

Accounting and Information Exchange Services on a Resource Management Portal

Diplomarbeit

der Philosophisch-naturwissenschaftlichen Fakultät

der Universität Bern

vorgelegt von

Calogero Butera

2004

Leiter der Arbeit:

Prof. Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Leiter der Arbeit:

Prof. Dr. Torsten Braun

Institut für Informatik und angewandte Mathematik
Rechnetze und Verteilte Systeme

Betreuer der Arbeit:

Marc-A. Steinemann

Institut für Informatik und angewandte Mathematik
Rechnetze und Verteilte Systeme

Zusammenfassung

Heutzutage stellen viele Schweizer Hochschulen Websites mit internetbasierten Kursen und entsprechenden elektronischen Hilfsmitteln zur Verfügung. Eine Website, die einen Ausgangspunkt oder Gateway zu anderen Ressourcen im Inter- oder Intranet liefert, nennt man Web Portal oder kurz Portal. Sie stellt den Anwendern typischerweise personalisierte Informationsquellen bereit und konzentriert sich auf ein spezifisches Publikum oder Themengebiet. Portale bieten Hilfsmittel wie Suchmaschinen, Verzeichnisse oder Diskussionsforen an. Sie kämpfen mit einigen Schwierigkeiten, da viele verschiedene Ressourcen zahlreichen Nutzern angeboten werden.

Im Jahre 2000 entstand in der Schweiz das Bundesprogramm *Swiss Virtual Campus* (SVC), welches das landesweite Angebot an Ressourcen erweiterte und durch Vernetzung eine bessere Zusammenarbeit auf Hochschulebene durchsetzte.

Angesichts einer grossen Zahl an Informationsquellen ist ein einfacher Zugang bzw. der Autorisierungsprozess für Nutzer wichtig, damit der Betriebsaufwand für die Anbieter verhältnismässig klein bleibt. Denn meistens stehen die Informationen nur einer bestimmten Auswahl von Nutzern zur Verfügung. Vor allem wenn es sich bei den Ressourcen um Online-Kurse handelt, ist es wünschenswert, dass die Studenten innerhalb des Portals interagieren und kommunizieren können. Damit lässt sich die Qualität der Informationsverarbeitung auf bemerkenswerte Art steigern.

Da mit den angebotenen Hilfsmitteln ein grosses Publikum angesprochen werden will, drängt sich auch der Wunsch nach einer statistisch-orientierten Kontrolle der Ressourcenzugriffe auf. Die Administratoren sollten Benutzeraktivitäten auf dem Portal beobachten und überwachen können. Die daraus ermittelten Informationen erlauben unter anderem eine Leistungsoptimierung und die Erarbeitung einer Verrechnungsstrategie. Besonders bei Portale mit hohen Wartungskosten ist es von Interesse diese finanzielle Aufwände zu decken, indem zum Beispiel einige Portalfunktionen kostenpflichtig gemacht werden.

Für einen vereinfachten Zugang für Informationsnutzer eignet sich eine sogenannte Authentifizierungs- und Autorisierungsinfrastruktur (AAI). Damit lassen sich Ressourcen zur Verfügung stellen, ohne den Zugriff auf einzelnen Nutzer anzupassen. Hier die wichtigsten Merkmale:

- Die Informationsnutzer, in diesem Falle die Studenten, sind jeweils bei der eigenen Institution bzw. Schule, die wir *Home Organization* nennen, registriert.
- Die Home Organization handelt mit den anderen Instituten aus, auf welche Ressourcen die eigenen Studenten Zugang erhalten.
- Die Ressourcenanbieter richten den Zugriff für Benutzergruppen mit gleichem Profil ein.

Um den Sachverhalt zu erläutern, geben wir hier ein Beispiel für ein Profil: „Geographiestudent im Hauptstudium an der Universität *A*“. Ein Ressourcenverwalter *B* kann den Studenten mit gleichem Profil den Zugriff auf die eigene Online-Datenbank *Geologie* gewähren. Wenn ein Student mittels Web-Browser auf diese Ressource zugreifen will, muss er sich nur bei der eigenen Home Organization – in diesem Falle die Universität – anmelden, welche dem Verwalter *B* das entsprechende Studentenprofil übergibt. Daraufhin erfolgt der Autorisierungsprozess.

Für den Einsatz einer AAI sind einige Anpassungen bei den Ressourcen notwendig. Zwischen der Authentifizierungs- und Autorisierungsinfrastruktur und der geschützten Ressource wird eine Komponente als Vermittler geschaltet.

Die Forschungsgruppe „Rechnernetze und Verteilte Systeme“ des Instituts für angewandte Mathematik und Informatik an der Universität Bern hat ein Ressourcenverwaltungsportal

entwickelt, welches eine Vermittlungsfunktion zwischen Portal und AAI übernimmt und andere Funktionalitäten zur Ressourcen- sowie Benutzerwaltung anbietet. Die Benutzer des Ressourcenverwaltungsportals können sich an die zur Verfügung gestellten Ressourcen anmelden. Die Ressourcenverwalter gestatten oder verweigern den Zugriff auf die Ressourcen.

Diese Diplomarbeit befasst sich mit der Integration von Hilfsmitteln in das Portal. Dadurch wollen wir a) die Kommunikation sowie die Interaktion zwischen den Portalbenutzern ermöglichen und b) durch Einbindung zusätzlicher Accounting-Funktionalitäten die Visualisierung und den Export der gesammelten Daten unterstützen.

Um die Interaktion zwischen Studenten zu ermöglichen, haben wir als Hilfsmittel ein Online-Dokumentenarchiv, ein Online-Forum und eine *Internet Relay Chat*-Anwendung (IRC) gewählt. Wir nennen diese Art von Software „Community Interaction Tools“. Für die Realisierung unserer Erweiterung haben wir uns für eine modulare Architektur entschieden. Die Möglichkeit weitere Software-Produkte zu integrieren sollte damit vereinfacht werden. Ausserdem hat jeder Ressourcenverwalter die Möglichkeit, seine eigenen Ressourcen seinen eigenen Anforderungen nach zu gestalten. Er kann also die Tools mit den Ressourcen auf beliebige Art verbinden.

Die Umsetzung unseres Konzepts ermöglicht dem Portal-Administrator das Community Interaction Tool für eine bestimmte Ressource nach Bedarf des Ressourcenadministrators zu installieren. Die Administratoren können das Community Interaction Tool vom Portal aus konfigurieren. Die Benutzer können dann auf diese Tools zugreifen. Wizards sorgen für eine benutzerfreundliche Installationsroutine und stehen auch für Konfigurationsschritte zur Verfügung.

Im zweiten Teil unserer Arbeit befassen wir uns mit der statistischen Überwachung der Zugriffe auf Ressourcen. Die Administratoren sollten Benutzeraktivitäten auf dem Portal beobachten und kontrollieren können. Die aus der Analyse der gesammelten Daten resultierenden Informationen ermöglichen unter anderem eine Optimierung der Systemleistung und die Erarbeitung einer Verrechnungsstrategie.

Die von uns vollständig integrierte Accounting-Komponente liefert Visualisierungsmöglichkeiten sowie Datenverwaltungs-Werkzeuge. Damit lassen sich Daten zu allen Aktivitäten sammeln, die im Ressourcenverwaltungsportal ausgeübt werden können. Diese Informationen können zu Auslagerungs- oder Visualisierungszwecken gesammelt werden. Jede erfasste Benutzeraktivität wird in der Portal-Datenbank gespeichert. Das Löschen solcher Daten darf nur der Portaladministrator vornehmen. Aktivitäten, welche Auskunft über Ressourcenzugriffe oder Community Interaction Tools geben, werden über ein Web-Interface visualisiert. Die Möglichkeit, solche Daten zu exportieren, bleibt ausschliesslich den Administratoren vorbehalten. Für Ressourcenadministratoren gilt dies nur für die eigens verwalteten Hilfsquellen und Anwender. Portaladministratoren hingegen stehen sämtliche Daten für die Auslagerung zur Verfügung. Damit wird die Privatsphäre der Benutzer geschützt. Diese Regelung gilt auch für die Visualisierung der Daten.

Unsere Arbeit befasst sich mit der Erweiterung eines Ressourcenverwaltungsportals. Die dadurch erzielte Verbesserungen der Kommunikation bekräftigen die Empfindung eines Treffpunkts für die Anwender. Besonders im Zusammenhang mit Online-Kursen können die Studenten von einem verbesserten Lerneffekt profitieren. Die Integration von Accounting-Funktionalitäten im Ressourcenverwaltungsportal bringt Vorteile für alle Parteien. Sie fördert die Überwachung des Portals sowie der Ressourcen und gewährt den Ressourcenadministratoren die Möglichkeit einer optimierten Gestaltung der Online-Kurse, was wiederum den Portalbenutzern zugute kommt.

Abstract

The implementation of an authentication and authorization infrastructure is a means of simplifying inter-organizational access to networked services. In order to profit from the authentication and authorization infrastructures, the resource providers must adapt their resources (*e.g.*, software applications, web sites, or networks). One approach which solves this issue is the creation of an entity which acts as a broker between the authentication and authorization infrastructure and the protected resource. The group “Rechnernetze und Verteilte Systeme” of the Institute of Computer Science at the University of Bern developed a generic resource management portal prototype which performs the mediation between any authentications and authorization infrastructure and any protected resource. In addition, the portal allows the management of the resources and provides user management capabilities.

The resource management portal prototype was implemented in an academic context, and therefore one of the potential applications is the adaptation of network resources with educational contents. In such applications the communication and interaction between users is fundamental because these activities increase the quality of educational resources enormously. The resource management portal is the ideal site to provide such services, since the students must use the portal in order to access the desired resource.

The resource management portal is designed to host a large user community. This requires a system which can provide valuable statistical information about resource visits. Administrators should be enabled to monitor, and survey the users on the portal. The result of the analysis of the collected data allows to optimize system performance, and to build a billing strategy.

The goal of the diploma thesis is to integrate the features enabling the users to communicate and interact with each other in the resource management portal, and to provide accounting capabilities collecting data about user activities on the portal, providing graphical data visualization features, and enabling data export.

The analysis of the resource management portal led us to design of an architecture based on modules which allow the installation, configuration, and access of web tools. These tools provide services to the resource management portal users, allowing them to exchange information (*i.e.*, communicate and interact with each other). The realized modules support an online file repository, a forum and an internet relay chat application (IRC). They permit to install, configure, and access the tools using the resource management portal for each hosted resource.

The evaluation of several accounting systems, led us to the design of an architecture for a proprietary accounting, fully integrated in the portal, and providing visualization and data management capabilities. The realized system allows collecting data for any performed action in the resource management portal, summarizing and visualizing the user access information, and exporting data.

Acknowledgements

First of all I would like to thank every body contributed to the success of my diploma. Especially I would like to thank Prof. Dr. Torsten Braun who he gave me the opportunity to carry out my diploma in his research group. I would like to express my gratitude to M. Alain Steinemann for coordinating my diploma work. There were many other people that supported me somehow during my diploma. The most important are all the components of my family: my father Carmelo Butera, my mother Maria Vancardo, my grandmother Concetta Butera and my sister M. Gabriella Butera. Then I would like to thank Thomas Spreng, Thomas Lenggenhager, Valery Tschopp, Stefan Zimmerli and Pino Zaccagnino (for their support and hints about technical and didactical aspects). A special thank goes to my friend Gianni Del Vecchio. He supported me and we had many interesting discussions about technical and conceptual aspects. Another friend which contributed to this work is Daniele Talerico. I want to thank him for the time he spend for me and for the time we spent together during our studies (without him the exams in mathematics would have been much more difficult).

During my studies I got the chance to know many people. We learned together, we suffered together during the preparation of our exams, and we had a nice time together trying to find the motivation to go on in order to achieve this and other important goals in our life. Those are Michele Lanza (he encouraged me all the time), David Vogel (always ready to motivate his friends), Stefan Martig (if one needs a hint he is always the best address), Georges Golomingi (he was really patient).

I want also to express my gratitude to my friends Arnaldo Mettimano, Dario Salituro, Stefano Russo, Maurizio Polselli, Paulina Molska, Michal Molski and his family, Caterina Zarro, Miriam Mannino, Luca D'Alessandro, Paola Salituro, Enrico Balistreri and his family, Matthias Wisler, Silvana Amato, Claudia Di Palma, Laura Ritter, M. Lorenzo and Carlo Di Loreto. They were near me when was stressed, when I wanted to give up and when I needed to speak about me and my studies. Their contribution seems to be minimal, but they helped me very much even without being aware of it. I would like to dedicate this work to all these people.

Contents

1. Overview	1
1.1 Introduction-----	1
1.2 Structure of the Document-----	2
2. Related Work	5
2.1 Introduction-----	5
2.2 Authentication and Authorization Infrastructures -----	5
2.3 Shibboleth-----	6
2.4 Adapting Resources to Authentication and Authorization Infrastructures -----	8
2.5 Resource Management Portal -----	9
2.6 Resource Management Portal's Architecture -----	9
2.7 Enhancements of the Resource Management Portal -----	14
2.8 Communities -----	14
2.9 Community Interaction in the Resource Management Portal -----	15
2.10 Community Interaction Tools -----	16
2.10.1 Synchronous Tools.....	16
2.10.2 Asynchronous Tools	17
2.11 Resource Management Portal in an Educational Context -----	17
2.12 Accounting-----	18
2.13 Accounting Systems -----	18
2.13.1 Accounting Concepts and Criteria	19
2.13.2 AAA Accounting	19
2.13.3 Apache HTTP Server's logging of client requests	19
2.13.4 Proprietary Accounting	20
3. Architecture	21
3.1 Community Interaction Tools -----	21
3.1.1 Architectural aspects	22
3.1.2 Instance Installation Utility	23
3.1.3 Instance Configuration Utility	23
3.1.4 Other Utilities.....	24
3.1.5 Tasks and Responsibilities	25
3.1.6 Proposed Architecture.....	25
3.2 Accounting-----	26
3.2.1 Architectural Aspects	27
3.2.2 Criteria for Data Gathering	27
3.2.3 Accounting Scheme	29

3.2.4	Access Limitation	29
3.2.5	Visualization	29
3.2.6	Other Tools	29
4.	Implementation	31
4.1	Community Interaction Tools	31
4.1.1	Integrated Community Interaction Tool Modules	31
4.1.2	Development	32
4.1.3	Database Modifications.....	33
4.1.4	Wizards	34
4.1.5	Community Interaction Tool Module Libraries.....	34
4.1.6	Implemented Utilities.....	36
4.2	Accounting Support	39
4.2.1	Developed Features.....	39
4.2.2	Database Modifications.....	39
4.2.3	The Accounting Library.....	40
5.	Configuration and Usage	45
5.1	Community Interaction Tools Configuration.....	45
5.1.1	General CIT Constants.....	45
5.1.2	Tool Specific Constants	45
5.2	Accounting Configuration	46
5.3	Overview of the Graphical User Interface.....	47
5.3.1	The Portal Administrator's Menu	48
5.3.2	The Resource Administrator's Menu	48
5.3.3	The User's Menu.....	49
5.4	Installation of a Community Interaction Tool.....	49
5.5	Configuration of a Community Interaction Tool	50
5.5.1	Single User Account Configuration.....	51
5.5.2	Multiple User Account Configuration	52
5.6	Access to a Community Interaction Tool.....	54
5.7	Uninstall of a Community Interaction Tool	55
5.8	Visualization of Accounting Data for the User	56
5.8.1	The Single Overview	56
5.9	Visualize Accounting Data for the Administrators	57
5.9.1	The Users' Overview	57
5.9.2	The Resources' Overview	58
5.9.3	The Community Interaction Tools' Overview.....	59
5.10	Accounting Data Management.....	60
5.10.1	Removal of Accounting Information	61
5.10.2	Accounting Data Export.....	61
6.	Discussion and Conclusion	63

6.1	Discussion	63
6.2	Conclusion	64
7.	Outlook	67
	Appendix	a
	List of Figures	c
	List of Tables	e
	References	g

1. Overview

1.1 Introduction

The implementation of an authentication and authorization infrastructure is a means of simplifying inter-organizational access to networked services. The core functionality of an authentication and authorization infrastructure is the tight coupling of the three basic interactions between a *user*, his or her home organization, and a resource during the authentication and authorization process. These interactions are user authentication, access request, and delivery of authorization attributes from the home organization to the resource [1].

In Switzerland exists a federal higher educational system with many universities, each of which has an own student authentication system. Initiated by the Swiss Virtual Campus [2], and by the demand for more inter-university teamwork and student mobility, the Swiss education and research network SWITCH [39] evaluated different authentication and authorization architectures [3], such as PAPI [4] or Shibboleth [5]. During this evaluation phase the question has raised about how to connect a protected resource to the evaluated authentication and authorization infrastructure. *A resource is an application, a web site, or a system that provides services to the users.* Besides the direct adaptation to the authentication and authorization infrastructure, other approaches were evaluated. One of the approaches was the development of an authentication and authorization infrastructure proxy server which is located between the protected resource and of the authentication and authorization infrastructure. Another approach consisted in the creation of a generic resource management portal. This portal can provide interfaces to any authentication and authorization infrastructure on one side, and to any kind of protected resources on the other side. The resource management portal was developed at the group “Rechnernetze und Verteilte Systeme” of the “Institute of Computer Science - University of Bern”

In addition to maintaining interfaces to authentication and authorization infrastructures for the resource users, the resource management portal provides the possibility for provisioning local user accounts. This allows accessing the resources without having the obligation of using the authentication and authorization infrastructure, if the access is granted by the portal administrators. The portal is deployed in a university environment; so it is often the case that the resources consist of web sites with educational purpose.

Many online resources do not provide any tools which enable the interaction between users. The integration of community interaction tools in the resource management portal, would offer the opportunity to provide user interactivity for such resources. The aim is to increase the communication and the didactical quality of the resource. Examples of community interaction tools are: online chat applications, bulletin boards, and online calendars.

The resource management portal is expected to have a large number of users. Therefore there is a need for a system which collects data about users’ activities in the portal. This allows surveying the resource management portal users. Furthermore, such a system allows making prognoses on the users’ behavior, eventually to optimize the system performance, or to create a billing strategy based on the collected data.

This diploma thesis focuses on the analysis, the evaluation, and the realization of a system which provides the opportunity to install, configure, and access community interaction tools within the

resource management portal. Furthermore, during this diploma, we want to analyze, evaluate, and realize a system which allows one to provide an accounting feature on the portal.

The analysis of the *community interaction tools integration* in the portal for resource management brought us to the following implementation criteria:

- The community interaction tool support is integrated in the portal to permit a modular installation of the tools.
- The resource administrators on the resource management portal are able to decide case-by-case, if a specific community interaction tool for their resources has to be installed. To allow this behavior we decided to apply the concept of single instance based installation: each resource has exactly one instance per community interaction tool; multiple resources cannot share a community interaction tool instance (one-to-one relationship).
- The resource administrators must be able to configure automatically the resource user accounts, and specific settings of the community interaction tool.
- The access for the resource subscribers to the available tools must be simple and cannot require any specific credential for the user beside the portal authentication. In fact, a manual login would stay in contradiction to the concept of Single-Sign-On which is one of the advantages for the authentication and authorization infrastructure users. *Using a Single-Sign-On technology, the user needs to provide his credentials only once in order to access more than one resource.*

The result is the implementation of three modules allowing ad hoc installation and configuration of open source community interaction tools, with an underlying framework developed in PHP [14] supporting specific database operations and file system functionalities. In particular, those modules integrate the resource management portal the “PHPBB Bulletin Board” [31], the “OWL, Document Repository” [30], and the “Aerial Chat Application” [32].

The analysis for the accounting support requirement in the resource management portal led us to the development of a proprietary accounting system fully integrated in the resource management portal. The criteria applied are the following:

- The accounting data to be gathered must be consistent, reliable and understandable.
- A graphical visualization on the portal must be provided, and has to preserve data confidentiality and privacy.
- The creation of data management utilities such as data export tools and data removal must be supported.

The resulting implementation is an accounting system which collects data for almost every user action on the resource management portal. As the selected data structure is very general, it allows further extensions of the resource management portal to be performed in a well defined and calculable way. The visualization features are fully integrated in the portal in the form of PHP pages supporting a navigation mechanism that allows browsing through the accounting information.

1.2 Structure of the Document

- In *Chapter 2*, after defining the concepts of authentication and authorization infrastructure, we present Shibboleth, the implementation of an authentication and authorization infrastructure which has become the authentication and authorization infrastructure used by most of the universities in Switzerland. At this point we discuss the

issue of connecting the resources to an authentication and authorization infrastructure, and we present the generic solution which is the resource management portal (the mediator between authentication and authorization infrastructure and resource). The architecture of the resource management portal and its needs for the enhancements done in this diploma thesis, are then introduced in order to discuss the concepts of community, and community interaction, the motivation for the needs for community interaction tools, the evaluation, and the analysis of their integration in the resource management portal. The second enhancement done in the resource management portal is providing an accounting support for the portal. The concepts of accounting, and the evaluation of the integration options is discussed at the end of this Chapter.

- *Chapter 3* presents all the architectural aspects, and the proposed architecture for the community interaction tools integration, and the accounting system implementation.
- *Chapter 4* presents the elaborated implementation of the community interaction tools integration, and the accounting system. Here we describe the implementation aspects of each developed component.
- The configuration and the usage of both implemented features are described in *Chapter 5*. The use of screen shots allows a better comprehension of the developed software in this diploma.
- *Chapter 6*, and *Chapter 7* discuss accurately the work done, evaluate the results, and propose future work in order to increase the resource management portal quality.

2. Related Work

2.1 Introduction

This Chapter describes all the work related to this diploma thesis. We first explain the authentication and authorization infrastructures, and present the resource management portal, also known as portal for resource integration to authentication and authorization infrastructures. Finally we discuss the concepts of community interaction, and present and the concepts of accounting.

The term authentication can be defined as the process of identifying an individual, usually based on a user name and password. The term authorization can be defined as the process of granting or denying access rights for a resource to an authenticated user. The authentication and authorization infrastructure consists of a software architecture with the goal to simplify inter-organizational access to networked services.

2.2 Authentication and Authorization Infrastructures

Authentication and authorization infrastructures are middleware systems, consisting of a set of protocols that enable the delegation of authentication and authorization issues to different instances. The infrastructures provide all the necessary mechanisms that let users, organizations, and resources collaborate in the system [3]. An authentication and authorization infrastructure requires three entities in order to provide its functionality (the home organization, the resource, and the user). Home Organizations are universities or other entities, where the potential users (*i.e.*, students, or staff members) are registered. The home organization is responsible for the authentication of their respective users. The users possess at least one account at a home organization. Resources exist in two forms: either they are web-based, or they act as services for authentication and authorization infrastructure users. If a user wants to access a resource, the authentication and authorization component of the resource asks the user's home organization for the authentication. Upon successful authentication, the home organization sends back a set of attributes containing the information about the user's home organization role. It is a precondition in authentication and authorization infrastructures that user information released by home organizations is reliable and accepted by the resource providers. Based on these attributes, the resource authorization system decides if the access should be granted or denied.

Authentication and authorization infrastructures bring advantages for all the involved entities in the process of accessing protected resources. The home organizations can increase the quantity of accessible resources by their members with a minimal overhead. The users take advantage of a simplified resource access procedure. The advantage for the resources consists in a decreased administration overhead (they do not need to create accounts for their users), and the reliable user information, since the attributes sent by the home organization are trustful.

An authorization process is based on the exchange of user authorization attributes in the form of name/value pairs. A policy which allows controlling this exchange of attributes is desirable. Some authentication and authorization infrastructures provide such a functionality which enables the users to make a trade-off between access and privacy. This mechanism is enforced by the Attribute Release Policy (ARP). In order to authorize users, resources need a policy which

defines a minimal amount of attributes they must receive. This policy is called Attribute Acceptance Policy.

Based on users attribute release policy and resources' attribute acceptance policy, users are able to access resources. Only users providing the required attributes are granted access to the resource.

2.3 Shibboleth

There are many implementations of authentication and authorization infrastructures; however Shibboleth [5] has become the inter-institutional authentication and authorization infrastructure for many universities in Switzerland. It bases on one of the most promising authentication and authorization architectures under development. It originates from a project of *Internet2/MACE* (Middleware Architecture Committee for Education). It aims at the development of an architecture for standard-based, and vendor-independent web access control infrastructures that can operate across institutional boundaries.

Shibboleth is able to interface with a directory exporting Lightweight Directory Access Protocol (LDAP) formatted attributes, and is designed such that programming interfaces to other repositories should be readily implemented. *LDAP consists of a set of protocols for accessing information directories*. This feature makes it easily adaptable to universities which maintain user directories.

Shibboleth features federated user administration – *i.e.* user management is delegated to home organizations –, resource access authorization based on user attributes, and active privacy management. *Attributes are generally name/value pairs (e.g., "Role=Life Sciences Student")*. *Active privacy management is a mechanism that allows users to determine their own attribute release policy*. It makes use of the Security Assertion Markup Language (SAML) [7], for the message and assertion formats, and of protocol bindings developed by the OASIS Security Services Technical Committee [6]. In Switzerland, SWITCH coordinates the Shibboleth deployment for the universities.

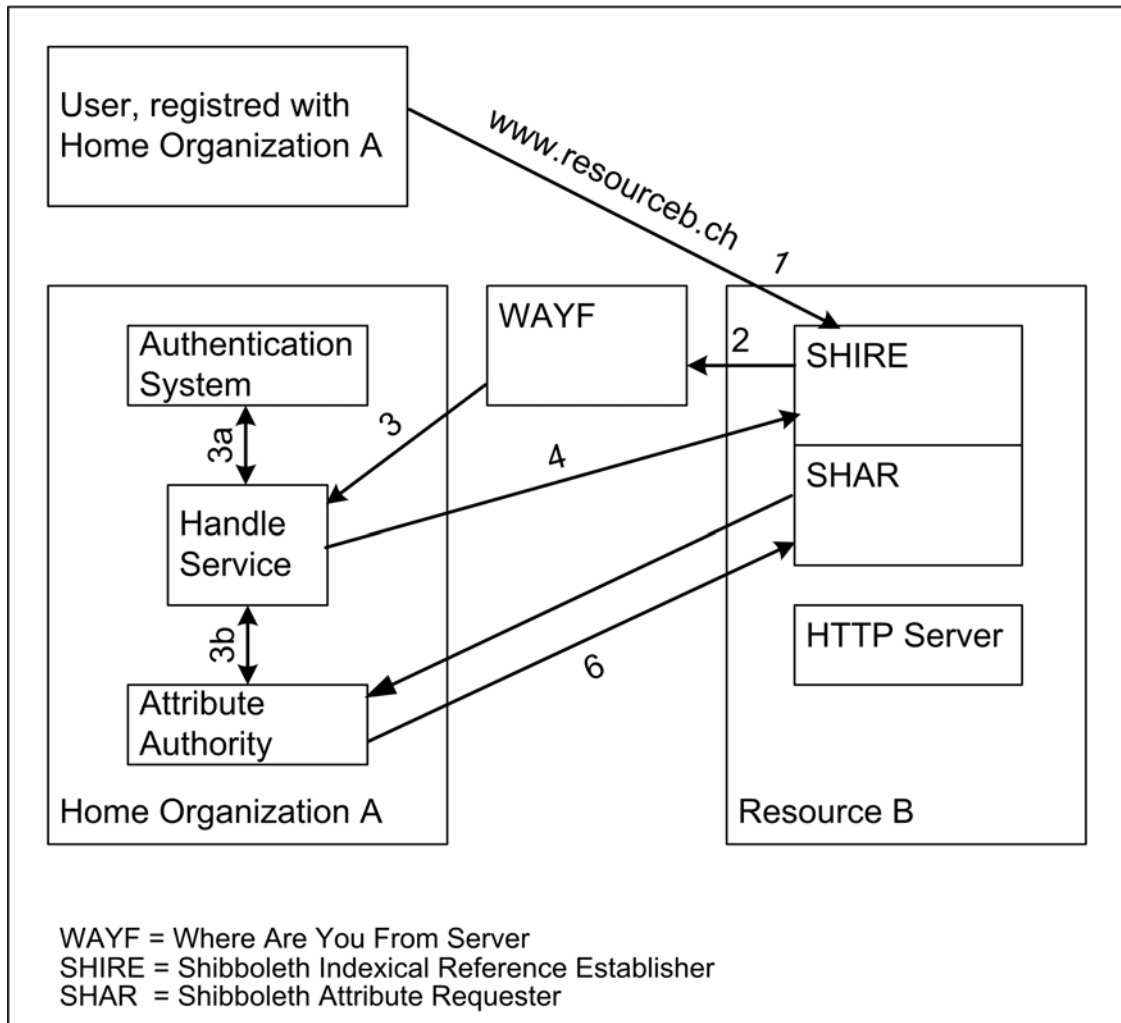


Figure 1: Shibboleth Architecture

Figure 1 depicts the user access process to a Shibboleth enabled resource. The Home organization incorporates the Authentication System, the Attribute Authority (AA), and the Handle Service (HS). The Authentication System is responsible for the verification of the user's identity. The attribute authority provides the means for users and administrators to specify attribute release policies, while the handle service provides the resource with a handle to be used for making attribute requests to the AA. The resource site includes two components. The Shibboleth Attribute Requester (SHAR) acquires attributes about a browser user from the user's attribute authority, while the Shibboleth Indexical Reference Establisher (SHIRE) is the component responsible for intercepting an HTTP request for a protected resource, and associating it with a handle suitable for attribute requests by the SHAR.

The steps of accessing a resource are described as follows:

- 1) A user accesses resource B, and gets to the Shibboleth Indexical Reference Establisher (SHIRE).

- 2)* The Shibboleth indexical reference establisher directs the user to the “Where Are You From” service (WAYF), a service located somewhere in the Internet that possesses full information about connected home organizations together with the handle acceptance URL, the URL where Shibboleth indexical reference establisher expects the answer from the Handle Service (HS), and the user’s desired target URL as parameters.
- 3) The WAYF service redirects the user to its home organization’s handle service together with the SHIRE parameters. The handle service makes sure that the user is authenticated.
- 4) The handle service sends back an opaque (not transparent) handle associated with the user to the Shibboleth indexical reference establisher's "handle acceptance URL" by means of an HTML form posting. The user's desired target URL is passed as an additional parameter. Impersonation counter information is presented as part of a "package" around the handle.
- 5) The Shibboleth Attribute Requestor (SHAR) sends an Attribute Query Message (AQM) to the Attribute Authority (AA).
- 6)* The Shibboleth Attribute Requestor receives an Attribute Response Message (ARM). Upon evaluation of the attribute response message the resource manager grants access to the protected resource.

2.4 Adapting Resources to Authentication and Authorization Infrastructures

There exist two groups of web resources.

Authentication-and-Authorization-Infrastructure-Enabled Resources

Authentication-and-authorization-infrastructure-enabled resources have specific interfaces to the authentication and authorization infrastructure, and are connected directly to the respective infrastructure. Enabling resources for authentication and authorization infrastructures is expensive and time consuming. Since specific authentication and authorization infrastructure implementations differ, resource interfaces must be re-implemented for each kind of authentication and authorization infrastructure and resource.

Non- Authentication-and-Authorization-Infrastructure-Enabled Resources

Non-authentication and authorization infrastructure-enabled resources cannot be directly connected to the authentication and authorization infrastructure as they lack the specific interfaces to the authentication and authorization infrastructure.

The problem of connecting resources to an authentication and authorization infrastructure concerns many *e-learning* projects such as the Virtual Internet and Telecommunications Laboratory of Switzerland (VITELS) [9] [10] and the Nano science laboratory Nano-world [11]. In some cases, where there is a high pressure from the user side and a resulting business case, resources get adapted to one or more authentication and authorization infrastructures. For example WebCT [12] was adapted to Shibboleth [5]. The question how to connect non-enabled web resources to an authorization and authentication infrastructure in a simple manner without bearing high costs seems to be justified.

*HTTP browser redirect

One approach to solve this problem is the implementation of a generic portal hosting both kinds of resources which connects them in a simple way to authentication and authorization infrastructures.

2.5 Resource Management Portal

The resource management portal is located between an authentication and authorization infrastructure, and a protected resource. Users belonging to an authentication and authorization infrastructure connected home organization access resources hosted by the portal. We see a resource management portal resource owner's site in Figure 2.

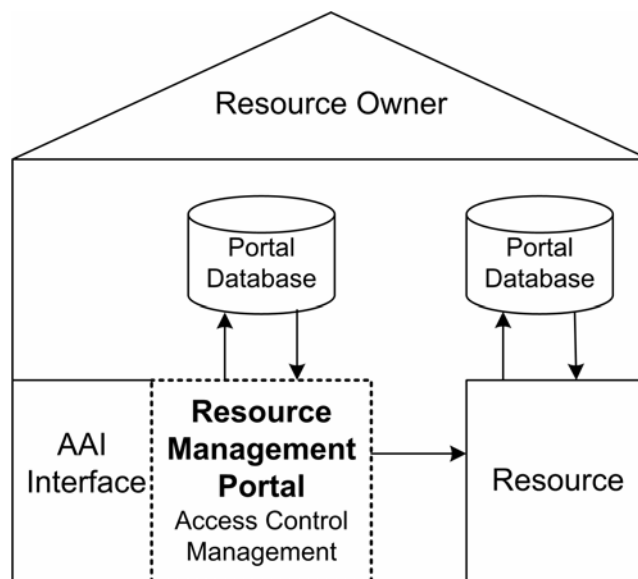


Figure 2: The resource management portal at the resource owner's site

The authentication and authorization infrastructure interface is responsible for the resource owner's site connection to the authorization and authentication infrastructure. The resource management portal receives user information attributes from the authentication and authorization infrastructure interface, and stores them in its database. Users that successfully subscribe to a resource management portal-hosted resource agree automatically with the transfer of selected user attributes to the resource. If required, the resource stores user data in its own database, or relays on the resource management portal's database. The resource may send back user information to the resource management portal. Besides the access management, the resource management portal provides enhanced user management. This allows resources to directly ask users for additional information, and enables users to manage their self-provisioned data.

2.6 Resource Management Portal's Architecture

In order to understand the resource and the user management features of the resource management portal, it is fundamental to introduce how users are classified inside the portal.

User Roles

The resource management portal provides three different user roles.

- The *super user* is the portal administrator. He is allowed to configure every feature of the resource management portal.
- The second user role is the *resource administrator* who acts as owner of resources hosted on the resource management portal.
- The *resource user* – sometimes simply called ‘user’ – has the third role. In this role one may access resources hosted on the resource management portal.

When accessing the resource management portal, a set of reliable user attributes is provided by the authentication and authorization infrastructure. In rare cases, users access the resource management portal through the built-in direct access interface. In such a case, their user rights have been manually defined by a portal administrator – *e.g.*, during the creation of the respective account.

The resource management portal’s resource management and user management enable the users to perform a set of activities depending on their user rights. Table 1 illustrates these activities related to the resource management on the left side, and user management on the right side.

Resource Management	User Management
<i>Create Resources</i>	<i>Grant User Access</i>
<i>Set Resources Visibility</i>	<i>Deny User Access</i>
<i>Open Resources for Subscription</i>	<i>Send Messages</i>
<i>Close Resources for Subscription</i>	<i>Add Information to User Profile</i>
<i>Subscribe Resources</i>	
<i>Suspend Resources</i>	
<i>Set Attribute Release Policy</i>	
<i>Delete Resources</i>	

Table 1: The resource management and user management activities on the portal

As showed in Figure 3, the resource management portal is divided into three distinct areas. Depending on the role, a user is allowed or not to access a determined area and perform some activities.

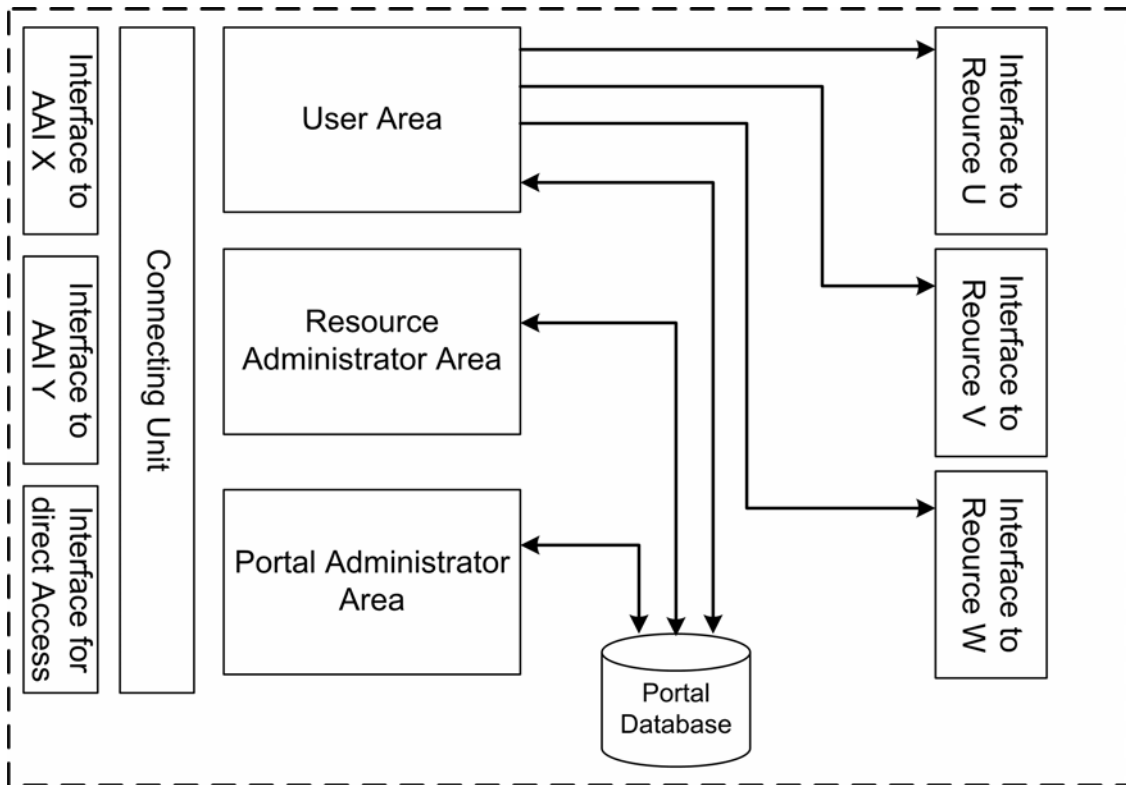


Figure 3: The functional parts of the resource management portal

The Interfaces

The resource management portal can be split up in its functional parts, as shown in Figure 3. On the left side, we see the interfaces of the authorization and authentication infrastructure for directly connecting users. The *connecting unit* links the interfaces to three user areas. On the right side, the interfaces for a set of hosted resources are depicted. There is no limitation to the number of hosted resources. Each of the interfaces, either for an authorization and authentication infrastructure or for a resource, has to be designed individually but can be reused whenever the same type of infrastructure or resource is connected to the resource management portal. The attribute acceptance policy of the authentication and authorization infrastructure interfaces is set to request all the user's attributes that the home organization can deliver. The resource management portal's database stores the collected data.

Technical Description

The resource management portal is a web application with a database backend. It supports several languages, using the I18N internationalization scheme [19]. The web front-end structure is subdivided into three parts: the *header area*, containing information about the user and the portal – such as user name, and name of the portal. The *menu area* allows classifying, and selecting the possible action given by the portal. The third part is the *content area* that displays the information selected in the menu area.

This classification enables the adaptation of the web design to the corporate design of any entity which wants to deploy the resource management portal. Figure 4 illustrates how the front-end is structured.

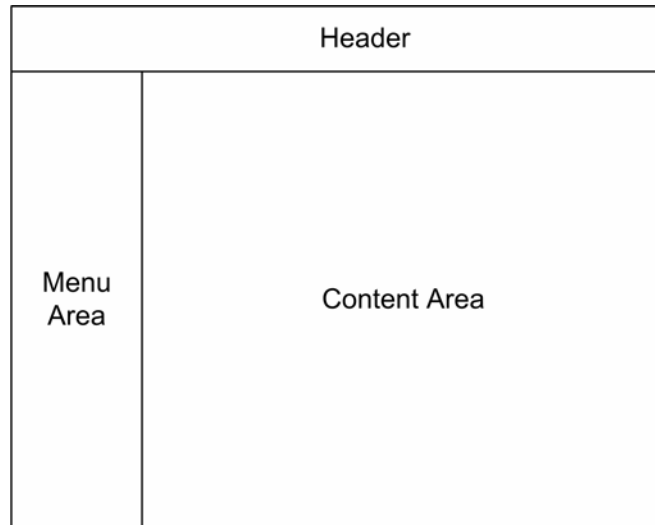


Figure 4: Web front-end structure

Database Design

The authentication and authorization portal's database contains eight tables. Table 2 lists the database table names with a short description.

1.	User	The user table holds the unique user id (<i>User_uid</i>) which must be provided by the authorization and authentication infrastructure. Each entry must have a privilege level from the <i>UserPrivilege</i> table.
2.	UserPrivilege	The <i>UserPrivilege</i> table has three standard entries, a normal user privilege, resource administrator privilege and a portal administration privilege. It is being used to distinguish normal users from administrators.
3.	Attribute	This table holds the attribute value for one specific attribute. Each user can have any number of attributes. Each attribute must have exactly one <i>AttributeType</i>
4.	AttributeType	The attribute value types are stored in this table and it is initialized with the standard authentication and authorization infrastructure attributes but it can also be extended by custom, string-encoded attribute types.
5.	Resource	The resource table stores the properties for each resource. Each resource entry must have exactly one user with at least resource administrator privilege.
6.	Policy	This table defines what <i>AttributeType values</i> are needed for each resource; technically it is a relation between the resource and the <i>AttributeType</i> table.
7.	ResourceAdapterParam	This table holds configuration parameters for resource adaptors.
8.	Subscription	The subscription table connects a user table entry to a resource. Like the policy table it is a relation between two other tables, user and resource. It holds some additionally data like access control information and so on.

Table 2: The authentication and authorization portal database tables

We describe the system just by analyzing the eight table names, listed in Table 2. Users (1) have privileges on the portal (2), and provide a set of attributes for resources (3). The attributes are classified by attribute type (4). On the other hand, the resource (5) requires a connection interface for the real resource. This interface contains a set of properties, *i.e.*, configuration parameters (7). Conforming to the own attribute acceptance policy, the resource must define which user information is required (6). In order to access a resource, the users need first to subscribe to it (8).

Implementation Technologies

The resource management portal is a dynamic open source web application with a database backend, licensed under “GNU General Public License” [13]. The implementation is realized with PHP 4.3.2 [14] including four modules: `<mhash>`, `<mysql>`, `<openssl>`, and `<session>`.

At the time of writing this document, the resource management portal is deployed under Apache 1.3.28 web server [15] and runs on a Linux (Redhat [17] or Debian [18]) operating system environment. The database backend is realized with the MySQL 3.23.x database engine [16]. The resource management portal is a typical LAMP application (Linux Apache MySQL PHP). In consideration of the technologies, the same application on the Microsoft Windows platform is conceivable.

2.7 Enhancements of the Resource Management Portal

The resource management portal was designed to be a mediator between the authorization and authentication infrastructure and protected resources. The target resources were web services with educational purpose. Since the students would use the resource management portal as a transit place to reach their educational resources the idea of extending the portal to a meeting point for its users raised. In this context, the term “Community Interaction Tool” (CIT) stands for the set of all web tools that provide services to the resource management portal users. Examples are: web calendar, discussion board, video/audio conferencing tools, or a simple list of hyperlinks. The integration of CIT support would give to the resource providers (*i.e.*, the tutor) the chance to use the resource management portal as a means to increase his course’s quality.

The collection of user data about their behavior on the resource management portal is interesting for the resource administrators and the portal administrator. Even the portal users can take advantage of this information. Therefore, the supply of valuable statistical information about the resource management portal users is required. There can be different motivations for the support of this feature. They start with monitoring the activities on the portal, and end with the creation of a billing model.

2.8 Communities

The term *community* is used in many contexts, and has therefore different meanings. Most definitions contain terms like “group of people”, “common”, and “sharing interests”. Based on this; we give here an informal definition:

A community is a group of persons that share interests, and interact in a common environment.

In the following, we add three definitions in order to give a short overview of commonly used terms. They offer a more formal approach and understanding for the next sections in this chapter.

Virtual Community

A virtual community is a group of people sharing common interests, ideas, and feelings over the Internet, or other collaborative networks. A possible inventor of this term and one of its first proponents is Howard Rheingold. He is the creator of one of the first major Internet communities, called "The Well". In his book, *The Virtual Community* [20], Rheingold defines virtual communities as social aggregations that emerge from the internet when people carry on public discussions. If long enough maintained and with sufficient human feeling, they form networks of personal relationships in cyberspace.

Learning Community

Rasmussen and Skinner [21] propose a very broad definition:

A learning community is a curriculum design which coordinates two or more courses into a single program of instruction.

They argue that the strength of learning communities lies in the integrated approach to education. Integrated educational experiences more closely parallel the way people learn and are more relevant to real world events. Students have the opportunity to see topics from multiple, sometimes even conflicting, perspectives, allowing more critical thinking [21].

Technological Learning Community

A technological learning community is a learning community where the members (e.g., students) are connected by technological means (e.g., computer networks); the educational content is offered online [22].

2.9 Community Interaction in the Resource Management Portal

The three circles in Figure 5 represent hosted resources in a resource management portal, while the people in the circles correspond to the resource subscribers. Since users can subscribe to more than one resource, they may be associated to more than one community. Therefore, resource management portal is a virtual location where people – basically the portal users and the resource administrators – form different communities. This circumstance encouraged us to think about an extension respectively an enhancement of the resource management portal with community interaction support. We outlined the following requirements:

- Community interaction tools should first of all give the chance to create a real community.
- We want different members of the communities to be able to interact and communicate with each other.
- The resource administrator assumes the tasks of a community manager. He disposes of means and decides on participation rights for users.

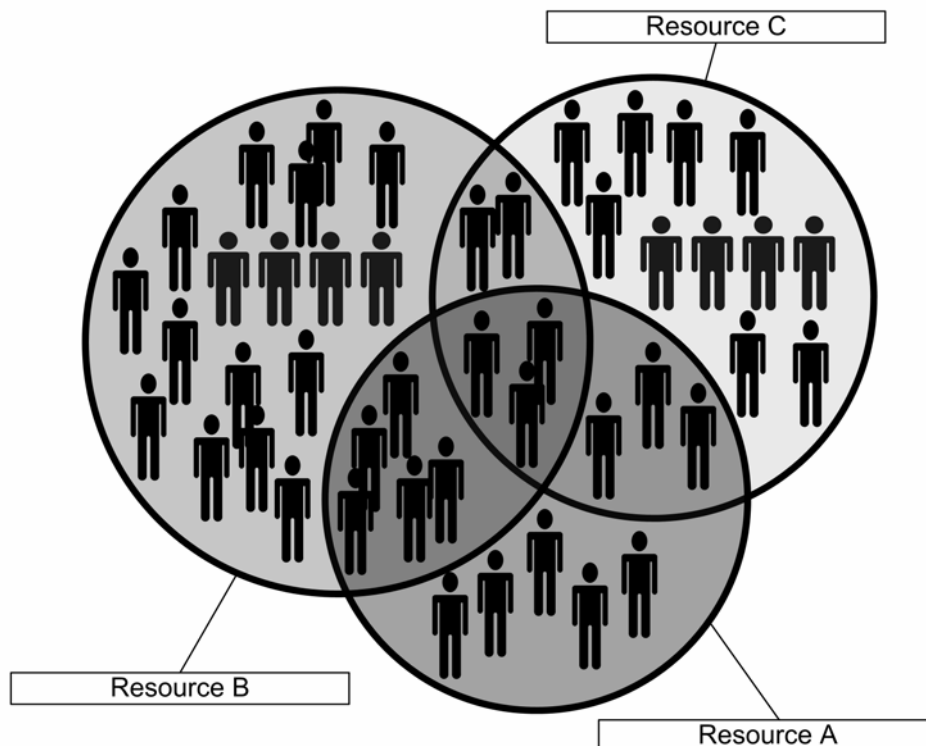


Figure 5: Resources and Communities

2.10 Community Interaction Tools

Community interaction tools (CIT) are web-based software components which allow a community to interact and communicate. We make a distinction between two different types: synchronous and asynchronous CITs.

Figure 6 illustrates the difference between both kinds of tools. Synchronous tools are used for direct communication such as instant messaging. They provide a meeting point for a direct interaction of community members. Asynchronous tools provide time shifted communication, as we know from discussion boards. Thus, we can say that they act as containers of exchanged information.

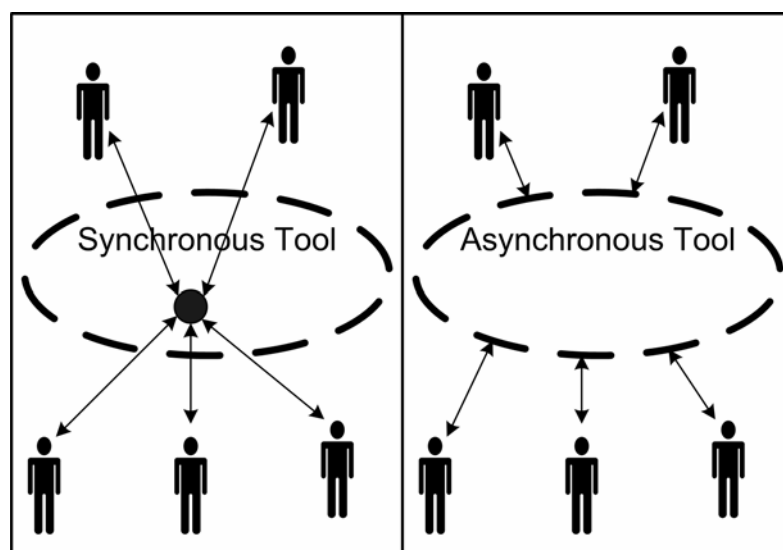


Figure 6: Synchronous tools vs. asynchronous tools

2.10.1 Synchronous Tools

Synchronous tools give tutors and students the possibility to directly interact with each other. In the following, we list commonly used synchronous tools:

- **Video/Audio Conferencing** allows close to reality communication between tutors and course subscribers. Conferencing sessions must be planned in advance and announced by other services. Big advantages are that discussions can be close to classroom reality and that students can communicate with each other.
- **Chat.** Tutors have the possibility to create internet relay chat rooms and act as moderators. Students have the opportunity to visit the respective rooms of their course. Non-moderated chat rooms can be created by students but only students subscribed to the same course are able to join.
- **Instant Messaging.** Students see the online availability of their course colleagues and are able to send messages that are instantly delivered. Tutors and students may see who is online without connecting to a service such as internet relay chat (IRC).

- **White Board.** This tool is normally in collaboration with chatting or other conferencing tools. It allows tutors to teach and draw on a board, at the same time. All students of the course can assist simultaneously. Students can also use the white board tool for small student groups.

2.10.2 Asynchronous Tools

Asynchronous tools give tutors and course subscribers the opportunity to communicate in a time shifted manner. Thus, they can carefully prepare their statements since they do not need to be simultaneously online. The following list presents typical applications:

- **Discussion Board.** Tutors may create a subsection of a forum with the name of their resource, and moderate the correspondent contributions. On the other hand, students get automatically access to the forum related to the course they have subscribed.
- **Calendar.** Each course tutor can add a calendar that announces conferences, deadlines, and other important milestones.
- **Pointers to Websites.** For every course a tutor can add a page with web links to related resources, such as other web pages, e-books, or any auxiliary means for download.
- **Group Announcements.** The tutor has the possibility to divide large student classes into groups. Group announcements affect information applications, such as e-mail, and short message service.
- **FAQ.** In most courses, students often encounter similar problems. In this case, tutors may publish frequently asked questions and answers.
- **Meeting Tool.** Group works are typical for courses. The meeting tool supports students in meeting each other. If desired, this can also be done anonymously.
- **SMS (Short Message Service).** This tool offers similar functionality as the e-mail tool. The difference lies in the availability of mobile phone users. Nowadays, they are easier to reach, and the communication performance is good.

2.11 Resource Management Portal in an Educational Context

The resource management portal presents some analogies to a university in real life. Table 3 lists these similarities between the portal and the educational institution. Of course, universities comprise more than classrooms, students, and administrations. Usually, we also find pin boards, rooms where people meet each other, like a cafeteria, and a library. These are all factors which support learning communities.

Resource Management Portal	University
Portal administrator	Vice chancellor
Resource administrators	Professors and assistants
User	Student
Host interface to resources	Classrooms
Maintain interface to online courses	Host classrooms
User management	Student administration
Resource management	Course administration
Subscription	Admission

Table 3: Resource management portal and a university

The integration of community interaction tools in the resource management portal is essential to create the technological counterpart of a conventional student community found at the universities.

2.12 Accounting

The creation of an accounting system is one of the required features in the resource management portal. *In information technology, accounting is the process of keeping track of a user's activity while accessing a network's resources. Accounting data is used for trend analysis, capacity planning, billing, auditing and cost allocation* [27]. The deployment of an accounting system in the resource management portal supports the *portal administrator* as well as the *resource administrator*. The first takes on the monitoring of the resource management portal, and surveys possible user misuses. The latter collects data about the user activities on his resources, (online courses). The gathered information helps to study the resource users' behavior and acquire notions about how to increase the courses' quality.

The application of accounting systems provides more benefits. The analysis of the user trends provides an indication of which components are more interesting to integrate in the portal, and supports the forecasting related to the system. Based on the accounting data, the organization that hosts the resource management portal together with the resource owners may develop a billing strategy.

2.13 Accounting Systems

There exist different ways for implementing accounting systems. Starting from the application of standard protocols, and ending with the creation of proprietary accounting solutions which are based on the customer's requests for specific accounting. In what follows, we first discuss concepts and criteria which must be supported by an accounting system, and then we present several accounting approaches.

2.13.1 Accounting Concepts and Criteria

Three important concepts underpin the preparation of any accounting method:

- **Going Concern:** accounting methods assume, unless it is explicitly stated, that a correct user's authentication happened first. This has important implications for the valuation of assets and liabilities.
- **Consistency:** accounting records and valuation methods are treated the same way from period to period. This allows making more meaningful comparisons of data over the time.
- **Prudence:** accounting records are not recognized as valid if they are not complete or broken.

Before it can be regarded as useful in satisfying the needs of an administrator, accounting information should satisfy the following key criteria.

- **Understandability:** this implies the expression, of accounting information in such a way that it will be understandable to administrators - who are generally assumed to have knowledge of the system.
- **Relevance:** to be useful, accounting information must assist an administrator to form, confirm, or maybe revise a view - usually in the context of making a decision (*e.g.*, should I block this user? Should I bill this user? Should I report a failure?)
- **Consistency:** this implies consistent treatment of similar items and application of accounting policies.
- **Comparability:** it comprises the ability for administrators to compare similar user activities, and to make performance comparisons over time.
- **Reliability:** this implies that the accounting information that is presented is trustful, accurate, complete (nothing significant missed out), and may be verified (*e.g.*, by a user or administrator).

2.13.2 AAA Accounting

An interesting accounting mechanism consists in the application of certain well known authentication, authorization and accounting protocols (AAA) [23]. Examples for AAA protocols are the Terminal Access Controller Access Control System (TACACS+) [28], or the Remote Authentication Dial-In User Service (RADIUS) [26].

This accounting allows tracking the services users are accessing as well as the amount of network resources they are consuming. When AAA accounting is enabled, the network access server reports user activity to the TACACS+ or RADIUS security server using accounting records. Each accounting record stored on the security server contains accounting attribute-value (AV) pairs.

2.13.3 Apache HTTP Server's logging of client requests

The module `<mod_log_config>` [29] of the Apache web server provides a flexible logging of client requests. Logs are written in a customizable format, and may be written directly to a file, or transferred to an external program. Conditional logging offers a filtering mechanism. Logs consider individual requests based on the requests' characteristics. Based on the logged

information it is possible to create an accounting system which supports the concepts and criteria described in Section 2.13.1.

2.13.4 Proprietary Accounting

The advantage of proprietary systems is the complete fulfillment of the customer's requirements. The data is provided and visualized in the desired format; it is possible to gather only the interesting information. Furthermore, if the accounting utility is part of the monitored software, there is no need to install new packages in the system.

2.13.4.1. Workflow

Figure 7 depicts the data flow in a system: after a user input, the system starts two parallel activities. On one side, it executes the required action, and returns an output. On the other side, it stores the relevant accounting data in a database.

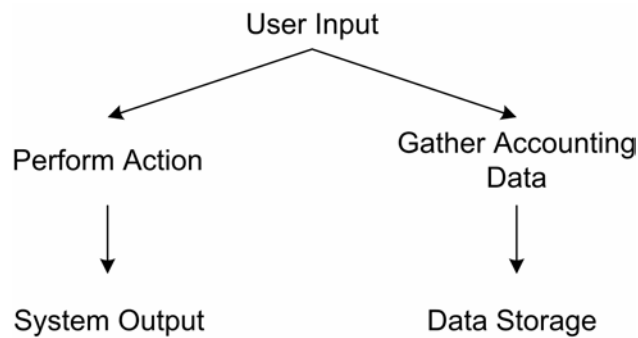


Figure 7: Proprietary Accounting

The data structure used in a proprietary accounting software should be kept as general as possible. This allows the modification of the gathering strategy, and grants more flexibility for later enhancements of the system.

3. Architecture

3.1 Community Interaction Tools

Before we begin to discuss architectural aspects, we want to specify a list of preconditions. These preconditions are based on the characteristics of the resource management portal, and have as main objective the maintenance of the system consistency. The preconditions are listed here below:

The resource management portal is an open source project [35]. The community interaction tools (CIT) in use must also fulfill this condition.

- The portal administrator does not need to be super user of the underlying operating system.
- The resource management portal shows its content dynamically. The installation of a community interaction tool must be modular.
- CITs are installed locally on the machine which hosts the resource management portal.
- The installation of a CIT module must be *single instance based* (explained below).

In order to explain the last point, we consider the case where the resource management portal comprises three community interaction tool modules (the modules CIT_1, CIT_2, and CIT_3). *The community interaction tool module (CITM) is a software package which enables the installation and the deployment of a determined CIT in the resource management portal.* There must be exactly one installation (instance) of the requested module for exactly one resource, see Figure 8. *The community interaction tool instance (CITI) is an installation of a CIT, unique and independent from other installations in the resource management portal. For the creation of a CITI, the correspondent Community Interaction Tool Module must be supported by the resource management portal.* In Figure 8, the Resource A can reference two different community interaction tool instances, but each one must represent a different module (CIT_1 Module respectively CIT_2 Module). Resource B cannot point to more than one instance of the same module. Two resources, Resource C and Resource D, cannot share the same instance of a CIT module. For instance, if the portal hosts n resources, the number of instantiations of one module is limited to n.

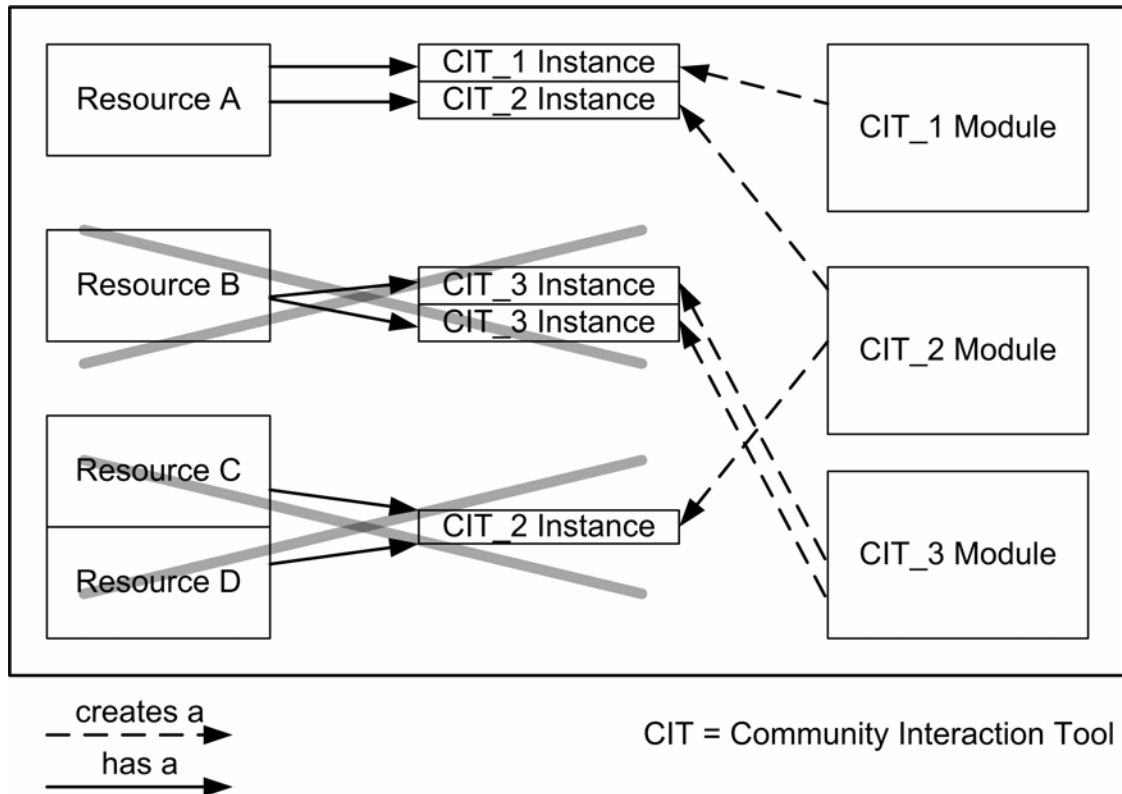


Figure 8: Single instance based installation of community interaction tools

3.1.1 Architectural aspects

The development of a community interaction tool module includes:

- The CIT software
- An instance installation utility
- An instance configuration utility
- An instance access utility
- An uninstall utility

According to the preconditions in Section 3.1, the *CIT software* – for example, a chat application – is an open source project. Its installation files must be stored locally on the resource management portal server.

As we see in the list above, it is necessary to develop four utilities. The first is the *instance installation utility*; responsible for the installation and the connection of the community interaction tool instance (CITI) to the corresponding resource. The *instance configuration utility* manages the configuration of the installed CITI. The configuration relies on the setting of user rights, and other CIT specific parameters of the CIT instance. The last two are the *instance access utility* which allows the users to log on the tool, and the *uninstall utility* which is responsible for the backup and the deletion of CITIs.

In the following sections, we describe the architecture of the presented components.

3.1.2 Instance Installation Utility

The installation of a community interaction tool is one of the portal administrator's tasks. Since the installation process is required to provide a high rate of automation, the implementation of a wizard is useful. A wizard allows a straightforward installation, because it interacts with the user and collects the needed information to execute the required task.

The implementation of an installation wizard can be realized in different ways. One option is the creation of a command line wizard which interacts with the portal administrator over a console. A different approach consists in the creation of a web-based wizard which can be accessed by the portal administrator with a standard web browser.

Command-Line Wizard

On one hand, the command line wizard can be easily implemented. On the other hand, it implies an overhead for the portal administrator. Typically, the use of a command-line tool requires a certain amount of know-how about the operating system. Given a web application; the main disadvantage for this kind of wizard lies in the addition of a heterogeneous component to the portal.

Web-Based Wizard

The implementation of a web-based solution is more complex than a command-line approach. The advantages of such an approach are: no further know-how is needed, the complexity of the task is encapsulated, and the resource management portal remains a fully web-based application.

In our solution, we propose to create a web-based wizard because it provides a straightforward approach that allows one to keep a unique user interface.
because it provides a straightforward approach that allows one to keep a unique user interface.

3.1.3 Instance Configuration Utility

The resource administrator is responsible for the configuration of the community interaction tool instance. Today, any community interaction tool provides the means to configure user accounts and CIT internal parameters – we call it a “manual configuration”. Because of this, the presence of a configuration utility in the community interaction tool module is not necessary for a correct deployment of the CIT.

For the development of this utility, we keep consistent with the wizard solution recommended in Section 3.1.2. Using a web-based configuration wizard gives the chance to create the same user interface for any CIT, and that way simplifies the configuration procedure for the user. The developed solution takes advantage of the information available in the resource management portal's database, since it utilizes an automated configuration process for the CIT instance. The manual configuration requires an individual creation for every account. Moreover, the usage of a wizard does not exclude the manual configuration.

Figure 9 illustrates a user account configuration. The configuration wizard of the *CIT_1* instance queries the data in the resource management portal database, and modifies the *CIT_1* database using the gathered data. Because of the correct configuration on its database, the CITI's user acceptance is straightforward.

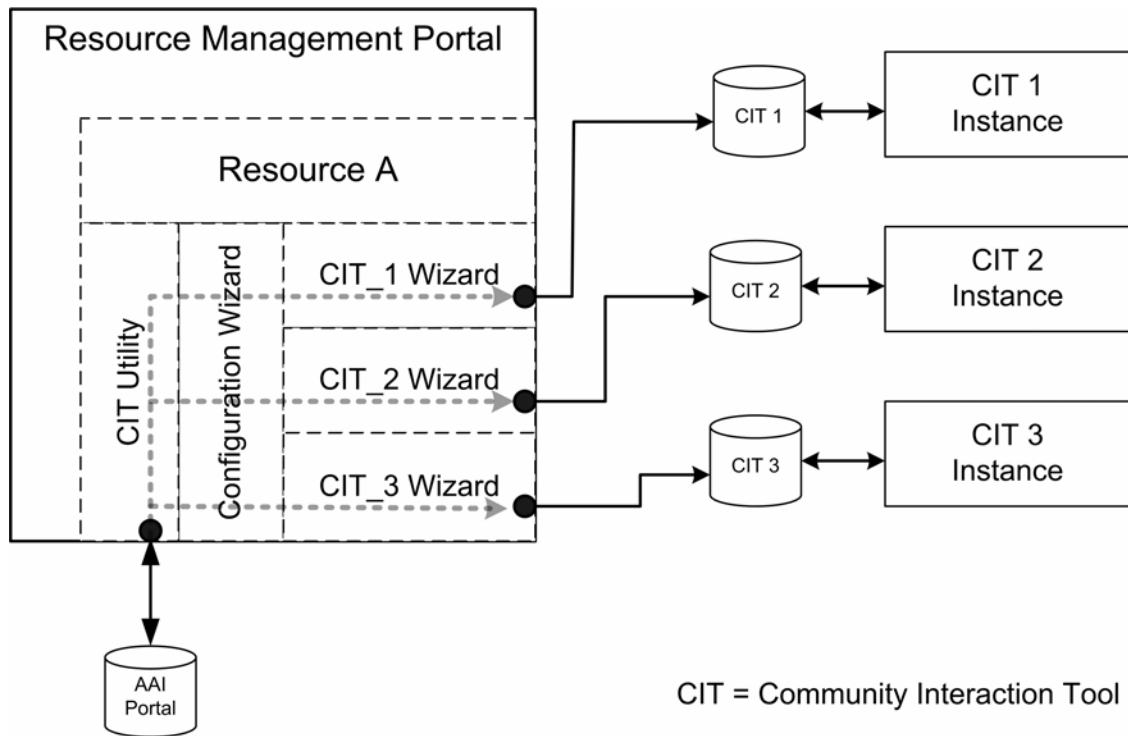


Figure 9: Configuration utility. The user account configuration in CIT instance

Setting the Permission Policy

A typical feature of the community interaction tool instance (CITI) configuration is the security management. The resource administrator manages the resources, the subscriptions, and is in charge of the CITI configuration. Therefore, he must take the decisions concerning the user access rights related to the CITI.

The configuration wizard has to provide a security management tool which is part of the configuration wizard. This utility can follow two different policies. The first is to grant or deny the access for 1) user groups, or 2) single users, and user groups.

While granting or denying of user groups access accelerates the configuration, the same task for single users allows one to be more flexible in the security management. We propose both functionalities in order to gain speed and keep flexibility.

3.1.4 Other Utilities

As considered in Section 3.1.1, the last two important features in a community interaction tool module are the user access tool, and the uninstall utility.

The first consists in a dynamic web interface. This gives the opportunity to list dynamically the links to the CITI. Thus, the user sees only the CITI-connections for which the resource administrator granted the user access rights. For the connection there is no need of credentials, since the user is already logged to the portal.

We propose an uninstall utility architecture analogous to the installation wizard (see Section 3.1.2). The wizard is accessible for the portal administrator and must perform the following tasks:

- Make a backup of the CIT instance file system.
- Make a backup of the CIT instance database.
- Reset/delete the CIT instance data and configurations.

3.1.5 Tasks and Responsibilities

There are three kinds of users in the resource management portal that have different roles. Each role includes tasks and responsibilities. It is necessary to define the operating range for each user role in the domain of the CITs. Table 4 illustrates the definition of the task responsibilities and functionalities for the portal user, the resource administrator, and the portal administrator.

Task, Responsibility, Functionality	Portal User	Resource Admin	Portal Admin
Installation of CIT Instances	∅	∅	✓
Removal of CIT Instances	∅	∅	✓
Activate installed CIT instances	∅	✓	∅
Configuration of CIT instances	∅	✓	∅
User and security management on CIT instances	∅	✓	∅
Accesses the CIT instances	✓	✓	✓

Table 4: Tasks, responsibilities, and rights of the resource management portal users

3.1.6 Proposed Architecture

The proposed architecture for the community interaction tool module implementation applies the concept of *single instance based* installation. Each resource gets a single installation of each available CIT module. All the implemented utilities must take into account this directive. Figure 10 depicts the recommended architecture.

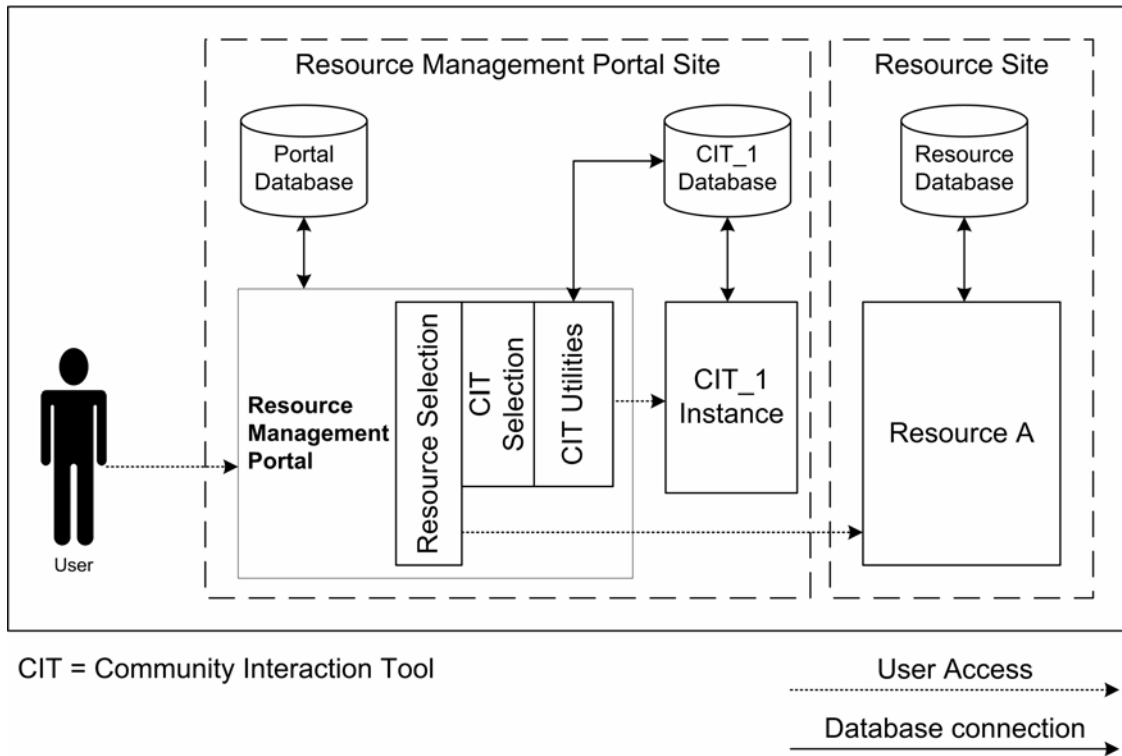


Figure 10: Community interaction tool support

We identify two sites: the *resource management portal site* and the *resource site*. In the resource management portal site we find the portal installation and a CIT instance with its database. The essential components in the portal are the *resource selection interface*, the *CIT selection interface*, and the *CIT Utilities* unit. While the first two are intended to direct the user to the desired target, the third contains all the utilities discussed above for the CITI installation and management. The utilities have access to the CITI-database, and the authentication and authorization infrastructure-portal-database. The resource selection guides the user to the wanted resource. The user may also choose a community interaction tool instance over the CIT access utility.

3.2 Accounting

The accounting utility is the basic tool needed for an accounting support. In the previous Section presented three different accounting tools. For the purposes of the resource management portal, we analyze and evaluate two of them: *Apache's logging for client requests*, and the *proprietary accounting* approach.

Apache's HTTP Server's logging of client requests

The logging capabilities of the Apache's HTTP server are very powerful. Since these functions belong to the web server's basic features, the performance is optimized. The server can log any HTTP request with its parameters. Thus, it can hold any activity defined in the data gathering strategy. Nevertheless, the goal of the resource management portal's developers is having an independent system from any external software different than the PHP interpreter. Therefore, creating an accounting tool which depends on the Apache's web server is inappropriate.

Proprietary Accounting Infrastructure

The implementation of a proprietary accounting infrastructure is the most customizable and flexible approach. It consists in a modification of the existing resource management portal software. Since the purpose of the accounting support in the resource management portal is the data collection concerning the resource management portal, the proprietary approach allows collecting the information exactly where it is requested. The direct access to the resource management portal's database simplifies this job. Furthermore, there is no need of additional know-how for the resource management portal's operator, and the portal remains a unitary entity.

We propose to implement a proprietary accounting support which is integrated in the resource management portal code. The advantages are: the accounting data is stored in the resource management portal database, no further installation or configuration of external tools is needed, the operator of the resource management portal is not bounded to a specific web server (Apache HTTP Server), and a full customization of the system is provided.

3.2.1 Architectural Aspects

The accounting component of a resource management portal aims at the data collection related to the users' traffic, and the activities on the resource management portal server machine. A monitoring of the remote resource sites is not required, since this is part of the responsibility of the resource site owner. The architecture for the accounting tool in the resource management portal covers the following aspects:

- The choice of the accounting tool
- The selection criteria for gathering data
- The choice of a data structure
- The data visualization
- The data management
- The data export tool.

In the following subsections we discuss each of these aspects, and propose the architecture for the support of accounting in the resource management portal.

3.2.2 Criteria for Data Gathering

The choice of a data gathering strategy consists in the decision of which activities in the resource management portal must be logged, and which ones are irrelevant. Since accounting is one means for the choice of a billing strategy or the survey of the portal, the decision must consider any requests of the administrators in the portal. For this reason, we propose to collect information for any activity which can be undertaken in the resource management portal. Table 5 shows that the user activities are subdivided into user roles. There are activities common to every role like *Login*, but there are also activities specific to a user role like *Remove CIT*. The availability of a data exporting tool makes it possible to evaluate the collected data according to individual needs.

Accounting Activities	Resource User	Resource Administrator	Portal Administrator
Login	✓	✓	✓
Logout	✓	✓	✓
Add Resource	✓	✓	✓
List my Resources	✓	✓	✓
List all Resources	✓	✓	✓
Modify Resource	✓	✓	⊘
Delete Resource	✓	✓	⊘
Add CIT	✓	⊘	⊘
Remove CIT	✓	⊘	⊘
Configure all CIT User Accounts	⊘	✓	⊘
Install CIT Accounts For all New Users	⊘	✓	⊘
Create Single CIT Account	⊘	✓	⊘
Modify Single CIT Account	⊘	✓	⊘
Change Password	✓	✓	⊘
Create User	✓	⊘	⊘
List all Users	✓	⊘	⊘
Delete User	✓	⊘	⊘
Accept or Resume Subscription	✓	✓	⊘
Reject Subscription	✓	✓	⊘
Suspend Subscription	✓	✓	⊘
Suspend Resource	✓	✓	⊘
Close Resource	✓	✓	⊘
Open Resource	✓	✓	⊘
Modify User	✓	⊘	⊘
List Pending Subscriptions	⊘	⊘	✓
Subscribe Resource	⊘	⊘	✓
Unsubscribe Resource	⊘	⊘	✓
Access CIT	✓	✓	✓
Access Resource	✓	✓	✓

Table 5: Accounting Activities on the resource management portal

3.2.3 Accounting Scheme

The accounting scheme is structured as follows:

- The data structure enables to store all the collected information according to the gathering strategy.
- The data structure used in the proprietary accounting software has to be kept as general as possible, which enables any kind of modification of the gathering strategy and future extensions of the resource management portal.

[**UserId, Timestamp, RemoteAddr, Activity, <Parameter, Value>, <Parameter, Value> ...**]

Figure 11: Proposed accounting scheme

Figure 11 depicts a possible accounting scheme. It provides a general definition of the data structure which is stored in the resource management portal database. It also contains information about the user (the user ID, a timestamp, the users IP address), and the performed activity (the activity name with a sequence of parameter-value pairs). Based on this scheme, accounting can be realized in an open and flexible way.

3.2.4 Access Limitation

The integration of an accounting system in the resource management portal raises the question of how to handle the stored data. The gathered data represents sensitive information about the users. Therefore, we want to define some limitations, so that not every user in the system has the permission to inspect such data. We call these restrictions the concept of “*Authorization and Confidentiality*”.

Authorization and Confidentiality

In the context of a resource management portal, there are three different user roles: the portal administrator, the resource administrator, and the user. Each user has access to his personal data, while the resource administrator disposes of the information about his resources and his resource subscribers. The portal administrator has access to all the available information. These rules support the protection of the users’ privacy.

3.2.5 Visualization

The AAI portal must provide the visualization support for the collected accounting data. Thereby, the users have access to the graphical representation of the accounting statistics on a web-page. This feature must take into account the authorization and confidentiality rules in order to protect the users’ privacy. The architecture of a visualization support must provide three different visualization areas which are mapped to the three user roles present in the resource management portal. Furthermore, the visualization must provide a data navigation utility which allows seeing more details about the collected data. We want to represent graphically only the data representing access to the portal, the community interaction tools, and the resources hosted by the portal.

3.2.6 Other Tools

The implementation of an exporting functionality is useful. Such a tool allows the users to import and elaborate the data using statistical software.

The portal administrator must be able to delete data older in order to preserve the privacy of former users and keep the database size under control. We call this tool the *accounting data management tool*.

4. Implementation

4.1 Community Interaction Tools

4.1.1 Integrated Community Interaction Tool Modules

The integration of community interaction tool modules requires an evaluation and a selection among the available open source community interaction applications. We decided to implement three modules. The two synchronous community interaction tools, that we want to create, are a file repository and a forum; while the chat represents the synchronous community interaction tool category. In sections 4.1.1.1, 4.1.1.2, and 4.1.1.3 we present the selected tools.

4.1.1.1. OWL Document Repository

An interesting module to develop within the resource management portal is a file repository. A valid choice among the available open source products is the “*OWL Document Repository (Version 0.71)*”. It consists in a multi user document repository (knOWLedgebase) system written in PHP 4 for publishing files/documents onto the web for a corporation, small business, or just for a group of people [30]. The main benefits of this software package are listed in the Table 6.

OWL Document Repository – Features
<ul style="list-style-type: none">• Multi-user environment<ul style="list-style-type: none">○ Folder and file permissions○ Group permissions• Size of the database is limited by the hardware• Searchable (integrated search tool)• Notifications for owner and group• Easy to use

Table 6: OWL Document Repository Features

4.1.1.2. PHP Bulletin Board (PHPBB) Package

Today we cannot imagine a community interaction environment without a forum. We selected “*PHPBB 2 bulletin board (Version 2.0.6)*” [31], because it is one of the best and the most popular online forums available on the internet. We can say that it is a high powered, fully scalable, and highly customizable open source bulletin board package which has a user-friendly interface, a simple and straightforward administration panel, and a helpful frequently asked questions list (FAQ). In addition PHPBB is based on the PHP server language and on the choice of the operator, it supports the MySQL, MS-SQL, PostgreSQL or Access/ODBC database servers. In Table 7, we see an overview of the key features of the “PHPBB Bulletin Board”

PHPBB 2 Bulletin Board – Features
<ul style="list-style-type: none"> • Supports popular database servers • Unlimited forums and posts • Multiple language interface • Private or public forums • Powerful search utility • Private messaging system • Complete customization with templates

Table 7: PHPBB Bulletin Board – Features

4.1.1.3. Aerial Chat

The last community interaction tool module to integrate in the resource management portal is a chat application. Among the available open source chat tools, we selected the “*Aerial Chat, Version 6.2*”. It is a powerful chat tool written in PHP which provides a support for the MySQL database engine [32]. Table 8 presents the key features of this chat tool.

Aerial Chat – Features
<ul style="list-style-type: none"> • Unlimited channels • Images support • Hyperlinks support • Mp3 support • Graphics effects support • IP Tracer • Encrypted passwords

Table 8: Aerial Chat - Features

4.1.2 Development

The development of a community interaction tool module requires the support of common functions needed for the basic tasks which are: the installation, the configuration, the removal, and the access to the integrated CIT. The support of community interaction tool modules requires the implementation of PHP pages which act as entry points to the functionalities listed above. Another aspect to consider is the modification of the available database. It is necessary to extend the database with a supplementary table which is responsible for the dynamic behavior of the application. Further, the creation of a set of PHP pages, and a library for the support of tool specific features is essential.

In what follows we present the database extension, the CIT supporting libraries, the task specific files, and functionalities.

4.1.3 Database Modifications

As already mentioned, the integration of community interaction tools involves the creation of an additional table in the resource management portal database. Figure 12 depicts the new database table called `CommunityInteraction`, and the already existing table `Resource`.

The table `CommunityInteraction` contains four columns: `cit_tool_name`, `cit_resource`, `cit_tool_title`, and `cit_active`. These columns are explained in Table 9.

<code>cit_tool_name</code>	is the unique name for the CIT Module (i.e., "OWL", "PHPBB2", "AECHAT") in the database
<code>cit_resource</code>	is the unique ID of the resource, which is going to be connected to the CIT instance
<code>cit_tool_title</code>	is the extended tool title (i.e., "OWL, the Document Repository")
<code>cit_active</code>	is a Boolean value; it is true if the community interaction tool instance is installed and active

Table 9: The resource management portal database. The "CommunityInteraction" table

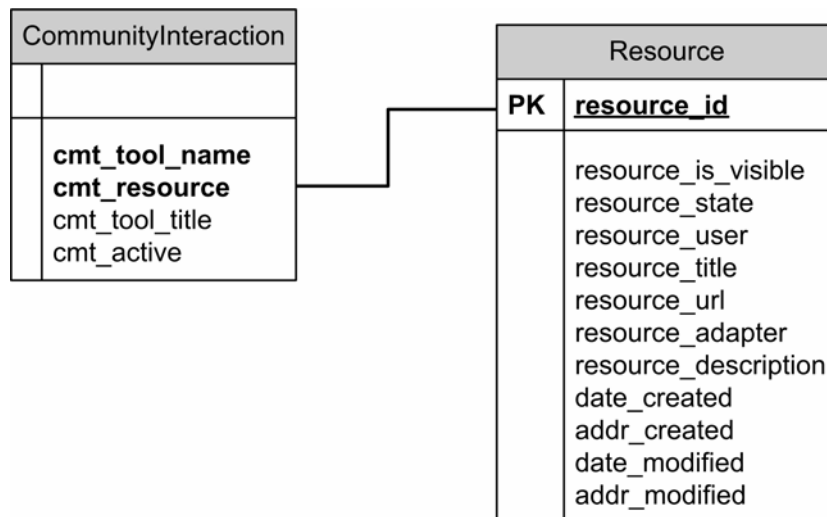


Figure 12: Resource management portal database - "CommunityInteraction" -new table definition

The information stored in the database table `CommunityInteraction` enables the resource management portal to display the right menus holding the dynamic behavior of the PHP pages.

4.1.4 Wizards

In order to encapsulate the complexity of implemented tasks we make use of wizards. They permit a straightforward approach to the installation and configuration of community interaction tools. Figure 13 depicts the installation wizard in form of a process flow chart. At the beginning, the process interacts with the user until the required installation parameters are complete. In a second step, the community interaction tool instance (CITI) installation is launched, and the process ends.

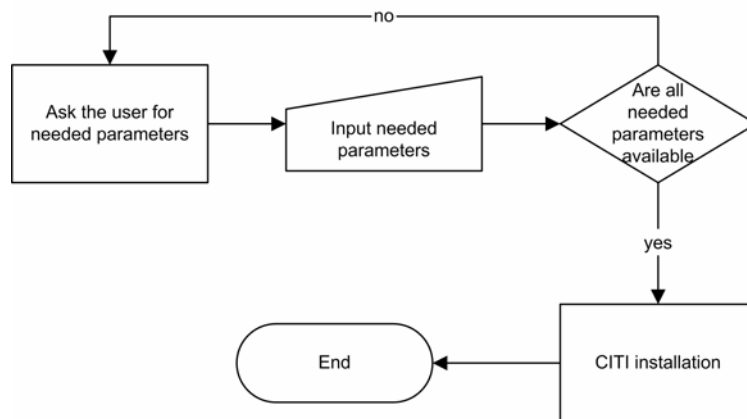


Figure 13: Flow chart of an installation wizard

4.1.5 Community Interaction Tool Module Libraries

The main functions of the community interaction tools framework are contained in the following files:

- `community.inc`
- `cit.class.inc`
- `citGeneral.inc`
- `cit_forms.inc`
- `phpbb2_forms.inc`

community.inc

This is the main library of the CIT framework. It contains the functions which handle with database specific tasks, file system specific functions, and methods which enable to prepare the output on the web browser. A part of these functions permits a general usability, while others

present CIT module specific functionalities. In what follows we present a selection of representative functions implemented in the file `community.inc`.

untar(\$fileName, \$fileLocation, \$targetDirectory)

The CIT installation files are provided by the manufacturer as tar archived files. `untar()` allows to open a file (`$fileName`) located in a directory (`$fileLocation`), and to extract the archived files to the target location (`$targetDirectory`)

replaceStringInFile(\$filename, \$searchPattern, \$replacePattern)

During the installation of software it is necessary to modify configuration files. This function gives is the automatic solution. It opens the file (`$filename`), searches a string in the file to modify (`$searchPattern`), replaces it with another string (`$replacePattern`), and store the file. This activity is done using regular expressions. The term of regular expression is defined in the following paragraph.

Regular Expression

A regular expression (sometimes abbreviated to "regex") is a way for a computer user or programmer to express how a computer program should look for a specified pattern in text and then what the program has to do with the found pattern matches. For example, a regular expression could tell a program to search for all text lines that contain the word "Windows 95" and then to print out each line in which a match is found or substitute another text sequence (for example, just "Windows") where any match occurs.

The best known tool for specifying and handling the incidence of regular expressions is `grep`, a utility found in Unix-based operating systems and also offered as a separate utility program for Windows and other operating systems [33].

backupAndDropDatabase(\$host, \$user, \$password, \$dbname)

This function is needed during the uninstall process of the CITI. Basically it creates a backup of the CITI database and stores it in the file system with a unique name. Then, it drops the CITI database. The input variables are needed to identify the database and set the needed user rights on it.

filterCitConfigurableResources(\$allResourcesArray, \$citShortname)

In order to select the CIT instance to remove, the system must inform the portal administrator which resources have a CITI of a determined kind (`$citShortname`; *i.e.*, OWL). In other words, it returns the resources which possess the CIT the portal administrator wants to uninstall. As already mentioned, there is exactly one resource for a CIT instance. In this context, this function takes an array containing the entire resources and tests for each one, if a corresponding CITI exists. The routine returns an array with the resources.

getCitDBNameFor(\$aResource, \$citShortName)

Since it is possible to install several CIT instances, the name of their databases must be unique. This function creates a database name using the resource (`$aResource`) and the selected CIT module (`$citShortName`) as input parameters. The result is a string composed by the `$citShortName`, the string "resource", and the resource unique ID (*i.e.*, `owl_resource_4`).

queryCitDatabase(\$resource_id,\$aQuery, \$citShortName)

This function executes a SQL query on a CITI database. The database is identified with the Resource ID (\$resource_id) and the CIT module (\$citShortName) parameters. The function returns the query result.

tarAndRemoveCitFiles(\$installDir,\$aBackupDirString)

This function creates a backup of the CITI file system and deletes it. It creates a compressed archive file for all the files contained in the directory (\$installDir) and stores them in the target directory (\$aBackupDirString). This routine is used when the portal administrator wants to uninstall a CITI.

cit.class.inc

This file contains the definitions of the classes `CitResource` and `CitUser`. They provide community interaction tool specific functionality for the users and resources in the resource management portal.

citGeneral.inc

Another necessary file is `citGeneral.inc`. This file contains general functionality like returning warnings, or hyperlinks in HTML format.

cit_forms.inc and phpbb2_forms.inc

The CIT framework displays many forms on the different wizard pages. They are needed for the installation, configuration, or just to pass values to the next PHP page. Often, it happens that the same forms must be displayed in several PHP pages just using another parameter. These two files provide the parameterized HTML forms. In particular, the file `phpbb2_forms.inc` provides forms that are specific to the “PHPBB Bulletin Board” module.

4.1.6 Implemented Utilities

4.1.6.1. Installation Utility

The CIT module must be instantiated in the resource management portal, before the resource administrator is allowed to configure his CITI. For this purpose, the portal administrator must create the CITI using the installation utility. The latter makes use of three PHP pages in addition to the functions present in the libraries. In order to illustrate the mechanism how the installation is realized, we choose the example of the “OWL Document Repository”.

add_cit.php

The file `add_cit.php` is a PHP page represents the entry point for the installation of a CIT instance in the resource management portal. From here it is possible to choose among the available CIT modules.

getAvailableCitWizardLinks()

This function takes the list of the available CIT modules in the resource management portal and creates the hyperlinks (in HTML format) to the install wizard page.

owl_install_wizard.php

This PHP page displays a form where the portal administrator provides the necessary information for the CIT instantiation. The collected values are sent to the installation page, which instantiates the requested CIT module.

filterInstallableResources(\$allResourcesArray,\$citShortname)

When the portal administrator wants to install the CITI, first he chooses the module to instantiate, and then he selects the corresponding resource. This function returns the list

of available the resources which have no CIT instance of the chosen module (*i.e.*, the “OWL Document Repository” module).

owl_installation.php

This PHP page is responsible for the installation process. Figure 14 illustrates the CIT installation process in a flow diagram.

OWL Document Repository Installation Flow

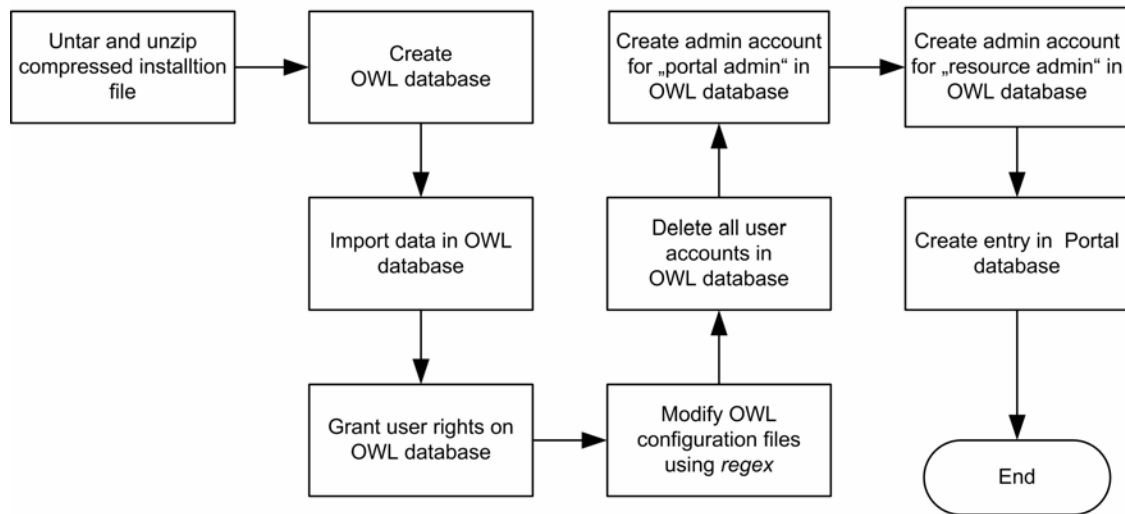


Figure 14: Community interaction tools installation process flow - OWL example

4.1.6.2. Uninstall Utility

The portal administrator can uninstall the instantiated CIT modules. The procedure is similar to the installation procedure. In this example we chose the “PHPBB 2 Bulletin Board” module.

The PHP page `remove_cit.php` is the entry point of the uninstall utility. From here the user selects the kind of community interaction tool he wants to remove. The file `phpbb2_remove_wizard.php` contains the uninstall wizard. The information collected in this form is sent to the file `phpbb2_remove.php` in order to backup the CIT instance, drop the database, and delete the CITI files.

4.1.6.3. Configuration Utility

The configuration utility contains a complex mechanism. The entry point is `config_cit.php`. The wizard is implemented in `aechat_config_wizard.php`, and the configuration is performed in `aechat_configuration.php`. The Figure 15 depicts how the configuration wizard works.

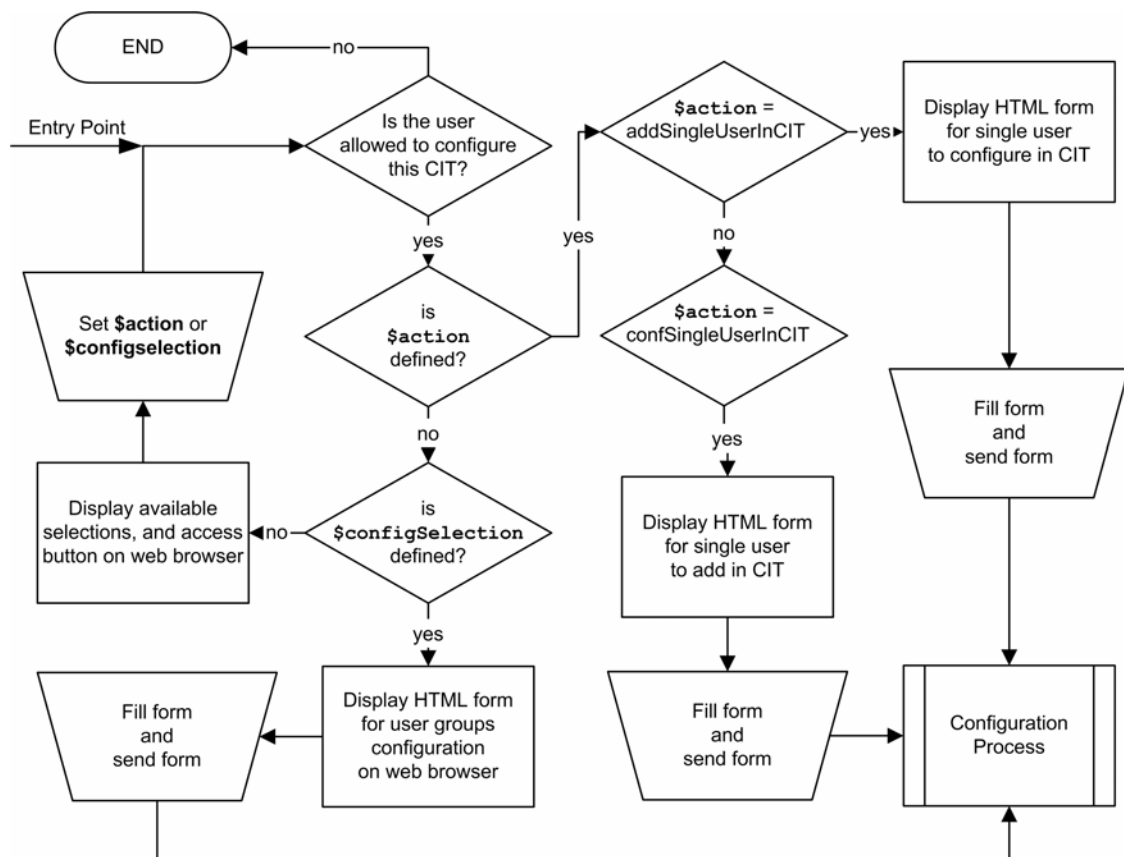


Figure 15: Configuration Wizard - Flow Diagram.

When the user opens the `aechat_config_wizard.php` page on his web browser, he views the list of possible actions (configure all accounts in the chat, install new accounts for all the new subscribers of the resource, modify the configuration of all the existing accounts, and install/configure a single account). After this choice the user can set the requested parameters in a form (*i.e.*, disable account), and send the data to the configuration process.

Configuration process

This process evaluates the collected data (sent by the wizard), and sets up the aerial chat database, in order to create one or more accounts, or modify the settings of existing ones.

4.1.6.4. Access Utility

The access utility is the last needed feature to create a CIT module. The PHP page responsible for it is `access_cit.php`. When the user opens this page he gets the list of the available CITIs in conjunction with login buttons. The user simply has to click on it, and the browser opens a new window with the CITI. The user does not need to log in manually. The PHP page contains an automatic login mechanism which uses the password generation.

Password Generation

When a user opens the “`access_cit.php`” with his web browser the program generates a one time random password. After clicking on the login button the access utility sets the generated password in the CIT database and the user is automatically logged in.

4.2 Accounting Support

4.2.1 Developed Features

The creation of a library is necessary to implement the following functionalities: a data gathering mechanism, the support of the available resource management portal database, and a visualization utility for the web browser. It is also needed to implement a data management utility. It allows one to export the accounting information, and remove old data stored in the resource management portal database. The library is written in PHP and is integrated in the resource management portal library.

4.2.2 Database Modifications

The resource management portal database needs an extension, in order to support accounting. The needed table is named `Accounting`. Figure 16 illustrates the `Accounting` table in relation to the `User` table.

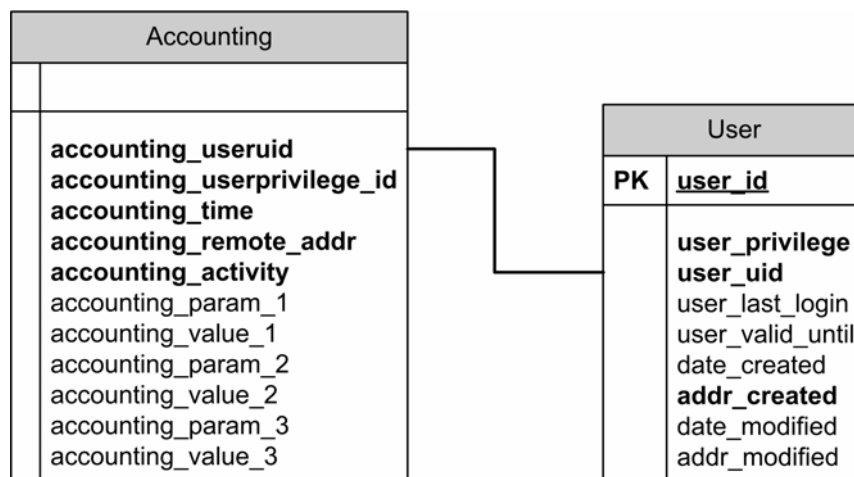


Figure 16: The resource management portal database - "Accounting" -new table definition

The table structure is associated with the accounting schema presented in Section 3.2.3. The columns present in the table `Accounting` are illustrated in Table 10.

<code>accounting_userid</code>	is the user unique id of the user which starts the activity. Corresponds to the column <code>user_uid</code> in the <code>User</code> table
<code>accounting_userprivilege_id</code>	is the identification of the user role. (portal administrator, resource administrator or user). Corresponds to the column <code>user_privilege</code> in the <code>User</code> table
<code>accounting_time</code>	is the timestamp taken at the moment of the activity launch
<code>accounting_remote_addr</code>	is the user IP address
<code>accounting_activity</code>	is the name of the activity (<i>i.e.</i> , "AccessResource")
<code>accounting_param_1</code>	is the kind of value 1 (<i>i.e.</i> , Resource)
<code>accounting_value_1</code>	is the value for the first parameter (<i>i.e.</i> , a number that identifies the resource ID)
<code>accounting_param_2</code>	is another parameter name. Some activities need more parameters. It can be <code>NULL</code>
<code>accounting_value_2</code>	is another parameter value. Some activities need more parameters. It can be <code>NULL</code>
<code>accounting_param_3</code>	is another parameter name. Some activities need more parameters. It can be <code>NULL</code>
<code>accounting_value_3</code>	is another parameter value. Some activities need more parameters. It can be <code>NULL</code>

Table 10: The resource management portal database. The items in the table `Accounting`

4.2.3 The Accounting Library

The class `Accounting` encloses the complete functionality of the accounting library. The class is stored in the file `Accounting.class.inc`. In Sections 4.2.3.1 and 4.2.3.2 we describe a selection of functions considered important.

4.2.3.1. Data Gathering Related Functions

The implementation of the data gathering process requires the modification of almost all the files present in the resource management portal. This enables the accounting system to collect the data when the activity is started (or literally when the user clicks on the event which should be logged).

`createEntry($activity, $param1, $val1, $param2, $val2, $param3, $val3)`

This function is responsible for the storage of the accounting data in the resource management portal database. Any logging function in the resource management portal calls this function to store the data. The mandatory input parameters are `$activity`, `$param1`, and `$val1`. The function takes these parameters, and stores them in the database with the other values gathered by itself (the unique user ID, the user IP address, and a timestamp).

`accAccessCit($citShortName, $resourceId)`

This function is used to log the user accesses to a CITI. The input parameters are `$citShortName` (the module identification), and `$resourceId` (the resource

identifier). These two parameters allow determining which CITI is accessed. During the execution this function calls `createEntry()` (function described above).

4.2.3.2. Data Visualization

The resource management portal must provide the visualization of the collected accounting data. Thereby, users have access to the graphical representation of the accounting statistics on a webpage. This feature must take into account the authorization and confidentiality rules described in Section 3.2.4.

Visualized Categories

The resource management portal must display information concerning the users, the resources, and the community interaction tools (CIT). In order to get an idea of the general state of these entities, we use the overviews. They display the accounting information about entity groups in the form of PHP pages or HTML tables. For example, a table which presents general information about all users contains a user list (each row contains data related to a user), and some basic data regarding the single user, like the *number of performed logins*. Table 11 shows the possible structure of an overview table for all users.

User Name	Number of Logins Today	Total Number of Logins
George Huston	3	234
Melanie Crosby	2	122
Bob Duster	4	145

Table 11: Accounting Visualization Concept. All Users Overview

Sometimes it is interesting to see data about single entities *e.g.*, a resource. Here it is more interesting to know for example, which people accessed the resource, and how many times. In order to display information about the single entities, we introduce this view. Table 12 illustrates what a single view could look like.

User Name	Total Accesses to Resource A
George Huston	25
Bob Duster	17

Table 12: Accounting Visualization Concept. Single resource view for resource A

A view represents data for a single entity, *e.g.*, a single resource. An overview contains data for a group of entities, *e.g.*, all resources. This concept permits to navigate from the overviews to the single views using hyperlinks. Here, an example: an overview shows all resources hosted on the portal together with the number of user accesses. Upon a click on the resource name a table displays which and how frequently users access this resource.

All Users Overview

This is a HTML table which contains the following columns:

- *User Name*: Given name, and the family name of the user
- *Login Name*: User unique id of the user
- *Last Login*: Last time when a determined user entered the resource management portal
- *Number of Logins*: Number of times the user logged in
- *Activity Rate*: Number of counted activities (see also Table 5)

All Resources Overview

The resource overview is a HTML table. Each row contains information about a resource. The table consists of the following columns:

- *Resource Name*: Name of the resource
- *Resource Provider*: Name of the resource administrator
- *Number of Accesses*: Total number of user accesses to a resource

All Community Interaction Tools Overview

An HTML table: each row contains information about a community interaction tool instance. It consists of the following columns:

- *Resource Name*: Name of the resource
- *Resource Provider*: Given name and family name of the resource administrator
- *CIT Name*: Name of the community interaction tool module
- *Number of Accesses*: Total number of user accesses to a CIT instance

Single User View

This view is a PHP page, and contains the following data:

- *User Name*: Given name, and the family name of the user
- *Number of Logins*: Number of logins the user did on the resource management portal
- *Resource Accesses*: Table containing each accessed resource and the number of accesses
- *CIT Accesses*: Table containing each accessed CIT and the number of accesses
- *Access History*: The complete list of the accesses to the portal, each resource, and each community interaction tool instance of the user

Single Resource View

The single resource view is a HTML table which contains following columns:

- *User Name*: The given name, and the family name of the user
- *Number of Accesses*: Number of accesses made by the user on the resource

Single Community Interaction Tool Instance View

An HTML table with following columns:

- *User Name*: The given name, and the family name of the user
- *Number of Accesses*: Number of accesses made by the user on the community interaction tool instance

The Users' Accounting Data Visualization Front-End

The visualization of the accounting data is done on the PHP page `accounting.php`. This script collects the data about the user, and displays the following information: the number of performed logins, the ranking of his used resources (which resource is accessed more), and the ranking of the accessed community interaction tool instances. At the bottom of the page the user sees the history of the accessed resources.

The Administrators' Accounting Visualization

Three files are essential to do the graphical visualization of the gathered data for the administrators. These are:

- `users-accounting.php`
- `resources-accounting.php`
- `cits-accounting.php`

`users-accounting.php`

This page shows a table representing all the users according to the rules of *Authorization and Confidentiality*, the number of user logins, and the activity rate, which tells how many activities were logged in the `Accounting` table. The user names are displayed as links. This gives the opportunity to get the detailed data for a single user.

`getAllUserAccountingOverviewString($adminUIId)`

This function is responsible to create the table displayed by the PHP page `users-accounting.php`. It shows a histogram for the number of logins, and one for the activity rate. The table also contains links to see further details about the users. The input parameter `$adminUIId` is used to determine which data the user is allowed to inspect.

`resources-accounting.php`

The functionality of this page is analogous to the case of the users' overview. This PHP page is responsible to create the overview on all resources. It shows a ranking list (with histogram) of which resources are most accessed and links for navigating into additional details about the resources. This job is done by the function `getResourcesGlobalAccessTable($adminUIId)` (see Section 3.2.5.)

`cits-accounting.php`

This page shows the overview with ranking for the community interaction tool instances. It displays links for the CITIs, and for the responsible resource administrators.

4.2.3.3. Data Management

The deletion and the export of accounting data in the database is given by the PHP page `export-accounting.php`. This allows the resource administrators to export data, concerning their resource users. The data is saved to a file in comma delimited format (CSV).

Beside the access to all the user data for exporting purpose, the portal administrator can delete the accounting data in the database that is older than a date defined by him. This feature is implemented in the file `export-accounting.php`.

5. Configuration and Usage

5.1 Community Interaction Tools Configuration

The configuration of the community interaction tool modules necessitates the adaptation of the file `CIT.constants.inc`. This file contains CIT module specific and general values.

5.1.1 General CIT Constants

CIT_INST_FILE_LOC

The installation files for the community interaction tools are located somewhere on the resource management portal server. This constant gives the absolute path of the directory where they stay.

Example:

```
define("CIT_INST_FILE_LOC", "/var/www/portal/community/CIT_Files")
```

CIT_PICS_PATH

The community interaction libraries make use of pictures to display some PHP pages on the web browser. This constant contains the *url-path* [34] of the pictures directory.

Example:

```
define("CIT_PICS_PATH", "/aai-portal/community/pics/")
```

5.1.2 Tool Specific Constants

The file `CIT.constants.inc` contains the settings for the installation and deployment of the available CIT modules. Each module needs at least four constants. We present the “Aerial Chat” related constants. The constant names for the other modules are similar. It is sufficient to replace the substring of the constant name “AECHAT” with “OWL” to get the corresponding constant for the “OWL Document Repository”.

CIT_AECHAT_TITLE

This constant is used to set the name of the module, which is displayed on the browser.

Example:

```
define("CIT_AECHAT_TITLE", "Aerial Chat")
```

CIT_AECHAT_SHORT_NAME

This value is crucial for the deployment of the community interaction tool support. It identifies the CIT module. This value is also used to ask the resources for their instantiated modules.

Example:

```
define("CIT_AECHAT_SHORT_NAME", "AECHAT")
```


CIT_AECHAT_INST_FILE

The installation file name of the “Aerial Chat” is defined in this constant.

Example:

```
define("CIT_AECHAT_INST_FILE", "aerial62.tar.gz")
```

CIT_AECHAT_CONF_FILE

During the installation process of the CITI, the configuration files of the community interaction tools are automatically modified. To do this the system needs the relative path and the name of the file which necessitates to be changed.

Example:

```
define("CIT_AECHAT_CONF_FILE", "chat/connect.inc.php")
```

CIT_AECHAT_PREFIX

To apply the concept of single instance based installation (explained in Section 3.1), we need to define a unique database name for the CITI on the MySQL server. The name is built by a prefix, the word “resource”, and the resource ID number on the resource management portal database (*i.e.*, aechatt_resource_12). It is possible to deploy several resource management portal installations on the same computer using the same database server. If so the prefix for the CIT modules must differ for each installation.

Example:

```
define("CIT_AECHAT_PREFIX", "aechatt")
```

CIT_AECHAT_ADM_PASS

The “Aerial Chat” supports only one administrator. Therefore, the portal and resource administrator make use of the same credentials to access the chat tool. The administrator password is stored in a configuration file of the chat application. This constant allows setting this password.

Example:

```
define("CIT_AECHAT_ADM_PASS", "LetMeIn67")
```

5.2 Accounting Configuration

The accounting system needs only two parameters to be configured. They define the maximal width (in pixel) of the histogram graphic, and the location of accounting related pictures on the server. The values are located in the file `constants.inc`.

ACC_MAX_HISTOGRAM_WIDTH

This constant defines the maximal width of a histogram. This value is used during the visualization of the accounting data.

Example:

```
define("ACC_MAX_HISTOGRAM_WIDTH", 150)
```

ACC_PICS_PATH

This constant defines the location of the pictures related to the accounting tasks. The value is the *url-path* of the directory which contains the figures.

Example:

```
define("ACC_PICS_PATH", "/aai-portal/pics/")
```

5.3 Overview of the Graphical User Interface

In Section 2.6, we mentioned that the web front-end structure is subdivided into three parts: the *header area*, the *menu area*, and the *content area*. Figure 17 illustrates the window of the resource management portal following these criteria. We also see the overview of all available community interaction tool instance (CITI) for the resource called *Rechtsmedizin*.

In this Section we often see the term “*Community Management*”. This term is equivalent to “*Community Interaction*”.

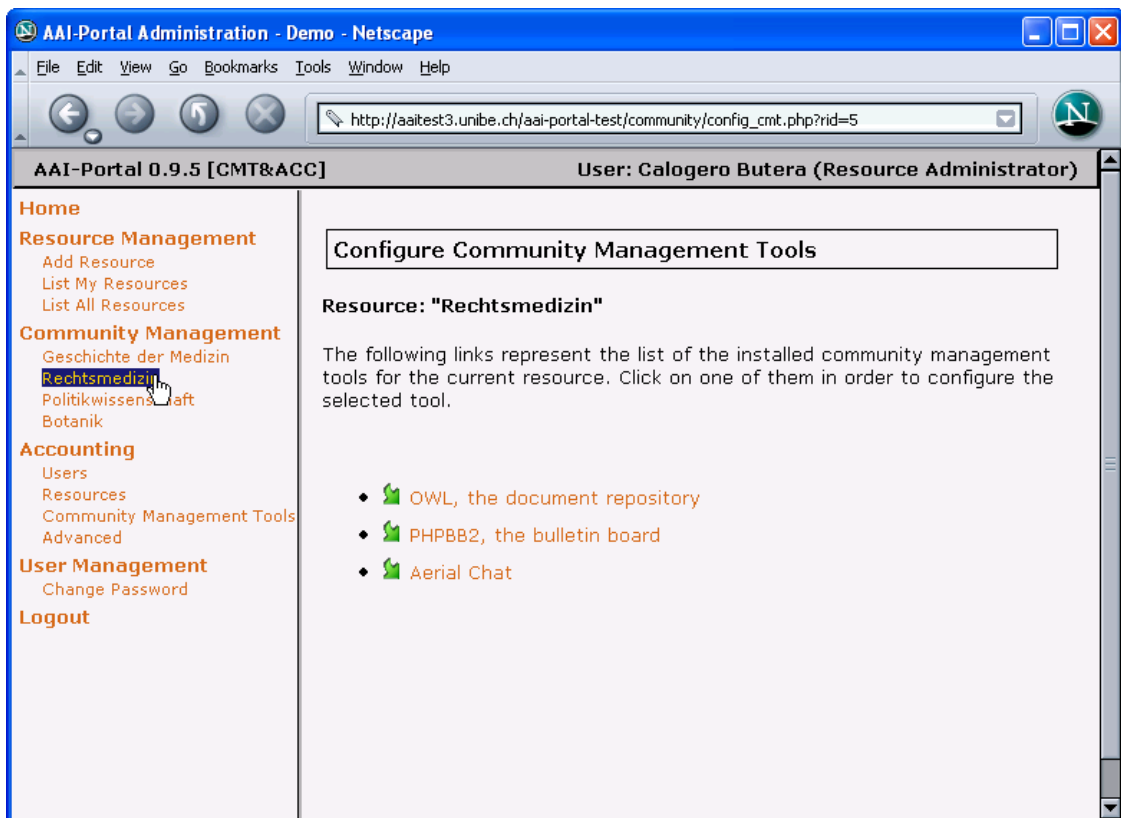


Figure 17: The resource management portal page - Overview.

With development of the community interaction tools integration, and the accounting system for the resource management portal we introduced new menu items. There is a new menu structure for each available user role in the resource management portal.

5.3.1 The Portal Administrator's Menu

In the domain of *Community Management* the portal administrator has two new responsibilities (*add*, and *remove* CITI). In the accounting menu he can select among the overviews for all users, all resources, or all CITI instances present in the portal. The menu item *Advanced* in the section *Accounting* allows to reach the *data management panel*.

Figure 18 shows the new menu structure for the portal administrator on the resource management portal.



Figure 18: New resource management portal menu for the portal administrator

5.3.2 The Resource Administrator's Menu

The new menu for the resource administrator contains two new sections. The section *Community Management*, and the section *Accounting*. Figure 19 depicts the new menu structure for the resource administrator.



Figure 19: New resource management portal menu for the resource administrator

The section *Community Management* lists all his resources as menu items. As we see in Figure 17, when the resource administrator clicks on the resource, he gets a list of the available CITIs. The Section *Accounting* lists the same items as for the portal administrator, but in opposition to him, the resource administrator gets the overviews only for his resources and his users. The data management is also limited to the export of the data concerning his resource users.

5.3.3 The User's Menu

For the needs of the resource management portal user we introduce a section *Community* and the menu item *Accounting* in the section *Profile Management*. Figure 20 illustrates the new menu structure.



Figure 20: New resource management portal menu for the user

In the section *Community* the portal lists all the resources the user subscribed (*Psychologie*, *Recht*, *Rechtsmedizin*, and *Politikwissenschaft*).

5.4 Installation of a Community Interaction Tool

The wizard based installation of the CITI encapsulates the complexity of this task enormously. The portal administrator clicks on the menu item *Add Tool*, selects the module to install (*i.e.*, OWL), completes the web form displayed on the screen, and clicks on the *Install* button. The installation process creates automatically the requested CIT instance. Figure 21 depicts the installation form for the “OWL Document Repository”.

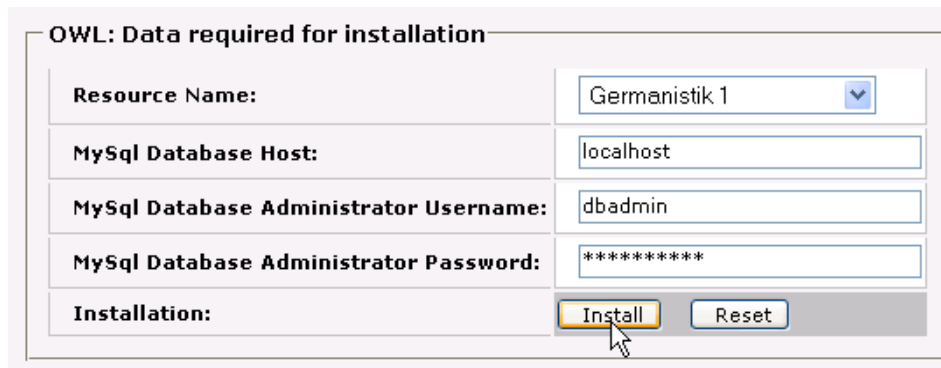


Figure 21: OWL installation wizard form

The installation process needs following data: the resource name, the host of the used MySQL database server, user name and password of the database administrator.

The Resource Name

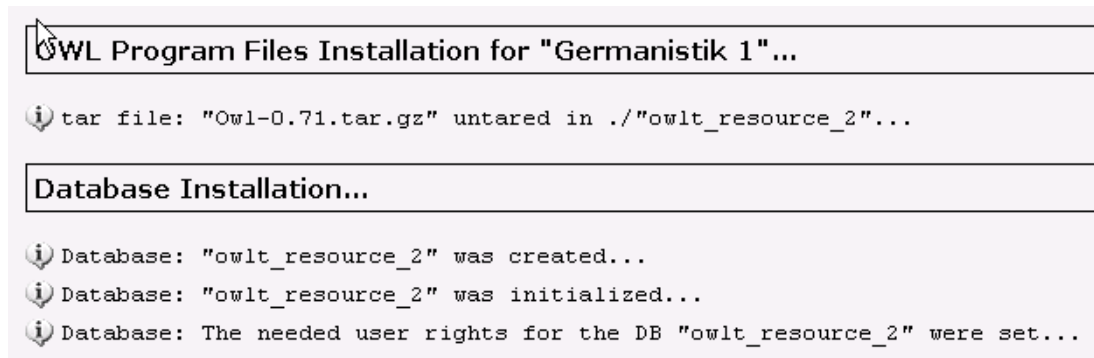
The instantiated CIT is mapped to a resource, the installation wizard form allows selecting one of the “free resources”.

The Database Host

It is the host of the database server. In most cases the value is “localhost”.

The Credentials of the Database Administrator

The installation process creates a new database and grants the user rights to the resource management portal database user. This is the reason why, this routine requires the credentials of a database user with enough user rights. Figure 22 illustrates the system output during the installation process. The compressed archived files are extracted, the document repository’s database is created and the user rights are set.

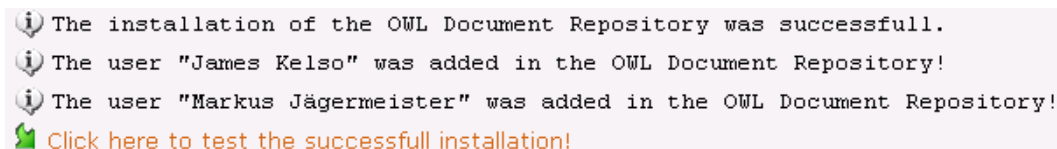


```
OWL Program Files Installation for "Germanistik 1"...
tar file: "Owl-0.71.tar.gz" untared in ./"owlt_resource_2"...

Database Installation...
Database: "owlt_resource_2" was created...
Database: "owlt_resource_2" was initialized...
Database: The needed user rights for the DB "owlt_resource_2" were set...
```

Figure 22: OWL document repository installation. The files, and database operations

Figure 23 depicts the output at the end of the installation process. The system creates two accounts: one for the portal administrator, and one for the resource administrator. Furthermore, it shows a hyperlink to the just installed feature.



```
The installation of the OWL Document Repository was successfull.
The user "James Kelso" was added in the OWL Document Repository!
The user "Markus Jägermeister" was added in the OWL Document Repository!
Click here to test the successfull installation!
```

Figure 23: Document repository installation completed. Account creation for the administrators

5.5 Configuration of a Community Interaction Tool

The resource administrator configures user accounts, and other options in the CITI. To access the configuration wizard, he must select the resource in the *Community Management* section, and choose the CITI to work with. Figure 24 shows the available options for the “Aerial Chat” configuration.

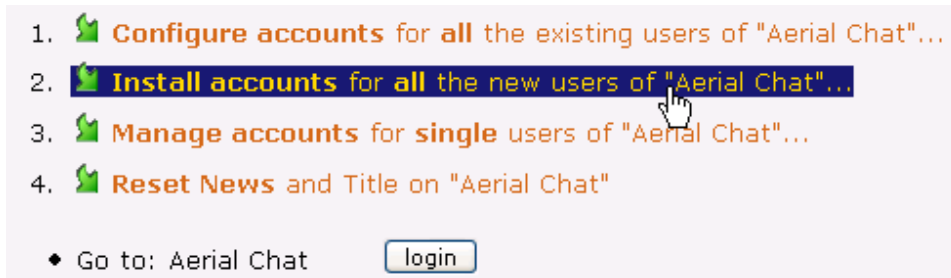


Figure 24: Configuration wizard for the "Aerial Chat" application

The configuration wizard of the CITI allows creating accounts on the tool, modifying the settings of existing accounts, or disabling them. The resource administrator is enabled to perform these actions for a single user, for all resource subscribers, or for the existing user accounts in present CITI. The wizard also provides the opportunity to modify general values on the CITI, or access the tool (see Figure 24).

5.5.1 Single User Account Configuration

As already mentioned, the resource administrator is able to manage single accounts on the CITI for his resource subscribers. For this purpose the configuration wizard displays the list of all subscribers in a table. This permits the administrator to select the account he wants to manage.

Figure 25 presents the overview table for the management of single user accounts in the CIT instance. It provides information about the user name, the user login name, the availability of an account on the CITI, and the state of the account (enabled, disabled, or "no account"). The button aside of the user data allows to create, or modify the user account.

Subscriber	Login name	OWL Account	OWL status	Action
David Vogel	dave	no	no account	Create Account
Franz Meier	franz	yes	disabled	Configure Account
Michele Lanzani	michele	yes	enabled	Configure Account
Demo Demo	demostudent@unibe.ch	yes	enabled	Configure Account
Martin Faller	29325@aaitest.switch.ch	yes	enabled	Configure Account

Figure 25: Overview Table for the management of single user accounts in the CIT instance

The resource administrator creates/configures a user account which is bound to a set of permissions and options. The configuration wizard displays them in a HTML form. Figure 26 depicts this form for the "OWL Document Repository".

Given Name	David
Surname	Vogel
Login	dave
Email	vogel@iam.unibe.ch
Language	german
Quota 0=Disabled	<input type="text" value="0"/>
Max Sessions	<input type="text" value="0"/>
Receive Notifications	<input type="checkbox"/>
Attach File	<input type="checkbox"/>
Disable Pref	<input checked="" type="checkbox"/>
News Administrator	<input type="checkbox"/>
Comment Notification	<input type="checkbox"/>
Disable User	<input type="checkbox"/>
<input type="button" value="Create user"/> <input type="button" value="Reset"/>	

Figure 26: The configuration wizard shows the single user settings form for the "OWL Repository"

The form includes on the top a read-only part, and below a part with input fields. The information shown in the read-only part is fetched in the resource management portal database, and cannot be modified here. The other settings are specific to the "OWL Document Repository", and are explained in the Table 13.

Quota	Maximum number of bytes a user can have in the repository
Max Sessions	Maximum number of concurrent sessions a user can have
Receive Notification	Does the user want to receive email notification of new and updated files
Attach File	Notification will include the file OR a link to the file
Disable prefs	Users do not have access to their preferences (change password, notification etc.)
News Admin	Is this a news administrator
Comment Notification	User will receive notification when a comment is attached to the file

Table 13: The "OWL Document Repository" user account settings

5.5.2 Multiple User Account Configuration

The creation/configuration of multiple user accounts is similar to the case for single users. The resource administrator needs to select the items "Install accounts for all New Users...", or "Configure Account for all existing Users...". The first allows creating user accounts in the CITI for all the resource subscribers that do not have an account on it. The second enables to modify

the configuration of all active accounts in the CIT instance. If the CITI does not present any user accounts, the configuration will inform the resource administrator as we see in Figure 27.

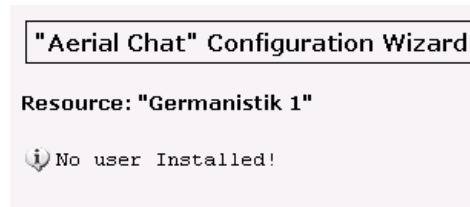
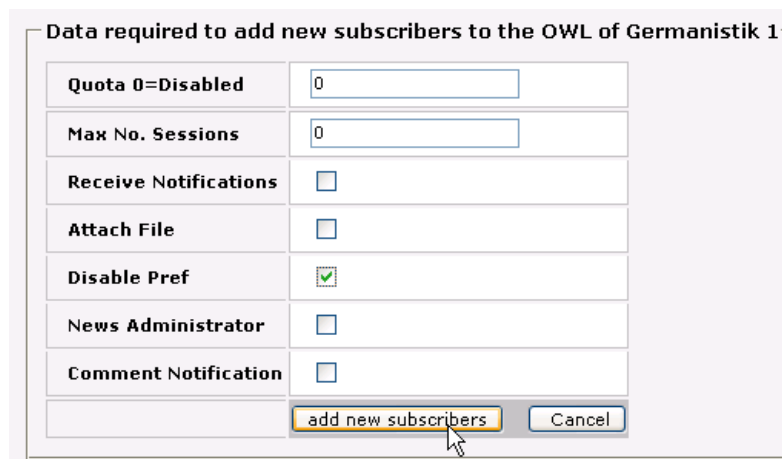


Figure 27: The configuration wizard cannot perform the configuration of multiple accounts

In the normal case, the configuration wizard displays a HTML form. The difference between this form and the form for single user accounts lies in the absence of a read-only part as depicted in Figure 26. Figure 28 shows the form in the configuration wizard (Creation of multiple accounts in the "OWL Repository"). The available options are described in Table 13.

The image shows a form titled 'Data required to add new subscribers to the OWL of Germanistik 1'. It contains several rows of settings:

Quota 0=Disabled	<input type="text" value="0"/>
Max No. Sessions	<input type="text" value="0"/>
Receive Notifications	<input type="checkbox"/>
Attach File	<input type="checkbox"/>
Disable Pref	<input checked="" type="checkbox"/>
News Administrator	<input type="checkbox"/>
Comment Notification	<input type="checkbox"/>

At the bottom of the form, there are two buttons: 'add new subscribers' and 'Cancel'. A mouse cursor is pointing at the 'add new subscribers' button.

Figure 28: The wizard shows the multiple user settings form for the "OWL Repository"

Figure 29 depicts the system output during the creation of new user accounts. The system displays the CIT instance name, the resource title, and each added user name.

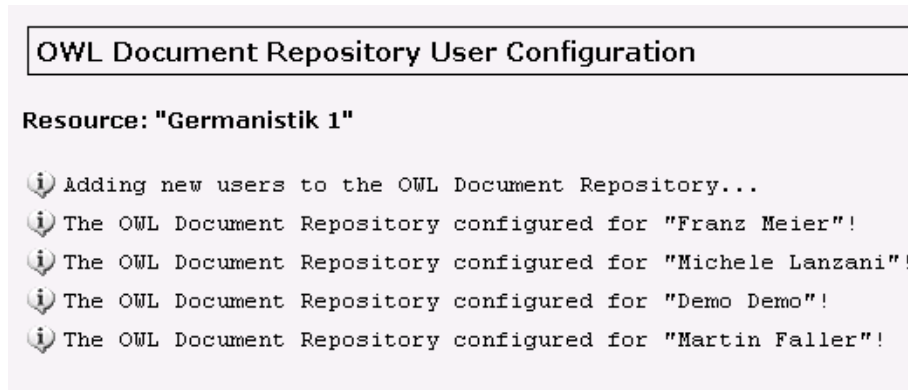


Figure 29: System output during the creation of multiple accounts

5.6 Access to a Community Interaction Tool

The resource subscriber is allowed to access the CITI for which he has an active account. In order to login into the CITI, he must select the resource in the *Community* section of the menu, and click on the "Login" button beside the desired community interaction tool. Figure 30 illustrates the PHP page which enables to access to the CITI of the resource called *Rechtsmedizin*.

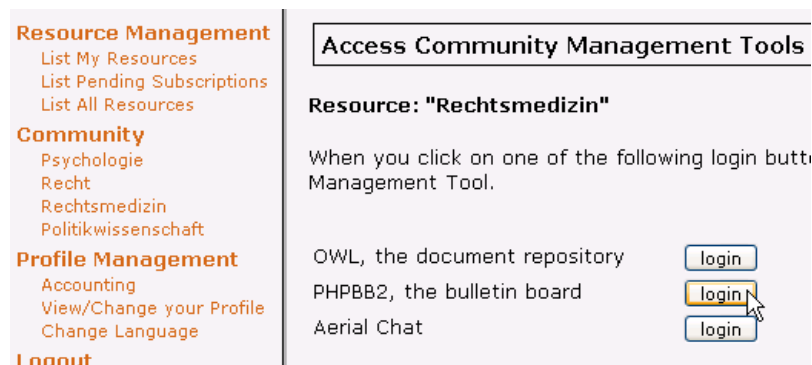


Figure 30: Community interaction tools access window for the resource management portal user

Sometimes the resource administrator does not need any community interaction tools for his resource, or he does not give the access rights to a resource subscriber. In these cases, the resource subscriber sees the warnings on his web browser like it is depicted in Figure 31 and Figure 32.



Figure 31: Warning: "No CIT available"

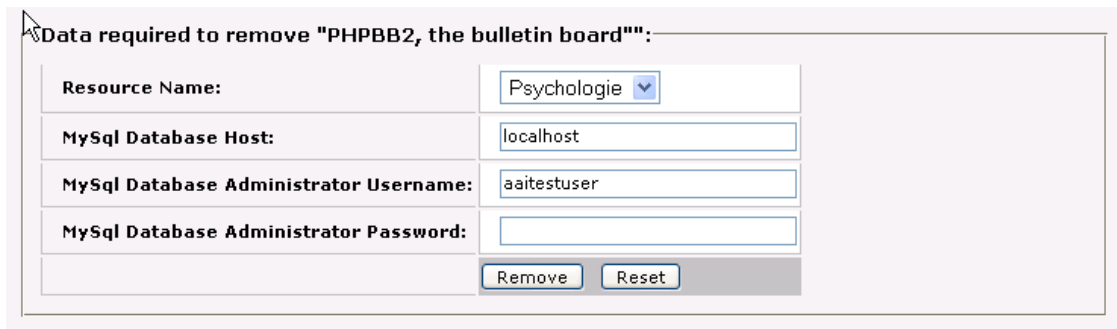
 WARNING: You don't have access to any existing Community Management Tool for "Geschichte der Medizin"!

Figure 32: Warning: "No account on any CIT available"

5.7 Uninstall of a Community Interaction Tool

The portal administrator can receive the request to uninstall a CITI. The resource management portal provides the uninstall wizard to perform this task. The procedure is similar to the installation routine. The portal administrator clicks on the menu item *Remove Tool* in the section *Community Management*, selects the kind of tool he wants to remove, fills the form presented by the uninstall wizard, and click on the *Remove* button.

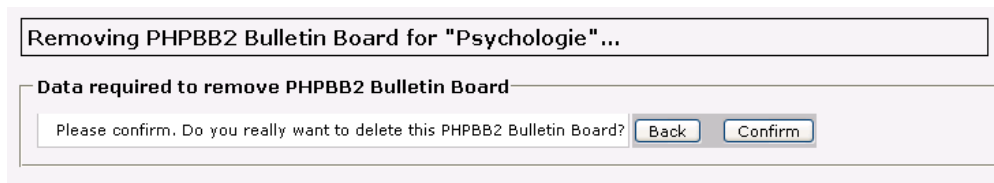
The form is filled following the criteria described in Section 5.4. Figure 33 depicts the uninstall wizard form for the "PHPBB Bulletin Board". The input fields are the same as for the installation wizard form.



The screenshot shows a web form titled "Data required to remove 'PHPBB2, the bulletin board'". It contains four input fields: "Resource Name" (a dropdown menu with "Psychologie" selected), "MySQL Database Host" (text input with "localhost"), "MySQL Database Administrator Username" (text input with "aaitestuser"), and "MySQL Database Administrator Password" (empty text input). At the bottom right of the form are two buttons: "Remove" and "Reset".

Figure 33: OWL uninstall wizard form

As soon as the form is sent to the resource management portal server, the system asks the portal administrator to confirm the deletion of the community interaction tool. For this purpose it displays a confirmation dialog. If the portal administrator is going to remove the wrong CITI, he can step back and choose the right one (see Figure 34).



The screenshot shows a confirmation dialog box with the title "Removing PHPBB2 Bulletin Board for 'Psychologie'...". Below the title is a section titled "Data required to remove PHPBB2 Bulletin Board". Inside this section, there is a text prompt: "Please confirm. Do you really want to delete this PHPBB2 Bulletin Board?". To the right of the prompt are two buttons: "Back" and "Confirm".

Figure 34: Confirmation dialog for the removal of a CIT instance

After confirmation the database backup and removal is done, the files of the CITI are tar archived, and deleted from the file system. Figure 35 shows the system output during the uninstall process.

```
Removing PHPBB2 Bulletin Board for "Psychologie"...

Database Backup...

Backup database "phpbb2t_resource_1"...
... done!
Drop database "phpbb2t_resource_1"...
... done!

Backup and remove files...

The resource "phpbb2t_resource_1" was archived and deleted!
The PHPBB2 Bulletin Board: removed successfully .
```

Figure 35: System output during the CITI uninstall process

5.8 Visualization of Accounting Data for the User

5.8.1 The Single Overview

The resource management portal user has access to the visualization of his accounting data. To view this information, he just needs to select the item *Accounting* in the section *Profile Management*. The displayed page shows the “single user view”. Figure 36 illustrates the top of this view. Here the user sees his name and the number of logins done.

```
Accounting

Personal Login Information of "Luca D'Alessandro":

The user "Luca D'Alessandro" logged in in the AAI Portal 10 times
```

Figure 36: Accounting visualization. The single user view displays the number of logins done

The middle of the single user view page, the resource management portal displays the statistic about the resource accesses, and the community interaction tools accesses (see Figure 37).

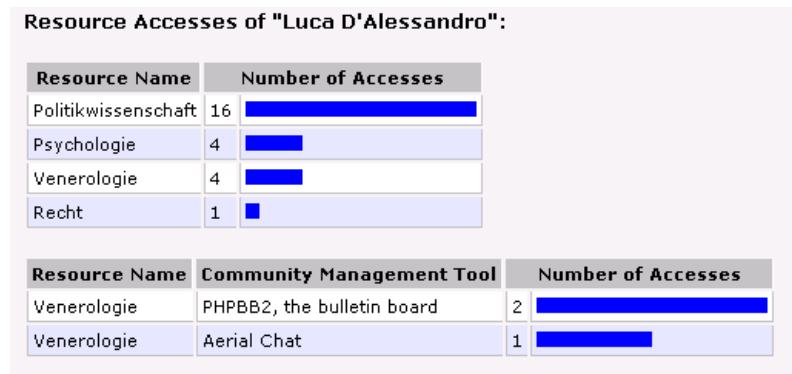


Figure 37: Accounting visualization. Resource and CIT statistics for the single user

While the access statistic is viewed in a ranking order (by number of accesses), the history in the last part of the single user overview shows every access to the resource management portal, any resource, and any CITI. Figure 38 depicts an example of the history list. The listed values are the data and time, the user name, the action, the accessed resource, and if the entry concerns a CITI, the name of the tool is also displayed.

Access History of "Luca D'Alessandro":

Date and Time	User	Action	Resource	Community Management Tool
08.04.2004 13:55:06	Luca D'Alessandro	Login	Portal	
10.04.2004 23:33:54	Luca D'Alessandro	Login	Portal	
10.04.2004 23:35:53	Luca D'Alessandro	Login	Portal	
10.04.2004 23:35:58	Luca D'Alessandro	AccessResource	Venerologie	
10.04.2004 23:40:44	Luca D'Alessandro	AccessResource	Venerologie	
10.04.2004 23:49:18	Luca D'Alessandro	Login	Portal	
10.04.2004 23:49:24	Luca D'Alessandro	AccessCmt	Venerologie	PHPBB2, the bulletin board
11.04.2004 05:11:15	Luca D'Alessandro	Login	Portal	

Figure 38: Accounting visualization. Single user history

5.9 Visualize Accounting Data for the Administrators

Even if the portal administrator and the resource administrator have access to the same menu items, and the same views for the visualization of the accounting data, the resource administrator has not access to all the data. According to the rules of *Authorization and Confidentiality* (see Section 3.2.4), the resource administrator has only access to the data concerning his resource users.

5.9.1 The Users' Overview

This overview allows visualizing in one sight all the users' data. This permits to get a first impression about the resource management portal users. The information is displayed in a table containing the user name, the login name, the date and time of the last login, the number of logins done, and the activity rate. The table is ordered by login name. As explained in Section 3.2.5, the

activity rate is the number of activities of one user stored in the resource management portal database. Beside the login number and activity rate, the table shows a histogram which enables the administrator to understand the users' behavior in one view.

Figure 39 illustrates the users' overview table. The user names are displayed as hyperlinks. Clicking on them enables the administrator to get a single user view (see Section 5.8.1).

Users Accounting						
Overview:						
User Name	User Login Name	Last Login	Number of Logins		Activity Rate	
Martin Faller	29325@aaitest.switch.ch	08.04.2004 13:40:15	2		46	
Calogero Butera	calogero	21.04.2004 23:30:16	28		90	
David Vogel	dave	21.04.2004 23:28:27	7		28	
Demo Demo	demostudent@unibe.ch	10.04.2004 23:34:55	5		12	
Franz Meier	franz	21.04.2004 21:56:23	15		62	
Franziska Ingold	franziska	09.04.2004 17:05:15	1		2	
Fritz Müller	fritz	21.04.2004 23:30:31	8		27	
Luca D'Alessandro	luca	21.04.2004 23:42:26	10		66	
Michele Lanzani	michele	16.04.2004 02:35:21	2		8	
Philippe Protasevicz	pippo	never	0		0	
James Kelso	portaladmin	22.04.2004 00:06:45	46		190	
Markus Jägermeister	resourceadmin	21.04.2004 23:19:15	20		93	

Figure 39: Accounting visualization. The complete users' overview

5.9.2 The Resources' Overview

The resources' overview allows to administrators to see a ranking of the accessed resources. This sight consists in a table with the resource names, the resource provider (the name of its resource administrator), and the number of accesses to the resource. The last is provided with a histogram which helps the observer to get an impression about the resource usage state. Figure 40 depicts the resources' overview. The resource names and the provider names are displayed as hyperlinks. This allows one to get more detailed information about the selected item.

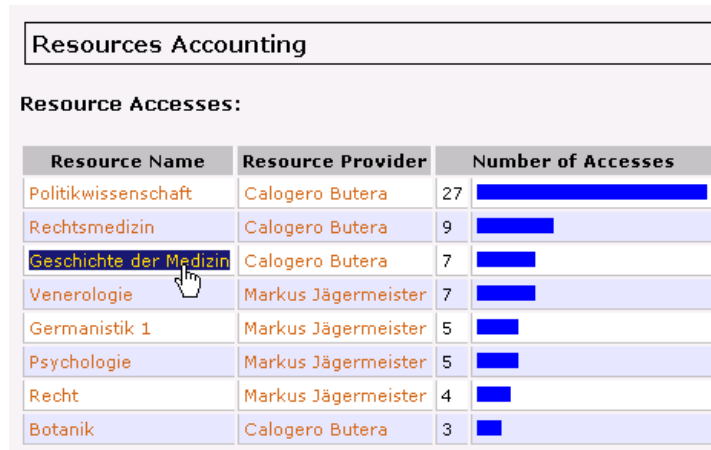


Figure 40: Accounting visualization. The complete resources' overview

Figure 41 depicts the single resource view for the resource named *Politikwissenschaft*. The table contains the ranking of the resource subscribers with user name, and number of accesses. The administrator can select a user, and get its single user overview.

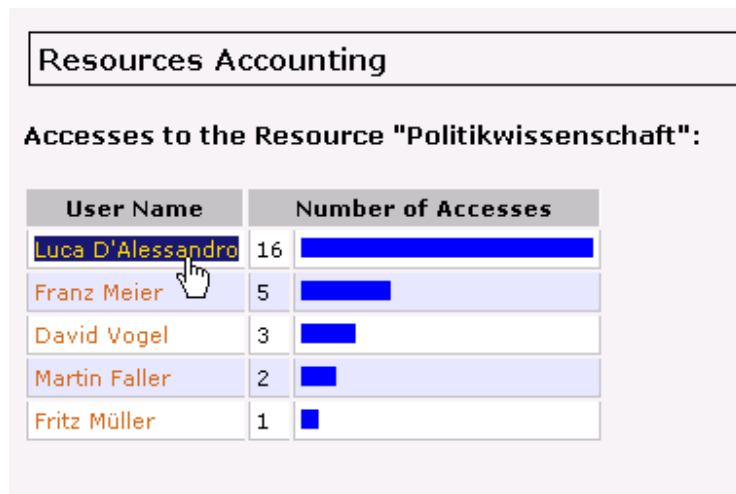


Figure 41: Accounting visualization. The single resource view

5.9.3 The Community Interaction Tools' Overview

The statistics about the community interaction tools access can be viewed in the CIT overview. The displayed table shows the resource name, the resource provider, the kind of CIT, and the number of accesses. The identification of a CIT instance necessitates two values: the resource ID and the CIT module name. This is the reason why, the resource name and the CIT module name are both displayed in the CIT overview (we assume that the resource name is unique). Furthermore, the table is ordered by number of accesses, and a histogram helps the better visualization.

Figure 42 illustrates the community interaction tools' overview. The resource names and the resource provider names are displayed as hyperlinks in order to enable the sight of the details about the community interaction tool instance, or the resource provider.

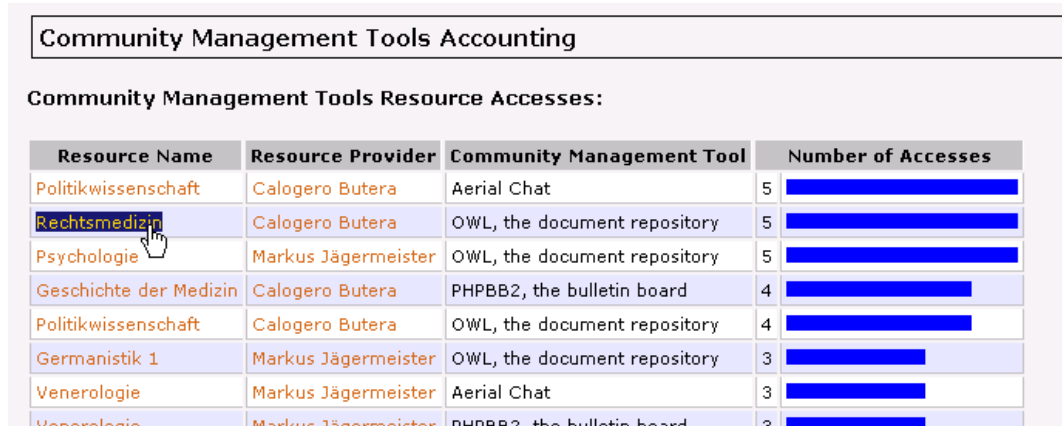


Figure 42: Accounting visualization. The community interaction tools' overview

The detailed view of the CITI is depicted in Figure 43. The user names and the number of accesses to the CITI are presented in a table with a histogram. The table is ordered by number of accesses, and the histogram helps the administrator to get a sight of the CITI state.

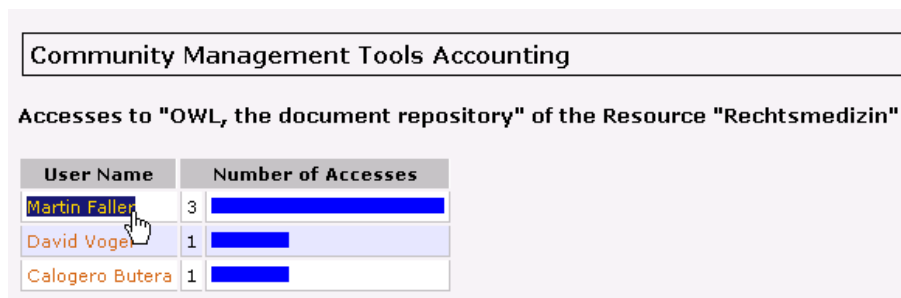


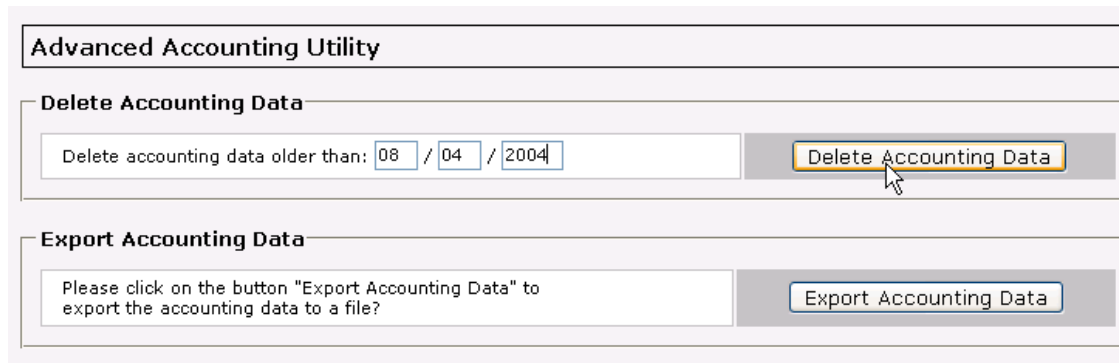
Figure 43: Accounting visualization. The single community interaction tool overview

5.10 Accounting Data Management

The accounting support for the resource management portal includes two advanced features which are related to the management of the available accounting data. The resource administrator is only allowed to export the accounting data. The portal administrator disposes of an additional tool which permits the removal of old data in the *Accounting* table of the resource management portal database.

5.10.1 Removal of Accounting Information

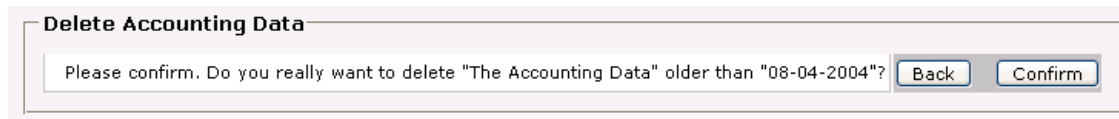
The graphical user interfaces for the deletion of accounting data are illustrated in what follows. Figure 44 shows the advanced accounting utility panel for the portal administrator. The user needs to fill the form with a date, and click on the *Delete Accounting Data* button in order to delete the accounting data older than the given date. The resource management portal displays then a confirmation dialog. This is a security feature, since the deleted data cannot be restored anymore.



The screenshot shows a web interface titled "Advanced Accounting Utility". It contains two main sections. The first section, "Delete Accounting Data", has a text input field with the value "08 / 04 / 2004" and a button labeled "Delete Accounting Data". A mouse cursor is pointing at this button. The second section, "Export Accounting Data", has a text input field with the text "Please click on the button 'Export Accounting Data' to export the accounting data to a file?" and a button labeled "Export Accounting Data".

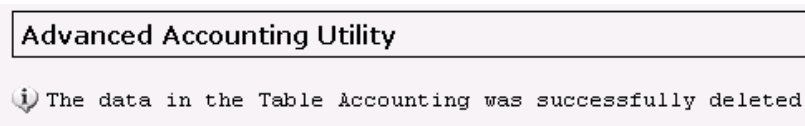
Figure 44: Accounting - advanced data management tools

After the confirmation by the user, the system displays the feedback about the successful deletion of the accounting data (see Figure 45 and Figure 46).



The screenshot shows a confirmation dialog titled "Delete Accounting Data". It contains a text input field with the text "Please confirm. Do you really want to delete 'The Accounting Data' older than '08-04-2004'?" and two buttons: "Back" and "Confirm".

Figure 45: Deleting the accounting data. The confirmation dialog



The screenshot shows a web interface titled "Advanced Accounting Utility". It contains a message box with a downward arrow icon and the text "The data in the Table Accounting was successfully deleted".

Figure 46: Delete accounting data. System output

5.10.2 Accounting Data Export

The export of accounting data allows the administrators to import these data into statistic software, in order to evaluate the information about all the activities performed by the users. Table 5 in Section 3.2.2 presents the complete list of logged activities by the accounting system. The data are exported to a text file in comma separated format (CSV) as illustrated on Figure 47. This format allows getting file which is compatible to any statistic software.


```
accounting_userid;accounting_userprivilege_id;accounting_time;account
calogero;2;20040418025936;130.92.65.4;Logout;User;calogero;;;;;
calogero;2;20040418025901;130.92.65.4;Login;User;calogero;;;;;
franz;1;20040418025856;130.92.65.4;Logout;User;franz;;;;;
franz;1;20040418025846;130.92.65.4;AccessCmt;Cmt;OWL;Resource;2;;;
franz;1;20040418025840;130.92.65.4;AccessCmt;Cmt;AECHAT;Resource;3;;;
franz;1;20040418025832;130.92.65.4;AccessCmt;Cmt;PHPBB2;Resource;4;;;
franz;1;20040418025826;130.92.65.4;AccessCmt;Cmt;AECHAT;Resource;6;;;
franz;1;20040418025820;130.92.65.4;AccessCmt;Cmt;AECHAT;Resource;26;;;
franz;1;20040418025817;130.92.65.4;AccessResource;Resource;6;;;;;
franz;1;20040418025816;130.92.65.4;AccessResource;Resource;6;;;;;
```

Figure 47: The exported accounting data

6. Discussion and Conclusion

6.1 Discussion

In the context of this diploma thesis we developed configurable community interaction tools and accounting mechanisms for the existing resource management portal prototype implementation version 0.9, starting with an analysis of the user requirements, going on with the elaboration of the concepts and ending with the implementation of the developed code into the resource management portal prototype.

At the beginning of this thesis we studied the authentication and authorization infrastructures with a particular attention to Shibboleth which is the authentication and authorization infrastructure deployed by most Swiss universities.

We installed a Linux server with Shibboleth and the resource management portal, and studied the portal architecture starting from the existing technical specification of the portal. These activities allowed us inspecting and analyzing Shibboleth and the resource management portal which was fundamental for the further work which had to be done.

The research about virtual communities, the existing community interaction tools, the accounting concepts, and the available accounting technologies helped us to collect the needed know-how for the integration of this community interaction and accounting in the portal.

The resource management portal was implemented in PHP and used a MySQL database engine. In order to work on the portal we learned these technologies.

Once we had the required know-how about the resource management portal technologies, we were able to analyze and develop the architecture for the integration in the portal of community interaction and accounting support.

The analysis of a potential community present on the portal showed requirements for communicative interactions between the users of the resources hosted on the portal. These interactions should lie in direct communication as well as in indirect communication and file exchange.

The resource management portal had a dynamic character, and the resource administrators were able to manage the users and the resources. This behavior led us to the concept of single instanced based installation. Using this concept the resource administrator was able to decide which community interaction tool should be applied to which resources.

The requirements regarding accounting lie in the possibility to statistically acquire all possible user actions on the portal and to visualize the respective usage data to the users. We have discovered the additional need to export accounting data for a further usage such as billing.

After evaluating accurately the available accounting tools, we decided to create an own accounting infrastructure into the resource management portal. This architectural decision was made, in order to be independent from external accounting software, and to be able to collect the data there where it is required.

The elaborated concept for the community interaction tools consists of modular pluggable specially adapted resources. These resources are directly configured by portal administrators. Portal users may use these tools in the same roles they possess on the portal.

Three open source resources have prototypically been implemented:

- The “OWL Document Repository (kn**OW**Ledgebase)”. It is a multi user document repository (knowledgebase) system written in PHP for publishing files/documents onto the web for a corporation, small business, or group of people.
- The “PHPBB 2 Bulletin Board”. It is a high powered, fully scalable, and highly customizable open-source bulletin board package.
- The “Aerial Chat”. It is a chat application implemented in PHP which supports, unlimited channels, private messaging, music and graphics effects.

The implemented utilities consist in an installation, a configuration, an uninstall, and an access utility. The first three make use of the wizard concept. During the uninstall process the corresponding utility creates a backup of the file system and the database of the community interaction tool instance. The support of these modules bases on the implemented framework which also provides the basic functionality for the migration of existing modules to new tool application releases, and the integration of new modules.

The concept for the accounting consists of a data gathering mechanism which is integrated into the portal. This mechanism allows acquiring all user actions on the portal, and displaying selected data to each user depending on the respective user role.

All the implemented features are configurable with enhanced graphical user interfaces of the resource management portal. Furthermore, the data representation is supported by the use of histograms.

The export utility allows the portal administrator to export all the data present in the database, whereas the resource administrators can only export data concerning their resource subscribers. The portal user is not allowed to export data. Accounting data may be exported as comma separated value files. The data management utility provides also the removal of old data. This task can only be performed by the portal administrator.

6.2 Conclusion

During this diploma thesis we learned why authentication and authorization infrastructures are useful and how their mechanisms work. The issue of simplifying the access to the online resources – shared by several institutions – without creating a big overhead had to be solved. The deployment of an authentication and authorization infrastructure solves this problem.

The resource management portal is the broker between authentication and authorization infrastructure, and the resources. It enables the institutions to adapt their resources to an authentication and authorization infrastructure in a simple manner.

During the thesis we learned the programming language PHP and the database engine MySQL. This was a further enrichment of our know-how. In fact PHP is a powerful language which permits to develop dynamic web applications. MySQL is a powerful database which is easy to manage and offers a high performance.

The implemented concept of the modularized and single-instance-based community interaction tools permits a flexible usage and integration of these tools.

The web-based installation procedure – which makes use of a wizard – enables also unskilled administrators to profit from the pluggable community interaction resources. Students and resource administrators take advantage of the added functionalities and the possibilities to use missing interaction tools of their e-learning resources.

The three implemented tools demonstrated that the developed concept works in practice. The developed software fulfills completely the initial requirements on the community interaction tool integration.

Furthermore the community interaction tool framework allows adapting the modules to new tool releases with very low overhead. This means if there is new release of a community interaction tool, and we want to adapt the module containing this tool, the framework offers functionalities to perform this action. Furthermore, the framework permits a simplified implementation of new community interaction tool modules.

The implemented accounting routines acquire all user actions and display the selected data graphically to the requesting user. The accounting data allow the administrators to get an impression about the state of the portal, of the resources, and of the community interaction tools. In addition, the users' behavior can be observed.

The export of the data permits the administrators to evaluate the complete accounting activities on external statistical software packages such as MS Excel [37], or SPSS [38]. The data removal feature permits to keep the database size under control.

7. Outlook

During the work on the community interaction tools and the accounting system for the resource management portal, we discovered missing features which could contribute to a higher usage quality on the portal.

It is desirable that further community interaction tool modules get integrated to provide a broad variety of tools which could suit the needs of the resource administrators. The community integration tool framework supports only Linux specific file system commands.

During the uninstall process of the community interaction tools the system makes a backup of the file system, and the database. The recovery of such a tool has to be made manually. It is conceivable to create a recovery tool.

The developed framework for accounting permits a future modification of the gathering strategy, although it could provide more functionality for data analysis which was not a requested feature.

The integration of a configuration tool could allow selecting the parameters which should be logged. The visualization tool could be configurable too. Therefore, the administrators could choose which parameter to see.

If the resource management portal is enhanced with additional functionality it could be interesting to gather information about the new activities.

The creation of a billing strategy based on the accounting data is also desirable. The deployment of any infrastructure bears costs, therefore selling the services provided by the resource management portal would provide to the operator an increased cost recovery.

Appendix

List of Figures

Figure 1: Shibboleth Architecture.....	7
Figure 2: The resource management portal at the resource owner's site.....	9
Figure 3: The functional parts of the resource management portal	11
Figure 4: Web front-end structure.....	12
Figure 5: Resources and Communities	15
Figure 6: Synchronous tools vs. asynchronous tools	16
Figure 7: Proprietary Accounting	20
Figure 8: Single instance based installation of community interaction tools	22
Figure 9: Configuration utility. The user account configuration in CIT instance	24
Figure 10: Community interaction tool support.....	26
Figure 11: Proposed accounting scheme.....	29
Figure 12: Resource management portal database - "CommunityInteraction" -new table definition.....	33
Figure 13: Flow chart of an installation wizard	34
Figure 14: Community interaction tools installation process flow - OWL example	37
Figure 15: Configuration Wizard - Flow Diagram.....	38
Figure 16: The resource management portal database - "Accounting" -new table definition.....	39
Figure 17: The resource management portal page - Overview.....	47
Figure 18: New resource management portal menu for the portal administrator	48
Figure 19: New resource management portal menu for the resource administrator.....	48
Figure 20: New resource management portal menu for the user	49
Figure 21: OWL installation wizard form	49
Figure 22: OWL document repository installtion. The files, and database operations....	50
Figure 23: Document repository installation completed. Account creation for the administrators.....	50
Figure 24: Configuration wizard for the "Aerial Chat" application	51
Figure 25: Overview Table for the management of single user accounts in the CIT instance	51
Figure 26: The configuration wizard shows the single user settings form for the "OWL Repository"	52
Figure 27: The configuration wizard cannot perform the configuration of multiple accounts.....	53
Figure 28: The wizard shows the multiple user settings form for the "OWL Repository"	53
Figure 29: System output during the creation of multiple accounts	54
Figure 30: Community interaction tools access window for the resource management portal user.....	54
Figure 31: Warning: "No CIT available".....	54
Figure 32: Warning: "No account on any CIT available"	55
Figure 33: OWL uninstall wizard form	55

Figure 34: Confirmation dialog for the removal of a CIT instance	55
Figure 35: System output during the CITI uninstall process	56
Figure 36: Accounting visualization. The single user view displays the number of logins done.....	56
Figure 37: Accounting visualization. Resource and CIT statistics for the single user	57
Figure 38: Accounting visualization. Single user history	57
Figure 39: Accounting visualization. The complete users' overview.....	58
Figure 40: Accounting visualization. The complete resources' overview	59
Figure 41: Accounting visualization. The single resource view.....	59
Figure 42: Accounting visualization. The community interaction tools' overview	60
Figure 43: Accounting visualization. The single community interaction tool overview..	60
Figure 44: Accounting - advanced data management tools	61
Figure 45: Deleting the accounting data. The confirmation dialog	61
Figure 46: Delete accounting data. System output	61
Figure 47: The exported accounting data.....	62

List of Tables

Table 1: The resource management and user management activities on the portal.....	10
Table 2: The authentication and authorization portal database tables	13
Table 3: Resource management portal and a university	18
Table 4: Tasks, responsibilities, and rights of the resource management portal users	25
Table 5: Accounting Activities on the resource management portal	28
Table 6: OWL Document Repository Features	31
Table 7: PHPBB Bulletin Board – Features	32
Table 8: Aerial Chat - Features.....	32
Table 9: The resource management portal database. The “CommunityInteraction” table	33
Table 10: The resource management portal database. The items in the table Accounting	40
Table 11: Accounting Visualization Concept. All Users Overview	41
Table 12: Accounting Visualization Concept. Single resource view for resource A	41
Table 13: The "OWL Document Repository" user account settings	52

References

- [1] Christoph Graf, Ueli Kienholz, Thomas Lenggenhager. *AAI - Authentication and Authorization Infrastructure, Results of the Pilot Phase*. 2004
- [2] Swiss Virtual Campus
<http://www.swissvirtualcampus.ch/>
- [3] Marc-Alain Steinemann, Thomas Spreng, Aljoscha Bachmayer, Calogero Butera, Torsten Braun, Christoph Graf, Martin Guggisberg. *Authentication and Authorization Infrastructure: Portal Architecture and Prototype Implementation*. 2003
- [4] Castro-Rojo R., López D.R.: *The PAPI System: Point of Access to Providers of Information*. Terena. 2001
- [5] Shibboleth Internet2 Middleware.
<http://shibboleth.internet2.edu/>
- [6] Oasis: Organization for the Advancement of Structured Information Standards.
<http://www.oasisopen.org/>
- [7] Security Assertion Markup Language (SAML) is an OASIS standard.
<http://www.oasis-open.org>
- [8] Graf Christoph et al.: *Architecture Evaluation*. SWITCH. Jan 20, 2003.
<http://www.switch.ch/aa/>
- [9] Steinemann M.-A. et al.: *Global Architecture and Partial Prototype Implementation for Enhanced Remote Courses, Computers and Advanced Technology in Education (CATE 2002)*, Cancun, Mexico, May 20-22, 2002, ISBN 0-88986-332-6, pp. 441-446
- [10] Virtual Internet and Telecommunications Laboratory of Switzerland.
<http://www.vitels.ch/>
- [11] Guggisberg M. et al.: *An Interdisciplinary Virtual Laboratory on Nanoscience. Electronic Notes in Future Generation Computer Systems*, Elsevier, Vol. 1 (2001)
- [12] Web Course Tool:
http://archives.internet2.edu/guest/archives/I2-NEWS/log_200212/msg00002.html
- [13] GNU GENERAL PUBLIC LICENSE
<http://www.gnu.org/copyleft/gpl.html>
- [14] PHP Hypertext Preprocessor
<http://www.php.net>

- [15] The Apache HTTP Server Project
<http://httpd.apache.org/>
- [16] MySQL an Open Source Database
<http://www.mysql.com/>
- [17] Redhat Linux
<http://www.redhat.com>
- [18] Debian GNU/Linux
<http://www.debian.org>
- [19] I18N *Character Model for the World Wide Web 1.0. W3C Working Draft* 2002.
<http://www.w3.org/TR/2002/WD-charmod-20020430/>
- [20] Howard Rheingold. *The Virtual Community*. MIT Press; 1st edition (Nov. 2000)
- [21] Rasmussen, G., & Skinner, E. *Learning Communities: Getting Started (ED433048)*. (1997)
- [22] Lynn Kirby. *T.E.A.M. Learning Community*. University of Wisconsin-Stevens Point
<http://www.uwsp.edu/education/lkirby/lc/>
- [23] AAA – Definition by the Webopedia Computer Dictionary
<http://www.pcwebopedia.com/TERM/A/AAA.html>
- [24] TACACS. Terminal Access Controller Access Control System [RFC 1492]
<http://www.ietf.org/rfc/rfc1492.txt?number=1492>
- [25] The Internet NG Project
<http://ing.ctit.utwente.nl/WU5/D5.1/Technology/tacacs+/>
- [26] RADIUS. Remote Authentication Dial-In User Service [RFC 2865]
<http://www.ietf.org/rfc/rfc2865.txt>
- [27] Accounting. Definition by the Webopedia Computer Dictionary
<http://wi-fiplanet.webopedia.com/TERM/A/accounting.html>
- [28] D. Carrel, L. Grant. *"The TACACS+ Protocol Version 1.78", Internet draft (work in progress), draft-grant-tacacs-02.txt*, Cisco Systems, January, 1997.
- [29] Module mod_log_config. Apache HTTP Server Version 1.3
http://httpd.apache.org/docs/mod/mod_log_config.html
- [30] OWL Intranet Engine
<http://owl.sourceforge.net/>
- [31] phpBB.com :: Creating Communities
<http://www.phpbb.com/>
- [32] Aerial Chat – Sourceforge
<http://sourceforge.net/projects/aerialchat/>

- [33] Regular Expression – Definition – Whatis.com
http://whatis.techtarget.com/definition/0,,sid9_gci517740,00.html
- [34] url-path - Uniform Resource Locators (URL) - 3.1. Common Internet Scheme Syntax
<http://www.w3.org/Addressing/rfc1738.txt>
- [35] The AAI Portal – an open source project of SWITCH AAI
<http://sourceforge.net/projects/aai-portal/>
- [36] Scott Cantor, Marlena Erdos. “Shibboleth-Architecture DRAFT v05” , 2002
- [37] Microsoft Excel, Software Package which enables to analyze data.
<http://www.microsoft.com/Office/Excel/prodinfo/default.msp>
- [38] SPSS Statistical Software
<http://www.spss.com/>
- [39] SWITCH, The Swiss Education and Research Network
<http://www.switch.ch>
- [40] Microsoft Windows Operating System
<http://www.microsoft.com/windows/default.msp>

