhoneGap to ensure its deployment on various platforms such as Android, Windows Mobile, IOS and BlackBerry.
Dependencies	
Source	

Notes	It is anticipated that some features will require platform specific code which means that some of the functionalities might only be available on those mobile platforms.
Priority	