LIBERTY ALLIANCE PROJECT WHITE PAPER
Liberty ID-WSF People Service – federated social identity

December 5, 2005

Editor
Paul Madsen

Introduction

In today's online world, the oft-repeated saying 'it's not what you know, it's who you know' is very apt. More and more, the 'who you know' matters. Applications are evolving appropriately. The first generation of online transactions/interactions were single-user, eg. online banking, travel booking, shopping etc – they allowed that user to access their resources at different providers. More and more however, our online interactions involve others than just ourselves. People want to use the net with other people. For instance

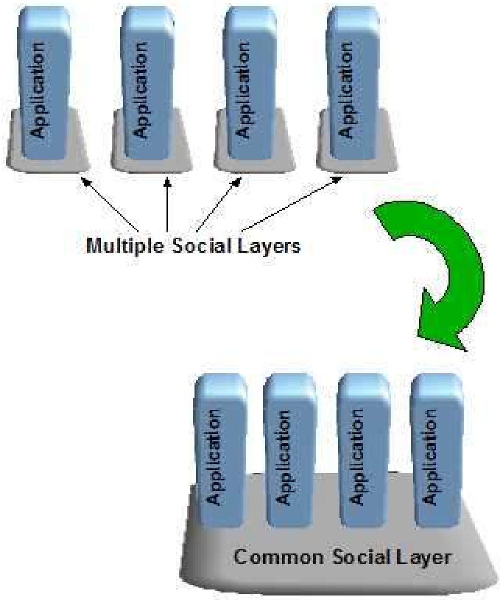
- the plethora of social network sites where a user can track and leverage their social network, ie. those people who they 'know' directly or indirectly through some other, for personal or business gain.
- 'Purchase Circles' on Amazon where focussed best seller lists are compiled based on some defined criteria, e.g. What people from a specific zipcode or company are buying.
- Instant Messaging lists of friends/buddies
- Contact Book software to keep your friends/colleagues informed when your details change after a move etc
- Sites where people are connected by shared interest, (e.g. in music or books) or some other criteria, (e.g. location, license plate numbers!)
- Social search engines give precedence to hits if 'recommended' by those in your social network (weighted by their proximity).

Whether it is communication, commerce, sharing, self-expression, or collaboration being enabled - all these interactions build on a social layer that connects individuals to others. Unfortunately, the current situation is that each of these applications generally builds its own view of a given individual's complete social network. This can result in duplication and undesirable management burden on those individuals, forced to maintain these multiple views. For instance, I can use one of the YASN (Yet Another Social Network) sites to track and grow my business network, use an integrated client with multiple buddy lists for IM, listen to a streaming music service, the channels to which I listen based partly on the preferences expressed by other users I specify, and can designate particular friends who can view my pics at my online photo storage.

Even ignoring the different interfaces and mechanics of the processes by which I build up my social network within each application, there is duplication and redundancy. For instance, I IM with various colleagues who are also part of the social network I maintain at my YASN yet I must maintain these connections in both applications. Additionally, any value extracted from one view of my complete network is non-transferable to another. For instance, if, through my YASN, I discover a potential contact, any relationship I establish with him/her through the YASN will not be automatically captured in my IM client.

Sometimes such separation between different aspects of your life is appropriate. Many people will want to keep work and family (and the different social networks for each) separate and compartmentalized. However, others may not, either for simplicity or because of the potential advantages of hooking together the networks. Fundamentally, such separation should be the individual's choice and not enforced by technological limitations.

The diagram below represents the current situation of 'social silos' – as compared to a future integrated social platform on which the different applications build.



The different applications (e.g. Bookmarks, blogging, photo sharing, IM) all plug into a common social layer for the relevant user.

These connections can be explicit (e.g. your friends, family, and collaeagues) or implicit (e.g. everybody who shares your passion for baroque flugelhorn music) and either direct or indirect (e.g. measured by the number of hops from yourself).

Many interesting interactions will involve those individuals who are both explicit and direct. For instance, you may wish to share your online photos with your family, or you may need to determine the network presence of your colleagues.

Enabling such direct interactions between users and their circle of friends is straightforward when both maintain an account at the same provider. On many online photo sites for instance, I can share my photos with others but only once they have established an account at the same provider. If the first user already knows the account name of the other, all that need happen is for that name to be supplied. If they don't know it, they might search existing accounts or, if necessary, have an invite sent to their friend encouraging them to create an account.

While this model is advantageous for boosting the number of account holders at the photo site, there are two significant implications:

1) both users must maintain or establish accounts at the same provider. Typically, the result of this requirement is that the friend being invited to interact (e.g. View vacation photos, etc) is forced to create an account (with associated logins and passwords to remember) at a provider where they might not otherwise choose to do so.

2) if some connection between two friends is established in the context of the photo site, it can't be leveraged in some other context (e.g. Calendar sharing) unless that provider happens to host both services.

Enabling such cross-user interactions such that the above two implications are addressed is the goal of the Liberty Alliance's People Service. The People Service provides a flexible, privacy respecting framework by which one user can manage/track the people they know - typically but not exclusively in order to assign them certain privileges for accessing certain resources owned by the first user. Providers query/manipulate this information through standard web service interfaces.

Before discussing the model and components of the People Service, we present a typical scenario in order to derive key requirements of the People Service functionality.

Scenario

Alice maintains her photos at photos.example.com. Upon returning from a vacation in the Caribean, she uploads her latest photos to her online store, creating a new album called

'Vacation Pics'.

Having previously been forced to sit through 3 hours of photos from her friend Bob's recent Grand Canyon vacation, Alice wants to return the favour and share her own pictures with Bob.

Any system to enable Alice sharing her photos with Bob will be constrained by the following preferences Alice and Bob have.

1. Alice does not want Bob to see all her other photos, just those within the 'Vacation pics' album.
2. Bob does not have an account at photos.example.com and does not wish to create one (not even for the chance of seeing Alice's inumerable blurry sunset shots).
3. While Alice knows Bob's email, she knows/expects that Bob does not want this shared with photos.example.com.
4. Alice expects that, in the future, she will likely grant Bob access to other of her online resources (.e.g. allow Bob to be informed of her online presence or allow his online contact book to automatically sync changes to Alice's contact info). Consequently, she wants to be able to record the fact of her 'connection' with Bob such that it can be leveraged in the future.

Requirements

From the above scenario, we extract the following requirements:

1) A resource owner must be able to assign permissions for their resources in a controlled and granular manner.

While Alice could simply have given her account name and password at photos.example.com to Bob, doing so would give Bob complete access to all of Alice's photos, and not just those in the 'Vacation Pics' album whch she wishes to share. Although no system based soley on passwords can prevent such sharing, it should not depend on it.

2) It must not be necessary for the user to whom permissions are being granted to establish a credentialled (i.e. with an account name and password) account at the service provider maintaining that resource.

IF Bob were to be encouraged to create an account at photos.examples.com, then the relevant permissions could be set against this new local account. The cost for Bob however would be yet another account for him to remember. The corollary for this requirement is that it must be possible for Bob to reuse an existing account held at some other provider in order to authenticated to the service provider.

3) It must be possible for the user to whom permissions are being granted to be notified of this fact without their contact info (e.g. email address, phone number) being disclosed to additional parties.

Alice does not provide Bob's email to photos.example.com and ask that Bob be notified of his ability to see the photos. Rather, Alice requests that photos.example.com create an invitation for Bob that Alice will send herself. The first option may be acceptable in some scenarios or for some users, it shouldn't however be the only supported mechanism.

4) Once a user has been granted permissions for resources maintained at one provider, it should be possible (but not required) for any subsequent invitation from other providers to leverage the 'connection' that was established.

After Bob acknowledges the invitation to view Alice's photos, it should be posible for Bob to allow the relationship (the fact of it but not its nature) between himself and Alice to be recorded so that the process of responding to subsequent invitations to access other of Alice's resources can be streamlined.

Solution Components

The requirements that were derived in the previous section have driven the definition of the People Service. We present here the different components and how they can meet these requirements.

People Service

A Liberty ID-WSF People Service (PS) provides SOAP-based interfaces into the list of other individuals with which a user wishes to enable some online interaction. Service providers interested in enabling such interactions can leverage this list rather than maintaining such information locally.

The PS allows a provider to query the list of members, add a new member, and delete an existing member (all of which assuming appropriate authorizations exist etc). A people service provider also allows users to categorize the members of their people service list into *groups*.

The following XML represents a SOAP request in which an SP is asking that a new group called 'Work Friends' be created.

```
<AddCollectionRequest>
    <Object>
        <DisplayName>Work Friends</DisplayName>
    </Object>
</AddCollectionRequest>
```

Once the above group had been created, individuals (as well as other groups) could be added to it as children.

Invitations

Invitations refer to a message communicated to a user indicating that another user wishes to interact with them online in some manner (e.g. allow the invited user to view some online resource, include the invited user in some grouping, etc).

Invitations are created by an provider on behalf of the inviting user and then communicated to the invited user - either by the inviting user or by that provider (if it is somehow able to determine an appropriate address to send the invitation to).

Invitations serve two purposes:

1. Inform the invited user of the details and potential implications of the desired inteeraction being enabled and obtain their consent.
2. Open communication with the invited user with the goal of federating them with their appropriate identity provider.

Invitations may contain a URL to which the invited user will respond to the SP. At this URL they will be able to obtain more information about the invitation, provide any necessary consent, and initiate any requried federations between their identity provider and other providers. If it happens that the invited user already has an account at that provider and are willing to use it, then the provider can simply assign appropriate privileges against that login name.

Alternatively, the invitation may contain a URL which, when presented by the invited user to their identity provider (likely by copy-and-pasting into some form) , will enable the establishment of the necessary federations, albeit from the 'other end'. This alternative invitation model will guard against the possibility of users being phished through imitation invites with embedded URLs.

Identity Federations

Whether or not the resource is theirs or that of a friend, if a user is to be able to reuse an existing account at some identity provider in order to access a resource maintained at some different service provider, the two providers must be able to agree on some identifier for that user. Once this agreement is in place, the user will, after authenticating to the identity provider, be able to navigate to the service provider and be granted appropriate permissions based on this shared identifier. This is often described as Single SignOn (SS0) – the user signs in once (at the identity provider) and is then, based on this authentication, able to be recognized at the service provider and be granted appropriate permissions. Liberty's protocols allow two providers to establish a shared identifier for a user (sometimes referred to as *federating* that user) between themselves.

By suppporting the identity provider of the invited user ebing able to federate that user with the service provider, this model ensures that that user can reuse their identity provider account and not be required to create and maintain a new account at the service provider.

Identity Mapping

If a particular user is a member of another's PS list, the implication is that the first user has been federated between their own identity provider and the second user's PS provider.

This allows these two providers to 'talk' about the relevant user. But, the people service provider itself will not be involved in any interesting interaction between this user and that whose people service list they belong to, any real interaction (e.g. Photo sharing, presence, contact book syncing, geolocation determination, etc) will be between some appropriate service provider and the user's identity provider.

Consequently, the people service provider has important roles to play both in facilitating the establishment of, and/or performing appropriate mappings, for the federated identifiers that exist between identity providers and the various service providers. For instance, when a user indicates to a 'Find a Friend' service that they wish some existing member of their people service list to be allowed to see their current geophysical location, the people service provider can facilitate the process by determining if the relevant user has already been federated with the 'Find a Friend' service, and if so, return the relevant identifier so that permissions can be assigned against it.

Groups

Often, the individuals that are added to a user's people service list will share some common characteristic such that it will be simpler if they treated and managed collectively as a group rather than separately. For instance, a user may set access control permissions for her online calendar as 'Let all members of my soccer team view my weekend entries' rather than defining separate rules for each member of the team. Or, a user may ask a party planning service to send an invitation for a Christmas party to 'everybody labelled as a work friend'.

Groups will provide significant value by allowing a user to assign access rights to a number of friends/colleagues in a single step. For instance, a user will be able to indicate to a service provider, 'Allow all members of my 'Work Friends' group' to have right access to my Work Calendar'. Subsequently, when some user tries to access the 'Work Calendar', the hosting service provider will ask the appropriate people service 'Is this user a member of the 'Work Friends' group. The answer will determine whether or not the access is granted or not. The alternative would be to assign this privilege individually, and then update it appropriately as group membership changes.

People service groups are similar to email client software distribution lists and contact book lists. The key difference is that an entry in a people service is not a static set of data, but rather a representation of a dynamic connection between the two users, manifested in a federated identifier established between the people service provider and appropriate identity provider.

Scenario Revisited

With the components of the people service model presented, we can reexamine the photo sharing scenario in a PS-enabled world. Many of the user interactions for Alice and Bob will be the same as in a non

1. Alice goes to photos.example.com site and indicates she wants to share a photo with a friend.
2. photos.example.com discovers the location of Alice's PS (through standard Liberty mechanisms) and, once determined, sends a query there for the members of Alice's list

of friends.

3. After determining that photos.example.com is authorized to act on Alice's behalf, Alices's PS returns the list of members to photos.example.com which displays the list to Alice.
4. As this list is composed of individuals with whom Alice has previously established an online connection, and this is the first time she has reached out to Bob, he is not in the list. Alice asks photos.example.com to request that 'Bob' be added to her list. photos.example.com sends the appropriate request to Alice's PS for Bob to be added.
5. Alice's PS returns a URL to which Bob should be directed if and when he responds to the (upcoming) invitation.
6. photos.example.com creates an invitation for Bob indicating Alice's desire to share her photos. This invite is made available to Alice (e.g. (in an HTML page for copying) so that she can communicate it directly to Bob. Alice emails the invite message to Bob. Alternatively, photos.example.com could have sent the invited directly if Alice had provided Bob's email.
7. Bob clicks on the URL within the invite message and is taken to photos.example.com where he can get more information about the nature of the invitation. If he consents to proceeding, and to allowing Alice to record the connection between them, he is directed to the URL Alice's PS previously supplied.
8. Alice's PS asks Bob if he would like to establish a linkage with an account he maintains at some other identity provider. If Bob consents (and assuming that the necessary business relationship exists), Alice's PS and Bob's identity provider establish this linkage (in the lingo, they *federate* Bob).
9. Alice's PS can now ask Bob's identity provider for a identifier that photos.example.com could use for Bob. Bob's identity provider generates such an identifier, encrypts it so that Alice's PS can't see it, and returns to Alice's PS.
10. Alice's PS delivers this encrypted identifier for Bob to photos.example.com which, after decrypting, assigns appropriate privileges to this new identifier.
11. The next time Bob accesses photos.example.com (by first signing on at his identity provider), photos.example.com will recognize him as somebody to Alice has assigned specific permissions, and he will be able to view the relevant photos.

Comparison to Other Social Initiatives

FOAF

Friend of a Friend (FOAF) is an activity of the Semantic Web. FOAF is a standard by which individuals can describe themselves, the things they do, and the people they 'know'. You create one or more FOAF files on your Web server and share the URLs so software can use the information inside the file.

Individuals maintain their FOAF files 'close' to themselves and, at least currently, must have a certain familiarity with web technology. At least currently, the FOAF file is typically created by hand in a text editor or generated from a tool (web-based typically) and then uploaded to a URL at which it can be discovered and parsed.

There is nothing inherent in the FOAF model that would prevent the details of these steps from being hidden from non-technical users. So, we could imagine FOAF-providers hosting FOAF files on behalf of users (and hiding the XML syntax from them).

YASNs

In Social Networks, individuals delegate the storage of their social relationships to a specialized provider. The provider maintains a similar set of relations between the various users who have accounts at that provider (likely as database tables rather than XML. Interestingly, more and more of the YASNs are adopting FOAF as an import/export format.)

All the users establish an account at the same social network provider and use this account to manage (e.g. explore, add, etc) their social connections. There is no need for individual FOAF files to be spidered in order to construct the social network, the provider's database maintains it in real time.

If any of the users were to maintain an account at another YASN, the different corners of their social network would be disconnected. For instance, if a user were to maintain their family/friends connections with one YASN, and their work connections with another, no connections could be inferred across the chasm (which may or may not be desirable). Additionally, for those connections that belonged in both worlds, they would need to be maintained separately in both YASNs.

Importantly, no 'pure' YASN provides a programmable interface into the data they store on behalf of their users. The implication of this is that if an application is to be built on top of the social layer, then it must be hosted by the YASN itself. So, for instance, there are providers that provide services such a photosharing, bookmarks, and blogging on top of a layer of social connections. Providers such as these more and more provide such APIs but they focus on their core business rather than probing the social layer itself.

People Service Model

The people service model can be imagined as somewhat of a hybrid of the FOAF and YASN models – a specialized provider helps individuals maintain and grow their social connections (the YASN aspect) but in a more distributed manner (the FOAF aspect).

As before, the same social connections (e.g. who knows who) are maintained. The people service provider will, similarly to a YASN, provide mechanisms/UIs by which the users can edit their networks. Similarly to the FOAF model however, the complete network is distributed across multiple providers and not centralized to a single one. This distribution of data (whilst still allowing a complete picture of the network to be established) provides improved privacy control versus the base single-provider YASN model.

Importantly, the people service provider, in addition to allowing the user themselves to interact with their social network, supports web service interfaces by which other providers can similarly interact (e.g. query and add).

Summary

Around the world, there appears to be a maximum value for the size of the social groups in which humans live and play. From hunter gatherer clans to communes to military units, groups normally top out around 150 members.

It seems that humans have a built in limit to the number of relations, friends, and colleagues we can conveniently keep track of on our own. Many evolutionary psychologists believe that this numerical limit on group size is coded into our brains – the large size of our brains relative to our primate cousins having evolved in order to allow us to keep track of such relationships (and, importantly, which among them are friend or foe).

Perhaps it's fortunate then that there was no People Service available to our ancestors as they descended from the trees onto the African savanna – who needs a brain when you have the Web?