

Social Networking: Power to the People^{*}

Stefano Bortoli, Paolo Bouquet, and Themis Palpanas

University of Trento,
Dept. of Information Science and Engineering,
Trento, Italy
{bortoli, bouquet, themis}@disi.unitn.eu

Abstract. Social networking sites have gained much popularity in the recent years, because of the opportunities they give people to connect to each other in an easy and timely manner, and to exchange and share various kinds of information. However, these sites are architected based on a centralized paradigm, which limits the mobility of their users, and ultimately, their chances for establishing new relationships and benefiting from diverse networking services.

In this paper, we argue for a decentralized paradigm for social networking, in which users retain control of their profiles, and social networking sites focus on the delivery of innovative and competitive services. Our position is that only in this environment will both the social networking sites and their users be able to develop to their full potential.

Keywords: social network portability, unique identification

1 A Different Position on Social Networking

Social networking sites have in the recent years gained considerable popularity in global scale, and have attracted a significant part of the online community. The above observations can be explained by the fact that these sites make it possible for people to connect to each other in an easy and timely manner, and to exchange and share various kinds of information among them.

1.1 The Centralized Paradigm of Today

Most social network services (like Facebook [1], MySpace [5] or LinkedIn [4]) are based on the following high-level architecture: a central repository in which they store data about users and their connections; and a set of modules offering users networking services (e.g., finding other people, sharing pictures/videos, exchanging professional information, etc.). We term the above architectural paradigm as *centralized* (see Figure 1). Under this paradigm, people use their client application (typically their browser) to create their personal and social profile and store it in the platform's central repository. Then they upload the information they want to

^{*} This work is partially supported by the FP7 EU Large-scale Integrating Project **OKKAM – Enabling a Web of Entities** (contract no. ICT-215032). For more details, visit <http://www.okkam.org>.

share with their contacts. A consequence of the centralized paradigm is the well-known *lock-in effect*: once a user joins a social network, then transferring their data (and their contacts) to another platform is a laborious and time consuming manual effort, and therefore, implicitly discouraged. In addition, it is necessary that all of the user's contacts do the same (in order to re-create the social environment). As a partial solution to this problem, today many social networks offer the functionality to search for contacts in a user's mailing address books (e.g., in the Gmail or Yahoo! Mail address books), and send to these contacts automatic invitations to join a new social network. However, this does not address and does not solve the problems outlined above, and it certainly does not give more control to people on their own data and contacts.

1.2 Our Proposal: The Decentralized Paradigm

The approach which we would like to propose to overcome this lock-in effect is to shift the control of this kind of personal data from the server-side to the client-side, thus, leaving the social networking platforms focus on the delivery of innovative and competitive services. The idea is quite straightforward:

1. data about users and their connections are decentralized and stored in locations which are not controlled by the social network application, and
2. social network applications focus on providing the spaces where people enjoy different (sometimes competing) services.

The intuition behind our approach can be explained through a simple analogy. *Imagine you go to a pub, and meet new friends. Even though the reason for meeting in that specific pub can be to drink a particular brand of beer, you and your friends are still the "owners" of your relationship, and such a relationship should be independent of the fact that you met in that pub. For example, one day you may decide to watch a football match with the same friends. Then you and your friends can simply meet at another pub that screens this particular football match. So the two pubs stay the owners of the services they offer (serving good beer, or having a large screen for football matches), and they compete on the basis of the services they offer. Nevertheless, pub customers are not locked-in, which means that they can decide to use other services (or even the same services, but of better quality) in another pub.*

The social network architecture based on the pub analogy is depicted in Figure 2. Users own their personal data, and may manage a collection of context-based relationships (e.g., professional contacts, friends and family, ex-classmates, etc.). This information is not centrally managed within any social network platform ("server-side"), but rather locally by some *client-side* tool-suite that allows people to easily create and update these data. This decentralized paradigm has advantages for both the social networking sites and their users (similar observations have also appeared in other studies [9]):

1. Data about users and their connections are under the control of the users, which means that they can in principle be uploaded into any social networking platform that is compatible with the format in which these data are structured.

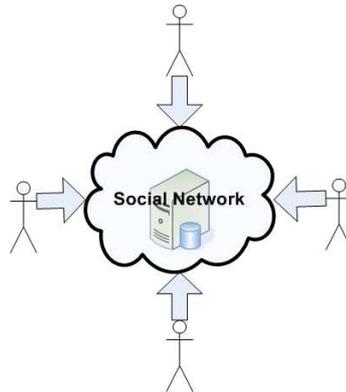


Fig. 1. Simple view of current state of Social Network Application

In this way, users can benefit from the different services offered by a variety of social networking platforms.

2. Social networking applications can focus on the task of providing value added services to users, and try to attract new users from a pool that is now much larger. By allowing users to establish relationships among different social network sites that cater to social networks with different interests and goals (eg, artistic, professional, etc), all social networking platforms can benefit from the increased mobility of users.

1.3 A First Step Towards the Decentralized Paradigm

Evidently, the vision outlined above is an elaboration of the well-known Friend-Of-A-Friend (FOAF) approach [3]. However, in order to implement a realistic client-side solution, any FOAF-like approach must be extended in at least three directions:

- FOAF is focused on providing a simple vocabulary for providing personal data and a generic "know" relationship with other people. We need to extend this vocabulary to allow some form of contextualization of social relationships: in general we don't want our relatives to be mixed with our professional contacts, or our professional contacts to be used for sharing pictures and videos among people who do not know each others and did not approve such a sharing.
- Another critical issue is that FOAF does not provide a real solution to the problem of identifying people (or any other type of entity, e.g. companies or locations) in a unique way across different local profiles. If a user stores their social contacts in a local file, then it would be desirable that the same entities (i.e., people, locations, organizations, events, etc.) are referred to by means of the same identifier. Otherwise the migration (or integration) of these data across different platforms becomes extremely cumbersome.
- FOAF profiles are stored in public web locations, so that any application can find them and use their data to any purpose. In a realistic setting, we need a much more fine-grained access control mechanism, so that the data in a local

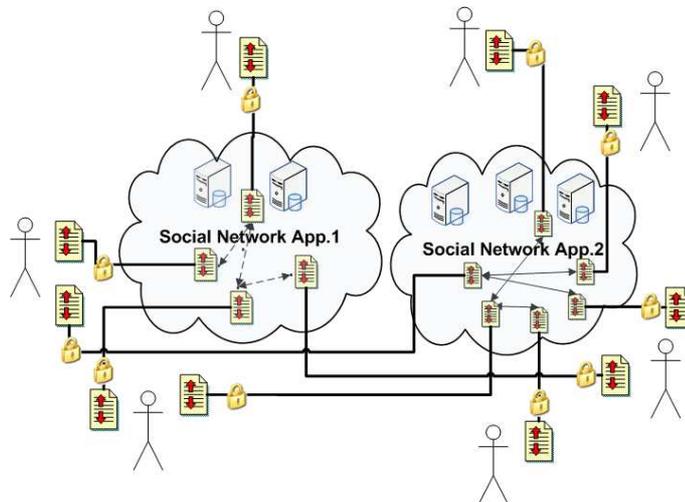


Fig. 2. Schematic view over our conception of client-side social networking

profile (including the social contexts) are used only by the applications which are authorized by the user, and according to specific rules.

Currently, as part of the OKKAM EU-funded project [6], we are developing an open service to address the second of the above issues in a very general and systematic way. A first attempt to solve the identification problem in FOAF has resulted in a web application enabling the integration of globally unique identifiers (based on the OKKAM Entity Naming System, which is briefly discussed in Appendix A) into FOAF profiles. This application, foaf-O-matic, is a first example of a tool supporting users in managing their social network data in a decentralized and “neutral” fashion. The application is currently under development, but a first usable version is already available [2]. We hope that this is only the first step towards the decentralized paradigm for social networking platforms.

References

- [1] Facebook. <http://www.facebook.com/>, 2008.
- [2] foaf-O-matic. <http://www.foaf-o-matic.org/>, 2008.
- [3] Friend of a Friend. <http://www.foaf-project.org/>, 2008.
- [4] LinkedIn. <http://www.linkedin.com/>, 2008.
- [5] MySpace. <http://www.myspace.com/>, 2008.
- [6] OKKAM Project. <http://www.okkam.org/>, 2008.
- [7] Paolo Bouquet, Heiko Stoermer, and Barbara Bazzanella. An entity name system (ens) for the semantic web. In *ESWC*, pages 258–272, 2008.
- [8] T. Palpanas, J. Chaudhry, P. Andritsos, and Y. Velegrakis. Entity Data Management in OKKAM. pages 729–733, 2008.
- [9] Raghu Ramakrishnan and Andrew Tomkins. Toward a peopleweb. *IEEE Computer*, 40(8):63–72, 2007.

A Entity Naming System

In the following paragraphs, we give a brief overview of the *ENS* (Entity Naming System) (a more detailed presentation can be found elsewhere [7, 8]), which we will use as the basis for our discussion. Note however, that our discussion is relevant to any system for entity identification management. The overall goal of the *ENS* is to handle the process of assigning and managing unique identifiers for entities in the WWW. These identifiers are global, with the purpose of consistently identifying a specific entity across system boundaries, regardless of the place in which references to this entity may appear (see Figure 3).

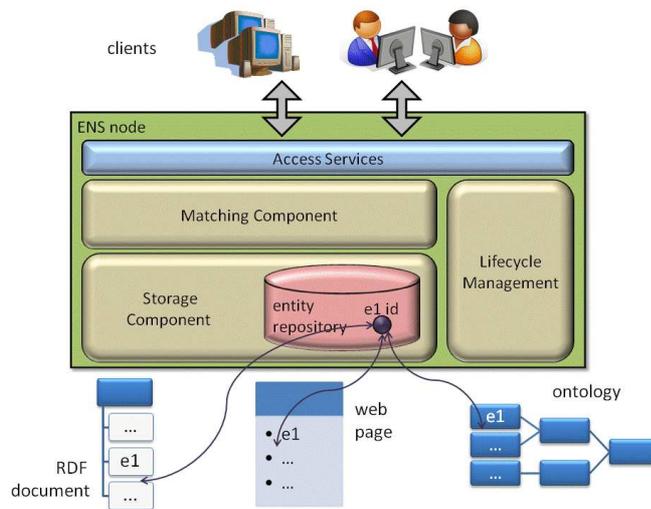


Fig. 3. Schematic of the *ENS* and its interactions.

The *ENS* has a repository for storing entity identifiers (note that this repository will be distributed and replicated) along with some small amount of descriptive information for each entity. The purpose of storing this information is to use it for discriminating among entities, not exhaustively describing them. Entities are described by a number of attribute-value pairs, where the attribute names and the potential values are user-defined (arbitrary) strings.

Clients interact with the system through the *Access Services* layer. Clients can be both human users and applications, and may inquire about the identifier of an entity by providing a set of attributes that describes this entity. If the entity exists in the repository, the system returns its identifier. Clients may also modify the state of the repository, either by inserting a new entity in the system, in which case the *ENS* returns the newly assigned identifier, or by changing some of the attributes of an existing entity. As shown in Figure 3, the end result is that all instances of the same entity (i.e., mentioned in different systems, ontologies, web pages, etc.) are assigned the same identifier. Therefore, integrating information becomes a much more simple and effective process than before.