An Architecture for Agent-Based Privacy-Preserving Information Filtering

Richard Cissée
DAI-Labor
Technische Universität Berlin
GOR 1-1, Franklinstraße 28/29
10587 Berlin, Germany
Richard.Cissee@DAI-Labor.de

ABSTRACT

Recommender Systems based on Information Filtering techniques are utilized to an increasing degree in order to provide personalized information, countering information overload. Due to the antagonism of personalization and privacy, however, current Recommender System architectures are not suitable for use with extensive and sensitive user profile data. We propose a novel approach to agent-based Information Filtering resulting in an architecture preserving the privacy of all participants. The proposed solution covers trust relationships between participants and utilizes privacy-preserving implementations of existing filtering techniques.

Categories and Subject Descriptors

H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval – *Information Filtering.* I.2.11 **[Artificial Intelligence]**: Distributed Artificial Intelligence – *Multiagent Systems*.

General Terms

Management, Security, Human Factors, Standardization.

Keywords

Multi-Agent Systems, Information Filtering, Privacy, Trust, Recommender Systems.

1. INTRODUCTION

The total quantity of information stored in electronic format is increasing exponentially. In order to counteract the resulting threat of information overload, human users accessing information utilize Information Retrieval (IR) technologies, such as search engines. Additionally, personalized information is provided by Information Filtering (IF) engines: Utilizing IF-based Recommender Systems, users are offered probably relevant items, (e.g. documents), based on similar items selected previously or further information stored in user profiles.

One of the central problems of the proposed approaches within this area is the antagonism of personalization and privacy [3][13]: In order to obtain personalized information, users have to reveal information about themselves to others, in many cases with unpredictable and undesirable consequences. In most Recommender Systems architectures, the propagation of extensive and sensitive user profile data is addressed insufficiently. As a consequence, the use of systems based on personalized information has not become widely accepted [18].

This paper proposes a solution balancing the privacy needs of all participants involved, resulting in an agent-based architecture for privacy-preserving Information Filtering. In contrast to current approaches, the filtering process is neither controlled by the user role nor the information provider role of the architecture. Therefore there is no need for any participant to reveal information to a potentially untrustworthy party. Thus, systems based on the proposed architecture are likely to gain a high acceptance in real-world applications.

While existing approaches require a high degree of trust between all participating roles, the agent-based design of the proposed architecture largely replaces trust by control, requiring a lower degree of trust between the participating roles which is consequentially established more easily.

The paper is structured as follows: The following section contains the problem description. Section 3 outlines the proposed approach to privacy-preserving Information Filtering. In Section 4, the design of the privacy-preserving Information Filtering architecture is described focusing on two central topics, namely trust relationships and privacy-preserving filtering techniques. Section 5 gives an overview of related work. Section 6 describes the implementation of the proposed approach. The paper is concluded by an outlook given in Section 7.

2. PROBLEM DESCRIPTION

In this section, the existing approaches to Information Filtering architectures are introduced, followed by a discussion of their respective drawbacks. Thus the necessity for a novel approach to Information Filtering is motivated.

2.1 Definitions

Recommender Systems are based on Information Filtering architectures. They produce personalized recommendations, based on a large amount of information of a certain domain, by utilizing filtering techniques. In all Information Filtering architectures, three participating roles can be identified. These roles are implemented as software systems or parts thereof (e.g. multi-agent systems):

- The *user* is the role for which recommendations are to be generated. Usually it represents a human user interacting, via interfaces, with the user role and with other roles. Information about users is collected in *user profiles*. A user profile may contain general preferences of the respective user, as well as certain items relevant for this user, such as previous recommendations.
- The *provider* is the role providing, within a *provider profile*, the information which is to be filtered. Usually, information providers act according to one of the following business models: They either offer information *per se* (e.g. news or documents) for a fee, or products based on the information provided (e.g. books or holiday trips). In the latter case, the information itself is usually provided free of charge.
- The *filter* is the role creating the recommendations, based on the information supplied by user and provider in the respective profiles. It utilizes *Filtering Techniques*, such as Feature-Based Filtering (FBF), Automated Collaborative Filtering (ACF), or a hybrid combination of different techniques. Filtering Techniques are discussed in detail in Subsection 4.2.2.

The profiles may be created and maintained in different ways: They may be based on information provided by other entities than the provider itself. User profiles may be stored in a centralized way within databases, by user agents, or by third parties (*infomediaries*).

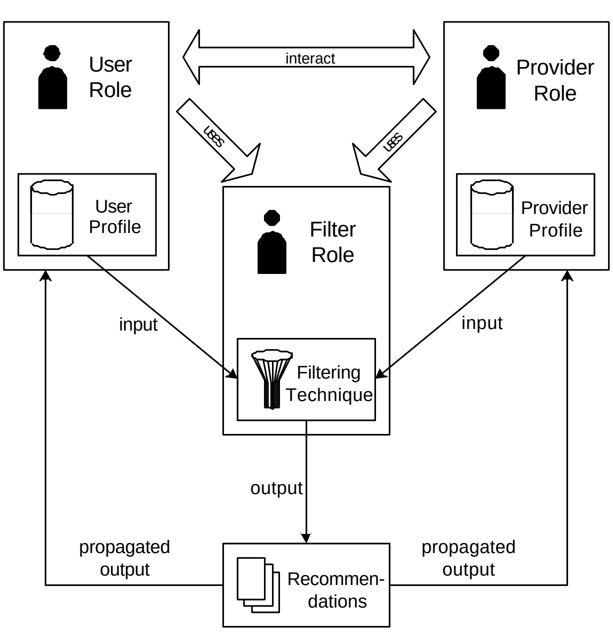


Figure 1: Participating roles and data components in Recommender System architectures.

All Information Filtering architectures are based on the three roles identified and the related data components (profiles and recommendations), as shown the abstract model depicted in Figure 1. Architectures differ, however, in the way the components are controlled by the roles, and the way roles are merged.

As we largely focus the discussion on privacy aspects of Recommender Systems, it should be noted that we use the term "privacy" in the sense of "informational privacy", following Westin's definition of privacy being "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others" [19]. According to this definition, the term is applicable to all roles that have been identified because all profiles contain information related to the respective role.

2.2 Provider-Controlled Information Filtering

Current approaches mostly focus on personalization, neglecting privacy aspects. The resulting architectures can be classified as *provider-controlled* Information Filtering, in the sense that the provider role manages all data components and has privileged access to them. User profiles are stored in a centralized way at the provider's side, and the filtering technique is not administrated by an independent filter role, but by the provider role as well (see Figure 2 for details of the architecture). The provider is responsible for propagating recommendations to the user, however, the user can not rely on the recommendations being complete and accurate. Furthermore, the provider is able to use the user profile for additional purposes without the consent of the user. Most Recommender Systems are currently based on this architecture.

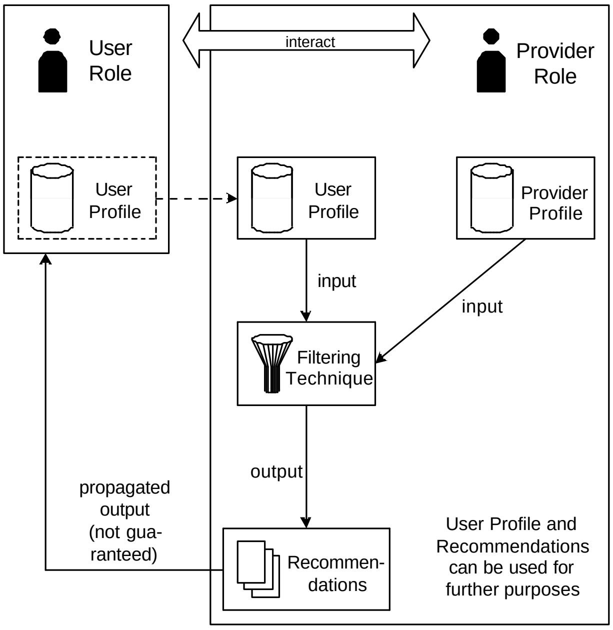


Figure 2: Provider-Controlled Information Filtering architecture (modification indicated by dashed element).

From the provider's point of view, the provider-controlled approach has the advantage that all data is easily accessible and controlled by himself. Certain filtering techniques, such as Collaborative Filtering in its original form, explicitly require this architecture. The provider, however, faces drawbacks as well: Due to the centralized character of this approach, the collected data is vulnerable to focused attacks which are likely to cause more harm than attacks on distributed architectures.

The main drawbacks of the provider-controlled approach obviously arise for the user: Users have no control over the further propagation of personal data, in fact they may not even be able to determine the amount and kind of personal information collected within the user profile s. Thus, the users' privacy is not protected at all.

To enhance the privacy of the user, the provider-controlled architecture is sometimes modified by storing user profile data on the user's side, for example within an agent (see again Figure 2). However, since the user still has to allow the provider to access the user profile, this approach does not constitute a significant improvement over the original approach. At least the security risks are balanced better than in the original approach, because the user profile data is distributed and attacks on single users yield less information than attacks on a centralized user profile storage mechanism would.

2.3 User-Controlled Information Filtering

To overcome the drawbacks of provider-controlled Information Filtering, further approaches protecting the user's privacy have been suggested. The resulting architectures can be classified as *user-controlled* Information Filtering, i.e. architectures in which the responsibilities of user role and provider role are largely switched, compared to the provider-controlled Information Filtering architectures.

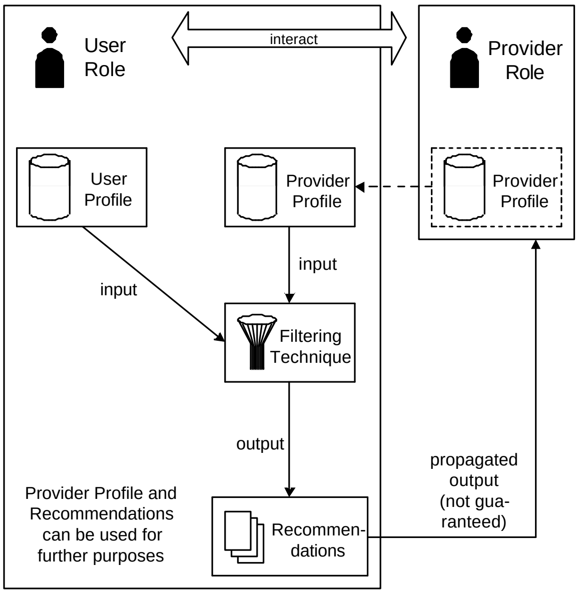


Figure 3: User-Controlled Information Filtering architecture.

In user-controlled Information Filtering approaches not only the user profile, but the filtering techniques as well are under the control of the user role, removing the necessity of propagating user profile data to the provider (see Figure 3 for details). This solution is generally not acceptable from the provider's point of view, as his needs are largely disregarded: He has to provide his entire profile to the user, which is usually economically infeasible. Furthermore, he may not obtain feedback regarding the provided information, based on which he could improve the quality of the information offered, because the user may have no incentive to propagate the filtered results to the provider.

Furthermore, the entire system controlled by the user role is difficult to manage and protect against attacks due to its complexity. A human user's trust in his own agent becomes an increasingly important issue, since the typical human user may not be able to verify that the user role performs all tasks as specified and desired.

2.4 Discussion

The introduced architectures are characterized by the following main drawbacks:

- **Imbalanced privacy protection:** At most the privacy of one side (user role or provider role) is protected, but there is no approach preserving the privacy of both sides, i.e. in no case user and provider role are both able to keep control of the respective profile information.
- **Uncertain Result Propagation:** The architectures do not ensure the correct propagation of the results of a filtering process to both user and provider role. This propagation is at best uncertain (in provider-controlled architectures) and generally rather improbable (especially in user-controlled architectures).

These drawbacks result in a lack of acceptance of the respective systems [18] mainly because the necessary trust between the participants cannot be established easily or at all. Obviously there are scenarios in which users are not concerned about privacy, because the information they provide is neither comprehensive nor sensitive. In these cases the existing approaches may be sufficient. The quality of personalized recommendations increases, however, with the amount of personal information they are based on. Therefore existing solutions are ultimately insufficient in more complex scenarios.

Removing the direct interaction between user role and provider role, e.g. by introducing an additional layer acting as a *portal* between them, does not solve the emerging problems either, since portal providers pursue their own goals and therefore cannot be assumed to be entirely trustworthy. The trust relationships between user role and portal role on the one side, as well as portal role and provider role on the other side are similar to the trust relationship between user role and provider role in the previous approaches. Thus they are characterized by the same problems. Therefore, this approach only complicates the architecture without solving any of the emerging problems.

3. APPROACH

To overcome the disadvantages of the architectures introduced in the preceding section, we propose a novel architecture for privacy-preserving Information Filtering (PPIF). Unlike existing architectures, it balances the requirements of user role and provider role, especially with regard to privacy. This balance is mainly accomplished by modeling the filter as a neutral and independent role which cannot be influenced directly by any other role.[1] Thus, a tradeoff between the user's and the provider's interests is achieved.

The approach is privacy-preserving in the following sense:

- No information about the provider profile, apart from the recommendations themselves, is passed on to the user role. This feature ensures the architecture to be acceptable from the provider role's point of view.
- Minimal information about the user profile is passed on to the provider role, or, in the ideal case, none at all. This feature ensures the architecture to be acceptable from the user role's point of view.

It is shown in the following section that these characteristics are actually fulfilled by the proposed architecture.

The PPIF architecture is designed as a multi-agent system for the following reasons: Certain requirements originate from the architecture, such as the ability of roles to operate in different environments, or the ability to limit the communication of other roles. All of these requirements may be fulfilled by an agent-based architecture, according to the widely accepted definition of agents as autonomous, pro-active entities deployed in decentralized, dynamic systems [21].

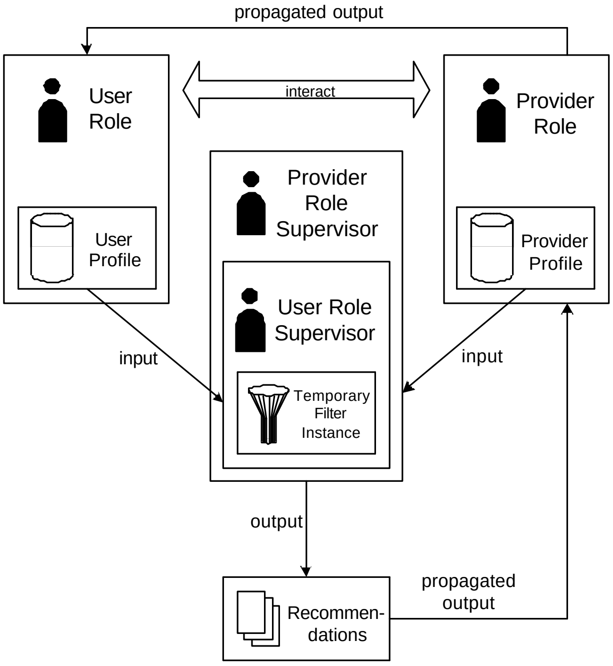


Figure 4: Privacy-Preserving IF architecture.

The actual privacy-preserving Information Filtering architecture largely resembles the abstract model shown in Figure 1, but is somewhat more complicated due to the trust relationships between the roles, which are not addressed by that model: A temporary instance of the filtering technique to be applied is encapsulated by two supervising agents, as shown in Figure 4. The rationale for this architecture is given in Subsection 4.1.

The propagation of results to the user role via the provider role may seem to be a limitation from the user role's viewpoint, compared to the abstract model. It allows the provider role to modify or entirely withhold the recommendations. Because the provision of information, however, is usually in the provider role's best interest, as its business model is based on it, this limitation is negligible in reality. The propagation of results to one role via the other role cannot easily be circumvented, because the results cannot reliably be propagated to both roles independently. This feature is caused by the inclusion of supervising agents within the architecture, which is motivated in Subsection 4.1.

It has to be noted that the issue of malicious hosts [17] is not explicitly addressed by the proposed approach. Suggestions on how to deal with this problem exist [12] and are applicable in this case. Straightforward ways to address this issue are to introduce trusted third parties acting as hosts, or to utilize tamper-proof hardware. To simplify the main architecture, however, hosts are assumed to be non-malicious.

The proposed approach is based on the following two main assumptions.

- The trust relations between the roles can be established more easily and with a higher degree of trust than in other architectures, even though three bilateral relations have to be considered instead of one bilateral relation between two roles. This assumption is necessary for the proposed architecture to have a decisive advantage over the imbalanced approaches.
- Filtering techniques which are applicable in an privacy-preserving way exist or may be found. Because it is infeasible to develop entirely novel filtering techniques within the scope of this work, it has to be determined whether existing filtering techniques may be used or adapted for the required tasks. This assumption is necessary for the proposed architecture to be actually realizable.

In the following section, both assumptions are shown to be valid with regard to the proposed architecture. Apart from the limitation related to the malicious hosts problem, the proposed architecture therefore actually provides "the enhancement of privacy and trust in electronic communities without having to resort to anonymity, pseudonymity, or trusted third parties", a goal stated as ultimately desirable in this context [11].

Users utilizing this architecture, however, should be aware of the fact that the quality of the recommendations is definitely not higher than in other architectures. A possible loss of quality has to be traded off against a definite gain in privacy.

[1] It should be noted, however, that the filter role and one of the other roles may still be provided by the same entity.

4. PRIVACY-PRESERVING INFORMATION FILTERING

Based on the two main assumptions made in the preceding section, the design of the privacy-preserving Information Filtering architecture is affected by the following two aspects: The trust relationships between the filter role and the other roles; and the privacy-preserving implementation of the filtering techniques. These central topics are described in the following subsections.

As the architecture is agent-based, all roles and parts thereof are represented by single agents or groups of agents. Because groups of agents may always be represented by one designated agent, we do not keep up this distinction and use the singular term (in the figures as well) in all cases. All agents run on platforms, i.e. execution environments controlled by a agent management system (AMS), according to the FIPA specifications [8]. Furthermore, agents are mobile, i.e. they are able to migrate from one platform to another.

4.1 Trust Relationships

The relationship of user and provider requires little trust with regard to private information, because such information does not have to be exchanged between these roles. Therefore, the proposed architecture has to focus on the trust relationships between the filter and the other roles. Generally, the necessity for trust is largely removed by preventing malicious acts of any role. The following three aspects have to be considered with respect to the trust relationships between the filter and the other roles:

- **Irrational maliciousness of the filter.** The filter role may act maliciously in an irrational way, i.e. without gaining any advantage by this action. For example, it could deliberately provide low-quality results. However, this aspect is less critical with respect to the trust relationships, since the quality of the results generally does not affect the privacy of the participants. Furthermore, a filter providing consistently irrelevant results is easily recognized and may be avoided in the future. Thus, this aspect is not addressed by the proposed architecture.
- **Rational maliciousness of the filter.** A larger threat arises from the fact that the filter may act maliciously in a rational way, e.g. by collecting information about the participants and using the information for profitable purposes (selling them, for example, to third parties). This aspect is addressed by the introduction of temporary filter instances, as described in Subsection 4.1.1.
- **Collusion between the filter and a participant.** Finally, the filter may collude with one of the participants, e.g. the provider role, by transmitting additional information about the other participant, e.g. the user role. Such a collusion would turn the architecture in effect into a provider-controlled IF architecture, an additional disadvantage for the user being the fact that he may not even be aware of the collusion. This aspect is addressed by the introduction of supervising agents, as described in Subsection 4.1.2.

4.1.1 Temporary Filter Instances

To prevent the filter role from collecting information about the other roles, based on their profiles, temporary filter instances are introduced. A temporary filter instance is provided by the filter role upon request and is represented by an agent implementing the required filtering technique functionality. When the recommendations have been propagated to the other roles, the temporary instance is terminated. Figure 5 shows the communication between the roles in this architecture. The numbers in parentheses in the following paragraph indicate the succession of the actions, which is shown in the figure as well.

A filtering process is initiated by the user role by sending a request for recommendations to the provider role (1). The provider role requests the filter role to provide a temporary filter instance (TFI) agent (2). After the TFI agent is instantiated (3), it queries the user role on the user profile (4), and the provider role on the provider profile (5). After determining the recommendations, the TFI agent returns them to the provider role (6), which in turn returns the recommendations to the user role (7).

Two facts have to be noted regarding temporary filter instances: First, the underlying filtering techniques must not rely on feedback, e.g. to improve their quality, because as part of temporary filter instances they are too short-lived to obtain a sufficient amount of feedback. Thus, only static filtering techniques may be applied. This limitation is taken into account in Subsection 4.2. Second, the temporary filter instance and the filter role itself are still able to communicate, resulting in the filter being able to act in a malicious way after all. If the other roles do not trust the filter role sufficiently, a solution is required preventing this kind of communication. This issue is addressed in the following subsection.

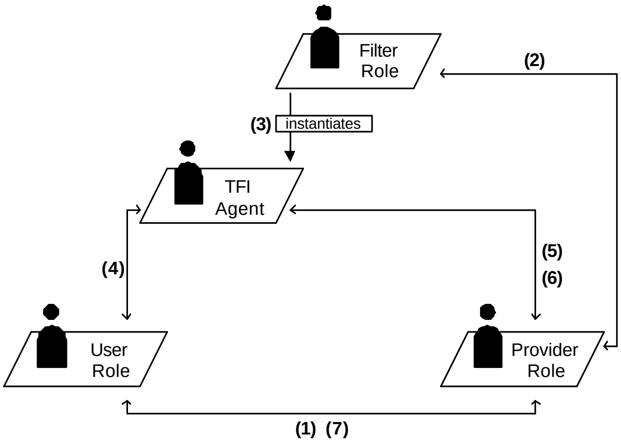


Figure 5: Communication flow in a privacy-preserving IF architecture including a temporary filter instance (TFI) agent.

4.1.2 Supervising Agents

The temporary filter instance (TFI) agent has to be prevented from communicating with the filter role itself as well as any other role or third party after it has obtained any profile information. Therefore the TFI agent is running on a special platform provided and controlled by a *user role supervisor* (URS) agent. The TFI agent may only communicate with the URS agent, who is able to terminate the platform and thus the TFI agent at any time. Communication attempts of any other kind are blocked by the platform manager of the platform the TFI agent is running on.

Upon receiving a message from the TFI agent, the URS agent decides whether to pass the message on to the originally intended recipient, and vice versa. Thus, the TFI agent is still able to obtain profile information in an indirect way, but it is prevented from passing on this information in a way unintended by the URS agent. Beyond limiting the TFI agent's communication capabilities, the supervising agent does not interfere with the TFI agent in any way.

Unfortunately, one supervising agent is still insufficient from the provider role's viewpoint, because provider profile information still may be passed on to the user role via the URS agent. Therefore, the URS agent itself, including the TFI agent, migrates to a special platform provided by a *provider role supervisor* (PRS) agent when it has received the user profile information. The relation between URS and PRS agent is equivalent to the relation between the TFI and the URS agent. In the resulting architecture, no information is passed on to any participant without the consent of any role, represented by the respective temporary or supervisor agent. Figure 6 shows the communication flow between the roles in this final architecture, the dashed lines indicating the platforms which are under the control of supervising agents. Again, the numbers in parentheses in the following paragraph indicate the succession of the actions, which is shown in the figure as well.

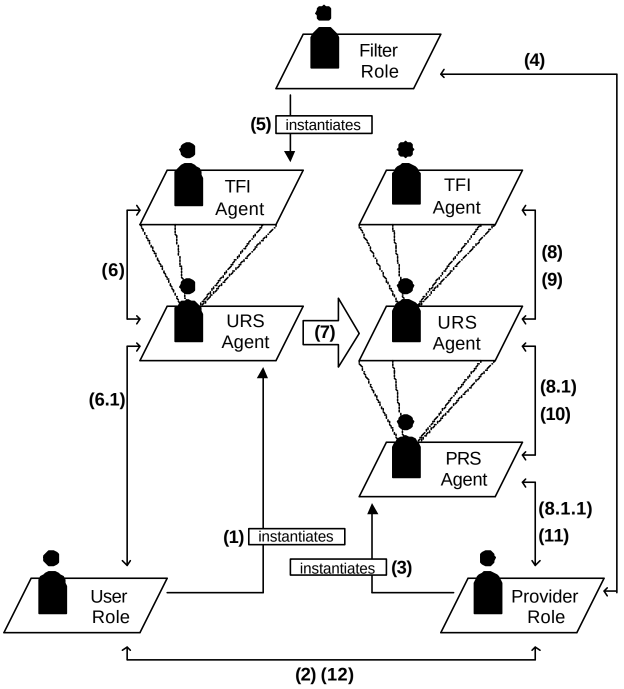


Figure 6: Communication flow in a privacy-preserving IF architecture including supervisor (URS; PRS) agents.

A filtering process is initiated by the user role by instantiating an URS agent on any platform (1) and sending a request for recommendations to the provider role (2). The provider role instantiates a PRS agent on any platform (3) and requests the filter role to provide a temporary filter instance (TFI) agent (4). After the TFI agent is instantiated (5), it queries the user role on the user profile, via the URS agent (6, 6.1). When the TFI agent has received all necessary information, the URS agent, including the TFI agent, migrates to the platform controlled by the PRS agent (7). Depending on the actual agent architecture used, this migration may be problematic because there may be no way to migrate entire platforms. In this case, the migrated URS agent has to initialize a new platform to which the TFI migrates separately.

After the migration has been completed, the TFI agent queries the provider role on the provider profile, via both supervising agents (8, 8.1, 8.1.1). After determining the recommendations, the TFI agent returns them to the provider role, again via the supervising agents (9, 10, 11), which in turn returns the recommendations to the user role (12).

The critical point in this succession of actions is the communication (8), because the TFI agent may utilize user profile information for the subsequent queries on the provider profile. To prevent this, the entire provider profile would have to be communicated, which is usually infeasible. Therefore the URS agent has to decide whether to accept specific queries from the TFI agent. To be able to make this decisions, queries have to be phrased in a certain format understood by all participants involved (e.g. for profiles stored in relational databases, SQL would be an obvious choice of format).

A limitation of this architecture is the fact that the TFI agent has only the possibility to obtain user profile information in a first phase, and provider profile information in a second phase. Iterations of obtaining information from different profiles are not possible, because they would violate the privacy-preserving characteristics stated at the beginning of Section 3. The consequences of this limitation are addressed in Subsection 4.2.

4.1.3 Discussion

To verify the validity of the first assumption stated at the beginning of this section (regarding the trust relations between the roles), we examine the relations between all roles in detail:

- **User role and filter role:** The user role does not have to trust the filter role at all, because it is able to prevent any undesired communication of the TFI agent via its supervising agent and is able to terminate the TFI agent at any time. The filter role does not have to trust the user role at all, because it provides no critical information at any time.
- **Provider role and filter role:** The relation is equivalent to the relation of user role and filter role.
- **User role and provider role:** The provider role does not have to trust the user role at all, because it is able to prevent any undesired communication of the URS agent via its supervising agent and is able to terminate this agent at any time. The user role does not have to trust the provider role to a high degree, because it provides no critical information at any time (this is ensured by the URS agent). The user role only has to trust the provider role to propagate the recommendations in a correct manner.

As noted in Section 3, all hosts of agents and platforms are assumed to act non-maliciously. Therefore, the TFI agent is able to protect the user profile it has obtained, because no other agent is able to manipulate the TFI agent in order to access the profile. Under this assumption, all agents may run on platforms hosted by one of the participating roles. If, however, the roles do not trust each other with regard to this aspect, each role would have to host its own agents as far as possible. In this case, the supervising agents and the temporary filter instance have to be hosted by a trusted third party, if the roles cannot agree on a host among themselves.

4.2 Privacy-Preserving Implementation of the Filter Role

The implementation of the filter role has to take the following aspects into account: The storage of meta-information, the underlying filtering techniques, and the specification of the filter's functionality. These aspects are addressed in the following subsections.

4.2.1 Storage of Meta-Information

A central requirement emerging from the trust aspects discussed in Subsection 4.1 is the following: Regardless of the actual filtering technique applied, the filter has to be modeled as stateless, i.e. no information about users or providers must be kept in between separate filtering processes. Generally, however, filtering techniques are based on long-term meta-information about user profile and provider profile information (such as classifiers or similarity graphs) in order to facilitate real-time filtering. The item models containing this meta-information may be created and updated by the filter role, but they have to be stored at the side of the respective participant. Creating and updating the item models is less problematic than the filtering process itself, because these operations only involve the filter role and one other role. Alternatively, these models may be created and updated by the respective role itself based on specifications given by the filter role. In either case, privacy risks with regard to item models may be eliminated completely.

4.2.2 Filtering Techniques

Several different filtering techniques have been suggested and applied successfully in the field of Information Filtering, such as Automated Collaborative Filtering (ACF) or Feature-Based Filtering (FBF)[2]. Together with the Demographic Filtering (DMF) approach, the former two constitute the class of learning-based approaches to Information Filtering. More recently, knowledge-based approaches such as Knowledge-Based Filtering (KBF) and Utility-Based Filtering (UBF) have been introduced. Hybrid techniques combining different filtering techniques have been suggested to improve the quality of the filtering. A detailed survey of the existing pure and hybrid approaches is given in [2].

Two requirements on filtering techniques are derived from the trust relations, as stated in Subsection 4.1:

- Iterations of obtaining information from different profiles are not possible. In a first phase, all necessary user profile information has to be obtained, in a second phase, all necessary provider profile information has to be obtained.
- The filtering technique must not rely on feedback from earlier filtering processes. This requirement does not necessarily conflict with learning-based approaches, even though this designation seems to imply an incompatibility, because the term "learning" refers to the profiles involved, and not the underlying filtering technique itself.

In the following, we examine the filtering techniques with regard to their general usability as well as these specific requirements.

Feature-Based Filtering

Among the pure approaches, Feature-Based Filtering is most suited to be applied in a privacy-preserving way, since it does depend neither on information about other users (in contrast to the other learning-based approaches) nor on detailed domain-specific knowledge (in contrast to the knowledge-based approaches). As its name implies, Feature-Based Filtering creates recommendations by comparing user profile and provider profile items, based on their characteristic features. The provider profile items most similar to items previously selected by the user are recommended. Similarity is determined directly or via meta-information stored in item models, such as decision trees or neural networks created from profile data. Because the actual algorithm used by this kind of filtering techniques is static, the first requirement is fulfilled. Because no iterations of obtaining profile information are required, the second requirement is fulfilled as well.

Knowledge-Based Approaches

Knowledge-based approaches are generally suitable as well, as they do not depend on information about other users. They are often based on Case-Based Reasoning mechanisms (for examples see [2]) and rely more on general user preferences than on specific items collected in the user profile. However, the following problem with respect to user profile information arises: Knowledge-based approaches often depend on data collected by user interaction, e.g. by relevance feedback mechanisms. This is contrary to the second requirement prohibiting this kind of iterated interaction. Therefore, the proposed architecture has to be extended if knowledge-based filtering approaches are to be utilized, e.g. by adding additional supervising agents.

Other Learning-Based Approaches

Approaches based on the similarity of users, i.e. ACF and DMF, are less suited for the proposed architecture, since no participant is keeping information about several users at any given point of time. These approaches provide recommendations by determining users most similar to a given user with regard to certain characteristics (such as demographic features in DMF, or any kind of user profile information in ACF). Items selected by similar users are then recommended to the current user.

To include these approaches in a privacy-preserving way, the proposed architecture has to be enhanced significantly: One possibility would be to store only the locations of all users' agents in the provider profile, and to retrieve the actual information from the user profiles for each filtering process via the user agents. Due to the high amount of communication between all participants, it may turn out to be infeasible to carry out such filtering processes in real-time. Further research is necessary to determine the feasibility of these approaches and hybrids thereof with regard to the proposed architecture.

[2] Also known under the less accurate terms "Collaborative Filtering" and "Content-Based Filtering" respectively.

4.2.3 Specification of the Functionality

For each filtering process, the user role and the provider role have to agree on a specific filter role to utilize. In order to assist the negotiation between these roles, filters have to describe their functionality. Based on this description, the other roles are able to determine whether a certain filter role meets their criteria. The specification of the functionality has to be based on a common ontology shared by all roles to be intelligible for all participants.

This ontology should minimally cover the following aspects:

- A plain text description of the filtering technique.
- The classification of the filtering technique, based on the main categories introduced in Subsection 4.2.2, and hybrids thereof.
- The domain on which the filtering technique is applicable.

The last aspect has not been addressed so far, but is especially relevant for the following reason: A filter may be classified either as generic or as domain-specific. The former type accepts input from any domain, while input to the latter has to be restricted to a specific domain. The quality of results computed by domain-specific filters is usually higher, because they are fine-tuned to the respective data structures. Domain-specific filters can be expected to be issued by the information providers situated within the respective domain, and thus generally may be assumed to be less neutral than generic filters. Therefore, generic filters may be more desirable from the user's viewpoint, because they may be trusted more readily, reducing the amount of supervision required by the respective agent.

5. RELATED WORK

Research in the area of Recommender Systems has largely focused on the filtering techniques themselves, mainly resulting in provider-controlled IF architectures disregarding privacy issues. Recently, however, some approaches have been proposed which take privacy aspects into account, suggesting modified provider-controlled IF and user-controlled IF architectures respectively. Concepts from these areas may be adapted for the proposed approach. In the area of privacy-preserving IF, related work covers partial aspects such as trust relationships or the agent-based user management.

5.1 Provider-Controlled Information Filtering

There has been some research on privacy risks in recommender systems, pertaining, however, only to a partial aspect of systems based on Collaborative Filtering, namely possibilities to infer identities of other users [15]. Privacy issues in the relations between user role and provider role, on which the proposed approach focuses, are not addressed. In provider-controlled IF architectures, if privacy issues are taken into account at all, they are addressed by the use of privacy policies. These policies are issued by a provider and state how personal information about users is collected, used and propagated. Guidelines for privacy policies have been issued e.g. by the US Federal Trade Commission [7]. Privacy policies are used with limited success in provider-controlled IF architectures: Because they are not legally binding, they are not necessarily adhered to, and therefore are insufficient for establishment of trust. Other approaches introduce anonymous or pseudonymous user identities, leading to recommendations based on non-identifiable personal information stored in the centralized user profiles [14]. If the user profile is stored at the user role's side, as in the modified provider-controlled IF architecture, access control of the user profile becomes a main issue. For Web-based information providers, the Platform for Privacy Preferences Project (P3P) Specification has been introduced [6]. P3P enables users to state their privacy preferences with regard to their personal information, which are matched against a provider's privacy policy. It has been noted that privacy preferences and requirements differ depending on the current user and his location, and therefore should be dealt with by applying dynamical solutions [13].

5.2 User-Controlled Information Filtering

Several user-controlled IF architectures have been introduced (e.g. [5, 9]). Generally, however, these architectures do not imply a single information provider with whom the user interacts, as in the proposed approach. Furthermore, they are often based on or related to Secure Multi-Party Computation (SMPC), which is not applicable if the function to be computed may not be known to the participants, as it is the case regarding the filtering technique in the proposed architecture. User-controlled IF approaches generally utilize ACF or related filtering techniques [3, 11]. Further related fields are Private Information Retrieval (PIR) [4] and Privacy-Preserving Data Mining [1]. PIR is theoretically applicable for querying the profiles, but is currently not sufficiently advanced to be used in real-life applications.

5.3 Privacy-Preserving Information Filtering

Several concepts for agent-based architectures involving three parties have been suggested, introducing a layer between user and provider, constituted by so-called brokers, mediators or middle agents. An overview is given in [20]. These kinds of agents are located between the user and provider and assist their negotiation of service usage. Thus, these concepts differ from the privacy-preserving IF scenario, in which user and provider interact directly. An agent-based management of user profiles, including access control mechanisms, has also been suggested [22]. In the area of trust establishment, reputation systems have been introduced as a mechanism for establishing trust between multiple parties [16]. Reputation systems may be applicable as an additional feature in the proposed architecture.

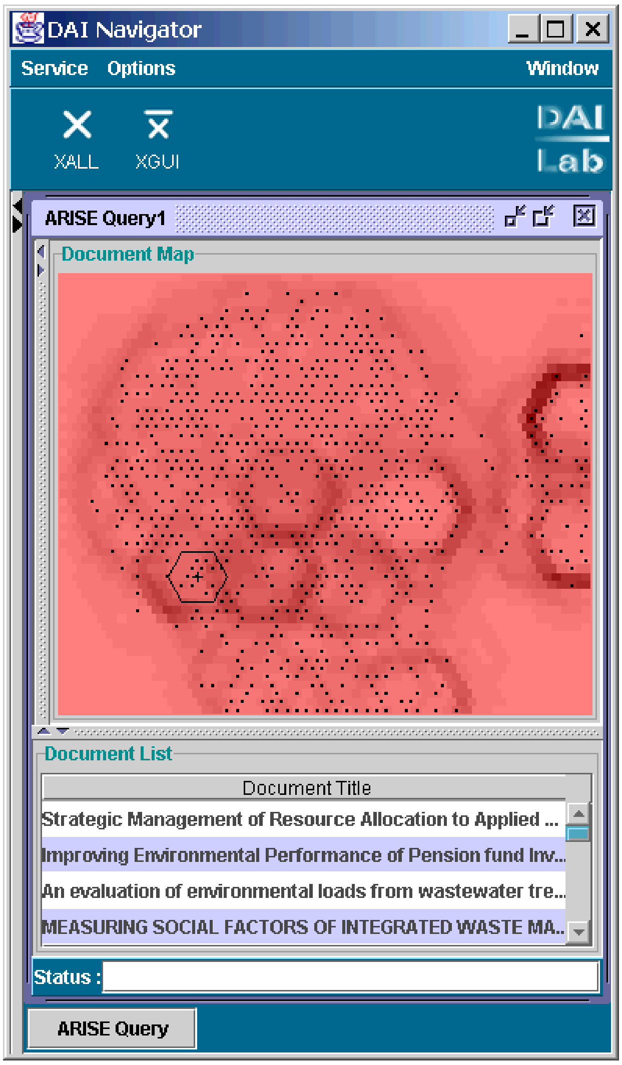
6. IMPLEMENTATION

To demonstrate the usability of the approach, a prototypical implementation of the architecture is currently carried out utilizing the serviceware framework JIAC [10]. JIAC (Java Intelligent Agent Componentware) is a framework for the efficient realization and deployment of intelligent, secure, and manageable agent-based services and applications. It provides a scalable architecture for the realization of different types of agents as well as a runtime environment supported by an extensive management infrastructure.

According to the main aspects of the proposed approach as stated in Section 4, the implementation provides a privacy-preserving implementation of existing filtering techniques as well as the privacy-preserving IF architecture itself.

The prototype currently includes filter roles utilizing feature-based filtering techniques, because its main goal is to prove the overall validity of the approach. Figure 7 shows a screenshot of part of the system, namely a user role's Graphical User Interface (GUI) presenting recommendations (in this case documents) to the user.

The privacy-preserving IF architecture is currently being implemented as part of the BerlinTainment project, which extends the JIAC serviceware framework by providing support for the realization of personalized, location based, and device independent services[3]. For the implementation of the supervising agents capable of blocking the communication of other agents, security mechanisms of JIAC are utilized and adapted.



Document Title
Strategic Management of Resource Allocation to Applied ...
Improving Environmental Performance of Pension fund Inv...
An evaluation of environmental loads from wastewater tre...
MEASURING SOCIAL FACTORS OF INTEGRATED WASTE MA...

Figure 7: Screenshot of the user role's GUI presenting recommendations, based on feature -based filtering of documents. The recommendations are visualized in a map clustering similar documents.

7. CONCLUSION AND FUTURE WORK

In this paper, a novel approach for an agent-based Recommender System architecture is introduced, based on a privacy-preserving Information Filtering architecture. It protects the privacy of all information provided by the participants, and is therefore suitable to be applied in cases where conventional approaches are insufficient due to their imbalanced nature. The proposed solution covers trust relationships between participants and utilizes privacy-preserving implementations of existing filtering techniques.

The implementation focuses on feature-based filtering techniques. An inclusion of other types of filtering techniques remains as future work. Furthermore, the implemented system will have to be evaluated, especially regarding tradeoffs between privacy and computational complexity, in order to determine in which real-world application scenarios the additional costs introduced by the architecture are actually acceptable.

8. REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, "Privacy-Preserving Data Mining", Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 439--450, 2000.

[2] Robin Burke, "Hybrid Recommender Systems: Survey and Experiments", User Modeling and User-Adapted Interaction, November 2002, Volume 12, Issue 4, Kluwer Academic Publishers.

[3] John Canny, "Collaborative Filtering with Privacy", IEEE Conference on Security and Privacy, Oakland CA, May 2002.

[4] Benny Chor, Oded Goldreich, Eyal Kushilevitz, Madhu Sudan, "Private Information Retrieval", IEEE Symposium on Foundations of Computer Science (1995).

[5] Lillian N. Cassel and Ursula Wolz, "Client Side Personalization", DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.

[6] Lorrie Cranor et al., "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification", W3C Recommendation 16 April 2002.

[7] Federal Trade Commission, United States of America: "Privacy Online: A Report to Congress" (1998). Online at: http://www.ftc.gov/reports/privacy3/index.htm.

[8] "FIPA Agent Management Specification", FIPA00023, 2002. Online at: http://www.fipa.org/specs/fipa00023.

[9] Leonard Newton Foner, "Political Artifacts and Personal Privacy: The Yenta Multi-Agent Distributed Matchmaking System", PhD Thesis, Massachusetts Institute of Technology (1999).

[10] S. Fricke, K. Bsufka, J. Keiser, T. Schmidt, R. Sesseler, and S. Albayrak, "A Toolkit for the Realization of Agent-based Telematic Services and Telecommunication Applications", Communications of the ACM, Vol. 44, No. 4, pages 43-48, April 2001.

[3] This project is funded by the German Federal Ministry of Education and Research (BMBF).

[11] Bernardo A. Huberman, Matthew Franklin and Tad Hogg, "Enhancing Privacy and Trust in Electronic Communities", Proceedings of the ACM Conference on Electronic Commerce, 78-80 (1999).

[12] W. Jansen, “Countermeasures for Mobile Agent Security”, Computer Communications, Special Issue on Advances in Research and Application of Network Security, November 2000.

[13] Alfred Kobsa, “Tailoring Privacy to Users' Needs”, Lecture Notes in Computer Science Vol. 2109: User Modeling (2001).

[14] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, “GroupLens: Applying Collaborative Filtering to Usenet News”, Communications of the ACM, Vol. 40, No. 3, pp. 77--87, 1997.

[15] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis, “When being Weak is Brave: Privacy Issues in Recommender Systems”, IEEE Internet Computing, 2001.

[16] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara, “Reputation Systems”, Communications of the ACM, 43(12), December 2000, pp. 45--48.

[17] Tomas Sander and Christian F. Tschudin, “Protecting Mobile Agents Against Malicious Hosts”, Springer-Verlag, Lecture Notes in Computer Science #1419, June 1998.

[18] J. Ben Schafer and Joseph A. Konstan and John Riedl, “E-Commerce Recommendation Applications”, Data Mining and Knowledge Discovery, Vol. 5, No. 1/ 2, pages 115-153, 2001.

[19] Alan F. Westin, “Privacy and Freedom”, New York, NY: Atheneum, 1967.

[20] S. G. Woods and M. Barbacci, “Architectural Evaluation of Collaborative Agent-Based Systems”, Technical Report, CMU/SEI-99-TR-025, Software Engineering Institute, Carnegie Mellon University, PA, USA, 1999.

[21] Michael Wooldridge and Nicholas R. Jennings, “Intelligent Agents: Theory and Practice”, The Knowledge Engineering Review, Vol. 10, No. 2, pages 115-142, September 1995.

[22] Wolfgang Wörndl, “Privatheit und Zugriffskontrolle bei Agenten-basierter Verwaltung von Benutzerprofilen”, TUM-I0106, Institut für Informatik, Technische Universität München, Nov 2001.