Weaving a Distributed, Semantic Social Network for Mobile Users

Sebastian Tramp, Philipp Frischmuth, Natanael Arndt, Timofey Ermilov, and Sören Auer

Universität Leipzig, Institut für Informatik, AKSW,
Postfach 100920, D-04009 Leipzig, Germany
lastname@informatik.uni-leipzig.de
http://aksw.org

Abstract. Smartphones, which contain a large number of sensors and integrated devices, are becoming increasingly powerful and fully featured computing platforms in our pockets. For many people they already replace the computer as their window to the Internet, to the Web as well as to social networks. Hence, the management and presentation of information about contacts, social relationships and associated information is one of the main requirements and features of today's smartphones. The problem is currently solved only for centralized proprietary platforms (such as Google mail, contacts & calendar) as well as data-silo-like social networks (e.g. Facebook). Within the Semantic Web initiative standards and best-practices for social, Semantic Web applications such as FOAF emerged. However, there is no comprehensive strategy, how these technologies can be used efficiently in a mobile environment. In this paper we present the architecture as well as the implementation of a mobile Social Semantic Web framework, which weaves a distributed social network based on semantic technologies.

1 Introduction

Smartphones, which contain a large number of sensors and integrated devices, are becoming increasingly powerful and fully featured computing platforms in our pockets. For many people they already replace the computer as their window to the Internet, to the Web as well as to social networks. Hence, the management and presentation of information about contacts, social relationships and associated information is one of the main requirements and features of today's smartphones.

The problem is currently solved solely for *centralized* proprietary platforms (such as Google mail, contacts & calendar) as well as data-silo-like social networks (e.g. Facebook). As a result of this data centralization, users' data is taken out of their hands, they have to accept the predetermined privacy and data security regulations; users are dependent of the infrastructure of a single provider, they experience a lock-in effect, since long-term collected profile and relationship information cannot be easily transferred. Increasingly, many people argue that

G. Antoniou et al. (Eds.): ESWC 2011, Part I, LNCS 6643, pp. 200–214, 2011.

social networks should be evolving. That is, they should allow users to control what to enter and to keep a control over their own data. Also, the users should be able to host the data on an infrastructure, which is under their direct control, the same way as they host their own website [3].

A possibility to overcome these problems and to give the control over their data back to the users is the realization of a truly *distributed* social network. Initial approaches for realizing a distributed social network appeared with *GNU social* and more recently *Diaspora* (cf. Section 6). However, we argue that a distributed social network should be also based on semantic resource descriptions and de-referenceability so as to ensure versatility, reusability and openness in order to accommodate unforeseen usage scenarios.

Within the Semantic Web initiative already a number of standards and best-practices for social, Semantic Web applications such as *FOAF*, *WebID* and *Semantic Pingback* emerged. However, there is no comprehensive strategy, how these technologies can (a) be combined in order to weave a truly open and distributed social network on the Web and (b) be used efficiently in a mobile environment. Also, the use of a distributed, social semantic network should be as *simple* as the use of the currently widely used centralized social networks (if not even simpler). In this paper we present the general strategy for weaving a distributed social semantic network based on the above mentioned standards and best-practices. In order to foster its adoption we developed an implementation for the Android platform, which seamlessly integrates into the commonly used interfaces for contact and profile management on mobile devices.

After briefly reviewing some use cases and requirements for a mobile, semantic social network application (in Section 2), we make in particular the following contributions:

– We outline a strategy to combine current bits and pieces of the Semantic Web technology realm in order to realize a distributed, semantic social network (Section 3),
– We develop an architecture for making mobile devices endpoints for the Social Semantic Web (Section 3),
– A comprehensive implementation of the architecture was performed for the Android platform (Section 4 and 5),

Furthermore, our paper contains an overview on related work in Section 6 and concludes in Section 7 with a discussion and outlook on future work.

2 Mobile Use Cases and Requirements

Before describing the overall strategy, the technical architecture and our implementation we want to briefly outline in this section the key requirements, which guided our work. These requirements are common sense in the context of social networks and are not newly coined by us. Unfortunately most of them are not achieved in the context of semantics enabled and distributed social networks, so we describe them especially from this point of view.

Make new friends. Adding new contacts to our social network is the precondition in order to gather useful information from this network. Maintaining our social network directly from your mobile phones means that we are able to instantly connect with new contacts (e.g. on conferences or parties). In the context of a distributed social network, this use-case also includes the employment of semantic search engines to acquire the WebID of a new contact based on parts of its information (typically the contacts name). In order to shorten the overall effort for adding new contacts, functionality for scanning and decoding a contacts business cards QR code[1] are also included in this use-case.

Be in sync with your social network. Once our social network is woven and social connections are established, we want to be able to gather information from this network. For a distributed social network this means, that a combination of push and pull communications is needed to be as timely updated as needed and as fast synced as possible. Especially this use-case is bound to a bunch of access control requirements[2]. where people want to permit and deny access to specific information in fine grained shades and based on groups, live contexts and individuals.

Annotate contacts profiles. It should be possible to annotate profiles of contacts freely, e.g. with updated information, contact group categorizations (e.g. friends, family, co-workers). These annotations should be handled in the same way as the original data from the friend's WebID except that this data is not updated with the WebID but persists as an annotation. One additional feature request in this use-case is to share these annotations across ones personal devices on the web, e.g. by pushing them to a triple store which is attached to ones WebID.

General requirements. The development of the Mobile Social Semantic Web Client was driven by a few general requirements which derived from our own experience with mobile phones and FOAF-based WebIDs:

- *Be as decent as possible:* Today's FOAF-based social networks are mostly driven by uploaded RDF files. In order to support such low end profiles, there should be no other required feature on a WebID than the availability as Linked Data[3]. All other features (FOAF+SSL, Semantic Pingback, subscription service) should be handled as optional and our client should require as little infrastructure as possible.

[1] QR codes are two-dimensional barcodes which can encode URIs as well as other information. They are especially famous in Japan, but their popularity grows more and more worldwide since mobile applications for decoding them with a standard camera can be used on a wide range of devices.

[2] A typical requirement: Disallow access to my mobile number except for friends and family members.

[3] In the meaning of Linked RDF Data defined at `http://www.w3.org/DesignIssues/LinkedData`

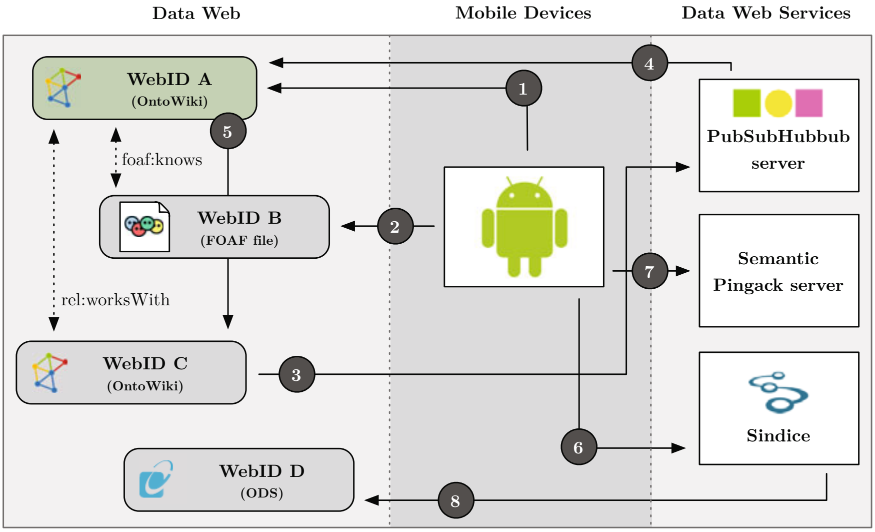

Fig. 1. Architecture of a distributed, semantic social network: (1) A mobile user may retrieve updates from his social network via his WebID provider, e.g. from OntoWiki. (2) He may also fetch updates directly from the sources of the connected WebIDs. (3) A WebID provider can notify a subscription service, e.g. a PubSubHubbub server, about changes. (4) The subscription service notifies all subscribers. (5) As a result of a subscription notification, another node can update its data. (6) A mobile user can search for a new WebID by using a semantic search engine, e.g. Sindice. (7) To connect to a new WebID he sends a Pingback request which (8) notifies of the resource owner.

- *Be as transparent as possible:* Mobile user interfaces are built for efficiency and daily use. People become accustomed with them and any changes in the daily work flow of using information from the social network will annoy them. The client we had in mind should work mostly invisible from the user, which means it should be well integrated into the hosting mobile operating system.
- *Be as flexible as possible:* This is especially needed in an environment where vocabularies are not yet standardized and are subject to changes and extensions. Our solution should be flexible in the sense that we do not want built-in rules on how to deal with specific attributes or relations.

Based on these preliminaries as well as based on the Social Semantic Web state of the art, we describe an architecture of a distributed social semantic network in the next section.

3 Architecture of a Distributed Semantic Social Network

In this section we describe the main ingredients for a distributed, semantic social network as well as their interplay. The overall architecture is depicted in

Figure 1. The semantic representation of personal information is facilitated by WebID. FOAF+SSL allow the use of a WebID for authentication and access control purposes. Semantic Pingback facilitates the first contact between users of the social network and subscription services allow obtaining specific information from people in ones social network as near-instant notifications.

WebID. WebID [16] is a best-practice recently conceived in order to simplify the creation of a digital ID for end users. Since its focus lies on simplicity, the requirements for a WebID are minimal. In essence, a WebID is an de-referenceable RDF document (including RDFa) describing its owner[4]. That is, a WebID contains RDF triples, which have the IRI identifying the owner as subject. The description of the owner can be performed in any (mix of) suitable vocabularies, but FOAF [4] emerged as the 'industry standard' for that purpose. An example WebID comprising some personal information (lines 8-12) and two `rel:worksWith`[5] links to co-workers (lines 6-7) is shown in Listing 1.

```
1  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2  @prefix foaf: <http://xmlns.com/foaf/0.1/> .
3  @prefix rel: <http://purl.org/vocab/relationship/> .
4  <http://philipp.frischmuth24.de/id/me> a foaf:Person;
5    rdfs:comment "This is my public profile only, more
         information available with FOAF+SSL";
6    rel:worksWith <http://sebastian.tramp.name>,
7      <http://www.informatik.uni-leipzig.de/~auer/foaf.rdf#me>;
8    foaf:depiction <http://img.frischmuth24.de/people/me.jpg>;
9    foaf:firstName "Philipp"; foaf:surname "Frischmuth";
10   foaf:mbox <mailto:frischmuth@informatik.uni-leipzig.de>;
11   foaf:phone <tel:+49-341-97-32368>;
12   foaf:workInfoHomepage <http://bis.informatik.uni-leipzig.de
         /PhilippFrischmuth>.
```

Listing 1. A minimal WebID with personal information and two *worksWith* relations to other WebIDs

FOAF+SSL. The more technical FOAF+SSL best-practice [17] aims to incorporate authentication functionality into the WebID concept. The main idea is to link an SSL client certificate to a WebID, thus allowing the owner of the FOAF+SSL enabled WebID to authenticate herself at 3rd party websites. Another goal of FOAF+SSL is to provide access control functionality for a social network shaped by WebIDs in order to allow access to different kinds of information for different groups of contacts (e.g. as presented with dgFOAF [14]). An example of a FOAF+SSL WebID extension is shown in Listing 2. This WebID

[4] The usage of an IRI with a fragment identifier allows the indirect identification of a WebID by reference to the (FOAF) profile document.

[5] Taken from *RELATIONSHIP: A vocabulary for describing relationships between people* at `http://purl.org/vocab/relationship`

now contains a description of an RSA public key (line 15), which is associated to the WebID by using the `cert:identity` property from the W3C certificates and crypto ontology (line 19).

```
13 @prefix rsa: <http://www.w3.org/ns/auth/rsa#> .
14 @prefix cert: <http://www.w3.org/ns/auth/cert#"> .
15 [] a rsa:RSAPublicKey;
16   rdfs:comment "used from my smartphone ...";
17   cert:identity <http://philipp.frischmuth24.de/id/me>;
18   rsa:modulus          [ cert:hex "C41199E ... 5AB5" ];
19   rsa:public_exponent  [ cert:decimal "65537" ] .
```

Listing 2. Extension of the minimal WebID from Listing 1: Description of an RSA public key, which is associated to the WebID by using the `cert:identity` property from the W3C certificates and crypto ontology

Semantic Pingback. The purpose of Semantic Pingback [18] in the context of a distributed social network is to facilitate the first contact between different people using the network. The approach is based on an extension of the well-known Pingback technology [8], which is one of the technological cornerstones of the overwhelming success of the blogosphere in the Social Web. The Semantic Pingback mechanism enables bi-directional links between WebIDs, RDF resources as well as weblogs and websites in general. It facilitates contact/author/user notifications in case a link has been newly established. It is based on the advertising of a lightweight RPC service, in the RDF document, HTTP or HTML header of a certain Web resource, which should be called as soon as a (typed RDF) link to that resource is established. The Semantic Pingback mechanism enables people but also authors of RDF content, a weblog entry or an article in general to obtain immediate feedback, when other people establish a reference to them or their work, thus *facilitating social interactions.* It also allows to automatically publish backlinks from the original WebID (or other content) to comments or references of the WebID (or other content) elsewhere on the Web, thus *facilitating timeliness and coherence* of the Social Web. As a result, the distributed network of WebID profiles, RDF resources and social websites using the Semantic Pingback mechanism can be much tighter and timelier interlinked than conventional websites, thus rendering a network effect, which is one of the major success factors of the Social Web. Semantic Pingback is completely downwards compatible with the conventional Pingback implementations, thus allowing the seamless connection and interlinking of resources on the Social Web with resources on the Data Web. An extension of our example profile with Semantic Pingback functionality making use of an external Semantic Pingback service is shown in Listing 3.

Subscription Service. The purpose of a WebID subscription service is to establish a publish/subscribe communication model to provide near-instant notifications of contact updates. The main idea here is to extend a WebID with a link to

```
@prefix ping: <http://purl.org/net/pingback/> .
<http://philipp.frischmuth24.de/id/me> ping:to <http://
    pingback.aksw.org>.
```

Listing 3. Extension of the minimal WebID from Listing 1: Assignment of an external Semantic Pingback service which can be used to ping this specific resource

a *PubSubHubbub* service[6] where any contact can subscribe to the WebIDs updates. Although such a behavior is described for SPARQL results in [12], there is currently no standardized solution for publishing RDF change sets through PubSubHubbub as well as for saving the incoming changes from all friends of a user in some kind of cache or proxy while the mobile device (the subscriber) is not online. As a consequence, our implementation (as described in the next section) does not yet support a full-fledged update subscription. As a fallback, updates are currently polled from the related WebIDs. This increases network bandwidth usage and might lead in some cases to slower user interfaces due to network latency. Please refer to Section 7 for a description of possible future work in this direction.

4 Implementation of a Mobile Interface

After describing the architecture of a distributed, semantic social network we now present our implementation of a mobile interface for this network.

4.1 Android System Integration

Figure 2 depicts the mobile social Semantic Web client consisting of two application frameworks, which are built on top of the Android runtime and a number of libraries. In particular, *androjena*[7] is one of those libraries, which itself is a partial port of the popular Jena framework[8] to the Android platform. Both frameworks provided by the client share the feature that they are accessible through content providers. The Mobile Semantic Web middleware (MSW) is responsible for importing Linked Data resources (in particular via FOAF+SSL) and persisting that data. It operates on triple level and provides access to the various triple stores through a content provider called `TripleProvider`. Each resource is stored separately, since named graphs are currently not supported. The Mobile Social Semantic Web middleware (MSSW) queries the triple data provided by MSW and transforms that data into a format that is more appropriate for social applications. It propagates two content providers, one that integrates well with the layout of contact information on Android phones (`ContactProvider`) and one that is suitable for FOAF based applications (`FoafProvider`).

[6] PubSubHubbub is an open, server-to-server web-hook-based publish/subscribe protocol realized as an extension to Atom: `http://code.google.com/p/pubsubhubbub/`

[7] `http://code.google.com/p/androjena/`

[8] `http://jena.sourceforge.net/`

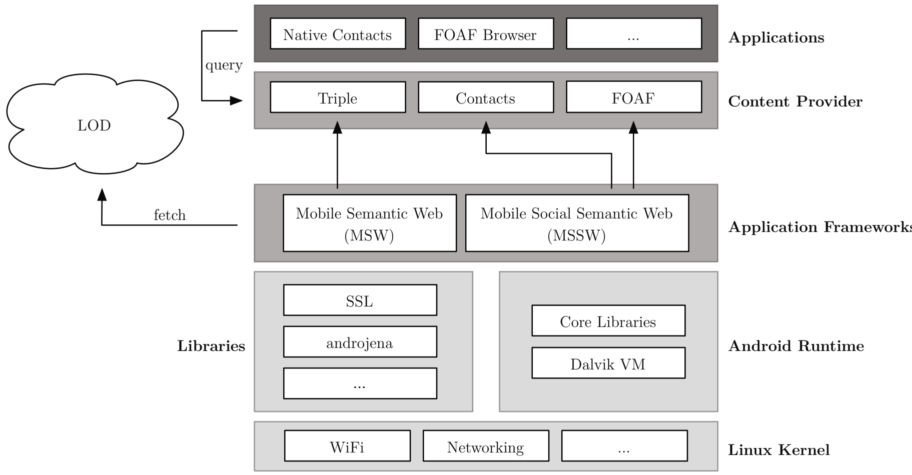

Fig. 2. Android Integration Layer Cake

4.2 Model Management

Since WebIDs are Linked Data enabled, they usually return data describing that resource. This circumstance makes it feasible to store a graph (referred to as a model here) for each WebID, since the redundancy between models is expected to be marginal. In reality MSW keeps more than one model per WebID for different purposes. On the mobile phones' SD-card we keep these models in the following subdirectories:

– `web` – This folder contains exact copies of the documents retrieved from the Web.
– `inf` – Models stored in this folder contain all entailed triples (more on this in Section 4.3).
– `local` – The user can annotate all WebIDs with personal information, which will be stored in this folder.

We decided to store all data as RDF files on a swappable SD-card, since we expect the following user benefits:

– Because SD-cards can be exchanged, the data is portable and can be reused on another phone or device. This makes the whole system more fail-proof.
– Most modern computers can handle SD-cards and hence data can be easily backed up.
– Other applications on the Android phone running the mobile Semantic Web client can access and modify the data stored on the card. Thus they can further annotate the information and the client can again take advantage of such annotations.

4.3 Rules and Data Processing

One of our initial requirements from Section 2 is flexibility in the sense that specific vocabulary resources should not be encoded in the source code of the WebID provider. In order to achieve this requirement, we decided to encode as much data processing as possible in terms of user extensible rules. Since we employ the *androjena framework*, we were able to use the included Jena rules engine as well. All rules processed by this rule-based reasoner are defined as lists of body terms (premises), lists of head terms (conclusions) and optional names[9].

```
1  @prefix foaf: <http://xmlns.com/foaf/0.1/>.
2  @prefix android: <http://ns.aksw.org/Android/>.
3  @prefix acontacts: <http://ns.aksw.org/Android/
       ContactsContract.CommonDataKinds.>.
4  @prefix im: <http://ns.aksw.org/Android/ContactsContract.
       CommonDataKinds.Im.>.
5
6  [jabber:
7    (?s foaf:jabberID ?o), makeTemp(?d) ->
8      (?s android:hasData ?d),
9      (?d rdf:type acontacts:Im),
10     (?d im:DATA ?o),
11     (?d im:TYPE im:TYPE_HOME),
12     (?d im:PROTOCOL im:PROTOCOL_JABBER)
13 ]
```

Listing 4. Example transformation rule: If a `foaf:jabberID` is present with a WebID (line 7), then a new blank node of RDF type `acontacts:Im` is created (line 7), which is of Android IM type `HOME` (line 11) and which gets an IM protocol as well as the IM identifier (line 12 and 10)

Since we also did not want our implementation to depend on the FOAF vocabulary (alternative solutions include RDF vCards [7]), we decided to create a native Android system vocabulary which represents the Android contacts database defined by the Android API. This vocabulary is deeply integrated into the Android system since it re-uses class and attribute names from the Android API and represents them as OWL class and datatype properties[10].

Based on this vocabulary, the given rules transform the downloaded WebID statements into Android-specific structures which are well suited for a straightforward import into the contacts provider. These structures are very flat and

[9] `http://jena.sourceforge.net/inference/#RULEsyntax`

[10] An example class name is `ContactsContract.CommonDataKinds.StructuredName`, which is represented in the vocabulary as an OWL class with the URI `http://ns.aksw.org/Android/ContactsContract.CommonDataKinds.StructuredName`. We published the vocabulary at `http://ns.aksw.org/Android/`. Please have a look at the Android API reference as well (`http://developer.android.com/`).

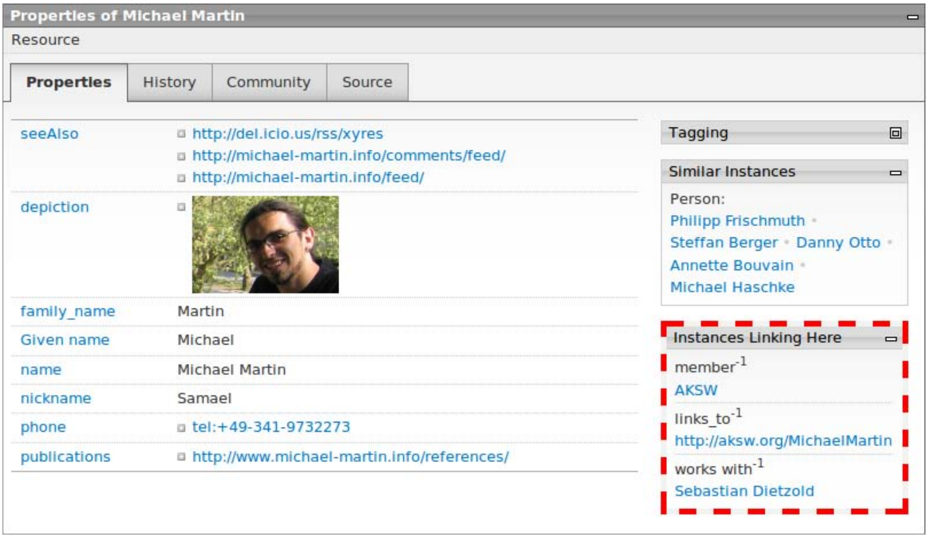

Fig. 3. Visualization of a WebID in OntoWiki: incoming backlinks (via Semantic Pingback) are rendered in the "Instances Linking Here" side box

relate different Android data objects (e.g. email, photo, structured name etc.) via a `hasData` property to a WebID. An example rule which creates an instant messaging account for the contact is presented in Listing 4.

After applying the given set of rules, the application post-processes the generated data in order to apply other constraints which we could not achieve with Jena rules alone. At the moment all `mailto:` and `tel:` resources are transformed to literal values, which is required for instantiating the corresponding Java class. In addition we download, resize and base64-encode all linked images. After that, the application goes through the generated data resources and imports them one by one.

4.4 OntoWiki

The mobile Semantic Web client supports arbitrary WebIDs, even those backed by plain RDF files. Nevertheless, some features require special support on the server-side. For our semantic data wiki OntoWiki [1] we implemented all functionalities required for a complete distributed Social Web experience. Any user can setup his own OntoWiki instance, which will then provide him with an enhanced WebID.

If configured properly a user can create a self-signed certificate with very little effort. Such a certificate contains the generated WebID as a Subject Alternative Name (SAN) and is directly imported into supported Web browsers[11]. From the browser the certificate can be exported in *PKCS12* format and stored on a

[11] A list of supported browsers is available at `http://esw.w3.org/Foaf+ssl/Clients`.

SD-card used by a mobile phone running the client. Since OntoWiki supports FOAF+SSL authentication, a user can split his data in publicly visible information and such, that is only accessible by people which have a certain relationship with the user (e.g. a `foaf:knows` relation).

Semantic Pingback is another technology supported by OntoWiki. Thus an arbitrary user can add a relationship to an OntoWiki backed WebID and as a result the WebID owner will be notified, enabling the user to take further actions (see Figure 3). In the use case of the mobile Semantic Web client this is especially useful for a first contact between users. In typical social network applications this step would be the "Add as a friend" step. In a distributed scenario, however, if one states that she is a friend of someone else, she would allow that person to view the data dedicated to be displayed by friends only. If both endpoints add that relation on their respective side, they can see each other's private data and thus are considered friends (in the Social Web sense). The Social Web has a very dynamic nature and information is changed frequently or new data is added. Hence, editing functionality is another important aspect and OntoWiki supports editing via SPARQL/Update.

5 User Perspective

The Mobile Semantic Social Web client implementation consists of two software packages - the *Android Semantic Web Core library* containing the triple store and the *WebID content provider for Android*. Both are available on the *Android Market* since August 2010 (cf. screenshot A in Figure 4). According to the market statistics, they were downloaded overall more than 400 times and are currently installed on more than 100 devices.

Once installed few initial configuration options have to be supplied. Screenshot B in Figure 4 shows the accounts and sync settings configuration menu, which allows a user to associate his WebID with his profile on the smartphone (the same way as adding an LDAP or Exchange account) and to configure synchronization intervals. Screenshot C shows an actual WebID with the last synchronization date and the option to trigger the synchronization manually.

After the user associated his profile with his WebID, information from linked WebIDs of the users contacts are synchronized regularly and the information are made available via the Android content provider to all applications on the device. During the import of the WebID contacts, they are merged based on the assumption of unique names. Independent of this automatic merge, the user can split and merge contacts manually in the edit view of these contacts. Screenshot D shows the standard Android contact application, where our WebID content provider seamlessly integrates information obtained from WebIDs. Information obtained from WebIDs is not editable, since it is retrieved from the authoritative sources, i.e. the WebIDs of the respective contacts.

Screenshot E shows the FOAF browser, allowing people to add contacts or to browse the contacts of their friends. In order to facilitate the process of connecting with new contacts the Android implementation also allows to scan QR-codes of WebIDs (e.g. from business cards) and to search for WebIDs using Sindice.

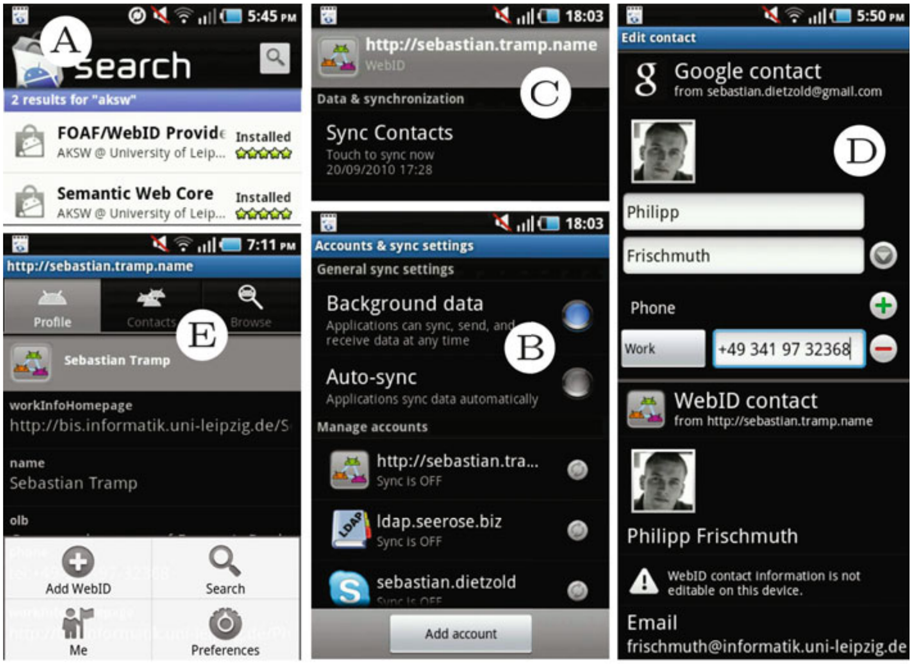

Fig. 4. Screenshots of the Mobile Social Semantic Web Client, the FOAF Browser and the Android components which integrate the WebID account: (A) The client as well as the triple store can be found in the official Google application market. (B) After installation, users can add a WebID account the same way they add an LDAP or Exchange account. (C) The account can be synchronized on request or automatically. (D) A contacts profile page merges the data from all given accounts. (E) By using the FOAF browser, people can add contacts or browse the contacts of their friends.

6 Related Work

Related work can be roughly divided into semantics-based (but centralized) social network services, distributed social network projects, mobile Semantic Web projects and mobile social network clients. A comprehensive overview is contained in the final report of the W3C Social Web Incubator Group [6]. In the sequel we present some related approaches along the four dimensions in more detail.

Semantics-based (but centralized) social network services. Evri[12] is a centralized social network based on RDF and mainly used for collaborative information storage. Evri also maintains mobile applications for iOS and Android to give their users access to these information.

[12] `http://www.evri.com` (formally know as Twine).

Distributed social network. The idea of distributed social networks appeared quickly after social networks became popular. A distributed social network traditionally refers to an Internet social network service that is decentralized and distributed across different providers, with emphasis on portability and interoperability. The most prominent representatives are Diaspora[13], NoseRub[14] and GNU social[15]. Although some of these networks make use of vocabularies (e.g. FOAF in the case of GNU Social) or certain elements of semantic technologies, their use of Semantic Web technologies and best-practices is rather limited.

Mobile Semantic Web applications. The application of Semantic Web technologies on mobile devices is not new - one of the earlier works dates back to 2003 [10]. However, this research area was not pursued very actively due to the large number of mobile device limitations common in that era. In the light of increasing processing power and data connectivity of modern mobile devices, the use of mobile Semantic Web technologies is becoming more feasible. There are a few publications which review the state of the mobile Semantic Web and the possibilities thereof to improve mobile Web (e.g. [9]). Also, there are publications on more complex systems, such as *SmartWeb* [15] which uses semantic technologies to enhance the backends of mobile web service. On the frontend side there are semantic mobile applications like *DBpedia Mobile*[2] or *mSpace Mobile* [19] that implement or utilize Semantic Web technologies directly on mobile devices. However, these frontend applications focus on very specific use cases - information about points of interest in the case of DBpedia Mobile and information for university students in the case of mSpace. Finally, there are publications which describe proof of concept applications as well as algorithms for consuming and replicating Linked Data and RDF in general [5,11,13].

Mobile social networking clients. All major social networking services (such as Facebook, Myspace, LinkedIn etc.[16]) have meanwhile clients for different mobile platforms, which are more or less integrated with the mobile phone platform itself. In addition to this, the Android market place lists more than 3500 applications in the category social networks with our implementation being one of them. However, up to our knowledge our MSSW client is the first to consequently employ W3C standards as well as Social Semantic Web best practices with regard to all aspects of data representation and integration.

7 Conclusion and Future Work

We see the work described in this article to be a further crucial piece in the medium-term agenda of realizing a truly distributed social network based on

[13] `https://joindiaspora.com`
[14] `http://noserub.com`
[15] `http://www.gnu.org/software/social/`
[16] An ordered list can obtained from Wikipedia: `http://en.wikipedia.org/wiki/List_of_social_networking_websites`

semantic technologies. Since mobile devices are playing an increasingly important role as clients and platforms for social networks, our realization focused on providing a extensible framework for social semantic networking on the Android platform. With this work we aimed at showcasing how different (social) Semantic Web standards, technologies and best practices can be integrated into a comprehensive architecture for social networking (on mobile devices).

With regard to future work we plan to further decrease the entrance barrier for ordinary users. A current obstacle is that users are required to have a WebID and - if they want to use authentication and access control features - a FOAF+SSL enabled WebID. In particular creating a FOAF+SSL enabled WebID is, due to the certificate creation, still a cumbersome process. A possible simplification of this process would be to enable mobile phone users to create and upload the required profile and certificates directly from their mobile device. We also plan to implement a more efficient and user-friendly way for subscribing to updates of contacts. These will include profile changes, status updates, (micro-)blog posts as well as updates retrieved from social networking apps. This feature would be facilitated by a proxy infrastructure, which caches updates until the device re-connects to the network after a period of absence (e.g. due to limited network connection or switched-off devices). A further important aspect to be developed is the standardization and realization of social networking applications, which seamlessly integrate with and run on top of the distributed social semantic network. Such applications would comprise everything we know from centralized social networks (e.g. games, travel, quizzes etc.), but would make use of FOAF+SSL and the other distributed social networking components for authentication, access control, subscription/notification etc.

Acknowledgments. This work was supported by a grant from the European Union's 7th Framework Programme provided for the project LOD2 (GA no. 257943).

References

1. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 736–749. Springer, Heidelberg (2006)
2. Becker, C., Bizer, C.: Exploring the Geospatial Semantic Web with DBpedia Mobile. J. Web Sem. 7(4), 278–286 (2009)
3. Berners-Lee, T.: Long Live the Web. Scientific American (December 2010)
4. Brickley, D., Miller, L.: FOAF Vocabulary Specification. Namespace Document September 2, FOAF Project (2004), `http://xmlns.com/foaf/0.1/`
5. David, J., Euzenat, J.: Linked data from your pocket: The Android RDFContentProvider. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, Springer, Heidelberg (2010)

6. Halpin, H., Tuffield, M.: A Standards-based, Open and Privacy-aware Social Web. W3C Incubator Group Report, W3C (December 2010)
7. Iannella, R., Halpin, H., Suda, B., Walsh, N.: Representing vCard Objects in RDF. W3c Member Submission, W3C (January 2010)
8. Langridge, S., Hickson, I.: Pingback 1.0. Technical report (2002), http://hixie.ch/specs/pingback/pingback
9. Lassila, O.: Using the Semantic Web in Mobile and Ubiquitous Computing. In: Proc. of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web, Jyväskylä, Finland. IFIP, vol. 188, pp. 19–25 (2005)
10. Lassila, O., Adler, M.: Semantic Gadgets: Ubiquitous Computing Meets the Semantic Web. In: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, pp. 363–376. MIT Press, Cambridge (2003)
11. Le-Phuoc, D., Parreira, J.X., Reynolds, V., Hauswirth, M.: RDF On the Go: An RDF Storage and Query Processor for Mobile Devices. In: Posters and Demos of the ISWC 2010, Shanghai, China (2010)
12. Passant, A., Mendes, P.: sparqlPuSH: Proactive notification of data updates in RDF stores using PubSubHubbub. In: SFSW 2010 (2010)
13. Schandl, B., Zander, S.: Adaptive RDF Graph Replication for Mobile Semantic Web Applications. Ubiquitous Computing and Communication Journal (Special Issue on Managing Data with Mobile Devices) (August 2009)
14. Schwagereit, F., Scherp, A., Staab, S.: Representing Distributed Groups with dgFOAF. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 181–195. Springer, Heidelberg (2010)
15. Sonntag, D., Engel, R., Herzog, G., Pfalzgraf, A., Pfleger, N., Romanelli, M., Reithinger, N.: SmartWeb handheld — multimodal interaction with ontological knowledge bases and semantic web services. In: Huang, T.S., Nijholt, A., Pantic, M., Pentland, A. (eds.) ICMI/IJCAI Workshops 2007. LNCS (LNAI), vol. 4451, pp. 272–295. Springer, Heidelberg (2007)
16. Sporny, M., Corlosquet, S., Inkster, T., Story, H., Harbulot, B., Bachmann-Gmür, R.: WebID 1.0: Web identification and Discovery. Unofficial draft (August 2010), http://payswarm.com/webid/
17. Story, H., Harbulot, B., Jacobi, I., Jones, M.: FOAF+SSL: RESTful Authentication for the Social W. In: SPOT 2009 (2009)
18. Tramp, S., Frischmuth, P., Ermilov, T., Auer, S.: Weaving a Social Data Web with Semantic Pingback. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS (LNAI), vol. 6317, pp. 135–149. Springer, Heidelberg (2010)
19. Wilson, M., Russell, A., Smith, D.A., Owens, A., Schraefel, M.C.: mSpace Mobile: A Mobile Application for the Semantic Web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, Springer, Heidelberg (2005)