Provider-Independent Online Social Identity Management -
Enhancing Privacy Consistently Across Multiple Social Networking Sites

Moritz Riesner
University of Regensburg
moritz.riesner@wiwi.uni-regensburg.de

Günther Pernul
University of Regensburg
guenther.pernul@wiwi.uni-regensburg.de

Abstract
The rising pervasiveness of social networking sites (SNS) poses new privacy risks and prompts adjusting one’s online appearance to the current context of usage. This includes the deliberate and targeted disclosure of selected information to a subset of one’s contacts, also known as social identity management (SIdM). Yet, when performing SIdM across multiple SNS, it is difficult to maintain a consistent representation of oneself. We introduce a global model to perform SIdM which is independent of particular SNS. Then we show how to apply modifications of the global model to particular sites. We evaluate the feasibility of the approach by surveying currently available settings and APIs in the most important SNS.

1. Introduction
Social Networking Sites (SNS) on the web such as the popular site Facebook are rapidly gaining importance both for personal use and in business settings. On SNS, users create personal profile sites, make connections to other users and traverse the resulting social graph [4]. SNS use is no longer limited only to college students [22] as there are professional and collaborative applications as well [20]. Beyond the mere publication of static information to contacts, SNS have introduced new, often semi-public communication channels, such as micro blogging, chat and commenting functionalities. With these new features and increased usage intensity, SNS go beyond merely representing aspects of reality and become a platform on which the real life partly takes place.

This rise of the importance of SNS has also prompted privacy concerns [1, 16]. Privacy considerations need to take into account not only service providers, as in previous research [14], but also the user’s contacts. As these are usually already known by name from other occasions, typical protection goals of privacy enhancing technologies such as anonymity and pseudonymity [13] are no longer applicable in such scenarios [24]. Instead, solutions for Social Identity Management (SIdM) are needed. SIdM, which is discussed in depth in [18] refers to the deliberate, context-dependent disclosure of certain personal information targeting selected groups of contacts.

SIdM becomes necessary because of the problems that are caused by the use of the same SNS account in multiple, possibly conflicting contexts [11, 19], such as a circle of friends and a group of coworkers [9, 22].

Performing SIdM requires partitioning one’s contacts into groups for selective disclosure of information, which is also known as audience segregation [2]. Subsets of one’s attribute values that are to be disclosed to certain audiences are referred to as personas or partial identities [9].

So far, SIdM has only been applied to the case of a single, self-contained SNS. In this paper we argue that there is a need to carry out SIdM in a provider-independent fashion, only considering contacts and presented attributes and not the particular SNS that establish the connection. We present a provider-independent model to aid performing SIdM and show how to decompose it into provider-specific models that can be used to apply and monitor SIdM decisions.

As the problem of provider-spanning SIdM has not yet been solved, we argue for employing a design-oriented research methodology [15]. Hence the paper is structured as follows: After discussing related work concerning SIdM in the following section, the need for provider-independent SIdM is justified in Section 3 by analyzing a sample case. The result is a set of problems that are to be solved by provider-independent SIdM. The proposed constructs and methods to facilitate this are introduced in detail in Section 4. Section 5 aims at validating these artifacts and our approach by surveying its applicability to existing SNS. Section 6 concludes the paper discussing our approach, possible improvements and remaining challenges.

2. Related Work
Defining features of SNS and a history of their development are given in [4]. Multiple authors argue that privacy is a growing concern as SNS usage has in-
creased over the years [16]. Further, it has been pointed out that previous conceptions of privacy protection need to be readjusted when applied to SNS [24]. Several studies examine user behavior on SNS regarding privacy, especially as their usage affects not only young college communities any more [22].

Existing privacy controls in SNS have been surveyed extensively in 2009 [3]. Also, a taxonomy to describe social networking data in privacy discussions has been introduced [21]. Several approaches to enhance privacy both in existing SNS and using newly designed architectures have been proposed.

In [1] and [17], client-side privacy managers are proposed that encrypt disclosed data before storing it at the SNS. While this protects the data from untrusted site operators, all contacts are required to use the client software for key exchange and decryption as well. The focus here lays on mere data protection through encryption instead of audience management. Also, while our approach leads to a client-side or web based solution as well, no further software is required to be used by the user’s contacts.

Decentralized social networking architectures aim at eliminating the need for a central site by distributing the functionality of common SNS among multiple possibly user-owned node servers [6, 7]. This would return data ownership and control to the user and allow for SIdM. However, as these approaches do not include the very popular existing centralized SNS, all contacts would be required to switch to such a solution as well.

The need for audience segregation and persona management as discussed in the previous section has been recognized by several authors [9, 23]. A prototypical SNS employing audiences and personas is shown in [2] and approaches to assist grouping contacts and attributes are discussed in [18] and [8]. Also aiming at privacy, formal models to express access control generically within a single [10] or decentralized SNS [5] have been proposed.

Unlike these approaches, our proposal is also applicable to the case of SNS-spanning SIdM. In [9], an approach to merge a user’s social network from multiple network accounts is presented, but the work is conducted from an external view and the focus does not lay on SIdM. Thus to the best of our knowledge, a solution similar to the work presented in the paper has not been presented before.

3. The Need for Provider-Independent Social IdM

As users are using multiple SNS to interact with their contacts [9, 4], new problems arise that have not been considered in previous research. To illustrate these issues, we introduce the following example:

Alice is using three SNS to contact over 200 friends and acquaintances. As these contacts originate from different areas of her life, she wants to share different personal information with them. To facilitate that, she groups her contacts into multiple Segregated Audiences (SeA) for selective attribute disclosure.

The first SeA contains 20 of her best friends. She decides to share certain profile information with these contacts, namely a profile picture showing her with a guitar, her cell phone number and a list of favorite artists. Thus she creates a persona containing these attribute values and links it to the SeA “friends”.

Another SeA consists of 30 colleagues from work. The corresponding persona contains a profile picture of her wearing a business dress and her work e-mail address. Also, future status updates regarding a company project are to be disclosed to members of the SeA “colleagues” to facilitate collaboration. She creates further SeAs and corresponding personas for the other areas of her life. As not all of those areas are completely separate, contacts may belong to multiple SeAs, hence making them overlapping.

Current SNS support SIdM as conducted by Alice to a limited extent by providing certain privacy controls [3]. For instance, Facebook allows grouping contacts into friend lists for selective disclosure of certain posts and attributes. Yet there are limitations. For example, for most attributes it is not possible to disclose different values to different contacts.

Also, these privacy controls are only available within an SNS. The members of the SeA “friends” for example are contacted using different SNS. Thus, once Alice decides to update an attribute in the persona that is linked to this SeA, she has to update it on multiple sites. Hence her attribute values are also represented on multiple SNS. Also, as she contacts some people using multiple sites, she has to be careful to maintain a consistent image of herself across these sites.

Figure 1 illustrates these problems by showing a user connecting with several contacts (c1,…,c5) using multiple SNS, of which only two are shown. Each SNS carries only a subset of the user’s own representation and her social network. For example, only contacts c1, c2 and c3 are contacted using SNS 1. Also, two of the three attributes of the user are represented there. In particular, the following issues arise in such situations:

- **Redundant online-representation of a single contact**: Users building connections with the same contact over multiple SNS results in multiple representations (see contact c3 in Figure 1).
- **Redundant online-representation of user personas**: The user’s personal information is stored on multiple SNS and is possibly contradictory (see the user’s attribute value 2 in Figure 1).
- **SNS-Spanning SeAs**: Connections to contacts from the same area of life are conducted through multiple SNS, thus leading to a possibly inconsistent representation of self to these contacts.

Figure 1. Problems when using multiple SNS

These issues easily lead to inconsistencies, making SIdM a tedious and error-prone task. SIdM-decisions have to be made separately for every SNS-account, yet at the same time they have to be consistent to each other.

To mitigate these problems and to enable conducting SIdM consistently across multiple SNS Platforms, we propose **provider-independent SIdM**. It is based on the idea of representing the user’s complete social network as a whole and independently of the particular SNS-platforms. In such a model, the user may perform SIdM without having to worry about the SNS-account through which the connection to a particular entity of the model is performed. It is analogous to using a conceptual model representing the state in all SNS platforms a particular user is using on a high level of abstraction while the particular SNS is responsible for carrying out the user’s SIdM decisions.

Our approach acknowledges the current fragmented SNS landscape. The assumption of users employing multiple SNS can be justified with the number of available general-purpose SNS, such as Facebook, Google Plus and regionally focused counterparts. Using a broad definition for SNS, a number of 5.7 different online profiles per person on average has been reported [16]. While some users might want to restrict themselves to only one SNS, they won’t be able to establish a connection to like-minded real-life contacts that only use a different SNS. Also, one might argue that the special purpose of an SNS implicitly leads to using it in the proper context, for instance using LinkedIn only with business contacts. However, this cannot be upheld for general-purpose SNS as shown in [22].

While possibly contributing to user data protection, recently proposed decentralized solutions [6] for social networking are still in early development. If implemented, they have only a small user base, suffer performance issues and do not provide the whole range of features that are offered by well-established sites [1, 7]. As our approach is provider-independent, it is applicable to both existing and future social networking solutions. It does not require the contacts on the other end to switch their SNS while still being open to future developments.

In the following section, we introduce a provider-independent model to express and define the entities of the user’s social network – namely contacts, audiences, attribute values and personas. This model is suitable as a basis for a future SIdM user interface. Further, we show its decomposition into multiple provider-specific models that are used to carry out the user’s decisions in the particular SNS.

### 4. Models for Provider-Independent Social IdM

We propose a solution comprising of a single **global provider-independent model**, representing all of the user’s contacts, audiences, attributes and personas, and multiple **local provider-specific models** consisting of subsets of the global model. As depicted in Figure 2, the user conducts SIdM-decisions using the global model, which is decomposed into **provider-specific target models** to be applied at the particular sites. To monitor if the settings were applied correctly, an **actual provider-specific model** is derived from the state in the SNS for comparison with the corresponding target model. Also, the initial global model is composed from the actual provider-specific models.

Figure 2. Proposed solution

As evident from Figure 2, there is a continuous cycle between user updates on the global model, applying these updates across the SNS, monitoring their
correct application and updating the global model if necessary.

The shaded areas on the left of Figure 2 indicate how a possible SIdM-solution could be implemented: A user interface represents the global model, implemented in a web application or a local client such as a browser extension. A customized connector for each SNS facilitates application and monitoring of the provider-specific models.

We define the global model in the following section and discuss SIdM actions on it in Section 4.2. The decomposition of the global model into provider-specific target models is described in Section 4.3. Deriving actual provider-specific models from the SNS and their composition to a global model is then discussed in Section 4.4.

4.1. Global Provider-Independent Model

The global provider-independent model contains contacts and user attributes regardless of through which SNS the interaction with the contacts is carried out and where the attributes are stored. Thus this model abstracts from particular SNS providers and allows a view for carrying out SIdM without having to care about the underlying providers.

In the model, contacts are allocated to groups representing SeAs. The user data consisting of attribute values is grouped into personas, which can in turn be assigned to SeAs to express that the attribute values of that persona are to be disclosed to the members of the linked SeAs. The model is illustrated in Figure 3, showing three SeAs that are linked to three personas.

More formally, let \( C = \{c_1, c_2, \ldots, c_n\}\) be the set of the user’s contacts, \( VAL = \{v_1, v_2, \ldots, v_m\}\) be the set of the user’s attribute values that are disclosed online and \( A = \{a_1, a_2, \ldots, a_k\}\) be a set of attributes to categorize these values.

\( CONF \subseteq A\) denotes conflicting attributes that can only have one value in the same persona. Showing multiple values of these attributes in the same persona and therefore to the same contacts may lead to contradictions. Examples are the attributes birth date and profile picture. Across multiple personas however, different values for the same conflicting attribute are possible. As pointed out in the example in Section 3, a user may choose a profile picture depicting her in a business suit for the persona linked to a professional SeA while using a more casual picture for other SeAs.

The global, provider-independent social network model of the user is a 4-tuple

\( M_G = (SEA_G, PER_G, ALLOC_G, ACCOUNT) \)

in which \( SEA_G \) is the set of possibly overlapping segregated audiences, with each being a subset of \( C \): \( SEA_G = \{SeA \mid SeA \subseteq C\} \). Figure 3 depicts three sample SeAs, for example representing “friends”, “colleagues” and “family” as shown in the example in Section 3.

\( PER_G \) is the set of personas, with each persona \( P \) being a set of 2-tuples representing an attribute and a set of corresponding values. There may only be one 2-tuple for each attribute per person and, as pointed out above, if the attribute is part of \( CONF \), there is only one corresponding value per persona:

\[ PER_G = \{P \mid P = \{(a, V) \mid a \in A, V \subseteq VAL, \forall (a_i, V_i), (a_k, V_i) \Rightarrow a_i \neq a_k, a \in CONF \Rightarrow |V| = 1\} \].

For that reason, for attributes that are not marked as conflicting, enumerations of multiple values in the same persona are possible. In the example in Section 3, this is needed to represent Alice’s list of favorite artists and her project-related status updates. In Figure 3, there are multiple values for the attributes \( a_1 \) and \( a_4 \) within the same personas. For \( a_1 \) and \( a_3 \), which are known as conflicting (part of \( CONF \)), there is only one value given within the same persona.

One could model further attribute properties, for example distinguishing between static user profile information and dynamically posted items such as status updates. Also, a distinction between self-published items and implicitly available information [21] could be made. However, such information is often very dependent on particular SNS and thus not suitable for a provider-independent model. Also, it is not necessary to consider these properties for the model to work.

\( ALLOC_G \) is the set of allocations between SeAs and personas, with each one being a 2-tuple containing an \( SeA \) and a persona:

\( ALLOC_G = \{(SeA, P) \mid SeA \in SEA_G, P \in PER_G\} \)

In Figure 3, the three allocations \( (SeA_1, P_1) \), \( (SeA_2, P_2) \) and \( (SeA_3, P_3) \) are shown as edges between SeAs and personas.

![Figure 3. Global provider independent model](Image)

The conflicting attributes require further consideration. In Figure 3, the requirement of only one value per conflicting attribute per persona is fulfilled. It would for instance be violated by the presence of another attribute value for attribute \( a_1 \) in persona \( P_3 \).
Contradictions can also occur when overlapping SeAs are linked to personas that contain different values for a conflicting attribute. For example, contact c₄ in Figure 3 is assigned to SeA₁ and SeA₂ and may thus see Personas P₁ and P₂, resulting in seeing the two values v₈ and v₇ for the attribute a₂. One way to resolve such problems is to treat any intersection between two SeAs as a new SeA and define a new, “virtual” persona as the set union of all personas linked to the original SeAs. Then the user needs to be advised of the two possibly conflicting attribute values.

The proposed global model presents the user’s entire online social network consisting of contacts and shared information in a unified manner, regardless of the platform on which these entities are contacted or stored. This is sufficient to perform SIdM decisions. However, to apply them, contacts and attribute values need to be linked to particular SNS accounts. Thus, for each SNS-account i, there is a set ACCᵢ containing those contacts and attribute values that are already carried by that account. These sets are members of the set \( ACCOUNT = \{ ACC \mid ACC \subseteq (C \cup PERG) \} \). Figure 3 contains ACCᵢ as a sample.

Note that certain SIdM actions may influence at which SNS contacts and attribute values need to be stored, thus changing the desired state of ACCOUNT. This is discussed together with the decomposition into provider-specific models in the following section.

The provider-independent model can serve as the basis for an SIdM interface. It employs constructs to model the notions of SIdM, namely forming SeAs by grouping contacts and constructing corresponding personas consisting of attribute values that are to be presented. If applied correctly at the particular SNS, it resolves the redundancy issues pointed out in Section 3. Further applications of the SIdM interface and thus the provider-independent model are possible. If more information about contacts is available, for instance relationships between them, the user could be assisted by automatically grouping contacts and attribute values [8, 18].

4.2. Performing SIdM using the Global Model

The global model is constructed initially from information extracted from the SNS accounts, which is discussed in Section 4.4. Also, the model is updated when SIdM-decisions are performed. Due to the requirements posed earlier, user actions concerning the main elements of the model need to be supported.

The user needs to be able to update, delete and move existing attribute values and add new ones. Updating a value does not interfere with the rest of the model, but changes need to be applied to the SNS. Creating a new attribute value in a persona requires either adding a new attribute-value pair (a, v) to a persona P or adding another attribute value to an existing attribute value-pair. When doing so, adherence to the previously stated rules regarding conflicting attributes is necessary. Removing attribute values is performed similarly by removing the value from P. Moving a value between personas corresponds to deleting it in the first persona and creating it in the second persona.

Moving contacts between SeAs does not need further consideration as no constraints are violated. Copying a contact to another SeA however results in overlapping SeAs and could thus lead to showing conflicting attribute values.

Assigning personas to SeA is performed by creating corresponding 2-tuples in ALLOC. Assigning a single persona to a single SeA is unproblematic. Special consideration is however necessary when two personas are disclosed to the same SeA and when overlapping SeAs are linked to different personas. In these cases, contacts may have access to conflicting values of the same attribute. Merging two personas is similar to moving all attributes from one persona to the other and deleting a persona is performed implicitly by removing all attribute values.

All relevant user SIdM decisions are well-defined using the global model. Most actions are carried out by simple reassignment of set entities. At this stage, only the constraints of the global model have to be considered. Particular circumstances of the SNS are resolved when the global model is decomposed and the settings are applied there.

4.3. Decomposition into Local Provider-Specific Target Models

A provider-specific target model is constructed for every account at a particular SNS that the user has. It contains the subset of those entities of the global social identity model that are or should be carried by that SNS account. It is used to actually execute the SIdM-decisions that are performed using the global model. Then, depending on the SIdM-decisions performed on the global model and the capabilities of the SNS to support them, it may become necessary to break down the model further and create new corresponding SNS-accounts.

We define the local provider-specific model for the SNS account ACC ∈ ACCOUNT as a 3-tuple

\[ M_{ACC} = (SEA_{ACC}, PER_{ACC}, ALLOC_{ACC}) \].

In a first step, for each SeAg in SEAᵣ in the global model, there is a corresponding SeAACC in the provider-specific model for account ACC. It consists of those contacts that are both in SeAg and contacted via account ACC. SEA∗ACC, denoting a tentative set of segregated audiences in MACC, is determined as follows:
∀(SeA ∈ SEA) ∃ SeAacc ∈ SEA∗ACC: SeAacc = SeA ∩ ACC.

Empty SeAs are dropped from SEA∗ACC, as they are not related to the currently processed SNS-account: SEAACC = SEASC∗ / { SeA | SeA = ∅ }. Then for each SeA in SEAACC, all corresponding personas need to be included in PERACC:

∀ SeA ∈ SEAACC, (SeA, P) ∈ ALLOCG: ∃ P ∈ PERACC.

Figure 4 illustrates the provider-specific model for ACC that was derived from the sample global model in Figure 3. Note that unlike SeA persons have to be added to the provider-specific model as a whole. This is because the same information has to be disclosed to contacts that are within the same SeA to ensure a consistent representation of self across sites. Thus there may be attribute values in the personas in PERACC that were previously not linked to the SNS account (not included in ACC) and that now have to be represented in the SNS account. This situation occurs for example when the user creates a new attribute value in one of the personas or when two contacts originating from different SNS are moved to the same SeA. Once the attribute values not yet included in ACC (UPERACC / ACC) are represented in the SNS, ACC in the global model has to be updated.

Figure 4. A provider-specific target model

Finally, the allocations in ALLOCG that concern both SeA in SEAACC and personas in PERACC become part of ALLOCACC and thus complete the provider-specific model.

So far, the provider-specific model MACC was constructed under the assumption that the SNS-provider ACC supports all facets of SmD. If this is not the case, MACC has to be modified. In particular, two distinctions regarding the provider’s support can be made:

**Distinction (A).** Does the SNS support partitioning the user’s contacts into SeA? For instance, Facebook supports creating “Friend Lists” for fine-grained privacy settings while there is no such feature on the microblogging site Twitter. A further distinction is whether one contact may be the member of more than one SeA. If there is no support for creating SeA, workarounds have to be employed, for instance creating a dedicated SNS-account for each SeA and networking with the corresponding contacts through that account. As it requires asking one’s contacts to confirm networking through a new account, this workaround is undesirable.

**Distinction (B).** To what extent does the SNS support disclosing different, well-defined personas to the previously defined SeA? For example, Facebook supports fine-grained access controls for some of its attributes such as religious beliefs, status updates and photo albums. However, for other attributes, such as the profile picture, this is not possible. Also, for most attributes it is not possible to show different values to each audience. Once again, a workaround for this problem is creating additional SNS-accounts.

In Section 5.1 we apply these distinctions to current SNS. To determine whether additional modifications to MACC are necessary, we propose the following steps, which refer to the illustration in Figure 5:

First, it needs to be distinguished whether contacts linked to the current SNS are assigned to multiple SeA (1 in Figure 5), which is the case when |SEAACC| > 1. The case that all of these contacts belong to the same SeA (2) is trivial: No further action regarding contact partitioning is necessary.

![Figure 5. Creating the provider-specific model](image)

If the contacts belong to multiple SeA, the further steps depend on the SNS support for multiple SeA (3) as described in distinction (A). If there is no support for SeA creation (4), the previously mentioned workaround of creating a dedicated SNS account for each SeA has to be applied. Thus for each SeA ∈ SEAACC, a new provider-specific target model MACC has to be created by decomposing MACC which can be performed similarly to decomposing the global model. No further action is required for the newly created provider-specific models, as they do not contain multiple SeA (2).
If the SNS does support SeA partitioning, it has to be examined if the corresponding personas can be mapped to the SNS account without alteration, as described in distinction (B). More precisely, for attributes with values in only one persona, the SNS needs to support hiding its values from contacts whose SeA is not linked to that persona. Also, for attributes with different values in multiple personas, the SNS needs to support showing different values for the same attribute depending on which contact is accessing them.

If proper and non-conflicting attribute representation is not possible, the workaround using multiple accounts has to be applied (4) to ensure complete integrity of the user’s SIdM decisions, which is of course an unfavorable result. If all attribute values of the corresponding personas can be represented properly in the SNS account, M\textsubscript{ACC} can be applied.

4.4. Deriving Actual Provider-Specific Models from the Current State in the SNS

The previously described provider-specific target models were derived by decomposing the global identity model and describe the state of the user’s account at a particular SNS as desired by the user. Their contents have to be applied to the sites using suitable means such as API-calls. Then the actual state in the SNS needs to be verified against the models to avoid mismatches and to ensure proper application of the SIdM decisions. Thus there is a need to extract the user’s social network information from the SNS, thereby creating actual provider-specific models.

Further, for building the global model initially, actual provider-specific models of the user’s social network need to be derived from the SNS accounts. This requires extracting the previously defined entities, namely contacts, user attributes and, if applicable, SeAs and personas. Also, it is beneficial to gather further information about the contacts to aid composition of the global model.

Creating the actual provider-specific model M\textsubscript{ACC} for a given SNS account ACC begins with gathering all contacts that are contacted via that account. If SeAs are modeled within the SNS, SEA\textsubscript{ACC}, the allocation of contacts to SeAs, is based on this information. For each SeA, a persona is created based on the attribute values that are disclosed to the SeA members, thus leading to PER\textsuperscript{ACC} and ALLOC\textsuperscript{ACC}. The model has to represent the factual disclosure of information to contacts by the SNS accurately. Thus if there are exceptions, for example that a certain attribute value is hidden from a contact, then M\textsubscript{ACC} has to be adjusted accordingly. Also, if there are no explicit SeAs modeled in the SNS, an SeA is formed for each group of contacts that has access to the same information.

A comparison of the actual provider-specific model M\textsubscript{ACC} with the desired state as expressed in M\textsubscript{ACC} indicates whether M\textsubscript{ACC} has been applied correctly. If so, the global model’s links between entities and their representation represented in ACCOUNT can be updated. If M\textsubscript{ACC} was not applied correctly, M\textsubscript{G} should be updated accordingly to indicate the actual state in the SNS and to notify the user.

Thus composing the global model M\textsubscript{G} from the factual provider-specific models serves two purposes: Firstly, to indicate desired changes in the SNS that could not be carried out and secondly, to build M\textsubscript{G} for the first time.

Building M\textsubscript{G} from the n actual provider-specific models \{M\textsubscript{ACC1}, …, M\textsubscript{ACCn}\} begins by aggregating all SNS-specific SeAs in M\textsubscript{G} (SEA\textsubscript{G} = SEA\textsubscript{ACC1} ∪ … ∪ SEA\textsubscript{ACCn}) and forming according links between contacts and the particular SNS in the set ACCOUNT. As it is possible that the same contacts are represented in multiple SNS, they may now also be represented redundantly in different SeAs in SEA\textsubscript{G}. Multiple representations of the same contact can be identified trivially by the name, by attributes that have equal unique values [12] such as phone numbers or by manual user allocation, if necessary. If redundant representations for a given contact are found in other SeAs, they are replaced by that contact, thus possibly leading to overlapping or identical SeAs. Also there are now links from that contact to multiple SNS, which have to be represented in ACCOUNT.

For each SeA ∈ SEA\textsubscript{G}, the corresponding personas are created similarly to the personas in the provider-specific models, resulting in PER\textsubscript{G} and ALLOC\textsubscript{G}. Matching similar attributes across SNS is accomplished using a predefined dictionary, and similar attribute values can be identified by mere comparison. Similarly to identifying redundant contact representations, equal values for the same attribute can be merged, which leads to overlapping personas.

5. SNS-Support for an SIdM Solution

Our greatest concern regarding the feasibility of the presented approach is its applicability to currently important SNS. The two major obstacles to a successful implementation of a provider-independent SIdM tool are a lack of provider-supported SIdM and a lack of full API-access.

To evaluate the applicability of SIdM solutions, we surveyed ten SNS for their support of SIdM. We selected the seven sites with the largest user base as listed in [3] that are still in operation. Instead of Google’s still available SNS Orkut we chose its newly introduced SNS Google Plus. Also, we included the largest business-related SNS LinkedIn, the popular
microblogging site Twitter and Diaspora as a representative for decentralized approaches. Relevant information was gathered as of May 2011 (August 2011 for Google Plus) from the sites’ online documentation and by creating test user accounts.

5.1. Provider-Support for SIdM

Provider-support for SIdM is shown in Table 1 and refers to the SIdM features outlined in Section 4.3. The column “SeAs possible” denotes whether grouping contacts in the SNS to form audiences is available. “Selective attribute disclosure” refers to the possibility of disclosing the values of an attribute to selected contacts or SeAs only. Lastly, “Differing attribute values” indicates whether it is possible to show different values for the same attribute to different contacts. The entry “posts only” indicates that the feature is available for dynamically posted items such as status updates and uploaded pictures, but not for static profile data.

Table 1. Provider-support for SIdM

<table>
<thead>
<tr>
<th>Site</th>
<th>SeAs possible</th>
<th>Selective attribute disclosure</th>
<th>Differing attribute values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Myspace</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Facebook</td>
<td>Yes, overlapping</td>
<td>Yes, some</td>
<td>Posts only</td>
</tr>
<tr>
<td>Friendster</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Tagged</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Google Plus</td>
<td>Yes, overlapping</td>
<td>Yes, some</td>
<td>Posts only</td>
</tr>
<tr>
<td>Hi5</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bebo</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Twitter</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Diaspora</td>
<td>Yes, overlapping</td>
<td>Posts only</td>
<td>Posts only</td>
</tr>
</tbody>
</table>

Our findings regarding the availability of SeAs concur with a survey of privacy settings in SNS [3] that was conducted in 2009 and identified nine sites which allow grouping contacts into “friend lists” for attribute access decisions. Among these, the two sites with the largest user base according to [3] are Facebook, the current market leader and Windows Live Spaces, which has been discontinued as of March 2011 and is thus not included in Table 1.

Three of the surveyed SNS allow creating groups of contacts. The other sites do not distinguish between contacts for selective attribute disclosure and allow more coarse-grained settings at most, such as contacts, their contacts’ contacts (“friend of a friend”) and registered users. If selective attribute disclosure is available, it is mostly limited: In Facebook for example, the profile picture cannot be hidden and for some attributes, disclosure settings can only be defined in groups of attributes. In Diaspora, only posts such as status updates may be hidden from certain contacts, while static attribute values are disclosed to all contacts. Google Plus has the most permissive settings, providing a distinct disclosure setting for every posted item and attribute (except for the profile picture). Also, users may hide subsets of their contact list.

Disclosing different values to different contacts for the same attribute is only possible rarely and in a very narrow sense. In none of the surveyed SNS, it is possible to disclose differing values for static profile information such as one’s picture. However, in some SNS it is possible to share certain dynamically posted items such as photos with only a subset of one’s contacts.

Richer support for SIdM can be expected, as previous features have been introduced incrementally as a response to a still growing consumer demand. Further, there is added competition between SNS providers. Only recently, Facebook has announced more fine-grained SIdM settings as a response to Google Plus.

5.2. Provider-Access Using API

A possible solution for provider-independent SIdM must be able to perform read- and write operations on the user’s SNS profile, as described in Section 4.4. The most desirable and error-free way to implement this is to use APIs that are provided by the SNS. Still, missing APIs may be circumvented by extracting information from the SNS’ HTML-code and by using form fillers or resorting to manual user entry.

Table 2 indicates which kind of access is available using available API for the ten surveyed sites. For the user’s contacts, attributes and SeAs, it is indicated whether create (C), read (R), update (U) and delete (D) operations are possible, for contacts also if it is possible to add or remove (rem) them. For user data, it is indicated whether visibility settings for subsets of one’s contacts are possible. It is once again distinguished between static profile information and dynamically posted items.

Five of the surveyed SNS implement support for a version of OpenSocial, a standard for external applications that integrate into SNS and access SNS data. It aims to facilitate development of social applications that are compatible with multiple SNS. OpenSocial mainly focuses on web-based applications. Desktop programs may exchange data with the SNS through the REST-API which is also part of the standard.

Even though the individual adoption of OpenSocial by each SNS varies by the supported feature set and version, general observations can be made. OpenSocial specify various methods and data items to retrieve contact information and to read and update personal data. Through the person data object, numerous attributes are available and further attributes may be specified. Also, there are dedicated data objects for
photos and posts. Yet, there is no setting specifying the visibility of any attribute value or data object.

OpenSocial specifies optional methods to create and delete group-objects to categorize contacts, which appears promising to represent SeAs. The standard does not however specify means to add or remove contacts to these groups and it is also not described how to use them for selective attribute disclosure. Thus, upon further advancement of the standard, its API support for SIdM remains limited. LinkedIn, which has been supporting OpenSocial, and Twitter support a subset of the functionality provided by OpenSocial’s API.

We expect more open and rich APIs in the future, even though there may be contrary business related reasons such as keeping customers locked in. Among the driving forces are an increased competition between SNS providers and legislation addressing data ownership, such as the EU data protection directive.

6. Discussion and Conclusion

We have presented an approach for conducting SIdM consistently across multiple platforms. Unlike other approaches for decentralized social networking, our approach acknowledges the current fragmented landscape of SNS providers. It allows the user to conduct SIdM with contacts using existing providers and upcoming SNS alike. Thus the approach benefits the transition to future, more privacy-aware and possibly decentralized solutions.

The problem scope has been described in a formal fashion, which can be extended and used to clarify further discussion. Also, model transitions and the application to actual SNS have been defined. A survey of currently used SNS has shown that the approach is applicable in some usage scenarios.

Still, the implementation of the proposed solution depends highly on provider support of SIdM and available APIs. The survey in Section 5.1 has shown that SeAs are available in some SNS, including the market leader Facebook. Yet, even where SeAs are available, contact-dependent disclosure is not available for all attributes. Thus it is likely that not all provider-specific models can be applied without modification and tedious workarounds. Also, SNS have a history of changing their APIs, therefore possibly requiring future adjustments to an implemented solution. However, as the global model is provider-independent, only the particular connector would have to be changed.

As of now, the global and provider-specific models focus on attribute sharing and do not represent all entities and contents of today’s SNS. Complex attribute structures, ownership of items posted by contacts and permissions granted to them are examples for notions in certain SNS that are not included in our models. There is also behavioral data that occurs implicitly when using an SNS [21]. Not representing such notions can be seen as a limitation. However, aiming at representing the state within certain SNS as accurately as possible would also result in the global model getting more dependent on platform-specific conditions.

Also, not all real-world complexities can be expressed in the models, as for example contradictions depending on the content of certain attribute values, which cannot be anticipated. CONF, the list of conflicting attributes only contains anticipated conflicts based on the attribute’s type and not on its content.

<table>
<thead>
<tr>
<th>SNS</th>
<th>Access Contact Data</th>
<th>Profile Data</th>
<th>Posts (e.g. status update)</th>
<th>SeA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access Set visibility</td>
<td>Access Set visibility</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facebook (Social Graph)</td>
<td>R</td>
<td>R</td>
<td>No</td>
<td>C, R, U, D</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>R</td>
<td>R</td>
<td>No</td>
<td>C</td>
</tr>
<tr>
<td>Twitter</td>
<td>R, add, rem.</td>
<td>R, U</td>
<td>No</td>
<td>C, D</td>
</tr>
<tr>
<td>Tagged</td>
<td>No API available</td>
<td>No API available</td>
<td>announced</td>
<td></td>
</tr>
<tr>
<td>Diaspora</td>
<td>No API available (announced)</td>
<td>No API available</td>
<td>announced</td>
<td></td>
</tr>
<tr>
<td>Google Plus</td>
<td>No API available (announced)</td>
<td>No API available</td>
<td>announced</td>
<td></td>
</tr>
</tbody>
</table>

In Facebook, external applications access SNS data through the Graph API. It is possible to create and modify SeAs through the FriendList data object. The API allows reading profile information of one’s contacts as well as oneself through the user data object. The API does not allow modifying one’s own static profile information and the visibility of certain attributes, even though this is possible manually through the website. The API allows posting items such as status updates, pictures and activities and restricting their visibility to certain users and friend lists.

The API survey shows limitations in all SNS regarding data accessibility. OpenSocial, while promising due to its standardization efforts, lacks access to attribute visibility options which are also scarcely available in the supporting SNS. Facebook’s API allows access to posted items and their visibility settings but lacks those means for static profile information. For Google Plus an official API has been announced, but it is not available yet. Thus, at this time it is possible to implement only some usage scenarios using available APIs while having to resort to the aforementioned workarounds in other cases.

We have presented an approach for conducting SIdM consistently across multiple platforms. Unlike other approaches for decentralized social networking, our approach acknowledges the current fragmented landscape of SNS providers. It allows the user to conduct SIdM with contacts using existing providers and upcoming SNS alike. Thus the approach benefits the transition to future, more privacy-aware and possibly decentralized solutions.

The problem scope has been described in a formal fashion, which can be extended and used to clarify further discussion. Also, model transitions and the application to actual SNS have been defined. A survey of currently used SNS has shown that the approach is applicable in some usage scenarios.

Still, the implementation of the proposed solution depends highly on provider support of SIdM and available APIs. The survey in Section 5.1 has shown that SeAs are available in some SNS, including the market leader Facebook. Yet, even where SeAs are available, contact-dependent disclosure is not available for all attributes. Thus it is likely that not all provider-specific models can be applied without modification and tedious workarounds. Also, SNS have a history of changing their APIs, therefore possibly requiring future adjustments to an implemented solution. However, as the global model is provider-independent, only the particular connector would have to be changed.

As of now, the global and provider-specific models focus on attribute sharing and do not represent all entities and contents of today’s SNS. Complex attribute structures, ownership of items posted by contacts and permissions granted to them are examples for notions in certain SNS that are not included in our models. There is also behavioral data that occurs implicitly when using an SNS [21]. Not representing such notions can be seen as a limitation. However, aiming at representing the state within certain SNS as accurately as possible would also result in the global model getting more dependent on platform-specific conditions.

Also, not all real-world complexities can be expressed in the models, as for example contradictions depending on the content of certain attribute values, which cannot be anticipated. CONF, the list of conflicting attributes only contains anticipated conflicts based on the attribute’s type and not on its content.
Future work lies in addressing these current shortcomings by extending the models where necessary and appropriate. Also, we plan to implement a solution based on the presented models and transition rules and validate its user acceptance in a field survey.

7. Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. The research in this paper is receiving funding from the European Regional Development Funds (ERDF). The content of this publication is the sole responsibility of the authors and in no way represents the view of the European Commission.

8. References


