Project Deliverable Report

Deliverable nr D4.1 – Design Document v1

Work Package       WP4
Task               T4.1
Date of delivery   Contractual: 01-10-2008      Actual: 27-11-2008
Code name          Version: 1.0
Type of deliverable Report
Security           PU
Partners           EMS, MORPH, UCY, HEIM
Contact Person     Argyris Kouloumbis
WP/Task responsible EMS
EC Project Officer Mr. Ch. Wilk
Abstract
(for dissemination) The first version of the design document for the idSpace platform.
Keywords List      System Design Document
# TABLE OF CONTENTS

List of Figures........................................................................................................................................ 4

1 Introduction .......................................................................................................................................... 5
  1.1 Position of idSpace platform in idSpace-project ................................................................. 5
  1.2 Work procedure for the definition of idSpace platform ..................................................... 6

2 The idSpace platform ..................................................................................................................... 7
  2.1 The existing background subsystems .................................................................................. 7
  2.1.1 Microcosmos .................................................................................................................. 7
  2.1.2 Kamala .......................................................................................................................... 8
  2.2 The Integration of the subsystems to the idSpace platform .............................................. 8

3 idSpace IMPLEMENTATION Concepts.................................................................................. 10
  3.1 General Philosophy ............................................................................................................... 10
  3.1.1 Modularity in idSpace .................................................................................................. 10
  3.1.2 Semantics in idSpace ................................................................................................... 11
  3.1.3 Context Awareness ..................................................................................................... 12
  3.2 System Design Concepts and Terminology ....................................................................... 13
  3.2.1 Workspaces ............................................................................................................... 13
  3.2.2 Area Structure .......................................................................................................... 13
  3.2.3 Role concept .......................................................................................................... 14

4 idSpace Features .......................................................................................................................... 16
  4.1 General idSpace features ................................................................................................. 16
  4.2 Support of Topic Maps contents .................................................................................... 16
  4.3 Multilingualism .............................................................................................................. 18
  4.4 Support of Multiple Portals/Sites .................................................................................. 20
  4.5 Search Facility ............................................................................................................... 20
  4.6 Templates (skinning) ...................................................................................................... 21

5 idSpace module architecture .................................................................................................... 22
  5.1 Overview .......................................................................................................................... 22
  5.2 Function modules .......................................................................................................... 22
5.2.1 Generic modules ................................................................................................. 23
5.2.2 Communication / Collaboration modules .......................................................... 25
5.2.3 Content management modules .......................................................................... 26
5.2.4 Content authoring modules .............................................................................. 26
5.2.5 Platform management modules ....................................................................... 27
5.2.6 Context Awareness Modules .......................................................................... 29
5.3 System Requirements ......................................................................................... 29

6 Communication interfaces ...................................................................................... 30

6.1 Overview ............................................................................................................. 30
6.2 Technology used for server communication ....................................................... 30
6.2.1 What are Web Services? .................................................................................. 31
6.2.2 Technologies involved ..................................................................................... 32
6.2.3 Why Web Services .......................................................................................... 34
6.3 Web Services interfaces in overview ................................................................. 35
6.3.1 Interface between Kamala and Microcosmos .................................................. 35

7 Summary / Conclusion .......................................................................................... 37

8 References .............................................................................................................. 39

Appendix I .................................................................................................................... 41

Appendix II .................................................................................................................. 43

Appendix III .................................................................................................................. 44
LIST OF FIGURES

Figure 1-1: Position of the idSpace Platform in the idSpace-project ......................... 5
Figure 2-1: idSpace-platform architecture ................................................................... 9
Figure 3-1: Example of Area structure definition ....................................................... 14
Figure 3-2: The permission for the user roles .............................................................. 15
Figure 4-1: Visual representation of the Topic-map ....................................................... 17
Figure 4-2: Multilingualism in the idSpace platform .................................................. 19
Figure 4-3: Skinning concept in idSpace ..................................................................... 21
Figure 5-1: idSpace platform core function modules in overview .............................. 23
Figure 6-1: Service invocation based on SOAP ......................................................... 31
Figure 6-2: The layers in web services communication ............................................... 33
Figure 6-3: Kamala communication model within idSpace ...................................... 35
1 INTRODUCTION

1.1 Position of idSpace platform in idSpace-project

The idSpace project has a research, a design and a development component. The idSpace platform will be the implementation of the findings of the research activities and the design activities.

The main objective of WP4 is to implement a web based platform that successfully incorporates synchronous tools for learning, exchange and storage of knowledge, as well as variety of asynchronous collaboration tools; the first version of the idSpace platform will be implemented through the integration of Microcosmos and Morpheus Software tools.

The first version will be evaluated by WP5 and the outcome of this process and the user requirements that will be collected will be used as input for the second version of the idSpace platform.

Furthermore for the second and third version of the platform the outcomes of WP2 and WP3 (ontologies for the creativity techniques and methods respectively the ontology for the context awareness) will be refined and translated into new functionality for the platform.

The outcome of WP4 will also be influenced by the research results of WP1 concerning issues such as creativity and pedagogical strategies as well as by the second evaluation phase conducted by WP5.
1.2 Work procedure for the definition of idSpace platform

The global work procedure for WP4 is that the 1st version of the platform and its design document will cover the merging of the two separate systems, which are Microcosmos by EMS and the Topic Maps technology and tools by Morpheus. This design document will thus only describe the two currently existing systems, their features and functionalities, the chosen technology for their communication as well as the process of merging the systems into one single platform.
2 THE IDSPACE PLATFORM

2.1 The existing background subsystems

The idSpace platform is based on pre-existing background technologies, provided by the technical partners. These two solutions build the cornerstones of the idSpace platform: Microcosmos, a communication and collaboration system provided by EMS. Kamala, a tool for capturing, sharing and integrating knowledge, information and ideas.

2.1.1 Microcosmos

Microcosmos is a collaboration and communication environment, especially created for educational environments. Microcosmos resulted from the School+ project, a 36-month educational project, funded by the EC within the “School of tomorrow” FP5 framework and from the UNITE project, another 30-month e-learning project funded by the EC within the “strengthen the integration of the ICT research effort in an Enlarged Europe” FP6 framework. Within idSpace, Microcosmos has evolved to support the following main features:

- Robust user permissions system (which can be fully edited and extended)
- Support of collaboration and communication
- Fully customizable visual interface
- Multi-language support (of both interface and content - for some modules)
- Ability to create pages and customize the functionality of each page
- Support of the creation of multiple portals using only a single installation
- Support of scheduled tasks via its internal scheduler
- Quick and robust search engine based on its internal content indexer

As one of the platform components, Microcosmos is named in the platform as idSpace-Microcosmos. idSpace-Microcosmos will be the front-end of the unified platform, upon which the other components will be seamlessly interfaced. idSpace-Microcosmos will be responsible for all communication / collaboration facilities. It will offer a wide range of functionality modules (ex. Workspaces, Announcements, Forums, Private Messaging etc.) with settings and options that will allow power users and administrators to fully customize them and place them anywhere on their pages.
2.1.2 Kamala

Kamala is a tool used for capturing, sharing, representing and integrating knowledge, which is also used as an ontology management environment. Kamala is built on Topic Maps technology (see 3.1.2).

Morpheus developed several Topic Maps applications for different domains and organisations: it developed innovative supporting tools for the Dutch Tax Office, the Police, for the Life-science domain but also for education. With the Dutch partner GeoActive it developed the MCIS (MAIUS curriculum information management) system. This system provides innovative solutions to quickly build and manage a curriculum by providing a subject-centric view so that teachers easily can see the subjects to cover, but also the overlap in different projects (Siebers, 2007).

Several organisations involved with education choose to use Topic Maps as a flexible knowledge management solution. One of the projects is a National Curriculum project (Befring et al, 2008, Brainbank Learning, 2008).

The advantages for these organizations are amongst others:

- A flexible index on the subjects of interest.
- The associative way of thinking that is reflected in Topic Maps. This also has been concluded in an eLearning project (Dichev, Dicheva, & Aroyo, 2004).
- The organizations can use Topic Maps to integrate with their existing systems and tools. In this respect idSpace shows also the quick and easily integration with Microcosmos.
- One can easily extend upon existing tools by just encapsulating the existing content and built a knowledge layer around it.

2.2 The Integration of the subsystems to the idSpace platform

The integrated platform is a heterogenic client-server platform (server-based architecture) which combines the benefits of each existing technology (the supported features are presented in detail in chapter 4, idSpace Features).

At least two servers are available to support the idSpace features:

- The idSpace-Microcosmos server provides support for general management, access rights, portal structure, communication and collaboration.
- The idSpace-Topic Maps server provides support for capturing knowledge and ideas in processes for e-learning, facilitating creativity, knowledge management and for associating distributed sources. Kamala provides support for displaying and managing the content of the Topic Maps used within idSpace.
The above mentioned servers communicate with each other via web service interfaces (see below). Each server provides services and is at the same time the client of the services provided by other servers. Based on the service-based architecture, further modules can be easily integrated via web service interfaces, independently from the location and the provider of the modules. The communication between the servers is performed in the background and is transparent to the end user.

The end user works with the platform using the graphical user interface provided by the platform.

![Diagram of idSpace-platform architecture](image-url)

Figure 2.1. idSpace-platform architecture
3 IDSPACE IMPLEMENTATION CONCEPTS

The idSpace Concepts describe the underlying ideas of the idSpace platform. Concepts of two different levels are described in this chapter. At the level of general philosophy, the overall principles followed in idSpace platform are described. At the level of realisation, the design concepts for selected aspects are described.

3.1 General Philosophy

3.1.1 Modularity in idSpace

The concept modularity describes the ability of configuring complex systems based on elementary, standardized parts. In idSpace, modularity occurs in different aspects:

- Modular system architecture
- Function modules
- Modular Topic Maps content

Modular system architecture

As opposed to monolithic proprietary solutions, the idSpace platform consists of individual components, which communicate via well-defined interfaces. Based on this approach, it is possible to create different instances of the platform by easy exchange of the components while the new components must just following the interface definition based on service oriented technology.

Function modules

The modular building blocks of the idSpace platform provide elementary functionality themselves. Following this concept, the platform can be easily expanded or adapted with minimal effort for the adapting.

Modular Topic Maps content

Kamala will be integrated in the Topic Maps module in the idSpace platform and will be responsible for:

- Managing the user interface
- Managing the administration of users and knowledge maps
- Managing the Topic Maps content

This subset of functions operates on top of the central Topic map-engine which provides the basic operations for storage and search facilities.
Technical realization of the modularity

Modules are the "building blocks" of idSpace. Each module is a logical program unit that contains its own independent functionality and serves a specific purpose (e.g. Forum, Topic Maps Viewer, Workspace etc.). Administrators can freely add modules to their idSpace site's pages using any common web browser and then configure those modules.

Currently there is a wide variety of modules available for the idSpace platform, and by using these modules, organisations are able to create custom sites that fulfil their needs (please refer to chapter 5).

Depending on the functionality of the modules, the modules provide their own user interface or web service interface when this module provides functionality used by other modules in the background. The modules, e.g. Topic Maps Viewer, can be added to the site, which is accessible only to administrators and allows them to perform various actions such as:

- Editing the module's content (i.e. if it is an Announcements module, you can add a new announcement from here)
- Configuring the module's Properties. These are general settings, common for every module. They include the module's title, view and edit permissions as well as the module's container.
- Cutting or copying a module and pasting it to another page.
- Changing the module’s position on the current webpage.
- Configuring the module's options, which are settings that are specific to each module (e.g. the number of announcements to display per page, required fields for User Registration, etc.).

Administrators can add modules to their pages by using the special Administrator Control Bar which allows them to choose from a list of all modules currently available.

3.1.2 Semantics in idSpace

In idSpace, knowledge, information and ideas will be represented using Topic Maps as a semantics technology. The creative process and its result will be captured and shared using Topic Maps.
Short explanation of the core concepts

Topic Maps is an open ISO standard (ISO 13250). The core concepts of Topic Maps are:

- **Topics** – Representing anything from people, countries, organizations and events to abstract concepts, files and documents.
- **Associations** – Representing the type of relationship between topics.
- **Occurrences** – Representing the resources and values relevant to a particular topic.
- **PSI’s** - Each topic, association and occurrence van be identified by a PSI (Published Subject Indicator). It contains an ID and a definition. With a PSI a topic can be identified across the border of systems, departments and even organizations.
- **Scope** – Scope is the context within which a topic is valid. It is defined in terms of ‘themes’. This theme is a ‘phrase’ describing a context. In other words, a theme is a topic that is used to limit the validity of a set of assignments
- **Merging** – Topics can be merged if they have the same PSI. This means that Topic Maps on different systems can be merged without lengthy programming efforts. Thus each Topic Map consists of the ontology and the content.

Using these concepts, an ontology-based knowledge map is constructed, which can than be queried. One of the main aspects of Topic Maps as a semantic technology is that it supports an associative way of thinking and communicating, which is closer to the way humans think and interact, than the relational model.

In short, by an associative way of thinking we mean that you don’t think in hierarchies and tree structures. Instead, like the human mind, you think by associating things, concepts and ideas with each other.

For more information about Topic Maps, please refer to:

   TAO of Topic Maps:  [http://www.ontopia.net/topicmaps/materials/tao.html](http://www.ontopia.net/topicmaps/materials/tao.html)

### 3.1.3 Context Awareness

The idSpace platform, amongst other characteristics, will be context-aware. Context is fundamental to the creative process. To support the creativity process the system has to keep track of the context, so as to be able to offer primarily suitable recommendations to users and/or teams. Only then can it be used to the team’s advantage.
Depending on the idSpace user’s competences, preferences, roles, history and ideas he is/was working on, etc. the system will be able to suggest suitable users to collaborate with and offer personalized assistance. Assistance can be in the form of relevant ideas previously developed, relevant resources (e.g. literature, URLs, etc.), suitable creativity techniques to use in a particular problem solving situation, experts to consult, etc.

3.2 System Design Concepts and Terminology

3.2.1 Workspaces

idSpace supports the creation and management of so-called workspaces, which promote the idea of a cross thematic approach. In a workspace, members from different scientific fields and backgrounds can participate to share their knowledge and communicate with each other.

Workspaces can be created by anybody who has the right to do so. Thus, not only administrators, but even simple members, who have acquired this privilege, can create workspaces and invite their peers, or any other interested party to participate, as well as create dedicated user groups for a particular workspace.

In technical terms, a workspace forms one of the main modules of the idSpace platform. It is a web structure (mini-site), that has its own pages, members and functionality modules, including its own repository (resource area). The workspace structure (i.e. pages and functionalities within a workspace) is not fixed. It can be modified by the site administrator to match the users or project needs. Every idSpace site can have a limitless number of workspaces.

Using workspace management, authorized users can create/edit/delete workspaces and assign users (individual users or groups) as their members.

The workspace owner is by default the user who creates the workspace. Only the workspace owners may invite members to their workspaces. A user can be member of one or more workspaces, but his/her user role doesn't change between workspaces (e.g. a user which has the role of "Expert" is always bound to that role).

3.2.2 Area Structure

In idSpace we distinguish collections of web pages based on the users’ allocated permissions, which form so-called Areas. We have Public Areas accessible by all users, and Private Areas accessible only by authorized users.
Following figure (3.1) illustrates how the default areas are laid out in idSpace. Nevertheless, Administrators have the ability to edit, delete or create new areas. For each available area, the “Users Allowed” indication shows which users may enter and view this area, the “Modules” indication shows the default area functionality, and the “workspace Owners” indication (for workspaces only) shows which users are allowed to edit/delete a workspace. In this example, the user Peter has access to the default Public Area as well as to the Private Area, from where he has further access to three workspaces.

![Area structure definition diagram]

**Figure 3.1. Example of Area structure definition**

### 3.2.3 Role concept

A role is a virtual set of characteristics, built to arrange and control the access rights of a idSpace user. Once a role has been assigned, the users gain all access rights offered by that role. These roles are comparable to keys. The more keys a user has, the more “doors” he can open.

A user in idSpace may have more than one role, thus possessing the union of the roles’ access rights. New custom roles can be created by the administrators.
In idSpace, there are five default roles:

- System Administrator
- Administrator
- Power User (Expert)
- User (Normal user)
- Unauthorized

Detailed description of each role can be found in Appendix I.

![Diagram showing permission for user roles](image-url)

*Figure 3.2. The permission for the user roles*
4 IDSPACE FEATURES

In this chapter, the general idSpace features are described in an overview. To characterise the platform, some specific features are described in detail.

4.1 General idSpace features

idSpace provides general management system features in the following aspects:

- User Management
- Content Management
- Collaboration and group work
- Topic Maps Editing and Management

4.2 Support of Topic Maps contents

The knowledge and ideas that are captured in the creativity sessions, are stored using Topic Maps technology (see 3.1.2.). The resulting knowledge maps can be viewed and edited using new Topic Map module.

The knowledge maps can be viewed as text or as a visual representation.

View as text

For each topic, association or occurrence that the user selects, all directly related other topics, associations and occurrences will be shown. Clicking on any of the related items results in the new item getting the focus and all of its own related topics, associations and occurrences will be shown. In this way, the user can navigate the entire knowledge map, without encountering a ‘begin’ or ‘end’ of the information.

View as visual representation

To visualize the contents and structure of a topic map, the “Topic-map Viewer” is used. Topics or topic types are represented by an ellipse while their relations are represented by lines connecting the ellipses. A very basic implementation of the “Topic-map Viewer” is shown in Figure 4-1.
**Editing the contents**

Information can only be edited when viewing the Topic Maps contents as text. All the information of the selected item that is shown can be edited in the same screen in which the user views the content. So while a user is navigating the topic map, he/she can also edit the information without leaving the screen.

The only restriction for editing content lies in:

- A user must have been assigned rights to edit the content
- A topic map contains its own constraints as information. In other words, the ontology contains content as well as information telling the system what a user can or can’t do. So the second restriction is that a user can’t edit information which violates these constraints.
4.3 Multilingualism

The idSpace platform supports the existence of multiple languages within a single environment. This means, that when a user selects a preferred language from the Language Selector menu, he or she is able to view the platform's interface and content (wherever available) in that particular language.

idSpace platform supports multilingualism in 2 levels:

- Multilingualism of the user interface
- Multilingualism of the content

**Multilingualism of the user interface**

The idSpace platform can be used in the local language of the user. All user interfaces are designed in a way that the language used for the user interface can be selected by the user individually.

The default sequence for setting the preferred language is given in the list below. The dropdown language selection has the highest priority. Once the user has changed his/her preferred language in the dropdown language selection, the whole system is switched to the selected language as far as the buttons and texts in the user interface are concerned.

1. Dropdown language selection
2. User language setting (from a saved Cookie)
3. Site language setting
4. Platform default language (EN)
5. Browser default language

The preferred language can be passed from the portal to all servers which need this information to generate the user interface.

During the project period, the English language will be used as the default language in the platform. Additional languages can easily be integrated later by the administrator.

In the idSpace portal, support for adding new language is integrated. The template for the language package is in the form of text files. Each module has a text language file that can be easily translated into any language. This file is used for presenting the module’s interface in the selected language.

To add a new language to the platform, a new language definition must be created by the administrator.
Multilingualism of the content

The idSpace platform also includes modules that have the ability to support their content in different languages—provided that the translated versions of this content are stored in the platform. These modules are:

- Announcements / Events
- Text_HTML
- Forums
- Topic Maps
- Journal / Blog
- Tasks
- Workspace Announcements / Workspace Events

When retrieving information (content), the user can view information in any of the languages in which information has been entered previously. When certain pieces of information are not present, the system can fall back to that same information in the default language.
4.4 Support of Multiple Portals/Sites

The platform can support multiple portals and sites with just a single installation. This allows the platform to be installed on a web server and support multiple web sites through a single System Administrator account.

System Administrator and Administrator Accounts

Due to the platform's support for multiple portals and sites, the distinction between System Administrator and Administrator is important. By differentiating administrative options between System (Host) level and individual portal or site level, the idSpace platform allows administrators to manage any number of portals or sites – each with their own look and identity – all under one System Administrator account. Each portal or site can be assigned to a different IP address on the server.

Logging in to idSpace as a System Administrator allows the user to manage settings related to site hosting (i.e. create new portals or sites, create new language definitions, install or uninstall modules etc.). The System Administrator has access to the list of all hosted portals or sites, their individual hosting information, and has the ability to edit, modify, or delete a specific portal or site.

Logging in to idSpace as an Administrator allows the user to manage settings related to his or her own individual portal or site. The Administrator may edit, modify, or delete settings specific to his or her own portal or site, but has no access to other idSpace portals or sites operating under the same host.

4.5 Search Facility

The idSpace platform provides a search facility through its search module. Like all other idSpace modules, the search module is installed in the Microcosmos part of the platform and is able to search for content locally in the Microcosmos server (Site search), or remotely (Topic Maps search), via web services. The search module offers two search options:

By selecting ‘Site search’, the user can search for keywords in the modules of idSpace platform that are able to hold content (See Appendix II).

In the background of this search facility, a search engine is integrated in idSpace platform. Text in idSpace platform is analysed and indexed in advance, so the search facility is fast and returns results in order of relevance.

By selecting ‘Topic Maps search’, the user can search for keywords in every listing of topics within the idSpace platform. The overall search functionality will search the entire Topic Map, displaying the topic results and some basic ontological knowledge about the topic.
4.6 Templates (skinning)

Most visual elements in the idSpace platform are based on templates. Templates are empty HTML files, which the platform fills with content wherever necessary. The use of templates allows separating the content from the layout. In this way, administrators can change the layout of the platform without affecting its content.

There are two types of templates in the platform:

A. Skins (The visual template of each idSpace webpage):
   - It defines the layout of that webpage, and consists of an HTML template file (webpage structure) and a style sheet file (colour scheme). The page skin divides a webpage into areas (so-called panes), in which modules are placed.

B. Containers (The visual template of a module):
   - It defines the general design and colour theme of a module, and consists of an HTML template file and a style sheet.

Once a page skin and its containers have been compiled together, the areas marked with [brackets] as shown in the following figure, are filled with the appropriate data from the idSpace database.

![Example of a Page Skin](image1)

![Example of a Module Container](image2)

Figure 4.3. Skinning concept in idSpace
5 IDSPACE MODULE ARCHITECTURE

5.1 Overview

idSpace is a fully modular platform, which means that its capabilities can be easily extended by installing additional modules to the ones that constitute the core of the platform.

Component architecture of the idSpace platform

The idSpace platform consists of two main components (see Chapter 2.1), which provide the following functionalities:

- Communication and collaboration
- Topic Maps management
- Content management

The communication and cooperation management are supported by the idSpace component, idSpace Microcosmos. The Content management is supported by the idSpace components, idSpace Microcosmos and idSpace Topic Maps Application. Finally, the Topic Maps management is supported by the idSpace component, Topic Maps Application.

This component architecture supports communication and collaboration as well as the storage and retrieval of content both locally as well as with (remote) users. To support users in different countries, the modules can be configured in different languages. To edit and manage the knowledge structure and the ontology the user will be supported by a Topic Maps editing tool, Kamala.

As shown in Figure 5.1 (below), each of the components, based on a set of function modules, provides specific functionality. The collaboration of the function modules ensures the functionality of the whole platform. In the following chapters, the core function modules are listed and described in detail.

5.2 Function modules

idSpace encompasses the following types of modules which will be described in more detail. Further modules required can be added to the idSpace platform in a later phase of the project.
- Generic modules
- Platform Management modules
- Content Management modules
- Context Awareness modules

![IdSpace-GUI Diagram](image)

**Figure 5.1.** idSpace platform core function modules in overview

### 5.2.1 Generic modules

**Login**
This module verifies registered users and if successful, allows them to enter the platform as members: a ‘remember me’ feature is also supported.
**Registration/User Info**

This module displays a registration form that allows users to complete their personal information in order to become registered members of idSpace. Upon registering, users must provide a valid username, password and email address. Users can choose if their (external) email address should be visible to other users, or not. Apart from the mandatory fields, the registration form can also include several optional fields, which can be defined by the administrators (for example birth date, phone number, city, etc). Once a registration form has been submitted, administrators verify and decide whether to grant access to this newly-registered user. Once a user is registered and logs in, he or she is able to view and edit his or her personal info.

In the background of the registration the user profile is managed.

**Personal Notes**

This module allows users to create or edit personal notes and specify a date or time on which the user wants to be alerted about that note, through their agenda or through a pop-up message. A HTML WYSIWYG editor is used for editing the personal notes.

**Linkbox**

This module renders a list of hyperlinks. These hyperlinks could be internal (to pages within the platform) or external (to other websites or external resources). Authorised users can insert any number of hyperlinks into this module. These hyperlinks are available to all users, which can click on and visit the pages that the links refers to. Each link consists of a link title (name) and the link's URL address. Internal links can be selected from a dropdown list of available pages. An interface is provided that allows the adding, removing or editing of links.

**Messaging Info Box**

This module is able to display short info from the user's Internal Messaging System: in more detail, it displays the total number of messages received, the number of new unread messages, the total number of messages sent by the user and the number of sent messages that remain unread.

**Workspace Viewer**

This module allows users to immediately view all workspaces that they are members of as a list with those workspaces’ titles. By clicking on a particular title, they can enter a particular workspace.
**Agenda**

The Agenda module, which acts as a regular calendar, allows users to get an immediate overview of all activities concerning them, for a specific time range. The users can choose a particular date, or a certain time range, and for that time range get a list of all activities concerning them in chronological order. Such activities can be: events taking place within this time range, personal notes written on any of the days within this time range, tasks they have been assigned to and have to submit by a specific date falling within this time range, etc.

5.2.2 Communication / Collaboration modules

**Forums**

This module allows workspace members to interact through a series of posts and replies. Authorised users can create or edit forum categories (sub-forums) and post/edit discussion topics within the forums. A topic may contain attached files (uploaded by the user, or copied/pasted from the Resource Area) and may also be of type announcement (always on top, does not allow replies), sticky (always on top, allows replies) or normal.

After a topic has been created, workspace members may post their replies and even attach files to their replies. idSpace users who are authorised to moderate forums can ‘close’ forums or topics (making them read-only) and also move topics from one forum to another (this is, for example, useful if a topic is posted by mistake in a wrong forum category).

**Internal Messaging System**

This module allows idSpace members to communicate and send messages to each other. Messages can be sent to single or multiple recipients, which can be selected from a list of available users. Apart from creating new messages, users can visit their Inbox and Outbox where they can read messages, reply to messages and view the status of messages that they have sent. Extra options include setting your, the user’s, personal signature to be included in every message as well as whether a copy of received messages should be forwarded to another external email address (outside of the idSpace platform).

**Users online**

This is the module which provides every user with information about the members, using the following features:

- Ability to display the number of registered members.
- Ability to display the number of registered users that are currently online.
5.2.3 Content management modules

**Topic Maps**

This is a multifunction module that is able to manage the topic maps as well as to author their content.

There will be two layers of content in the topic maps. Firstly there is the meta-content, containing information about the supported creativity techniques and context information. Secondly there is the actual set of knowledge and ideas.

The power of Topic Maps is that these two layers are blended into a single topic map, which can freely and seamlessly be browsed by the user. Therefore, managing the two layers of content down to managing a single topic map.

The combination of user rights and constraints incorporated in the ontology can provide a broad range of possibilities to direct the content management. Both the user rights and constraints are part of the underlying ontology.

For example, it is possible that only privileged users can edit the information concerning the creativity techniques and that normal users can only edit information from creativity sessions in which they participated themselves.

5.2.4 Content authoring modules

**Topic Maps**

Please refer to § 5.2.3

**Journal**

The Journal module is very similar in functionality to a common web blog. A list of usernames is provided from which users can select and view each other’s journal. A Word-style WYSIWYG editor is used for editing the journal.

**Text HTML Box**

This module is able to render a snippet of HTML or text: In more detail it allows authorised users to edit HTML or text snippets directly through a Word-style WYSIWYG editor, which is included. The Word-style WYSIWYG editor allows users to format HTML or text snippets, to insert images, tables, bulleted lists, links and paste content from other sources. This module is used in a general context to create an HTML text.
**Announcements / Events**

This module allows authorised users to create or edit announcements and events using an HTML WYSIWYG editor. A start and end date can be set for each announcement or event which defines when they will be displayed and when they will expire. Reoccurring events will appear every year on the same date. Additionally, events can be viewed from the agenda by order of their start date.

**5.2.5 Platform management modules**

**Role management**

The role management allows Administrators to create, edit or delete roles which are then assigned to the users of the platform.

**File Management**

The File Management area, which uses a ‘Windows Explorer style’ layout for displaying files and folders, allows Administrators to manage files which visually shape an idSpace site beyond its default visual elements. Such visual elements are:

- Containers (i.e. module templates)
- Images
- Skins (i.e. page templates)
- Style Sheets

The directories that hold these elements (i.e. containers, images, skins and styles) are protected by the platform and therefore cannot be deleted.

**Site Management**

System Administrators can edit settings related to site hosting (i.e. create new portals or sites, create new language definitions, install or uninstall modules etc.). They also have access to the list of all hosted portals/sites, their individual hosting information, and have the ability to edit, modify, or delete a specific portal/site.

Site Administrators can edit general settings of their site such as the default skins, containers, language, time zone, home page etc.

**Page Management**

This module provides an interface for creating, editing or deleting pages for an idSpace site. Pages are displayed in a tree list format allowing administrators to also change page
position and hierarchy. Administrators can choose the appearance (skin) as well as the view/edit permissions of each page.

**User Management**

This module allows Administrators to create, edit or delete users. Administrators have the option of banning users when necessary. A search field is provided to quickly find a specific user.

**Group Management**

Groups can be created, edited or deleted and users assigned to them.

**Disposal Management**

Deleted items such as forums, pages, files, modules can be restored to their previous state or permanently deleted (shredded).

**Workspaces Management**

This module allows Administrators to create, edit or delete a Workspace and assign users to these Workspaces as their members. A brief description can be provided for each Workspace. Before creating the first Workspace, administrators have to define the general structure of Workspaces.

**Log Management**

This module displays user actions within a idSpace site. Information such as the date/time, action type, IP address and description of each user action is provided. Each action entry is colour-coded, based on the action type. Fields are provided to allow filtering of user actions by type, username, IP address and date/time.

**Module Management**

This module displays all available modules and their information. Modules can be installed or uninstalled from a idSpace site.

**Language Management**

This module displays a list of available language definitions. System Administrators can add edit or delete language definitions.
5.2.6  Context Awareness Modules
These will be a set of modules that will monitor context elements (User, Task/Idea, Social Environment and System) and based on these will provide recommendations concerning: people to join a group for solving a particular problem, peers/experts to consult, relevant resources to explore, related ideas previously developed, suitable creativity technique, etc.

5.3  System Requirements
The idSpace Microcosmos components are implemented in the programming language PHP while the idSpace Topic Map components are implemented in the programming language JSP. Both technologies run on an Apache Server.

The following system requirements are identified for the 2 parts of the platform:

**Software requirements for server**
- Web server Apache >= 2.2.0
- Support of PHP >= 5.2.5
- Data base MySQL >= 5.0.51a

**Hardware requirements for server**
- Minimum 256 MB RAM for Microcosmos
- Minimum 512 MB RAM for Topic Maps
- Minimum 50 MB free space on hard disk for each part of the platform and an additional space for the contents
6 COMMUNICATION INTERFACES

6.1 Overview

The idSpace platform is a composition of function components. Each of the components builds a client-server architecture. Beside the client server communication within the components, the definition of the communication between the components is one important design task of the integration.

The communication between the components takes place on different levels:

- The communication between the user interfaces.
- The communication between the server components.

The communication between the user interfaces

The user interfaces are

- The idSpace portal
- Kamala

There is only communication between the portal components, for example, between the workspace GUI and Kamala. To initialize the Topic Maps Viewer, the GUI needs to call a Kamala JavaScript function with the appropriate parameters that provide context information. The Kamala scripts will then, using its own server side Java code, display the Topic Maps Viewer within the workspace GUI.

The communication between the server components

For the communication between the server components, the standard technology web service should be used. In the following chapter, the web service interface will be described in more detail.

In the next section, the server side communication is described in more detail.

6.2 Technology used for server communication

idSpace platform is a heterogeneous system. The server components are based on different technologies. The idSpace Microcosmos server is based on PHP and the Topic Maps server is based on a set of java scripts on the front end, communicating with the back end using Java and a Topic Map specific query language.
To support the interaction between the servers using different technology, the web services technology is selected for the server-side communication.

### 6.2.1 What are Web Services?

The W3 Group defines a web service as a software system designed to provide a standard means of interoperating among different software applications, running on a variety of platforms and/or frameworks (W3C, 2004). Its ultimate goal is to make software functionality available over the Internet, which means that different programs based on different technologies, e.g., PHP, ASP, JSP, JavaBeans, COM, can make a request to a program running on another server (a Web Service) and use that program’s response in a website, WAP service, or other application (W3Schools, 2008).

One of the goals of software development in recent years has been the need of connecting people, information, and processes across different platforms. The first step in this direction was made with the development of the Web – finally, all major platforms could interact, using Web browsers.

Web services made it possible for applications to publish not only data, but to have also its functions or methods published, located, and invoked across the Web (Manes, 2008). Based on the client-server architecture, the procedure is simple: the web service provider defines a format for requests for its service and the response the service will generate. Once this format is known, a computer can make a request to the web service across the network, whereby it passes a parameter data with the request, calling the service over corporate intranets, extranets, or across the Internet. The web service performs an action, generates a result and sends the response back.

![Figure 6.1. Service invocation based on SOAP](image-url)
6.2.2 Technologies involved

The broad acceptance of web services is due to a simple fact: they rely on a core set of standards that describe the syntax and semantics of software communication, making it easy to jump across the widespread UNIX/Windows divide. The basic platform is simple eXtensible Mark-up Language (XML) based access over plain HTTP. XML was designed as the universal language for representing and transmitting structured data that is independent of programming language, software platform, and hardware (Cooney, 2002). While XML provides a meta language which can be used between different platforms and programming languages and still express complex interactions between clients and services, the HTTP protocol is the most used Internet protocol (HTTP, 2008).

Web services use XML to code and decode the data: servers convert the XML message to a middleware request, process it and convert the results back to XML.

Other technologies involved:

SOAP: the Simple Object Access Protocol (SOAP) forms the foundation layer of the Web services stack. SOAP is an open, XML-based transport protocol used to exchange data with calling clients. It defines a uniform way of passing XML-encoded data. HTTP is the underlying communication protocol (Wikipedia, 2008a).

WSDL: the Web Services Description Language (WSDL) is a template for how services should be described and bound by clients. It provides a mechanism to describe what a web service can do, where it resides, and how to invoke it (Didio, 2003).
REST: the Representational State Transfer (REST) is another way of providing web services (Wikipedia, 2008b; Mitchell,, 2008). REST was proposed by Roy Fielding (Fielding, 2000), one of the principal authors of the HTTP protocol (Fielding, 2008). It is therefore that REST is based on HTTP in its purest form. Principals of REST are:

- Application state and functionality are abstracted into resources
- Every resource is uniquely addressable using a universal syntax for use in hypermedia links
- All resources share a uniform interface for the transfer of state between client and resource, consisting of
  - A constrained set of well-defined operations
  - A constrained set of content types, optionally supporting code on demand
- A protocol which is:
  - Client-server
  - Stateless
  - Cacheable
  - Layered

The abstraction into resources produces (when applied correctly) well formed URL’s that specify a unique resource to be used when processing the request. The response to a REST request is a representation of the requested resource. The representation supplied is dependent on the requested data form (XML, text, HTML, JSON, etc).

![Diagram](image)

Figure 6.2. The layers in web services communication
6.2.3 Why Web Services

Although web services are a relatively new solution to a well-known problem, they are, by no means, the only one. As the challenge of accessing programs running in remote computers or distributed applications has existed for quite a while, several technologies have been developed to address the problems of interoperability with low cost and platform independence.

Adding complexity to the specific case of the idSpace platform, the technology used has to guarantee free communication flow between partners without the danger of blockage by firewalls.

Some existing alternatives were considered and discarded because they did not satisfy all the requirements.

Traditional RPC-style middleware like RPC, CORBA, RMI, and DCOM are fragile and expensive, have high maintenance costs, and are hard to use. They rely on very brittle tightly coupled connections (Rampally, 2006) that can break if any modification is made to the application, creating a maintenance nightmare.

In contrast, web services support loosely coupled connections, which minimizes the impact of changes to applications. Web services are self-describing software modules which encapsulate discrete functionality. As a web service interface provides a layer of abstraction between the client and server, a change in one doesn't necessarily force a change in the other. Because the abstract interface makes it easier to reuse a service in another application, loose coupling reduces the cost of maintenance and increases reusability.

Web service messages are transmitted as SOAP or REST formatted messages using HTTP over port 80, an open port for Web server firewalls. As SOAP and REST are designed to serialize both simple and complex data types, SOAP or REST messages are in XML format, simply text with no binary data, meaning SOAP or REST can be used to easily transmit complex data from the client to the server and back again.

And finally, in regard to interoperability, web services are both platform-independent and language-independent. All the three fundamental layers (Service Transport, XML Messaging and Service Description) use open and well-known protocols (Powell & Lyon, 2002) meaning web services can be easily developed using any language and be implemented on any platform. They can also be accessed by any other application, regardless of either's language or platform.
6.3 Web Services interfaces in overview

In this chapter, the interfaces for web services communication are described in their usage context. The formal specification of the web services interfaces is defined in the Appendix III.

6.3.1 Interface between Kamala and Microcosmos

Communication between Kamala and Microcosmos is done by using web services. The communication layer of Kamala is based on the REST web services technology. To extend Microcosmos’s functionality to include Topic Maps, these services are routed through Microcosmos. For a client this seems to be a single server, providing REST and SOAP services.

The Topic Maps are stored and maintained on a separate server, containing the necessary software to operate Topic Maps.

When using Kamala as the displaying tool for the Topic Map content, Kamala uses AJAX technology to request data from the Topic Maps server. This request is a REST request. The request is send to a local idSpace-Microcosmos address, which is then redirected by a proxy to the Topic Maps server.

The description of the services being provided through the proxy are work in progress, and the latest versions can be found at http://docs.mssm.nl/kamala-rest-0.1dev/.

![Diagram](image-url)
Exchange of information between Microcosmos and Kamala will be done by using the services provided. For example, Kamala can request the workspace users and authorization information from Microcosmos using the SOAP services. The same applies in reverse order.
7 SUMMARY/CONCLUSION

This document described the many ways in which the idSpace platform enhances and integrates the technologies of its two main components: idSpace Microcosmos and idSpace Topic Maps.

The idSpace platform uses a modular architecture to create building blocks for the idSpace system. Any module that serves a specific purpose can be added, deleted and managed by administrators using any common web browser, thus making the idSpace platform a very flexible and adaptable system to work with for any organisation.

idSpace supports the creation and management of so-called workspaces, which promote the idea of a multidisciplinary approach. In a workspace, users or groups of users of the platform can participate. Permissions to all modules can be managed flexibly using roles. These roles give access to users, or groups of users, as and when required for each module.

The (meta-) data in the idSpace platform will be stored using the Topic Maps-technology. Meta-data, a combination of constraints and an ontology specifying topic types and their possible relationships, is used to create the structure for a flexible semantic network. This network contains data such as creativity techniques, context information and ideas, which can easily be linked to one another. The combination of data and metadata can be used to suggest relevant information or show related problems, in order to enhance creativity.

The knowledge captured can be viewed and edited using tools incorporated in the idSpace platform. Restrictions and access control for users or user roles is specified in the Topic Map ontology itself and can therefore be edited using the same tools.

Depending on the idSpace user’s competences, preferences, roles, ideas he is/was working on, etc. the system will be able to suggest suitable users to collaborate with and offer personalized assistance. Assistance can be in the form of relevant ideas previously developed, relevant resources (e.g. literature, URLs, etc.), suitable creativity techniques to use in a particular problem solving situation, etc.

A further benefit of the idSpace platform is its being developed as a multilingual platform. This will allow users to learn and communicate in their native tongue, or their preferred foreign language.

The idSpace platform integrates the two existing technologies mentioned throughout the document to create a coherent system by using a mixture of XML web services between the servers.
Exchange of information between Microcosmos and Kamala will be done by using the services provided. For example, Kamala can request the workspace users and authorization information from Microcosmos using the SOAP services. The same applies in reverse order.
8 REFERENCES


Appendix I

*idSpace default roles and their default access rights*

**System Administrator**
- Has full access permission to everything

**Administrator**
- Has full access permission to everything except for access to System Administrator Management pages (Host Management, Database Management, Module Management, Lang Management, Topic Maps Management).

**Power User (Expert)**
- Access to Login form
- Create/Edit/Delete Users
- Create/Edit/Delete Groups
- Create/Edit/Delete Roles
- View/Create/Edit/Delete Personal Notes
- View/Create/Edit/Delete Workspaces
- View/Create/Edit/Delete Tasks Units, Tasks, Task submissions, Evaluations & Comments
- View Trace Viewer
- View Agenda
- View/Create/Edit/Delete Journal
- View/Create/Edit/Delete Private Messages
- View/Create/Edit/Delete Announcements
- View/Create/Edit/Delete Events
- View/Create/Edit/Delete Forums, Topics & Replies
- View/Create/Edit/Delete/Upload files in the user’s My Resource Area
- View/Create/Edit/Delete/Upload files in the public Resource Area
- View/Create/Edit/Delete/Upload files in Resource Area bound to Workspaces he owns
- Access to Search
- View Topic Map ontologies
- Edit Topic Map ontologies
- View Topic Maps
- View/Create/Edit/Delete Topics, Associations & Occurrences
**User**

- Access to Login form
- View/Create/Edit/Delete Personal Notes
- View Workspaces
- View Tasks Units, Tasks, Evaluations
- View/Create/Edit/Delete Task submissions & Comments
- View Agenda
- View Forums
- View/Create/Edit/Delete Topics & Replies
- View Topic Maps
- View/Create Topics, Associations & Occurrences
- Edit/Delete Topics, Associations & Occurrences created by the user
- View/Create/Edit/Delete Journal
- View/Create/Edit/Delete Private Messages
- View Announcements
- View Events
- View/Create/Edit/Delete/Upload files in the user’s My Resource Area
- View/Create/Edit/Delete/Upload files in Resource Area bound to Workspaces he is a member of
- View files in the public Resource Area
- Access to Search

**Unauthorized**

- Access to Login form
- Access to Registration form
- View Public Announcements
- View Public Events
- View files in the public Resource Area
- Access to Search
Appendix II

The following table shows what is returned by each module after a ‘Site search’.

<table>
<thead>
<tr>
<th>MODULES</th>
<th>SEARCHABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generic</strong></td>
<td></td>
</tr>
<tr>
<td>Announcement/Events</td>
<td>Search results are returned</td>
</tr>
<tr>
<td>Linkbox</td>
<td>No search results are returned</td>
</tr>
<tr>
<td><strong>Private</strong></td>
<td></td>
</tr>
<tr>
<td>Personal Notes</td>
<td>Search results are returned from the current user's Personal Notes only</td>
</tr>
<tr>
<td>Journal</td>
<td>Search results are returned</td>
</tr>
<tr>
<td>Messaging System</td>
<td>Search results are returned from the current user's incoming and outgoing messages only</td>
</tr>
<tr>
<td>Trace Viewer</td>
<td>No search results are returned</td>
</tr>
<tr>
<td>Workspace Viewer</td>
<td>Search results are returned from the current user's Workspace list only</td>
</tr>
<tr>
<td>My Resource Area</td>
<td>No search results are returned</td>
</tr>
<tr>
<td>User Info</td>
<td>Search results are returned</td>
</tr>
<tr>
<td>Agenda</td>
<td>No search results are returned</td>
</tr>
<tr>
<td><strong>Workspace</strong></td>
<td></td>
</tr>
<tr>
<td>Workspace Announcement/Events</td>
<td>Search results are returned only for Announcement/Events of Workspaces which the user is a member of</td>
</tr>
<tr>
<td>Workspace Resource Area</td>
<td>No search results are returned</td>
</tr>
<tr>
<td>Tasks</td>
<td>Search results are returned only for Tasks of Workspaces which the user is a member of</td>
</tr>
<tr>
<td>Forum</td>
<td>Search results are returned only Forums of Workspaces which the user is a member of</td>
</tr>
</tbody>
</table>

Search results depend on context
Appendix III

Introduction

In this section, you can find the application programming interface of the web services in the idSpace platform. The web services are provided by the idSpace server components and are presented by the server components as service provider. The idSpace modules communicate with the server via the HTTP interface and the web services interfaces.

API of the web services provided by idSpace Microcosmos

<table>
<thead>
<tr>
<th>Method Name</th>
<th>validate_session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Returns if session is valid</td>
</tr>
<tr>
<td></td>
<td>Return values: 1 if session ID is valid or 0 otherwise</td>
</tr>
<tr>
<td>Input Parameters</td>
<td>Array('session_id' =&gt; 'xsd:string')</td>
</tr>
<tr>
<td></td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>'session_id' : the current session ID</td>
</tr>
<tr>
<td>Output Parameters</td>
<td>Array('is valid' =&gt; 'xsd:int')</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Name</th>
<th>get_user_language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Returns current user language assigned to a user</td>
</tr>
<tr>
<td></td>
<td>Return values: language key if session ID is valid i.e. 'en-GB' or -1 otherwise</td>
</tr>
<tr>
<td>Input Parameters</td>
<td>Array('session_id' =&gt; 'xsd:string', 'user_id' =&gt; 'xsd:int')</td>
</tr>
<tr>
<td></td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>'session_id' : the current session ID</td>
</tr>
<tr>
<td></td>
<td>'user_id': User ID</td>
</tr>
<tr>
<td>Output Parameters</td>
<td>Array('language key' =&gt; 'xsd:string')</td>
</tr>
</tbody>
</table>
### get_username

**Documentation**

Returns the username of a user if session ID is valid or -1 otherwise.

**Input Parameters**

```
Array('session_id' => 'xsd:string')
```

*Where*

'session_id': the current session ID

**Output Parameters**

```
Array('username' => 'xsd:string')
```

---

### get_ws_name

**Documentation**

Returns workspace name if session ID is valid or -1 otherwise.

**Input Parameters**

```
Array('session_id' => 'xsd:string', 'ws_id'=> 'xsd:int' )
```

*Where*

'session_id': the current session ID  
'ws_id': Workspace ID

**Output Parameters**

```
Array('ws_name' => 'xsd:string')
```

---

### get_userid_from_username

**Documentation**

Returns user ID from username, if session ID is valid or -1 otherwise.

**Input Parameters**

```
Array('session_id' => 'xsd:string', 'user_name'=> 'xsd:string' )
```

*Where*

'session_id': the current session ID  
'user_name': the user’s username

**Output Parameters**

```
Array('user_id' => 'xsd:int')
```
<table>
<thead>
<tr>
<th>Method Name</th>
<th>add_log_item</th>
</tr>
</thead>
</table>
| Documentation | Logs an action  
Return values: 1 if session ID is valid or 0 otherwise |
| Input Parameters | Array('session_id' => 'xsd:string', 'type' => 'xsd:string',  
'legend_code' => 'xsd:int', 'description' => 'xsd:string',  
'item_id' => 'xsd:int') |
| Where         | 'session_id': the current session ID  
'type': type of log  
e.g. 'Create Announcement' or 'Topic Restored'  
'legend_code': Legent Code  
'description': log description  
'item_id': item ID that is logged (or NULL) |
| Legent Code   | D_LOG_ITEM_CREATED: 1  
D_LOG_ITEM_UPDATED: 2  
D_LOG_ITEM_DELETED: 3  
D_LOG_OPERATION_SUCCESS: 4  
D_LOG_OPERATION_FAILURE: 5  
D_LOG_ITEM_VIEWED: 6  
D_LOG_ITEM_RESTORED: 7  
D_LOG_ITEM_SHREDDED: 8  
D_LOG_MODULE_UNINSTALLED: 9  
D_LOG_CONTROL_DELETED: 10  
D_LOG_CONTROL_SHREDDED: 11  
D_LOG_CONTROL_RESTORED: 12  
D_LOG_MODULE_INSTALLED: 13 |
<p>| Output Parameters | Array('retval' =&gt; 'xsd:int') |</p>
<table>
<thead>
<tr>
<th>Method Name</th>
<th>get_roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Returns roles assigned to a user</td>
</tr>
<tr>
<td></td>
<td>Return values: role ID if session ID is valid or -1 otherwise</td>
</tr>
<tr>
<td>Input Parameters</td>
<td>Array('session_id' =&gt; 'xsd:string', 'user_id' =&gt; 'xsd:int')</td>
</tr>
<tr>
<td>Where</td>
<td>'session_id' : the current session ID</td>
</tr>
<tr>
<td></td>
<td>'user_id': User ID</td>
</tr>
<tr>
<td>Output Parameters</td>
<td>Array('xml_roles' =&gt; 'xsd:string')</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Name</th>
<th>get_groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Returns the groups a user is assigned to</td>
</tr>
<tr>
<td></td>
<td>Return values: xml structure of group IDs if session ID is valid or -1 otherwise</td>
</tr>
<tr>
<td>Input Parameters</td>
<td>Array('session_id' =&gt; 'xsd:string', 'user_id' =&gt; 'xsd:int')</td>
</tr>
<tr>
<td>Where</td>
<td>'session_id' : the current session ID</td>
</tr>
<tr>
<td></td>
<td>'user_id': User ID</td>
</tr>
<tr>
<td>Output Parameters</td>
<td>Array('xml_groups' =&gt; 'xsd:string')</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Name</th>
<th>get_my_workspaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Returns current users workspaces</td>
</tr>
<tr>
<td></td>
<td>Return values: xml structure of workspace IDs if session ID is valid or -1 otherwise</td>
</tr>
<tr>
<td>Input Parameters</td>
<td>Array('session_id' =&gt; 'xsd:string')</td>
</tr>
<tr>
<td>Where</td>
<td>'session_id' : the current session ID</td>
</tr>
<tr>
<td>Output Parameters</td>
<td>Array('XML_results' =&gt; 'xsd:string')</td>
</tr>
<tr>
<td>Method Name</td>
<td>get_user_workspaces</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
</tr>
</tbody>
</table>
| **Documentation**| Returns a users workspaces  
Return values: xml structure of workspace IDs if session ID is valid or -1 otherwise |
| **Input Parameters** | Array('session_id' => 'xsd:string', 'user_id'=> 'xsd:int')  
Where  
'session_id' : the current session ID  
'user_id': User ID |
| **Output Parameters** | Array('XML_results' => 'xsd:string' ) |

<table>
<thead>
<tr>
<th>Method Name</th>
<th>check_ws_permission</th>
</tr>
</thead>
</table>
| **Documentation**| Returns if user is member of the workspace  
Return values:  
• 1 if the user is a member of the workspace  
• 0 if the user is not a member of the workspace  
• -1 otherwise (not valid session ID) |
| **Input Parameters** | Array('session_id' => 'xsd:string', 'ws_id'=> 'xsd:int' )  
Where  
'session_id' : the current session ID  
'ws_id': Workspace ID |
| **Output Parameters** | Array('is_valid' => 'xsd:int' ) |

<table>
<thead>
<tr>
<th>Method Name</th>
<th>check_session</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td>Returns user’s ID for a valid session or -1 otherwise</td>
</tr>
</tbody>
</table>
| **Input Parameters** | Array('session_id' => 'xsd:string' )  
Where  
'session_id' : the current session ID |
<p>| <strong>Output Parameters</strong> | Array('user_id' =&gt; 'xsd:int' ) |</p>
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Documentation</th>
<th>Return values:</th>
<th>Input Parameters</th>
<th>Output Parameters</th>
</tr>
</thead>
</table>
| **verify_phone_number**  | Returns session ID if the user is a member of the required workspace and if his phone number is verified | • session ID on success  
• 0 if the user is unverified or banned  
• -1 otherwise | Array('phone_number' => 'xsd:string', 'ws_id'=> 'xsd:int' ) | Array('session_id' => 'xsd:string' ) |
| **verify_email_address** | Returns session ID if the user is a member of the required workspace and if his email address is verified | • session ID on success  
• 0 if the user is unverified or banned  
• -1 otherwise | Array('email address' => 'xsd:string', 'ws_id'=> 'xsd:int' ) | Array('session_id' => 'xsd:string' ) |
<table>
<thead>
<tr>
<th>Method Name</th>
<th>login</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td>User login, returns session id</td>
</tr>
<tr>
<td></td>
<td>Return values:</td>
</tr>
<tr>
<td></td>
<td>• 0 if the user is not a member of the platform</td>
</tr>
<tr>
<td></td>
<td>• 1 if the user is an unverified member of the platform</td>
</tr>
<tr>
<td></td>
<td>• 2 if the user is an banned member of the platform</td>
</tr>
<tr>
<td></td>
<td>• session ID for a valid, verified and unbanned member of the platform</td>
</tr>
<tr>
<td></td>
<td>• -1 otherwise</td>
</tr>
<tr>
<td><strong>Input Parameters</strong></td>
<td>Array('username' =&gt; 'xsd:string', 'password' =&gt; 'xsd:string')</td>
</tr>
<tr>
<td></td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>'username': the user’s username</td>
</tr>
<tr>
<td></td>
<td>'password': the user’s password</td>
</tr>
<tr>
<td><strong>Output Parameters</strong></td>
<td>Array('session_id' =&gt; 'xsd:string')</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Name</th>
<th>logoff</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td>User logoff</td>
</tr>
<tr>
<td></td>
<td>Return values: 1 for success or 0 otherwise</td>
</tr>
<tr>
<td><strong>Input Parameters</strong></td>
<td>Array('session_id' =&gt; 'xsd:string')</td>
</tr>
<tr>
<td></td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>'session_id': the current session ID</td>
</tr>
<tr>
<td><strong>Output Parameters</strong></td>
<td>Array('retval' =&gt; 'xsd:int')</td>
</tr>
</tbody>
</table>