W. Prinz, M Jarke, Y. Rogers, K. Schmidt, and V Wulf (eds), *Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work, 16-20 September 2001, Bonn, Germany*, pp 319-338

Community Support and Identity Management

Michael Koch, Wolfgang Wörndl
Technische Universität Munchen, Germany
{kochm,woerndl}@in tum de

Abstract. Computer based community support systems can provide powerful support in direct exchange of information and in finding people for information exchange Such applications usually make use of information about the user (user profile information) for personalization and for supporting contact management As in real life, a user will interact with different communities (community support applications) hosted by different providers With the current approach users have to provide and update information about their identity and interests for each community independently That results in cold-start problems with new community support applications and in inconvenience for the user In this paper we discuss user-centric identity management for community support applications and concentrate on a platform for using user profiles in more than one application We also propose mechanisms to address privacy issues in this framework

1. Introduction

Community support and virtual communities gain more and more attention in various areas from marketing to knowledge management. In this context community support includes a large variety of functions supporting groups of people with some kind of commonality and an ongoing rhythm of interaction (Mynatt et al., 1997). Identity plays a key role in virtual communities. In communication, which is the primary activity in communities, knowing the identity of those with whom you communicate is essential for understanding and evaluating an interaction (Donath, 1998) In addition to this, a community support application could offer special personalized services.

The Webster English Dictionary describes the word *identity* as. "*1) the condition or fact of being the same or exactly alike (sameness, oneness), 2a) the condition or fact of being a specific person or thing (individuality), b) the condition of being the same as a person or thing described or claimed*" (Webster, 1988).

For community support applications the aspect of identity as proving to be a specific person is not as important as the aspect of identity as all information that describes a specific person in the real world Hence, we regard identity more in the context of user profile, a set of information representing a user or clearly related to a user or role in the digital world In the rest of the paper we will use the terms user profile and identity synonymous

Managing which information is available for which application is called *identity management* Identity management is something we do in normal conversation everyday when we decide on what to tell one another about ourselves In interactions with others we consider the situational context and the role we are currently acting in as well as the respective relationship with the interaction partners This results in different sets of information being released to different interaction partners Sometimes this leads to the situation that a person is known under different names in different contexts, e.g. by using special names, nicknames or pseudonyms suiting the occasion (Köhntopp and Bertold, 2000)

Also or especially in the digital world people are using different (digital) identities. When interacting with different applications from different providers and using different identities it becomes hard to keep track of the information which service stores which information, and to keep the information in the services up to date

An identity management system would allow people to define different identities, roles, associate personal data to it, and decide whom to give data and when to act anonymously An identity management system would empower the user to maintain their privacy and control their digital identity. For community support systems a user-centric identity management system would make it easy for the user to use different communities and thereby lower the entry barrier to online communities.

In this paper we tackle the technical aspect of user-centric identity management After reviewing requirements for identity management systems (Section 2) we will present a general architecture for identity management that is developed in the context of the Cobricks project at Technische Universitat München (Section 3). Then we will focus on the privacy issues in this architecture (Section 4). The paper concludes with a brief overview of the implementation status and prototypes (Section 5), some comments on related work (Section 6), and a summary and look-out to future work (Section 7).

2. Identity Management Requirements

2.1. Functionality

One of the main motivations for identity management is to enable different services reuse user profile information The user of an online service should no longer have to enter a lot of data for registration or wait a long time until the application has learned his preferences and can provide properly personalized services Therefore, identity management support first has to provide functions for creating, storing and accessing digital identities (user profiles). Here it has to be stressed that in contrast to different kinds of customer relationship management systems, identity management has to clearly treat the user providing information as the owner of the profile and not the services using the information.

Identities are created by a user for himself or by a certification authority for the user, and are accessed by services the user is interacting with or has interacted with (e.g. community applications or e-commerce services). For enabling services to access identity information the user has to provide the service with an identity identifier. This identifier could either be transmitted automatically by the user's browser or be entered by the user upon registration or login.

In addition to reading data from the identity, services should also be able (with permission) to add new information to the identity. This can relieve the user from entering all information himself.

Usages for digital identities are

- Authentification – the identity provides a way for the service to authenticate the user (e.g. password checking function or public key certificates).
- Providing data for different functionalities like
  - (e-commerce) transactions (e g delivery address, payment data),
  - personalization, or
  - configuration/initialisation (e g. email address for sending push information, buddy lists for configuring awareness and communication functions)

For authentification the identity management support has to provide functions that enable a service to check the identity of a user who is directly interacting with it.

Access to the identity has to be logged to provide the owner of the identity with reports about which service possesses which personal data

The owner of an identity should have the possibility to determine which interaction partner should see what information. This could be done by creating different digital identities or by defining and negotiating special access rights to one identity In our opinion both solutions have to be provided by an identity management system since it is not possible to foresee every usage of the information

by defining a special sub-identity for it. A connection between the two solutions and a functionality to help users managing several identities can be introduced by providing the possibility to link identities to each other. Such a link could define a data path that determines how updates to one identity are automatically forwarded to another identity

2.2. Privacy

As already mentioned, the identity management system has to enable the identity owner to specify which services can access and write which data. This is one aspect of privacy (Information) privacy refers to the claims of individuals that information about themselves should generally not be available to other individuals or organizations, and that, where data is possessed by another party, the individual must be able to exercise a substantial degree of control over that data and its use (Clarke, 1999)

Surveys among Internet users show that they are concerned about their privacy. For example, 87% of the participants in a survey conducted by Ackerman et al (1999) were somewhat or very concerned about threats to their personal privacy while being online Respondents were less inclined to provide information when personally identifiable information was requested·

"In a scenario involving a banking Web site, 58% of respondents said that they would provide information about their income, investments, and investment goals in order to receive customized investment advice. However only 35% said they would also supply their name and address so that they could receive an investment guide booklet by mail " (Ackerman et al., 1999, p 5).

There is an obvious need for mechanisms allowing users to specify and enforce their personal preferences regarding privacy. More precisely, requirements for a privacy infrastructure in our scenario are.

- Flexible access right control system, e g through rules and negotiation
- Possibility to monitor access rights and accesses
- Possibility to use a pseudonym instead of real identity
- Purpose binding of data accesses
- Possibility to allow access for temporary use
- Possibility to revoke granted access rights
- Control whether user data can distributed to other services (and users)
- Integration of cryptographic techniques for anonymous data transfers
- Possibility of support from privacy authorities

Online services and businesses could also benefit from a powerful privacy architecture because users with less fear of risking their privacy are likely to make more and better personal information available to services (Köhntopp and Bertold, 2000) Our ideas for a technical solution to ensure privacy in this framework are presented in Section 4

3. Identity Management Architecture

To introduce a user-centric identity management system one has to tackle the following issues.

1) infrastructure/architecture of the system
2) standards for representing and storing user profile information (for making multiple usage and exchange among applications possible)
3) cryptographic means for storing and authenticating identities and pseudonyms
4) privacy issues (this includes negotiation and the possibility to use more than one identity)

In this section we will mainly discuss architectural issues (Issues 1 and 2) The following section (Section 4) will address the privacy issues.

3.1. Server- versus client-side storage of user profiles

Today, user related information is stored on different servers For example, amazon com or bn.com keep track of users' interests to make recommendations based on previous transactions or interest specifications users have made available. However, there are several problems with this server-side approach:

- A user has to log on to different community applications manually and provide his profile information again and again; there is no possibility to distribute new information to different communities (i.e. community applications) in one step
- Information gained by one community cannot be used by another
- Privacy considerations· users have little or no control over what personal information is collected

Alternatively, user profile information can be stored on the user's computer. This could lead to higher trust because personal information is located near the user and because the usage of profile information can be controlled and monitored. The user information can be reused for several communities Client-side user profile storage is usually implemented by so-called *infomediaries* Infomediaries are (small) applications on the client computer, which maintain user profiles and offer services such as automatic fill-in of Web forms (Cranor, 1999).

The main problem of client-side storage of user profile information is that it is not portable (Mulligan and Schwartz, 2000). Personal information stored on one computer (e g at work) cannot be easily transferred to another one (e g. at home) Therefore, a promising solution might be a *ID-Repository.* It offers server-based storage of user information for different services under the control of the user A network of repository servers operated by different companies might be the best approach because it adds scalability and allows the user to choose one or more server operators to store her personal data

3.2. User Profile Repository Network

Our technical approach is to separate user profile information from services that make use of it and store it in a central place where it can be maintained by the user and be accessed by different services (with permission of the identity owner).

The core component in our architecture is a user profile repository service (ID-Repository) that stores information about an identity and offers the identity owner and authorized services interfaces to access this information (see Figure 1)

The server offers a functionality to store more than one identity and to link identities to each other (defining data propagation paths)

For the repository we have several possibilities placed between the following two extremes.

- one central identity server for storing all identities of all people
- one or even several servers per person storing different identities

We imagine that in the real world there will be identity providers – services that operate servers (see Figure 2). These services might also offer certification services for profile information.

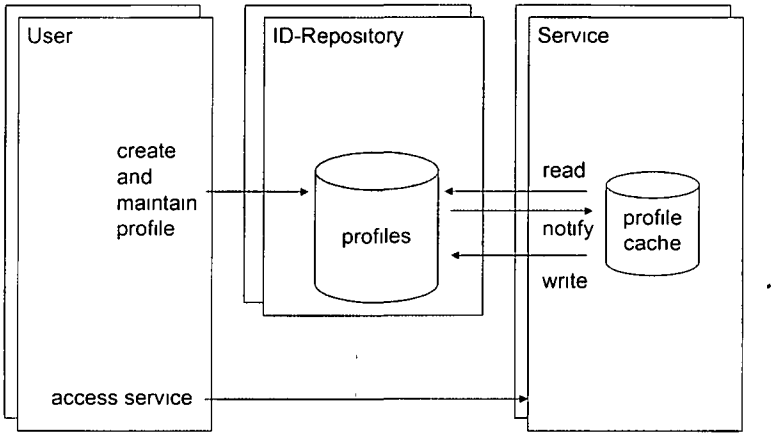

Figure 1. Identity Management Architecture Components.

For accessing the profile information we need a means for name resolution. Here we have to answer the question how a service can find the user profile server? There are two major possibilities for that issue. First we could use a service dependent identity identifier that includes the contact address of the identity repository server Since it should be possible to move identities between servers this solution is not flexible enough The only reasonable alternative is to introduce a name service or broker that is presented with the identifier and forwards requests to the correct server or returns the server's name or address

The services that read the profile information should have a possibility to cache this information for some time Here we need a means for keeping the cache up to date (and for the user to request deletion of the cached copy). After the negotiation of the basic lease this whole process can be seen as replication of the data with a master copy.

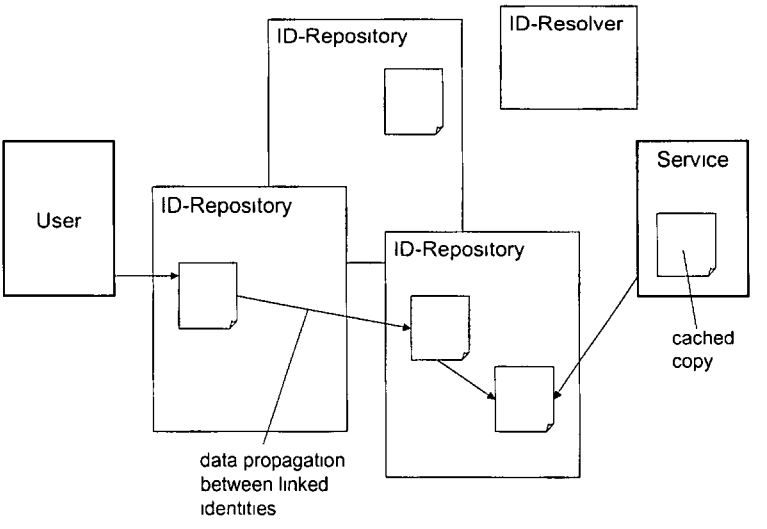

Figure 2. Identity Management Network

3.3. Profile schema

Now that we outlined the general architecture for storing and accessing profiles, there is the question of how a profile should be structured to be of general use and allow interoperability.

Some information standards have been defined in the past for user profile information. Examples are the vCARD standard (Howes et al., 1998) or the standard included in W3Cs P3P specification (P3P, 2000). These approaches mainly choose hierarchically structured sets of attribute value pairs, i e there are attribute names like "personal.address zip" and values of different data types stored for these attributes. So called ontologies are used to define the attribute and data type names and hierarchy.

When reviewing information needs in community support applications the following types of information can be identified.

- basic and demographic attributes like "name" or "gender"
- information about interests This can be represented by correlations with predefined clusters or stereotypes (e.g. in iFAY (www.ifay.com)), by ex-

plicit attributes (e.g. "interest.music = 'hip hop'") or by collaborative interest definitions (correlations with other users). The source for all of this can be ratings given by the user to information (implicit by visit or explicit).

- information about relationship networks: colleagues, buddies, .

Some of this information can be stored in a standard way using attribute value pairs, but not all of it Therefore, our approach extends the standard approach by new data types mainly for ratings and for relationships Additionally, there is the possibility to have multiple values in any place in the hierarchy. This is needed to store sets of values for an attribute (e.g. "personal spokenLanguages = ('de', 'en', 'fr')") or to provide several data sets (e g "personal.address(1) street", "personal address(2).street")

The main features in our approach can be summarized as follows (see also Figure 3 for an overview).

- hierarchical attribute space
- values at any level can be sets (multiple values)
- domain specific standard set and additional application specific attributes
- special types for relationships and ratings
- ontology to define attribute hierarchy, attribute names and data types

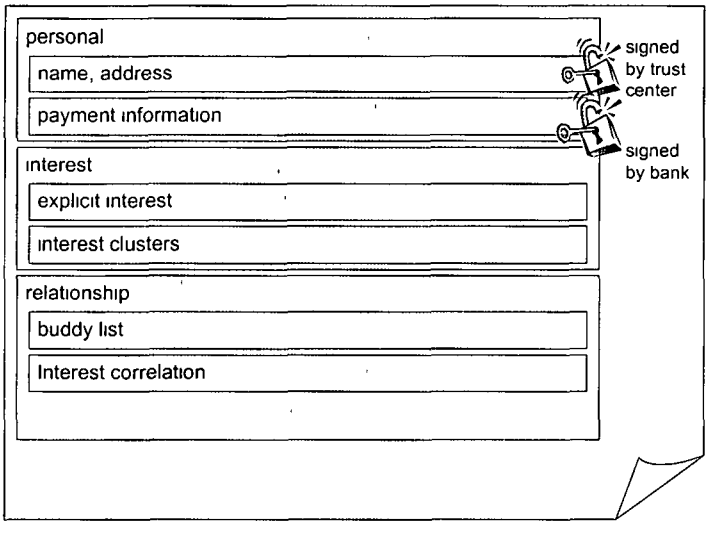

Figure 3 User Profile Structure and Signatures.

Since attributes can be set by different sources they have to store meta information about who has stored them. In addition to knowing who has set some data it might often be necessary to have a prove for this If attributes are to be used for

one application only, the service could store these attributes locally (and only store the other attributes globally in the identity management service) – but there might be needs where attributes should be exchanged among services and still have to be trusted (e.g. attribute that user has bought for more that $1000 at one e-commerce site which entitles him for special discounts in other services). The two solutions to this issue are that the identity management service itself guarantees the source of the data or that the data source digitally signs the data so that anybody can check the origin and the integrity We have chosen the second possibility The repository servers offer a possibility to sign any sub-hierarchy or sub-set of attributes in the repository and store the signatures

4. Privacy

4.1. Overview

The presented ID-Repository network allows reuse of user profile information for different services but does not necessarily improve the privacy situation for the user at first A powerful access control system to the user profile information is essential in this framework.

Existing access control systems (Sandhu and Samarati, 1994), such as role- or group-based solutions may be suitable within a community where administrators and community members may have different access rights and are all well known. With regard to user profile access, there really are only two groups the user who should have total control over his profile and services or other users that may access part of the profile Some communities are more trustworthy to users than others and it is not practicable for users to define different access rights to her profile information for different, not necessarily known, communities. Therefore, a more flexible approach is needed

Our proposal includes two phases (see Figure 4):

1. Negotiation of access rights using privacy policies and preferences and generation of an access ticket (Section 4.2)
2 User profile data access with the access ticket (Section 4.3.)

4.2. Negotiation of access rights

The negotiation of access rights is based on the Platform for Preferences Project (P3P) of the World Wide Web Consortium (W3C) (P3P, 2000) P3P is a project to define a standard way for web sites and user agents to communicate about privacy practices. Its goal is to enable the development of tools for making informed decisions about when personal information should be revealed when surfing the WWW

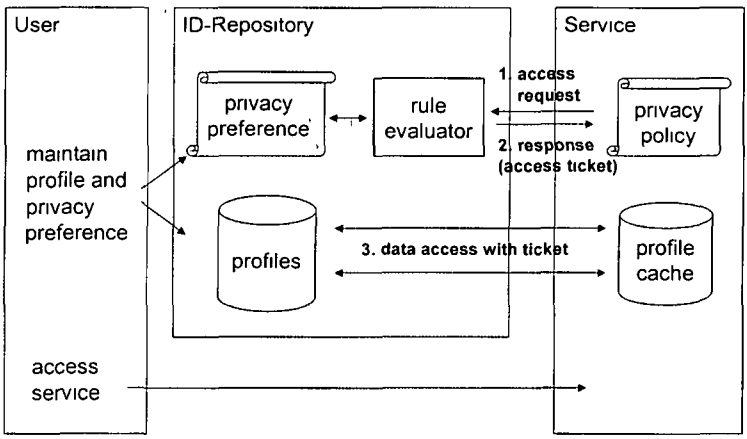

Figure 4 Negotiation of access rights

Services or communities express their privacy policies and profile access requirements in machine-readable form. The P3P specification defines a vocabulary for describing data practices of a service For example, a community can make a statement regarding whether information about the interests of a user are explicitly made available to other members of the community. Also, the purpose of data accesses can be modelled.

The user agent, respective ID-Repository, can then check the conformity of the privacy policy of a community with the user's privacy preferences. P3P includes a standard language for encoding the user's privacy preferences called "A P3P Preference Exchange Language (APPEL)" (APPEL, 2000) APPEL rules allow the expression of preferences over anything that can be expressed in the P3P schema It defines four standard actions "accept" (privacy policy is compliant to privacy preference), "reject" (privacy policy is not acceptable), "inform" and "warn" (information or warning should be provided to the user).

In our framework, a rule evaluator uses the P3P privacy policy and the APPEL rules in the users' privacy preferences to determine access rights for the requested profile attributes allowing/disallowing access or requesting user interaction. This access information is stored in an *Access Ticket* (see Section 4.3).

We are currently working on extending P3P and APPEL for our purposes and implementing the rule evaluator which will be the core of our privacy architecture P3P's data format specifies some commonly used data schemas, such as a person's email and postal addresses This data format can be extended to match our user profile format described in Section 3 3 In addition, P3P provides a mechanism to extend its syntax and semantics using the "<EXTENSION>" tag.

The meaning of the data within the "<EXTENSION>" element can be defined by the extension itself

Negotiation was contemplated in earlier draft versions of P3P but omitted in the final recommendation and might be implemented in future versions In our framework, negotiation is implemented by mandatory and optional parts in privacy policies and preferences and by exchanging different proposals

Users do not necessarily have to formulate the privacy preferences (or access rules) themselves Reasonable rule sets could be provided by trusted organizations and a user could choose a suitable set with to option to manually adjust some parameters or rules. A suggestive user interface is also very important in this regard.

4.3. Access Tickets

The actual data accesses do not need negotiation but the access ticket that represents the result of the negotiation The access ticket manifest the access rights of a certain community application for the user profile information The ticket must be presented by the community with each data access.

The access ticket can be compared to the XML Tickets proposed in (Fujimura et al., 1999) or XML languages for digital rights management such as XrML (Extensible rights Markup Language, www xrml org) It is digitally signed by the ID-Repository on behalf of the user and contains the following information

- Ticket issuer
- Validity date
- Ticket owner: a ticket is usually valid for one service only but may contain rights to distribute information
- Access modes for user profile attributes which include "Read", "Write", "Delete", "Read Once", "Read & Distribute" and "Read & Subscribe"

A released access ticket may be revoked at any time, e g if the user changes his mind about certain rules in his privacy preferences. The access ticket may be passed over from the ticket owner to another service If the ticket states that some information is distributable (access mode "Read & Distribute"), other services can access this information without further negotiation.

The access control system is independent from caching If a community has the right to read (or write) data, it may cache data or not do so unless the ticket states something else (e.g access mode "Read Once") Caching is allowed until the access ticket expires or is revoked. On the other hand, communities need an access ticket to write or collect user data locally, even if the data is never actually written to the ID-Repository.

Since the ID-Repositories handle the specific access rights of every service, it is possible to notify services when information has changed (access mode "Read & Subscribe"). This might not be the case with a role-based access control sys-

tem In addition, functionality in the user interface can be implemented that allow users to check and monitor not only their access rules but also the granted access rights at any time.

4.4. Anonymity

Anonymity is another important issue in any privacy architecture. Users do not want to reveal their true identity to all services Levels of anonymity/identification include:

- (Real) anonymity (transactions cannot be associated with a user)
- Pseudonymity (transactions can be linked to a pseudonym but not to a particular individual)
- Identification (a user's identity is revealed and authenticated by a certificate)

Anonymizing components in our framework act as intermediaries between user and community components and provide the requested degree of anonymity The negotiation of the level of identification is part of the negotiation process described in Section 4.2. users can specify rules in their privacy preferences with regard to the favoured level of anonymity, e.g. "do not reveal true identity unless the privacy policy of the service does meet certain criteria".

Anonymity is not always desirable. For example, transmission of credit card information over a secure channel (e.g Secure Socket Layer, SSL) might necessitate client identification Also, communities have contrary requirements regarding anonymity of their members, users may have to identify themselves to use certain services

Several projects try to achieve unobservability and anonymity in open networks, including CROWDS (Reiter and Rubin, 1997) or ONION ROUTING (www.onion-router.net) and anonymizer services such as www.anonymizer com. These anonymity tools are often based on *mix networks* (Chaum, 1981) A mix network is a collection of routers – the *mixes* – that use layered public-key encryption to conceal the path of a message through the network. Anonymization in the underlying communication network can be integrated in our framework but we will not describe this aspect in more detail in this paper.

4.5. Trust

The storage of user profile information under control of the user and negotiation of access rights based on privacy policies and preferences is a promising technical approach but it cannot solve problems of trust how to control that the privacy policy is really observed by the service? Organizations such as TRUSTe (http //www truste org) or BBB Online (http //www bbbonline org) check the compliance of online services with credible privacy policies and provide privacy seals that are part of a P3P privacy policy The privacy preferences of a user

might include the following rule: "grant access rights only if a privacy seal by a trusted organization is presented by the service"

Services might be able to short-circuit the negotiation and access right process and enhance their information of a user without the user's consent, e.g. by tracking web accesses. However, it could be part of the privacy policy that no other information about the user is collected. A violation of the privacy policy can lead to legal consequences (or loss of the privacy seal, at least)

5. Implementation and Prototypes

In the Project IMC/Cobricks[1] we have implemented first parts of the ideas outlined above – the basic repository service is operable and identity caches are finished for different community support applications The privacy negotiation mechanisms discussed in Section 4 are work in progress and not yet implemented. However, users can already create different identities and link them together, and thereby get different levels of anonymity when giving away access to the data

In this section we will give a brief overview of the implementation and of the applications which are currently using the identity management system.

5.1. Identity Management Infrastructure

ID-Repository

As described in Section 3, the core component of the identity management system are servers storing user profiles (identities). In our case this are the ID-Repository servers A server stores user profiles associated with an identity identifier, which is an anonymous global identifier, and offers the possibility to link identities to each other (see Figure 5)

The identity identifier does not include information about where the identity is stored. To resolve this there is a ID-Resolver in the system that acts as a central directory server and knows which ID-Repository is storing which identity. In this context we are currently working on a de-central solution and on other solutions including the usage of existing directory server infrastructures like X.500 and LDAP

In addition to the profile information (including the meta information about origin of the data and the signatures) the server maintains a log of all access and provides the profile owner with reports from this log. Using the log the profile owner can determine which service currently holds personal information

[1] Cobricks ((Software) bricks for supporting communities) is a project at Technische Universitat Munchen aiming at building a infrastructure for communityware (see (Koch and Lacher, 2000)) – The identity management system is a central part of this infrastructure

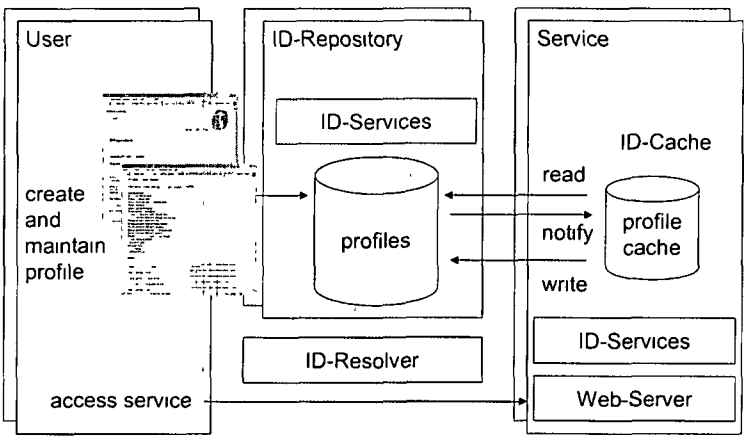

Figure 5 Identity Management Infrastructure.

The server is implemented in Java and offers the following interfaces.

- Corba and RMI interfaces for service-server interaction, for client-server interaction, and for server-server interaction
- ACL (FIPA-ACL interface): we are experimenting to implement the negotiation protocols discussed in Section 4 using the FIPA agent communication language (see (Koch and Lacher, 2000))
- Web (HTTP/HTML) interface for profile owner

We have created XML based schemas for defining the ontologies. The schemas are used in the web interface for dynamically creating the user interface for entering information, and for entering meta information

ID-Cache

On the community support application side we have built custom made stub solutions all based on a common ID-Cache module for storing data and handling the interaction with the ID-Repository. The module will support the negotiation of access rights and already handles synchronization with the ID-Repository.

We are also starting to make stub modules available for widely used generic community platforms So we just have finished implementing a user module for the Cassiopeia Community Server (see www.cassiopeia com)

ID-Services

There are different usages for identity data. We are currently working on a modular approach to this. The so called ID-Services will either be installed in the ID-Repository or on the service side (using the ID-Cache to access data) and perform different services like personalization or generalization on the user data.

5.2. Prototype Applications

The identity management infrastructure is used in several community support applications we are developing and operating in our group Namely these are our different community portals like the Informationsdrehscheibe Informatik, the knowledge management tool CommunityItemsTool and the recommender application CoMovie (see Figure 6 for the Web user interfaces of the applications).

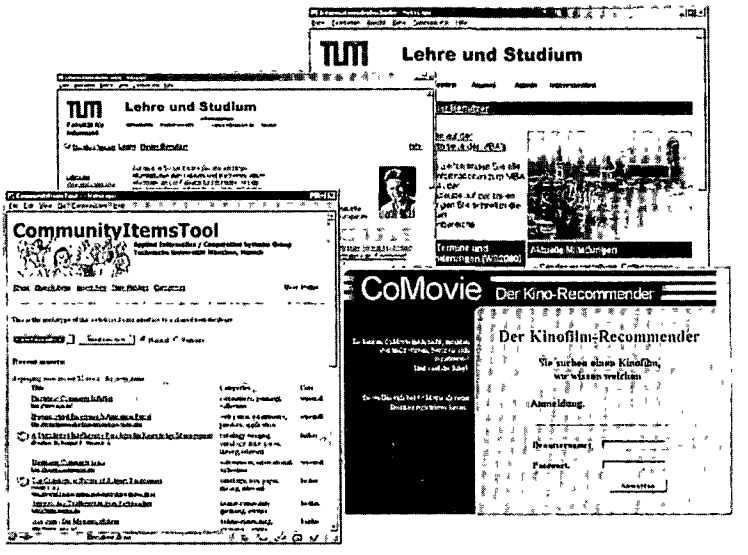

Figure 6. Prototype Community Support Applications.

Informationsdrehscheibe Informatik[2]

This application is a web portal (information source) and online community for the computer science department at Technische Universität München The application uses a lot of information about the user for personalizing the information displayed – some of which are explicitly about user interests. The user has either the possibility to set the user profile information explicitly or the have them derived from his click stream Currently, about 1500 registered members of the faculty and students are using this application Other community portals based on the same technology have been implemented for other departments of for special interest groups at Technische Universitat München

[2] Available at http //drehscheibe in tum de/

CommunityItemsTool

The CommunityItemsTool is a knowledge management application that helps people to publish and exchange bookmarks and bibliographic references (Koch et al., 2001) Users can assign items to user-specific folders to build personal categorizations and use recommendations based on keyword searches Personalization features include configurable layout options and notification services

CoMovie

CoMovie is a classical movie recommender system It offers users the functionality to rate movies and to add comments to the movies (Koch and Lacher, 2001) The information is used to calculate correlations among users (relationship data), and these relations are then used to make recommendations (collaborative filtering, see (Grasso, 1999) and (Resnick, 1994) for more information)

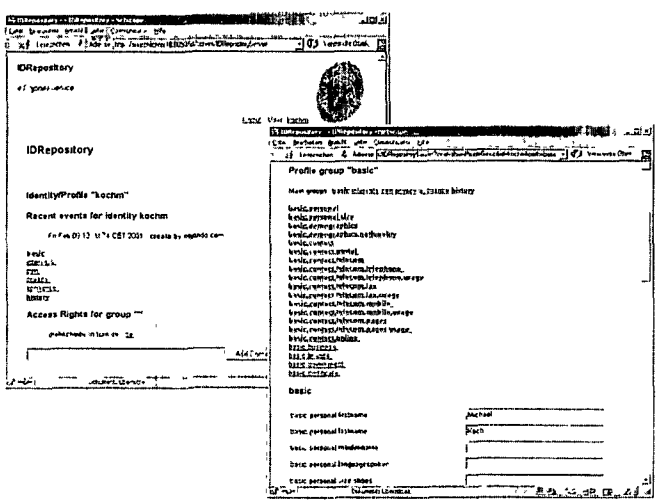

Figure 7. Web User Interface of IDRepository

Usage of Identity Information

The identity framework first offers a single sign-on solution for all these applications. Using the global identifier one can easily register or log in to the services Both passwords and client side certificates are supported This single sign-on also includes the reuse of basic configuration information like name and email address (plus mobile phone number for the emerging mobile services in the community support applications) User profiles can be created and maintained in the community support applications or by using the Web user interface of the IDRepository

(see Figure 7). First results show that the single sign-on solution and possibility to reuse demographic information such as email and postal addresses has been found very valuable by users.

For the other attributes used by the applications we are currently developing ways how to most generally label the information for making reuse possible. Successful reuse was already possible for relationship data (for collaborative filtering and for buddy lists) and for interest information derived from ratings or explicitly entered by the user In the latter case however it showed that the current applications are too different in their focus to make large scale reuse possible – there are only some overlapping areas, e.g. leisure activities.

As a result of this finding we are currently setting up new test applications for the identity service. One of them will be a Munich-wide Mobile Lifestyle Community (see www cosmos-community org for more information) and another one will be an Entrepreneurship Community that especially focuses on relationship networks (see www telekooperation.de/tibid/ for more information).

6. Related Work

First we have to mention work dealing with what "identity" is and how the identity is used or determines interaction in online communities. Examples for work in this area are from Donath (1998) on Newsgroups and from Churchill and Bly (1999) on MUDs.

On the technical side one can find different approaches helping users to manage their online identities by collecting identity information like the infomediaries Jotter (www.jotter com) or Persona (www.persona.com). Most of the infomediaries allow the user to store information in a personal data store and use it in conjunction with automatic form filling features. Some have additional features for automatically sharing information with marketers of products or services they have expressed interest in. The P3P standard described in Section 4.2 is also being integrated in some infomediaries.

Server-side solutions are digitalme from Novell (www.digitalme.com) and Microsoft's passport (www passport com). These systems are very similar to what we have in mind for communities, however they concentrate on delivery information (name, address) and payment information only, and not on personalization.

For personalization one can find different services like iFAY (www.ifay.com) or Yodlee (www.yodlee.com) which support clustering users and making the information about the affiliation to clusters available for personalization. However, these services either do not offer sufficient user control or are too concentrated on marketing and personalisation issues so that they cannot be used for configuration or initialisation.

Köhntopp and Bertold (1999) discuss the feasibility of using P3P as a basis for identity management Their work is more focused on studying legal implications of privacy enabling technologies than technical infrastructures.

For single sign-on there are various solutions based on public-key cryptography and directory services. iPlanet for example is explicitly advertising its directory server and certification authority for this purpose and has extended its servers to accept client side certificates for authentication (see http://docs iplanet com/docs/manuals/security/SSO/sso htm).

Finally, there are services for easily replicating user data. These services currently are used for sharing address information with different peers. Once the information is changed by the owner the business card changes at all places An interesting approach in this area that can also be extended further is the solution by Onename (www.onename.com) and XNS (www.xns org)

7. Summary and Future Work

In this paper we have discussed the issue of identity management in community support applications. With every user using more and more applications, and these applications making use of more and more information about the user this issue becomes important both in groupware and communityware.

We presented an identity management infrastructure which separates the identity management from the service applications This separation is important for making community support applications easier use In real life people take their identities with them all the time and implicitly give or communicate it to people they interact with This also has to be possible in online communities to lower the entry barriers and make rich (personalized) services possible.

Our current work is mainly about implementing the privacy negotiations mentioned in Section 4 and about defining interoperable user profile data in different application areas. Here we also look into the application of the identity management service for internet appliances, i e. tools the user is working with and that have network access These tools will need more and more information about the user in the future and surely lack proper means for entering this data

References

Ackerman, M S , Cranor, L F and Reagle, J (1999) 'Privacy in E-Commerce Examining User Scenarios and Privacy Preferences', *Proc ACM Conference on Electronic Commerce*, Nov 1999

APPEL (2000) 'A P3P Preference Exchange Language', W3C Working Draft, 2000

Chaum, D (1985) 'Security Without Identification Transaction Systems to Make Big Brother Obsolete', *Communication of the ACM*, Vol. 28, No 10, Oct 1985, pp 1030 – 1044

Chaum, D (1981) 'Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms', *Communications of the ACM*, Vol 24, No 2, Feb 1981, pp 84 – 88

Churchill, E and Bly, S (1999) 'Virtual Environments at Work ongoing use of MUDs in the Workplace', *Proc Intl Joint Conf On Work Activities Coordination and Collaboration*, 1999, pp 99 – 108

Clarke, R (1999) 'Internet Privacy Concern Confirm the Case for Intervention', *Communications of the ACM*, Vol 42, No 2, Feb 1999, pp 60 – 67

Cranor, L F (1999) 'Agents of Choice Tools that Facilitate Notice and Choice about Web Site Data Practices', *Proc 21st Intl Conf on Privacy and Personal Data Protection*, Sep 1999, Hong Kong, China

Damker, H , Pordesch, U and Reichenbach, M (1999) 'Personal Reachability and Security Management – Negotiation of Multilateral Security', in Muller, G and Rannenberg, K (eds). *Multilateral Security in Communications – Technology, Infrastructure, Economy*, Proc Multilateral Security in Communications, Jul 1999, Stuttgart, Addison-Wesley-Longman, pp 95 – 111

Donath, J S (1998) 'Identity and deception in the virtual community', in Kollock, P and Smith, M (eds) *Communities in Cyberspace*, London Routledge, 1998

Fujimura, K , Nakajima Y and Sekine, J (1999) 'XML Ticket Generalized Digital Ticket Definition Language', *Proc W3C Signed XML Workshop*, Apr 1999

Grasso, A , Koch, M and Rancati, A (1999) 'Augmenting Recommender Systems by Embedding Interfaces into Practices', *Proc Intl Conf On Supporting Group Work (GROUP'99)*, Phoenix, AZ, Nov 1999

Howes, T , Smith. M and Dawson, F (1998) 'MIME Content-Type for Directory Information (vCARD Specification)', *RFC 2425*, Sep 1998

Koch, M and Lacher, M S (2000) 'Integrating Community Services – A Common Infrastructure Proposal', *Proc Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, UK, Sep 2000, pp 56-59

Koch, M and Lacher, M S (2001) 'The Comovie Movie Recommender – An Interoperable Community Support Application', *Proc HCI International 2001 – 9th Intl Conf on Human-Computer Interaction*, New Orleans, LA, Aug 2001

Koch, M , Lacher, M S and Worndl, W (2001) 'Das CommunityItemsTool - Interoperable Unterstutzung von Interessens-Communities in der Praxis', *Proc Liechtensteinisches Wirtschaftsinformatiksymposium – Informationsmanagement – Herausforderung und Perspektiven*, May 2001

Kohntopp, M and Bertold, O (2000) 'Identity Management Based on P3P', *Proc Workshop on Design Issues in Anonymity and Unobservability*, Jul 2000, Berkeley, CA

Mulligan, D and Schwartz, A 'Your place or mine? Privacy Concerns and Solutions for Server and Client-side Storage of Personal Information', *Proc Computers, Freedom and Privacy*, Apr 2000, Toronto, ON, Canada

Mynatt, E D , Adler, A , Ito, M and Oday, V L (1997) 'Design for Network Communities', *Proc ACM SIGCHI Conf On Human Factors in Computer Systems*, 1997

Reiter, M K and Rubin, A D (1997) 'Crowds Anonymity for Web Transactions', *Technical Report 97-15, DIMACS*, Aug 1997

Resnick, P , Iacovou, N , Suchak, M , Bergstrom, P and Riedl, J (1994) 'GroupLens An Open Architecture for Collaborative Filtering of Netnews', *Proc Intl Conf On Computer Supported Cooperative Work*, Chapel Hill, NC, 1994

P3P (2000) 'The Platform for Privacy Preferences 1 0 (P3P1 0) Specification', W3C Candidate Recommendation, Dec 2000

Sandhu, R and Samarati, P (1994) 'Access Control Principles and Practice', *IEEE Communications Magazine*, Vol 32, No 9, Sep 1994, pp 40 – 48

Webster (1988) 'Webster's New World Dictionary of American English', Third College Edition, Cleveland Webster's New World, 1988