Privacy and Access Control in Federated Social Networks

Animesh Pathak[1], George Rosca[1], Valerie Issarny[1],
Maarten Decat[2], and Bert Lagaisse[2]

[1] Inria Paris-Rocquencourt, France
firstname.lastname@inria.fr
[2] iMinds-DistriNet, KU Leuven, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. Online social networks (OSNs) are increasingly turning mobile and further calling for decentralized social data management. This trend is only going to increase in the near future, based on the increased activity, both by established players like Facebook and new players in the domain such as Google, Instagram, and Pinterest. The increasing adoption of social networks in the workplace has further led to the development of corporate social networks such as those provided by Yammer, which was recently acquired by Microsoft. As individuals from different companies will need to interact as part of joint teams in these federated social networks, questions of privacy and access control arise. This chapter identifies the challenges concerning the above aspects, surveys the state of the art, and identifies directions of future research.

Keywords: social networks, access control, privacy, federation.

1 Introduction

As recent trends show, online social networks (OSNs) are increasingly turning mobile and further calling for decentralized social data management. This trend is only going to increase in the near future, based on the increased activity, both by established players like Facebook and new players in the domain such as Google, Instagram, and Pinterest. Modern smart phones can thus be regarded as *social sensors*, collecting data not only passively using, e.g., Bluetooth neighborhoods, but actively in the form of, e.g., "check-in"s by users to locations. The resulting (mobile) social ecosystems are thus an emergent area of interest.

The recent years have seen three major trends in the world of online social networks: *i)* users have begun to care more about the privacy of their data stored by large OSNs such as Facebook, and have won the right (at least in the EU [1]) to remove it completely from the OSN if they want to; *ii)* OSNs are making their presence felt beyond casual, personal interactions to corporate, professional ones as well, starting with LinkedIn, and most recently with the purchase by Microsoft of Yammer, the enterprise social networking startup [2], and the launch of Google Plus for enterprise customers [3]; and *iii)* users are increasingly using the capabilities of their (multiple) mobile devices to enrich their

M. Heisel et al. (Eds.): Engineering Secure Future Internet Services, LNCS 8431, pp. 160–179, 2014.

social interactions, ranging from posting cellphone-camera photos on Instagram to “checking-in” to a GPS location using foursquare.

In view of the above, we envision that in the near future, the use of ICT to enrich our social interactions will grow (including both personal and professional interactions [4]), both in terms of size and complexity. However current OSNs act mostly like data silos, storing and analyzing their users’ data, while locking in those very users to their servers, with non-existent support for federation; this is reminiscent of the early days of email, where one could only email those who had accounts on the same Unix machine. The knee-jerk reaction to this has been to explore completely decentralized social networks [5], which give the user complete control over and responsibility of their social data, while resorting to peer-to-peer communication protocols to navigate their social networks. Unfortunately, there are few techniques available to reconcile with the fact that the same user might have multiple devices, or that it is extremely resource-consuming to perform complex analysis of social graphs on small mobile devices.

Our view lies somewhere in the middle of the two extremes, taking inspiration from the manner in which users currently use email. While their inboxes contain an immense amount of extremely personal data, most users are happy to entrust it to corporate or personal email providers (or store and manage it individually on their personal email servers) all the while being able to communicate with users on any other email server. The notion of **Federated Social Networks** (FSNs) —already gaining some traction [6]— envisions a similar ecosystem where users are free to choose OSN providers which will provide storage and management of their social information, while allowing customers using different OSN providers to interact socially. Such a federation can be beneficial in three major ways, among others: *i)* it allows users to enjoy properties such as reliability, availability, and computational power of the hosting infrastructure of their choice, while not being locked down in terms of whom they can communicate with; *ii)* much like spam filtering services provided by modern email providers, that are tuned by feedback from their users, FSN users can benefit from the behavior of others sharing the same OSN provider[1]; and *iii)* this fits perfectly with enterprise needs, where ad-hoc teams can be formed across corporate OSN providers of two organizations to work on a joint project.

1.1 Illustrative Example and Challenges

As an example of the circumstances discussed above, let us consider two organizations, companies A and B, which already use social networking platforms internally, but want to allow some of their employees to collaborate together as part of a joint team in order to achieve some goal (Working group B in Figure 1). This will involve exchanging messages, publishing shared contents, documents, but also participating in events, etc. Additionally, Company A uses a third party solution for behavior analysis of their employees based on their socializing logs

[1] This also gives an incentive to commercial OSN providers to provide value-added services.

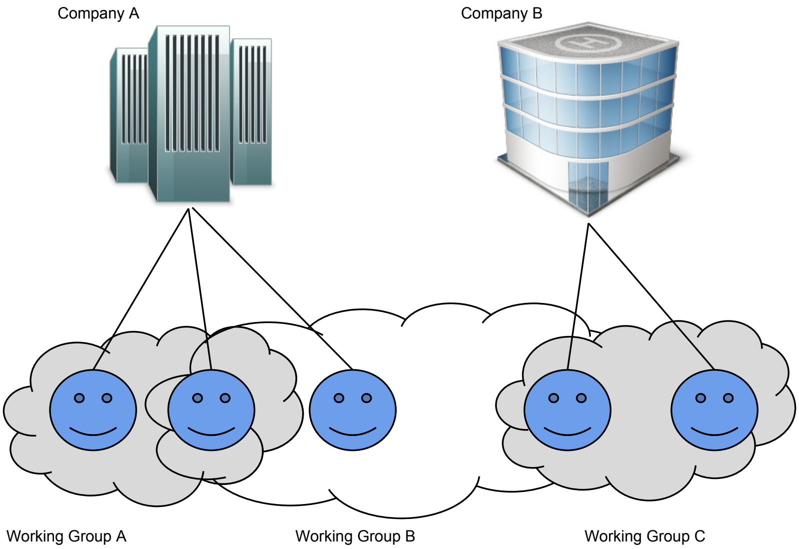

Fig. 1. An Example for Federation Among Enterprise Social Networks

(e.g., for suggesting team constitutions to managers for future projects). Alice from company A and Bob from company B are assigned to this working group, and add each-other to their contacts. Later, Alice shares a private document with her contacts. From a trust and privacy perspective, the following questions may arise:

- Can the system warn of leaks caused by interaction of certain type? Remembering that Alice shared the document with her contacts, from which Bob is part of, should he be able to see it? How can the system either warn or prevent such an incident?
- If Bob adds comments to the private document which Alice shared, are those comments subject to the same access control policy? Can we pre-determine that the social networking platform of company B will restrict access to these comments only to those members of company B who received Alice's initial post?
- Can company B be assured that their employees interaction within the common project will not be analyzed by A's third party solution?
- Is sharing information outside the company a decision held by the user itself or network administrator?

Clearly, for addressing questions such as those posed above, expressive, flexible, yet easy-to-specify privacy and access control policies are needed so that users can feel safe as more and more of their (social) life is made available online, thanks in large part to mobile clients for OSNs. As we have discovered in the course of our recent work, most current policy and trust frameworks are unable to adequately address the complexity of social networks, resorting to simple role-based access control. The need of the hour is a privacy and access control framework founded on the clear data and interaction models discussed earlier. Such a framework will also allow OSN providers to adequately evaluate whether or not a certain data of a user should be shared with another OSN provider (e.g., replies to a Facebook wall post have the privacy settings of the original post, while replies to a status message on Twitter have those of the user posting the reply, thus rendering them incompatible). Equally important is the availability of such techniques in mature, ready-to-deploy software platforms.

This chapter presents the reader with a set of requirements (Section 2, followed by a survey of the state of the art in social networking solutions, with a special focus on their ability to support rich privacy and access control policies in federated settings (Section 3). Through this extensive analysis we offer a broad vision on existing social networking platforms, protocols involved but also their privacy and access policies. By doing so, we identify the main components of a federated social platform together with presenting the current trends in standards and security paradigms underlying actual open source solutions which offers their implementation. Section 4 provides recommendations on constructing such systems. We then conclude in Section 5 with directions for future research.

2 Social Networking Platform Requirements

Before presenting the survey of social networking platforms, proposed in literature as well as available on the market, we introduce the criteria which form the basis of our assessment:

- **Person to Person links**: We will distinguish between symmetric and asymmetric 'friend' relationships among resources of type 'Person' (users). Needless to say, in order to semantically describe social ties between people we address a more realistic approach of being able to model both symmetric friendship like those seen in traditional OSNs like Facebook (where one is "friends" with all their friends) but also asymmetric links (e.g.'follow', 'knows', etc.).
- **Ease of Application Development on the Platform**: We highlight, if needed, the programming language, license, the API offered, native mobile support and the object model. The object model here refers to what kind of social resources and the connections between these the system utilizes (e.g. groups, events, etc.) with an emphasis on the ease of use but also the ability to extend this model when it comes to creating applications on top of the platform.
- **Federation Support**: Allowing the interaction between various decentralized systems raises the need to establish or make use of existing open protocols on which all these systems must comply in regards to information exchange. These protocols must provide identity, data interoperability and real-time communication.

- **Privacy and Access-control Policies**: Between individuals or communities access policies must be defined. Towards this direction a decentralized system must support a comprehensive set of mechanism which enable fine-grain control over the users who will have access to the data generated within such systems.

3 Existing Platforms

We now describe, based on the criteria identified previously, existing platforms together (summarized in Table 1). Broadly, we categorize social networking platforms as follows:

3.1 Siloed

Siloed social networks are the most common type found in commercial social networks open to the public. In the systems below, all the users share the same social networking service provider, and can not usually interact with users of another provider.

Facebook. Currently one of the leading commercial online social networking platforms, Facebook [7] offers a high level of API maturity allowing a large variety of application to be built on top of it, both online but also mobile specific. It offers a predefined data model which does not allow class extensions, offering the ability only for resource of type 'content' to be customized based on one's needs as depicted from their Open Graph API ('custom stories'). Between users the notion of friendship is symmetric while it allows support for asymmetric 'follow' links acting as a subscription which aggregates data on the main activity feed ('timeline'). It provides native application support for mobile environments so that applications build on top of the Facebook platform can benefit from the Single Sign On feature for authentication while also enabling traditional OAuth [8] through thin clients as well. It has a full-fledged mature API client and search capabilities but it does not allow federation since users of this platform are limited to interact with other users under the same centralized authority.

Privacy and Access Control. Since Facebook does not support federation no Server-to-Server rules are supported; it provides a robust access control mechanism for both the user and his data but also policies for third party applications build on top of the Facebook platform which might use sensitive user information. In regards to sharing data it offers a role-based policy mechanism (e.g. share with custom list), while in terms of data re-sharing it preserves the originator's policy. Note that in Facebook, tagging people in pictures will extend the visibility of those causing a leak of information, though when a private content is shared tagged comments will not have the same effect.

Twitter. Twitter's [9] online social networking platform is considered to be a device-agnostic real-time message-routing infrastructure which relies on the well known Redis [10] framework. Its object model is rather limited, it does not have events or groups but the friend relationship is asymmetrical ('follower' or 'followee'), while for authentication it offers OAuth support. The challenges which Twitter as a platform addresses are real-time syndication of content among connected users. It offers the ability to build applications on top of their platform providing only thin web clients for mobile and desktop environments.

Privacy and Access Control. In Twitter social networking platform the user can control by whom is he followed and each individual post's visibility which can be either public or visible by the ones who are following the user. However when a 'tweet' is private that can not be re-tweeted which means that re-sharing of the data is, in some ways, protected according to the originator's policies.

Mosco. Though Mosco [11] as a social platform is intended to be for portable devices ('middleware for mobile social computing') its architecture (see Figure 2) is mixed between cloud (Google's App Engine) and mobile implementation, having a rather limited basic model (stored in databases) with the ability to extend. The entities model is depicted in Figure 3 which better shows the connection between them and also which entities can be extended: `AbstractPrivacyData` for enriching the privacy policy access control manager and `AbstractData` for new object types with no ability however to define new connection between the resources available. The `AbstractData` extensions can be then accessed via SQL-like queries.

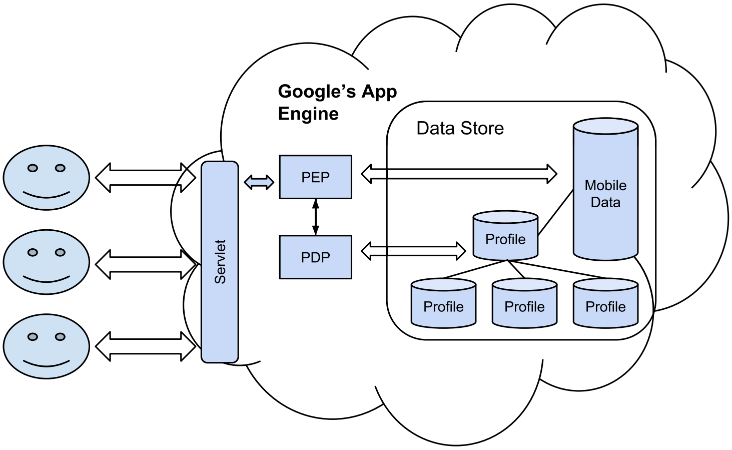

Fig. 2. Mosco Architecture Overview

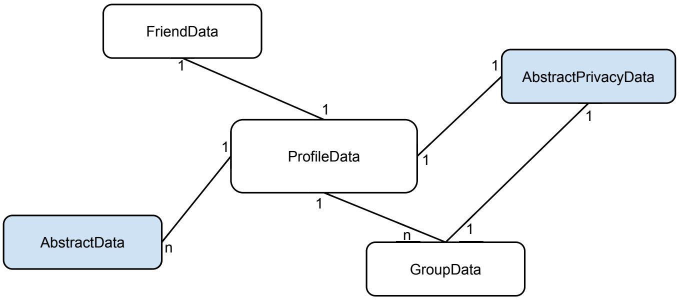

Fig. 3. UML diagram representing main entities in the data model present in Mosco platform. The shaded entities are to be extended when implementing a new application.

Privacy and Access Control. The complex and flexible access control policy manager of Mosco is an extension of the popular XACML [12] with a set of predefined policies suitable for social computing. As an example it allows users to create context-specific policy rules like sharing the current location with people in the immediate proximity or patient records when some threshold is reached. In order to define new privacy rules application developers must extend `AbstractPrivacyData`.

3.2 Social Networking as a Service

These platforms employ the Software as a Service (SaaS) paradigm, thus enabling organizations to define their own networks or domains, enabling individuals from different organizations to co-operate. That said, the data and logic is hosted in most cases under the control of the service provider.

Google Plus. Google's online social networking platform is indeed similar with other of its kind (e.g. Facebook), but it is the one which introduced the notion of *circles*, differentiating itself with the asymmetric relationship between users. The object model has no option of extending it but only to customize the objects maintained. It allows application building on top of their platform both online and mobile, having also native support for the most popular portable systems, but also web clients to use based on one's needs. It also offers a Domain API for enterprise social networks in that sense being similar to Yammer (see below), offering domain name support for individuals but more specifically for companies and enterprises. The authentication method which applications can use is OAuth.

Privacy and Access Control. The access control policy mechanism is similar to other social networking platforms of its kind (role-based) offering also support for establishing per-domain access control rules which can scope the visibility of content within a domain. Google Plus mixes between traditional OSNs and Enterprise Social Networks (ESNs) by offering the ability for network administrators to specify domain specific policies as well as domain-wide delegation of authority. In that direction the scope of posts within a domain can be limited to be only visible inside the organization. In terms of data re-sharing outside the organization it is believed that the decision to allow data outside the domain should reside with the user rather than administrators. When a private content is being shared a simple comment with a tagged person will extend the content's visibility making the comments but also the original post available for the one who has been tagged.

Ning. Ning [13] is an online social networking platform which allows people or organization to create their own customized micro-blogging network which will primarily be hosted on a subdomain of Ning. In terms of social resources it has a limited model containing user profiles, groups, pictures, messages, contents without the ability to extend or define new resource types. It implements OpenSocial [14] protocol which allows the creation of applications which are able to interact with the platform. It allows applications to be build for mobile using Javascript and HTML5, so no real native support is available. Since it is a commercial software it enables easy creation of networks inside Ning without needing any programming experience (drag and drop) allowing users from different networks (or subnetworks) to interact as if they were in the same network.

Privacy and Access Control. From a privacy perspective, Ning offers its users fine-grain access control, providing granular content moderation allowing anyone, just friends or members of the same network to view information. Also, in the same manner, users are able to choose who can comment on their content or even moderate which comments can appear attached to their content or information.

Yammer. Yammer [15] is the leading software in enterprise social networking platforms. It offers an Open Graph API with an actor-action-object structure (as described in Figure 4) which is extensible and allows the description of any kind of fact, offering the ability to describe new object types under different namespaces. It offers virtual storage for companies and easy deployment and installation with further interaction between users on different companies further maintaining their privacy policies (NDA). From a UI perspective Yammer maintained the same pattern as Facebook which they have identified as being the 'DNA for socializing' so that user adoption will be much easier. Cross-domain collaboration can be achieved allowing companies to establish communities with their customer in a secure manner thus providing federation among deployments of Yammer.

Privacy and Access Control. It offers support for SAML [16] 1.1/2.0-based Single Sign On mechanism supporting also OAuth both for desktop and mobile environments. It provides TLS encrypted e-mail transport, session management and built-in logical firewalls for the data centers. What is different from Google Plus Domain API is the fact that the user starts in a private network and they can collaborate with other corporate networks if invited.

```
{
    "activity":{
        "actor":{
            "name":"Sidd Singh",
            "email":"sidd@xyz.com"
        },
        "action":"create",
        "object": {
            "url":"https://www.sched.do",
            "title":"Lunch Meeting"
        },
        "message":"Hey, let s get sushi!",
        "users":[{
            "name":"Adarsh Pandit",
            "email":"adarsh@xyz.com"
        }]
    }
}
```

Fig. 4. Example actor action object JSON code structure in Yammer

3.3 Federated Social Networks

Federated social networks are networking services that allow interactions between users across distinct social networking service providers. However, their architecture is not completely distributed since the users in each network still depend on servers whom they must trust regarding the processing of sensitive data.

Status.net. One of the most powerful microblogging social networking platform, Status.net [17] (formerly Laconica) is a ready-to-deploy decentralized solution, written in PHP, which can be accessed via multiple standard protocols including e-mail, sms, XMPP. Formerly it has been supporting identi.ca [18] and pump.io [19], but since late December 2012, the latter decided to change its infrastructure to NodeJS from performance reasons, while maintaining the same concept of microblogging making use of ActivityStrea.ms. It also implements OStatus which allows notifications of status updates between distributed social platforms including Friendica. For discovery it offers an implementation of WebFinger [20] protocol. Its data model contains groups, asymmetric relationships between people, being extensible through ActivityStrea.ms. It also provides 1 - 1 messaging support. It also offers support for updates through XMPP, cross posting to Twitter, Facebook integration. It also implements the Salmon protocol which allows the unification of conversation through content from different servers to happen. There is no native support for mobile environment but it has an Open Source client for both Desktop and Mobile based on the Appcelerator [21] platform.

Privacy and Access Control. It implements OpenID [22] for identity, but offers support for Apache Authentication which allows any kind of such mechanisms to be integrated. The access control policies are limited to role-based policies as well as domain specific policies.

Friendica. Friendica is a decentralized open source social networking platform which provides fully distributed protocols for secure communication such as DFRN [23] or Zot [24], the two complementing each other. It supports LDAP [25] for authentication having a limited data model which can only be extended by the support of server side plugins. As an example of the latter it offers plugins for displaying locations on the map or connectors for popular social networks such as Twitter or Google Plus. It does not offer native mobile support but since their API is similar to Status.net, the latter's mobile clients can be used along with existing Friendica's available clients. Since it was intended for small networks, in order to solve the scalability problem they have introduced Red [26] which is addressed for companies and organizations in which case it dramatically reduces the abilities in cross-service federation.

Privacy and Access Control. In terms of security, Friendica offers both server-to-server but also one-to-one advanced message encryption, while all the items (messages, posts, etc.) are controlled by a fine grained access control mechanisms. Groups can also have specific policies which are applicable to all the members contained, profile visibility can as well be controlled by the individuals, together with its data which can easily be backed-up on home computers.

Diaspora. The open source decentralized social networking platform Diaspora [27] addresses the privacy concerns related to centralized social networks allowing users or developers to deploy their own server solution thus interacting with other users from other deployment. It offers social aggregation facilities by importing data from Twitter, Tumblr and Facebook. Written on Ruby on Rails under AGPLv3 license, Diaspora has a fixed social data model without the ability to extend it. Regarding mobile integration there is no native support but there are a couple of web clients which can be used, without allowing applications to be developed on top of the middleware. In terms of federation, Diaspora facilitates this by providing an implementation of Salmon [28] protocol and for discovery it provides support for the WebFinger open protocol.

Privacy and Access Control. Diaspora offers a fine grained aspect-oriented access policy mechanism. This provides the ability to control posts' visibility to either public or limited, which is the traditional role-based but named 'aspects' in this case. It has some already built-in 'aspects' such as friends, family or co-workers but other lists can be constructed as well.

OneSocialWeb. OneSocialWeb [29] is an interesting social networking platform licensed under Apache 2.0, having a communication layer relying on XMPP which allows federation to be achieved much easier. Though the code base is not maintained anymore, it allows an already to deploy solution on the server side with the possibility to use existing clients for mobile devices. It implements ActivityStrea.ms protocol as for data modeling and an activity based policy mechanism which ensures flexibility in terms of storage, offering an implementation of OpenID for authentication.

Privacy and Access Control. Like in any other social networking platform OneSocialWeb offers the ability to control the access for individual posts, profile items or even relationships. It is interesting to note that their mechanism is fine grained in the sense that you can define the subject or 'accessor' of the information which can be either a contact, a group, people from a certain domain, everyone or a specific individual. Also, the action performed on the data can be customized which can be read, write, delete, update or append. Some real examples would include: a post visible to everyone but only friends can add comments or a public photo album which only family can edit.

Buddycloud. buddycloud [30] is a decentralized open source social platform, licensed under Apache 2.0. Working in collaboration with W3C, Mozilla Foundation and XSF [31], they offer multiple open standards such as ActivityStrea.ms, ATOM syndication format and XEP [32] which is an extension of XMPP protocol offering useful functionalities such as discovery. They offer an easy to install federated server side code base, written in node.js (offer a version in java as well) and as for mobile support an Android client is provided which relies on Backbone (JavaScript library). It offers messaging support including a couple of other useful social engines such as recommendation, real-time search, resource discovery and push notification. The data model is rather limited (e.g. it does not contain events or groups) having an asymmetric relation between users, but it does give you the ability to extend the basic model in some ways by making use of ActivityStre.ms. Users will authenticate via traditional basic HTTP method with the option for using a secure connection. A summarization of the platform architecture is depicted in Figure 5.

Privacy and Access Control. Users can share almost anything through media channels having the ability to limit posts visibility through a rather simple access control mechanism which allow black/white listing. Also, it provides a 'butler' which enables users to securely share their location with friends. It provides support for SSL/TLS communication for both client - server and server - server communication so that user's privacy will be preserved.

ELGG. ELGG [33] is one of the most popular PHP open source social networking software platforms which is easy to deploy and configure, providing a large variety of components for individuals and companies having an already stable community with lots of already-made plugins for different purposes. Its architecture is decentralized in the sense that multiple federated ELGG server

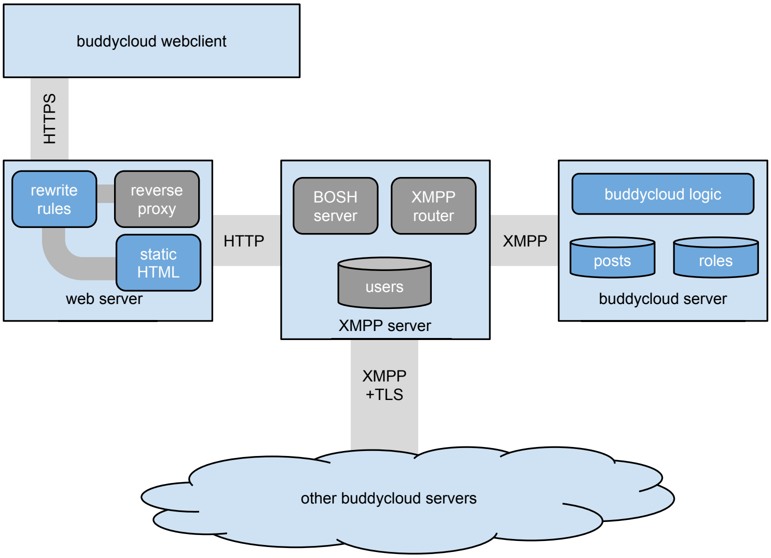

Fig. 5. buddycloud Architecture Overview

instances can communicate while it is still preserving the traditional online social networking paradigm where all the data is stored on the server. It is lacking of any mature client API for mobile devices thus applications build on top of the platform will reside on the server as plugins. The data model is rather limited but enough for its purpose containing groups, asymmetric user relationship, messaging support (only 1 - 1), contents which can be attached to groups. The data and policy model are extensible only through server side plugins while from the authentication perspective it provides a powerful pluggable authentication module (PAM) which allows the implementation of any sort of authentication.

Privacy and Access Control. Users can control the visibility of profile information, posts or groups so that the data can be private, accessed by friends, logged in users or even public. By making use of plugins, enhanced authentication mechanisms can be added such as logging in using a Twitter account or even LDAP credentials.

3.4 Decentralized

Decentralized networks are federated social networks which do not depend on any central authority in order to function. Consequently, user data is completely out of the cloud residing on user's devices (which can be one or many).

Musubi. As a mobile social networking platform Musubi [34] offers a comprehensive peer-to-peer (P2P) encryption mechanism, both 1 to 1 but also multiple peers key exchange, between users who authenticate themselves using e-mail but also OAuth. While the social relationship between agents is symmetrically mapped it is interesting to note that its communication layer is centered around the notion of ‘feeds’. So that is why groups are modeled as a multi-party feed list, making it easy to support group chats. Even though the access control mechanism is rather limited offering just simple black/white listing, and events as a social resource are missing from the basic model, it offers the ability to extend the latter through subclassing the `Obj` class defined in Musubi which are then stored in a database on the owner’s device. The SDK exposes a complete mobile collaborative application middleware which provides identity, group formation, reliable group messaging allowing a facile manner of applications development. Its architecture is depicted in Figure 6.

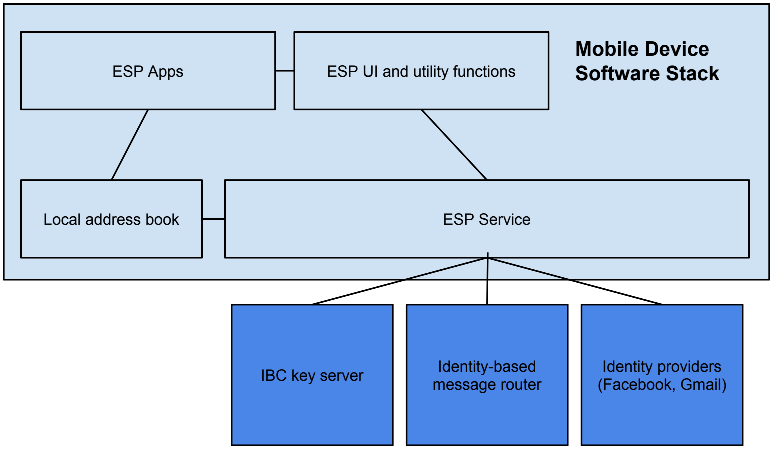
Mobile Device Software Stack
ESP Apps
ESP UI and utility functions
Local address book
ESP Service
IBC key server
Identity-based message router
Identity providers (Facebook, Gmail)

Fig. 6. Musubi’s Egocentric Social Platform (ESP) Architecture Overview

Privacy and Access Control. Though the current platform depends on reliable but not fully trusted traffic relay services to achieve P2P communication over the Web, it provides encryption on data transfer, key management, as well as it describes a Trusted Group Chat Protocol which involves a multi-peers key exchange. The access control mechanism is simplistic thus error prone, but it does not protect against data re-sharing.

Yarta. Yarta [35] is a flexible decentralized mobile social platform (see Figure 7) which keeps all of users data out of the cloud, on their devices in a semantic manner (RDF) which offers a high level of information re-usability across applications built on top of the platform by using the inherited inference from ontology models. Moreover all these data is shared using a semantic aware access control manager which allows the creation of complex policy models which can

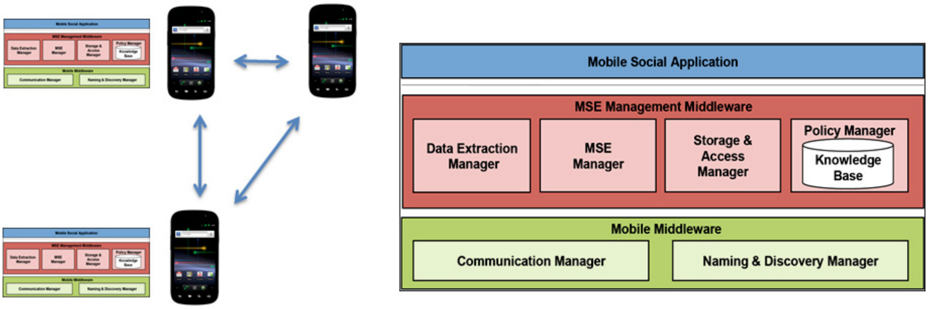

Fig. 7. Yarta Architecture Overview

also include context information which are gathered from mobile sensors. As for the authentication Yarta currently provides an OAuth example but this can be easily replaced with any flavor of one's needs.

Privacy and Access Control. Yarta offers an extensible, powerful and comprehensive semantic based access control mechanism [36] allowing the description of semantically defined policy rules which sits at the gate of owner's device before sharing any data. Still there is the problem of data re-sharing, since another peer which gathered the data might either not be aware of the sensitivity of that information nor it should be trusted.

4 Recommendations for a Privacy-Aware Federated Social Networking Architecture

Based on our survey above, we believe that there is a need for clear identification of the components needed to create federated social networking platforms, with a special emphasis on privacy and access control. Notably, in a federated social ecosystem each entity participating in the production or consuming information should comply to open standards which will further allow the integration of heterogeneous systems. To that end, we identify below the main components needed for a federated social network, as well as the currently existing solutions for each. The overall architecture is shown in Figure 8, and includes the following components:

- **Storage**: Whether it is present locally on the user's device or in a trusted federated server it is clear that the storage of a system needs to be done in such a fashion that will allow the description of existing social resources (e.g. person profiles, textual and multi-media contents, messages, etc.), but also allow the ability to extend the model through defining new concepts, complex data structure but also novel connections between these.
- **Access Control**: Users should be able to express rich policies in terms of their social context, links, groups and domain, which should be enforced before granting access to their data.

Table 1. Summary of major social networking platforms

Architecture	Network	Mobile	Groups	Person-Person links	Events	Storage	Access control	Authentication
Siloed	Facebook	Online	Yes	Symmetric	Yes	Fixed	Limited	OAuth
	Twitter	Online	No	Asymmetric	No	Fixed	Limited	OAuth
	Mosco	Online	Yes	Symmetric	No	Extensible	Extensible	OAuth
Social Networking-aaS	Google Plus	Online	Yes	Asymmetric	Yes	Fixed	Limited	OAuth
	Ning	Online	Yes	Symmetric	Yes	Fixed	Limited	OAuth
	Yammer	Online	Yes	Asymmetric	Yes	Extensible	Limited	OAuth
Federated	Status.net	Online	Yes	Asymmetric	Yes	Extensible	Extensible	Apache
	Friendica	Online	Yes	Symmetric	Yes	Fixed	Limited	LDAP
	Diaspora	Online	Yes	Asymmetric	No	Fixed	Extensible	HTTP
	buddycloud	Online	No	Asymmetric	No	Extensible	Extensible	HTTP
	OneSocialWeb	Online	No	Asymmetric	No	Extensible	Extensible	OpenID
	ELGG	Online	Yes	Asymmetric	No	Extensible	Extensible	PAM
Decentralized	Musubi	Mobile	Yes	Symmetric	No	Extensible	Limited	OAuth, Email
	Yarta	Mobile	Yes	Asymmetric	Yes	Extensible	Extensible	OAuth

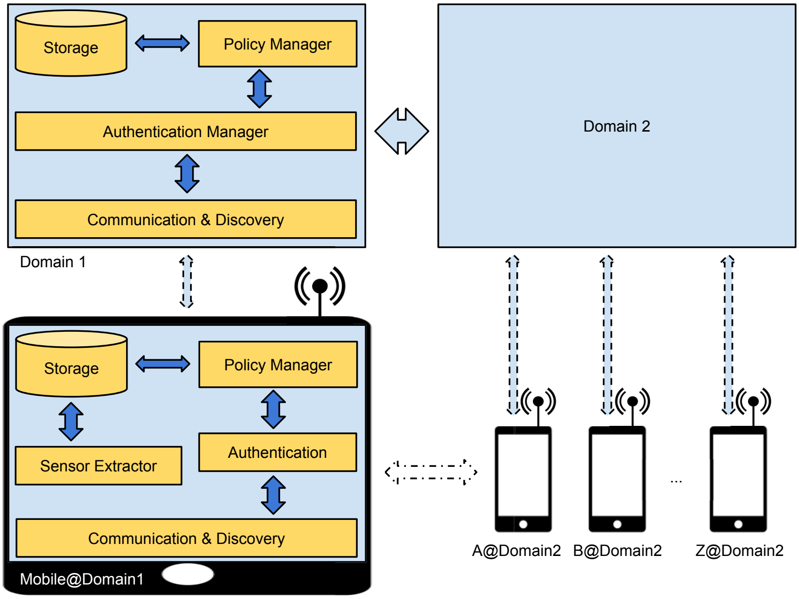

Fig. 8. Federated architecture which identifies the main components on the server side together with those present on client side devices along with the interaction which can happen between different peers

- **Authentication**: Authentication should be enabled whenever any peer communicates with a different one, allowing the problem of 'to whom I am speaking with' to be solved in an easy manner offering trust and security through an advanced cryptographic system.
- **Communication**: The communication module can be either part of the system itself or an independent external module but should enable the communication of any two peers which have social connections. In case the communication is an external third party component then it is imperative to employ adequate security measures to ensure data integrity as well as user-anonymity, when needed.
- **Discovery**: This is an essential mechanism to allow resources be defined over the web and also enables their discovery.

We discuss below the alternative solutions —both open source and commercial— which can be adopted for each component in part, noting that according to W3C's Federated Social Web group [37] the main trends towards federation would be to adopt open standards.

4.1 Storage

In terms of storage current social networking trends are moving towards extensible mechanisms such as mapping social resources as ontologies or JSON based actor - action - object format as seen in Yammer's case or ActivityStrea.ms which has been adopted by many open source platforms. The latter is similar, in terms of semantics, with RDF storage schema, since it implies the existing of a subject, a predicate and an object such as triples.

If we are to consider storage of social information as triple stores then we would have plenty of solutions which enables such capabilities, both open source but also commercial, such as Parliament [38], AllegroGraph [39] and Mulgara [40]. If we would consider a distributed synchronization of such models then tools like RDFSync [41] would come in handy.

One standard which has been adopted by many open source federated social networks, ActivityStrea.ms is becoming more and more popular. It provides an extensible manner of activity description. Implementations can be found in many open source projects such as OneSocialWeb, buddycloud, Status.Net or eXo [42] platform.

4.2 Authentication and Access Control

From the authentication perspective OAuth and OpenID are becoming more and more popular and has been adopted by the majority of social networks for which there exists several implementation for both server and clients. Source code in most popular programming languages can be found on each protocol's website. It we are considering federated authentication and authorization then Shibboleth [43] and Gluu [44] are two interesting tools which we might consider working with, noting that they both offer an open source implementation of SAML protocol.

Access control have become an important aspect of nowadays social ecosystem. As seen in [45], if we are to consider social networks as a SaaS, then both the provider and the tenant should be able to express their privacy policies in a secure manner since the latter has to disclose sensitive information. Access control mechanisms should be able to describe both traditional policies but also complex ones making use of context information as well. For simple access mechanism one can choose an open source implementation of Access Control List (ACL) protocols, more advanced ones such as XACML[2] or an implementation of the semantics-based policies of [36] which provides a highly extensible, generic yet expressive access control policy management solution. In order to approach the problem of re-sharing information one should consider sticky security policies solutions described in works such as [46,47].

4.3 Communication and Discovery

For communication various open protocols can be adopted such as XMPP, Salmon, PubSubHubbub or OStatus to achieve federation since those have been adopted by many open source social networking platforms. More, one can make

[2] Open Source XACML: `http://sunxacml.sourceforge.net/`

use of faster, light-weight communication middlewares such as MQTT [48], iBICOOP [49] which provides transportation relays between devices over the Internet.

Coupled with the above, there are options for discovery which include open ones such as WebFinger which has been adopted by Diaspora and Status.net, XEP from XMPP, mDNS protocol which can be found in the Bonjour commercial software, or even UPnP media discovery and of course iBICOOP.

5 Future Directions

It is evident that the future will see increased adoption of social networking, and it will not all be managed by a single entity. Consequently, support for federation among social networks emerges as a necessary functionality, something that currently available systems are not able to provide in a comprehensive manner. We believe that in order to enable the federated social networking platforms of the future, empowered with strong privacy and access-control policies, the community should *i*) Adopt open standards for the necessary components as much as possible, in order to prevent reinventing the wheel and speed-up adoption; *ii*) Use semantic techniques for modeling of social knowledge, enabling the easy and extensible re-use of data, both by applications executing on these platforms and other social networking providers; and *iii*) provide rich privacy and access-control mechanisms, preferably semantically-based sticky policies so as to provide adequate protection to the users' and organizations' sensitive information. Following the above should lead to interoperable social networking platforms which will gain wide acceptance.

References

1. European Commission: Commission proposes a comprehensive reform of the data protection rules, `http://ec.europa.eu/justice/newsroom/data-protection/news/120125_en.htm` (accessed January 2014)
2. Microsoft: Microsoft to Acquire Yammer, `http://www.microsoft.com/en-us/news/press/2012/jun12/06-25msyammerpr.aspx` (accessed January 2014)
3. Ho, R.: Google+ is now available for Google Apps, `http://googleenterprise.blogspot.it/2011/10/google-is-now-available-with-google.html` (accessed January 2014)
4. Hinchcliffe, D.: Today's Collaboration Platforms for Large Enterprises, `http://www.zdnet.com/the-major-enterprise-collaboration-platforms-and-their-mobile-clients-7000018519/` (accessed January 2014)
5. Narayanan, A., Toubiana, V., Barocas, S., Nissenbaum, H., Boneh, D.: A critical look at decentralized personal data architectures. CoRR abs/1202.4503 (2012)
6. Esguerra, R.: An introduction to the federated social network, `https://www.eff.org/deeplinks/2011/03/introduction-distributed-social-network` (accessed January 2014)

7. Facebook: Online Social Networking Platform, https://www.facebook.com/ (accessed January 2014)
8. OAuth: Secure authorization open protocol, http://oauth.net/ (accessed January 2014)
9. Twitter: Online Social Networking and Microblogging Service, https://twitter.com/ (accessed January 2014)
10. Redis: Open source advanced key-value store, http://redis.io/ (accessed January 2014)
11. Tuan Anh, D.T., Ganjoo, M., Braghin, S., Datta, A.: Mosco: A privacy-aware middleware for mobile social computing. Journal of Systems and Software (2013)
12. XACML: eXtensible Access Control Markup Language (XACML) Version 3.0, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (accessed January 2014)
13. Ning: Build and cultivate your own community, http://www.ning.com/ (accessed January 2014)
14. Foundation, O.: OpenSocial protocol, http://opensocial.org/ (accessed January 2014)
15. Yammer: Enterprise Social Network, https://www.yammer.com/ (accessed January 2014)
16. SAML: Security Assertion Markup Language (SAML) v2.0, https://www.oasis-open.org/standards#samlv2.0 (accessed January 2014)
17. Status.net: Free and open source social software, http://status.net/ (accessed January 2014)
18. Identi.ca: Open source social networking service, https://identi.ca/ (accessed January 2014)
19. pump.io: Open source social stream server, http://pump.io/ (accessed January 2014)
20. WebFinger: Personal web discovery protocol, https://code.google.com/p/webfinger/wiki/WebFingerProtocol (accessed January 2014)
21. Appcelerator: Portable software development platform, http://www.appcelerator.com/ (accessed January 2014)
22. OpenID Foundation: The Internet Identity Layer, http://openid.net/ (accessed January 2014)
23. Macgirvin, M.: DFRN - The Distributed Friends and Relations Network, https://macgirvin.com/spec/dfrn2.pdf (accessed January 2014)
24. Zot: Secure decentralised communications framework, https://github.com/friendica/red/wiki/zot (accessed January 2014)
25. Wahl, M., Howes, T., Kille, S.: Lightweight Directory Access Protocol, https://www.ietf.org/rfc/rfc2251.txt
26. Friendica: Red design documentation, https://github.com/friendica/red/wiki/red (accessed January 2014)
27. Diaspora: The Community-run, Distributed Social Network, http://www.joindiaspora.com/ (accessed January 2014)
28. Salmon: Real-time Commenting Protocol, http://www.salmon-protocol.org/ (accessed January 2014)
29. OneSocialWeb: Creating a free, open, and decentralized social networking platform, http://onesocialweb.org/ (accessed January 2014)
30. buddycloud: Federated social network, http://buddycloud.com/ (accessed January 2014)
31. XMPP: XMPP standards foundation, http://xmpp.org/about-xmpp/xsf/ (accessed January 2014)

32. XMPP: XMPP extension protocols, http://xmpp.org/extensions/xep-0001.html (accessed January 2014)
33. Elgg: Open Source Social Networking Engine, http://elgg.org/ (accessed January 2014)
34. Dodson, B., Vo, I., Purtell, T., Cannon, A., Lam, M.: Musubi: Disintermediated interactive social feeds for mobile devices. In: Proceedings of the 21st International Conference on World Wide Web, pp. 211–220. ACM (2012)
35. Toninelli, A., Pathak, A., Issarny, V.: Yarta: A Middleware for Managing Mobile Social Ecosystems. In: Riekki, J., Ylianttila, M., Guo, M. (eds.) GPC 2011. LNCS, vol. 6646, pp. 209–220. Springer, Heidelberg (2011)
36. Hachem, S., Toninelli, A., Pathak, A., Issarny, V.: Policy-based Access Control in Mobile Social Ecosystems. In: Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks, Pisa, Italy. IEEE computer society (June 2011)
37. W3C: Federated social web community group, http://www.w3.org/2005/Incubator/federatedsocialweb/wiki/Main_Page (accessed January 2014)
38. Parliament: High-performance triple store, http://parliament.semwebcentral.org/ (accessed January 2014)
39. AllegroGraph: RDFStore Web 3.0's Database, http://franz.com/agraph/allegrograph/ (accessed January 2014)
40. Mulgara: Open source scalable rdf database, http://www.mulgara.org/ (accessed January 2014)
41. Tummarello, G., Morbidoni, C., Bachmann-Gmür, R., Erling, O.: RDFSync: Efficient remote synchronization of rdf models. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 537–551. Springer, Heidelberg (2007)
42. eXo: Open Source Enterprise Social Network, http://www.exoplatform.com/ (accessed January 2014)
43. Shibboleth: Federated identity solutions, http://shibboleth.net/ (accessed January 2014)
44. Gluu: Open source access management, http://www.gluu.org/ (accessed January 2014)
45. Decat, M., Lagaisse, B., Van Landuyt, D., Crispo, B., Joosen, W.: Federated authorization for software-as-a-service applications. In: Meersman, R., Panetto, H., Dillon, T., Eder, J., Bellahsene, Z., Ritter, N., De Leenheer, P., Dou, D. (eds.) ODBASE 2013. LNCS, vol. 8185, pp. 342–359. Springer, Heidelberg (2013)
46. Mont, M.C., Pearson, S., Bramhall, P.: Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. In: IEEE Proceedings of the 14th International Workshop on Database and Expert Systems Applications, pp. 377–382 (2003)
47. Fatema, K., Chadwick, D.W., Lievens, S.: A multi-privacy policy enforcement system. In: Fischer-Hübner, S., Duquenoy, P., Hansen, M., Leenes, R., Zhang, G. (eds.) Privacy and Identity 2010. IFIP AICT, vol. 352, pp. 297–310. Springer, Heidelberg (2011)
48. MQTT: Machine to machine connectivity protocol, http://mqtt.org/ (accessed January 2014)
49. Bennaceur, A., Singh, P., Raverdy, P.G., Issarny, V.: The iBICOOP middleware: Enablers and services for emerging pervasive computing environments. In: IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, pp. 1–6. IEEE (2009)