We are now here: Location-Based Social Services

Philipp K. Jaquet - philipp.jaquet@uni-muenster.de
Karsten Jeschkies - karsten.jeschkies@web.de
Bettina Budeus - budeusb@uni-muenster.de
Prof. Dr. Klaus Hinrichs
Department of Computer Science
University of Münster
http://viscg.uni-muenster.de

Abstract

The arrival of the digital era results in the dissociation of a human in at least two identities: Apart from the well-known "'offline identity"', nowadays every user possesses several "'online identities"': social media bundle the interests, skills and photos to an online counterpart to virtually every Internet user. This work identifies the different online identities of a user and presents an idea on how to combine the online identities with the offline counterpart. This is done with an application that combines as much online identities as possible and provides them with spatial data derived from a GPS drive in a current smartphone. With this application, new ways of interaction between users are possible. We show that there are lots of interesting and exciting topics for further investigation and point out that the connection with social services changes the way we perceive the web and, in turn, enables the web to add semantics to its contents.

Introduction

The introduction of the Internet in the 1990s did not only change the way we perceive and use technology, but also had a great impact on our self-concept and on the way we define our identity. Vossen and Hagemann [16] describe the phenomenon known as "Web 2.0" as a confluence of three streams: The "application stream" describes the development of applications which are available online, the "technology stream" summarized developments on hardware and software, and the "User participation and contribution stream" describes the perception and interaction of a user with the Internet. The way how a user perceives the Internet has fundamentally changed during the recent years: While the Internet started as a collection of more or less static information-conveying pages, a modern web page normally provides very little (if any) information by itself, but works as a platform for users to contribute their own content. According to Leverne [10] Social Networks add a dimension to the Internet which exceed mere linking between pages: Social Networks add links between people or groups. Those links seem to add a new awareness for other people: The successful online service Twitter, for example, enables anyone to share short messages, encouraging the users to tell others what he is currently doing or where he is at the moment. [7]

Online and offline identity

This novel way of expressing oneself leads to a novel way of self-conception: The well-known 'offline identity' is being complemented by an online counterpart: The 'online identity' summarizes the user's needs and interests, reflects his social network and has to be well-tended: For example, the usage of web-technologies plays an important role in the recruitment process of many companies [3]. The online- and the offline identity are typically not congruent: For example, the online identity might list more people as 'friends' than the offline counterpart, and most certainly the offline identity has certain friendships or social bounds (such as grandparents) that are unknown to the online counterpart. However, this situation is not always desirable. For example, if two people meet at a conference, they might want to stay in contact afterwards - the current situation forces both to exchange some sort of identifier like the e-mail address or a phone number. This identifier has to be unique, otherwise it will be hard to find the correct person online (See Figure 1) Even with business cards the user has to pre-select the identifiers he wants to share without even knowing who he will be meeting. Vice versa, with current social graph-based networks - networks that display "'friends of friends"' in addition to the contacts - a user's online identity might be aware that the person next to him shares the same interests or might even be a mutual friend, while the offline identity doesn't. Even the instant access to the online counterpart does not help as it is only familiar with the online identity of the other person - and not with its offline identity next to the user.

Hybridization and interaction

Current Smartphones like Apples iPhone are by default equipped with a GPS module which allows the accurate localization of the device. However, even without GPS-Devices, it is still possible to estimate the user's position using the information provided by surrounding hot spots. This development does not only allow the user to know where he is, he is also enabled to add a spatial coordinate to his online identities. This results in a hybridized identity, where the offline identity carries his online identities around in his pocket, as shown in Figure 1. This does not only solve the problems previously described, but allows new means of interaction between two users, using the benefits of both "worlds". The application provided is not more than a small "proof-of-concept", further ideas are yet to be implemented and are listed in the "Discussion and Future Work" - Section of this work.

Related Work

The idea of combining social media with location based data is currently a very active area: Some of the big social networks such as Facebook or Twitter are currently working on the integration of location based data in their services. On the scientific side, there is also a lot of research going on: Kjeldskov and Paay [6] described a system that allows the context-aware collaboration of users via smartphones, but without integrating information derived from social networks. Yamamoto et al. [18] evaluated the benefits and drawbacks of location based social networks based on the social connections among students, but without any community functions that form an important part of current social networks. In 2008, Singla and Richardson stated that people who interact often via instant messaging are more likely to share interests and have similar web searches [15], a finding which motivates the work on a semantic web using social services. The works of Mika [11], Wu et al. [17], Niepert [12] and others already show promising results.

Figure 1: A human and his online identities In the era of social networks, most users have one or more online profiles, which represent their needs and interests. Using positional data, those online identities can be combined in order to hybridize the online- and the offline-identity. If two persons meet, they have to share identifiers such as a e-mail address or the full name to enable the other person to connect with him.

System Description

The provided system is a Client-Server-Architecture with the client software written in Objective-C and the Server-Software written in Python and Java. On the server side, the open source software "Openfire" by "Ignite Realtime" has been used. This software is a free Java-Program that complies with the Extensible Messaging and Presence Protocol (XMPP) [13]. This protocol is nowadays often used in chat-applications such as Jabber, but provides many extensions that match the needs of this project. The Openfire-Server was used with default settings and talked to a program, the backend. The backend is a program based on the Twisted framework [9] and is written in python. It is responsible for storing and aggregating the user data as well as maintaining connections among the users. The client software was written in Objective-C and programmed for Apples iPhone. It used an existing XMPP-Framework [5] for the communication with the server. Figure 2 shows the three parts of the system, their responsibilities and the communications.

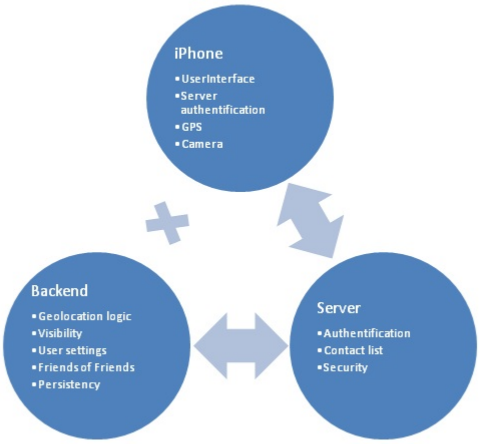


Figure 2: The three parts of the system The system consists of one backend, several clients (iPhones) and a Jabber server. The server provides the messaging functionality as well as user registration, authentification and the contact list. The clients provide the user interface and the hardware such as the GPS device. Every important action is performed by the backend. The client send messages via the server to the backend which performs a calculation, if necessary, and sends the result to one or more clients via the server.

The XMP-Protocol

The XMP-Protocol is an XML-based protocol developed in 1999 for the use in real-time chat programs [13]. It consists of a core protocol and various Extension Protocols (XEPs), which unveil the true potential of the protocol: there is a XEP for virtually every need, which means that in order to write an application that supports certain features, it simply has to implement the corresponding XEPs and can then easily interact with any other application that used this XEP. Additionally, with a corresponding Service Discovery XEP, the communicating applications do not need to implement every XEP of each other. The XEPs used in this work are listed in Figure 3, an example XEP is shown in Figure 4).

XEP Number	Name
XEP-0255	Location Query
XEP-0009	Jabber-RPC
XEP-0008	IQ-Base-Avatar
XEP-0244	Attention

Figure 3: The XEPs used in this application. This table shows the XEP which have been used in this application. The Location Query defines a protocol for broadcasting and retrieving the location of a user. The Jabber-RPC (Remote Procedure Call) is used to execute a command on the server and retrieve the results, for example while changing the user settings. The IQ-Base-Avatar XEP is used to set the images for the "'BuddySight"' - Feature (see Figure 11). XEP-0244 aims to attract the attention of a user and has been used in the "'Nudge"' - Feature (see Figure 8).

```
<iq from='hamlet@maraudersserver.com/phone'
    id='q01'
    to='back.maraudersserver.com'
    type='get'
    xml:lang='en-US'>
  <locationquery xmlns='urn:xmpp:locationquery:0'>
    <lat>57.0501862</lat>
    <lon>9.9188746</lon>
    <accuracy>35.6</accuracy>
    <street>Jomfru Ane Gade 13</street>
    <locality>Aalborg</locality>
    <country>Denmark</country>
    <accuracy>20</accuracy>
    <uri>http://shakespeare.lit/places/</uri>
    <text>King's␣Head␣Pub</text>
␣␣</locationquery>
</iq>
```

Figure 4: XEP 0255: Location Query. With this XEP, the user "'hamlet"' sends his location to the Openfire server, which then sends it to the backend according to the address in the "'to"' field. From all nodes below "'locationquery"', only "'lat"' and "'lon"' are required. The values of the other fields are derived by reverse geocoding, if possible. The field "'accuracy"' describes the error in GPS measurement. See `http://xmpp.org/extensions/xep-0255.html` for the whole specification.

Openfire

Openfire [8] is a free open source XMPP Server written in Java. It handles the basic protocol and thus makes up a full Jabber server. This includes the registration of new users, authentication and simple messaging. Openfire has also an add on to run on a cluster system which leads to a great scalability. The XEPs were implemented within an XMPP component using the Python framework Twisted. An XMPP component is something like a client with many privileges. Many instances of a component can connect to one Openfire server. Openfire distributes the incoming messages to the instances using the Round Robin paradigm. This approach ensures, again, a high scalability.

Using an open source implementation of XMPP ensures also a high degree of safety.

In conclusion Openfire allowed the developers to concentrate on the core feature of the program and not on low level details.

Twisted

Twisted [8] is an event-driven networking framework written in Python. It supports a lot of protocols including XMPP. The event driven asynchronous approach allows a short response time to requests. The implemented protocols can also be extended fairly easily. This in combination with the scripting language Python leads to a fast implementation of the logic of the program. In fact, it took unexpectedly less time to develop the server than the iPhone client.

The iPhone

The iPhone is a smartphone developed by Apple, Inc. It is equipped with a multi-touch input device, a GPS module, an orientation sensor and is capable of connecting to the Internet using wireless lan as well as 3G technology. The relevant hardware details of the iPhone are listed in Figure 5. Apples iPhone Developer SDK comes with a vast API, Cocoa, as well as the "Human Interface Guidelines" [1], which enables programmers to create applications which are both attractive and easily usable as Apple encourages programmers to re-use styles and metaphors known from other iPhone applications. As the application's target group is clearly dominated by non-professional default users of an iPhone, great care was taken to comply with the Human Interface Guidelines. An open source library, XMPPFramework [5], has been used to provide basic XMP-Functions such as establishing and maintaining a connection. Every XEP used apart from that had to be implemented manually.

Feature	**Model**
Multi-Touch Display	3G, 3GS, iPod touch
Camera	3G, 3GS
GPS	3G, 3GS
Compass	3GS
WiFi	3G, 3GS, iPod touch
Orientation Sensor	3G, 3GS, iPod touch

Figure 5: Hardware Details of the compatible devices. This table shows the key features of Apples iPhone (Models "'3G"' and "'3GS"') and iPod Touch. The application was built to be compatible with all three devices. To use every function of the application, a iPhone 3GS has to be used.

Application

The application's interface consists of three parts ("tabs"): The Map, the Event List and the Preferences.

Preferences

The preferences tab allows the user to change their settings. This includes application-specific settings as well as information specific for any social network. As a placeholder for social networks and other online identities, the user's mail address can be entered in the application. Furthermore, it can be triggered whether the user wants to forward certain events to his e-mail address, an analogon to the possibility to publish certain events directly to a social network.

Visibility Settings

The most important application-specific settings are the visibility settings. First, the application distinguishes between Friends, Acquaintances and Foreigners. Friends share a direct connection to the user. This means that one of both the user and the friend had to query a "friendship request", which had to be accepted by the other person. According to the user's privacy settings, the other user may be able to see the mail address of the user, a prototype for seeing all the networks a user belongs to (or even establishing a connection in a shared social network, which depends on the API provided by the social network). An Acquaintance is not directly connected to the user, but shares a friendship connection with another user who is, in turn, friends with the user himself. A Foreigner is a person who is neither a friend nor an acquaintance. The main idea behind the visibility settings is adapting the concept of body size and using it as a metaphor within the application. In the real world, large objects are more likely to be seen than smaller ones. While humans do not vary much in their height, this concept makes sense with other objects - for example steeples, which can be seen from nearly every point in a small city, while somebody on the top of the steeple might not be able to see all the people who look at the steeple, as they might be too far away or too tiny to recognize. In the application, the user is able to choose his size - or, from a different point of view, he chooses the area around him in which he is visible to either friends, acquaintances or foreigners. See Figure 6 for details.

Events & Contacts

The Events & Contacts part of the application allow the user to access the available information in a structured, table-based way. Everything that is displayed at the map is also displayed here, with additional information such as the distance from the current location to the location of the list item and the address of the list item (derived by reverse geocoding, a

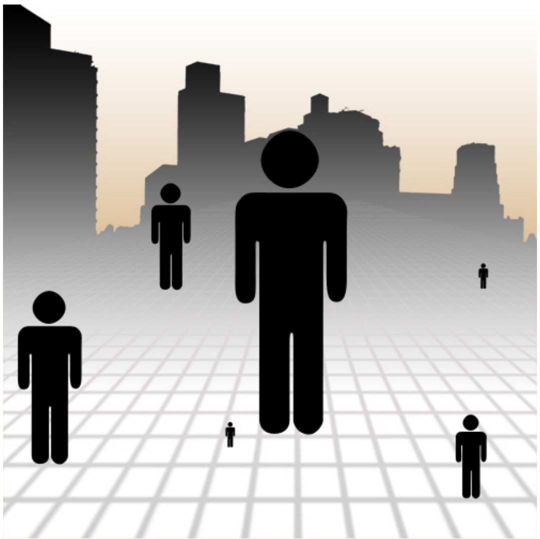
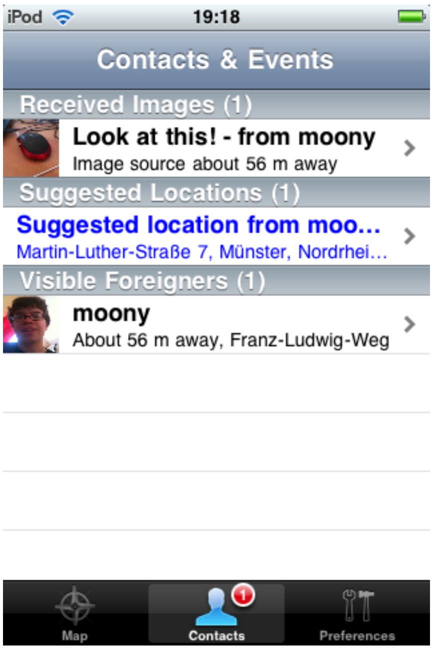


Figure 6: Visibility and body height This image shows the context of visibility used in this application: Larger persons are more likely to be seen than smaller ones, but do not necessarily see more: Even if they are close, very small persons may be overlooked by larger ones.

concept which derives a dictionary of human-readable address information from a given Geo location. One approach is described in [4].). The information are divided in different categories for visible users, distinguishing between Friends, Foreigners and Acquaintances as well as their visibility status and a category called "events", for other information that have been received, such as images or locations. The Events & Contacts list is shown in Figure 7

The Map

As the name suggests, the last tab of the application provides an isometric view of the area around the user. The map centers on the user by default, but since the iPhone provide a compass, the application also enables him to automatically orient the map according to the orientation of the device. The map shows other users with different colors according to their relationship to the user (Friend, Acquaintance, Foreigner) and their names. Other events, such as shared locations by other users, are displayed as a small pin. As described earlier, the user is only able to see users who have set their "size" high enough for the user to be able to see him - as soon as the user leaves this radius (or, respectively, the other user moves away), the corresponding icon disappears from the map. If both users share a "friend" - connection, the user is still able to use the "Events & Contacts" List to see that there is a friend who is logged in but currently invisible. The map is displayed in Figure 8



iPod 19:18

Contacts & Events

Received Images (1)

Look at this! - from moony
Image source about 56 m away

Suggested Locations (1)

Suggested location from moo...
Martin-Luther-Straße 7, Münster, Nordrhei...

Visible Foreigners (1)

moony
About 56 m away, Franz-Ludwig-Weg

Figure 7: Contacts and Events Every information displayed on the map is also displayed in the Contacts & Events list. New events are highlighted in blue and promoted in a red badge on the tab icon that leads to this list until they are selected for the first time. The positional information of the events update with each GPS update performed by the iPhone.

Interacting with other users

By selecting another user - either from the map or from the Events & Contacts List - the program presents various means of interacting with the user. Interaction methods combine common social network features such as the ability to share an image with Geo coordinates provided from the iPhone. The application supports only a small fraction of all possible interaction methods, as it was intended to be not more than a prototype. Figure 9 shows an overview of all implemented interaction methods. The ability to get the attention of another user is shown in Figure 8, sharing a location is shown in Figure 10. As it is not possible to stream videos with the iPhone, a more sophisticated interaction method called BuddySight was implemented. Every 10 seconds, it takes a picture of the current location and broadcasts it to other users, as shown in Figure 11. The receiver's iPhone stores the last 10 images as well as the last 10 positions, which allows the user to re-view where the sender was and what he saw on his way.

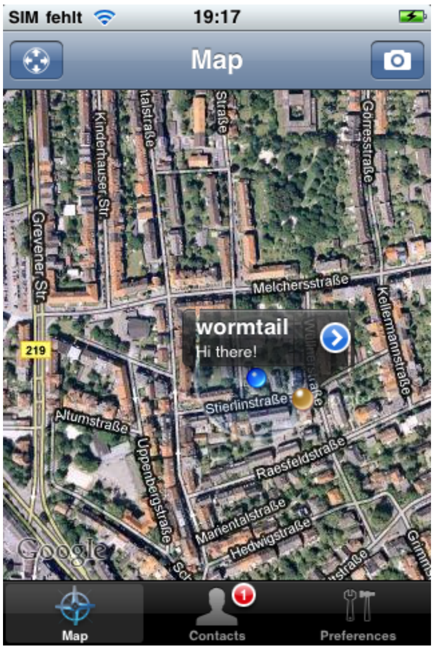
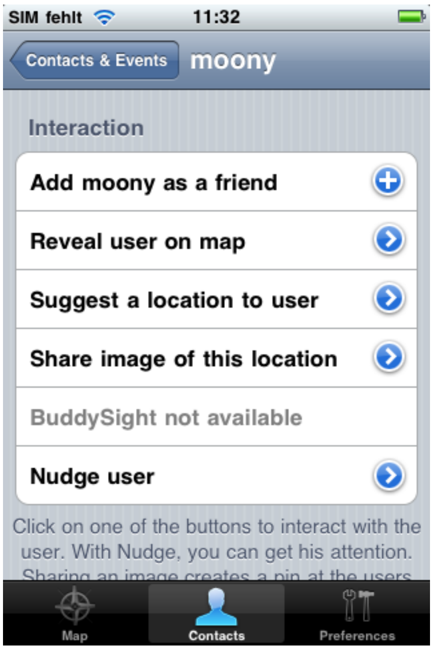


Figure 8: The Map The map provides a bird's eye view of the user (represented by a blue icon) and other users. The yellow icon represents foreign users, i.e. a user to which no friendship connection exists. In this image, the user ”‘wormtail”’ has performed a ”‘nudge”’ to get the attention of the current user. This is also indicated by the small number on the ”‘Contacts”’ Button. The button on the navigation bar above the map allow the user to change his view mode, where the map automatically aligns itself with the iPhone's compass.



Click on one of the buttons to interact with the user. With Nudge, you can get his attention. Sharing an image creates a pin at the users

Figure 9: Interacting with other figures After selecting another person, the user can choose an interaction. Adding a user as a friend establishes a connection between both users. ”‘Reveal on map”’ centers the map on the user. ”‘Suggest a location”’ is explained in Figure 10. ”‘Share image of this location”’ takes a picture, which is then send with the current location to the receiver. On the receiver's map, the image's location is represented by a pin. The BuddySight-Feature is shown in Figure 11, nudging other users is explained in Figure 8.

Sending Geolocations

The key part of the application is the ability to send and receive geolocations. Using the iPhone's GPS module and its ability to perform reverse geocoding, the GPS data from the iPhone are translated into Human-Readable addresses and are then transmitted using the XEP shown in Figure 4 . As Apple prohibits the excessive use of the reverse geocoder, the human-readable address is only updated after the device has significantly changed its position. The whole process is repeated every 5 seconds. Figure 12 shows the process of sending and receiving geolocations in more detail.

Discussion and Future Work

Evaluation of the application

As mentioned before, the application developed in this work is not more than a demonstration of the main ideas of the concepts, such as body height as a metaphor for a visibility range. From a programmer's point of view, the use of the XMP-Protocol is cumbersome as it requires a lot of overhead: For example, a simple remote method invocation on the backend requires a translation to a XML stanza, which is then send to the backend which, in turn, has to revert the translation process to obtain the name and the parameter list of the method that should be called. Figure 13 shows an example of this situation. On the other hand, once implemented, the XMP-Protocol provides a reliable base of the client-server-connection and can easily be extended.

Human-Interface-Guidelines for maps

While Apple provides good guidelines on how to design applications on the iPhone in general, it still lacks sophisticated guidelines for the interaction with an information-enriched map. As a map provides information distributed on a plane with very large extents, therefore the ability to zoom the map as well as scrolling in every possible direction is crucial. Apart from this, there are currently only very little approaches on how to organize and display the informa-

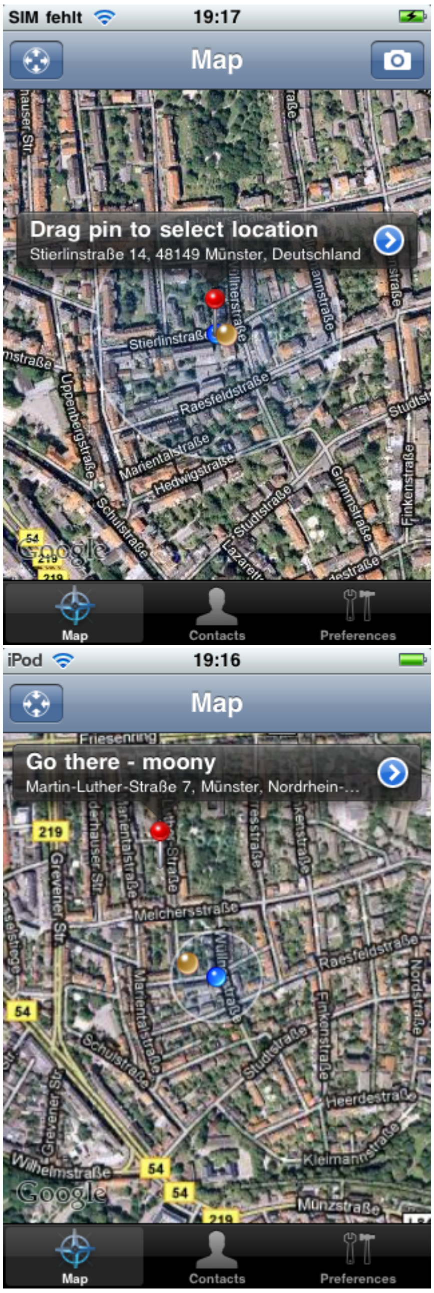

SIM fehlt 19:17
Map
Drag pin to select location
Stierlinstraße 14, 48149 Münster, Deutschland
Map
Contacts
Preferences
iPod 19:16
Map
Go there - moony
Martin-Luther-Straße 7, Münster, Nordrhein-...
Map
Contacts
Preferences

Figure 10: Share a location To share a location, the user drags a pin to the location he wishes to share (first image). He is able to add a text to the location before sending it. The other user receives the pin as well as the provided message (second image).

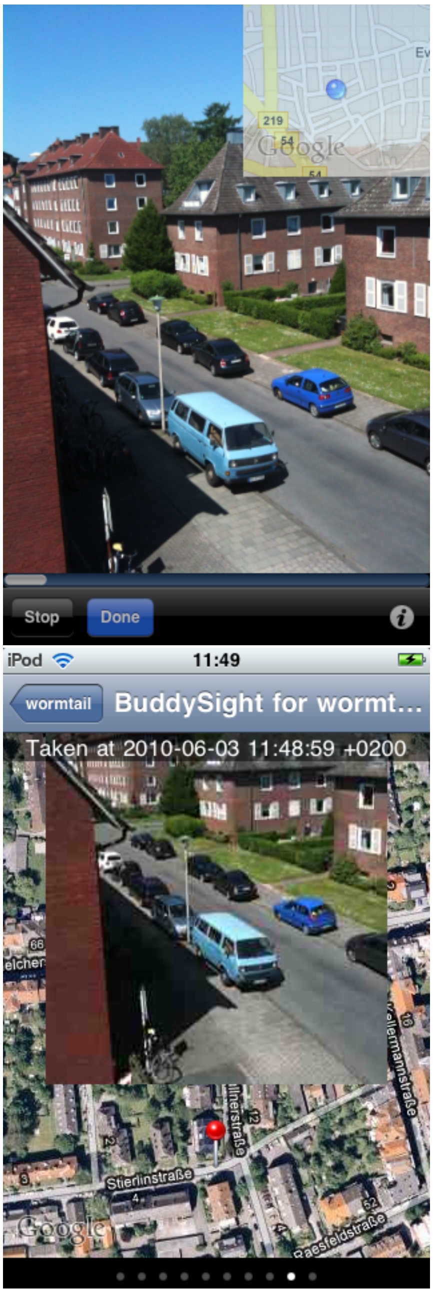


Figure 11: BuddySight The BuddySight - Feature takes a picture every ten seconds and sends it to everyone who is able to see the user (first image). The second image shows how these images are displayed on the receiver's device: the map shows where the image has been taken. This feature is a replacement for video streaming, which was not possible with version 3 of the Apple iPhone Developer Kit. The last 10 images and positions are stored on the receiver's iPhone and make it possible to re-view the way a user walked and what he saw.

tion in a coherent, user-friendly manner on compact displays such as a mobile phone.

Enriching the environment

One of the most interesting ideas that still has to be evaluated is not only to include social service identities into the application, but to extend this concept by other online interactive features. For example, web polls are widely spread over the Internet. With a spatial coordinate, a user could be able to set up a location-based poll with which he is able to see how

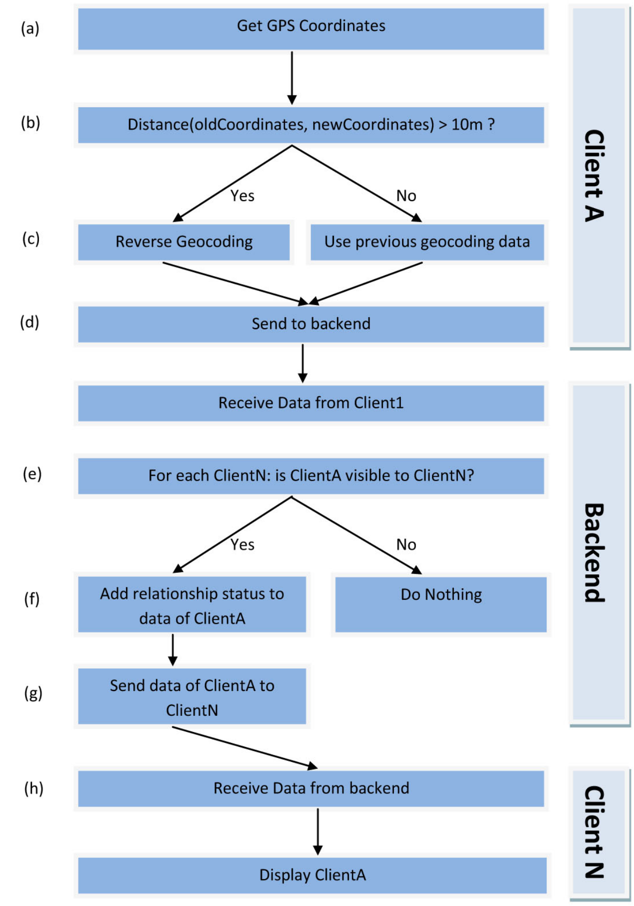
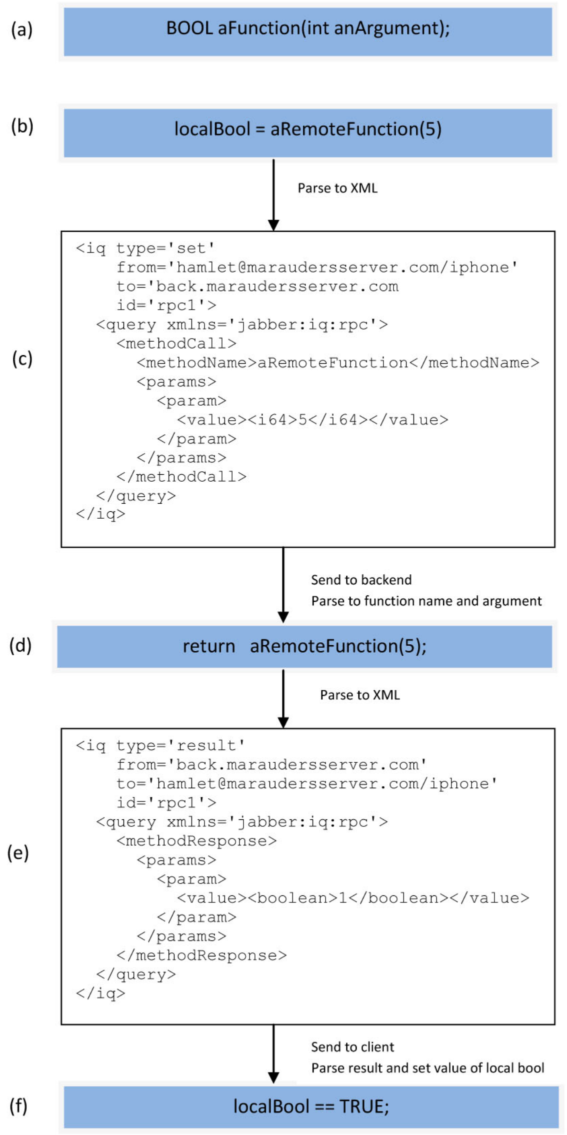

(a)
Get GPS Coordinates
(b)
Distance(oldCoordinates, newCoordinates) > 10m ?
Yes
No
(c)
Reverse Geocoding
Use previous geocoding data
(d)
Send to backend
Receive Data from Client1
(e)
For each ClientN: is ClientA visible to ClientN?
Yes
No
(f)
Add relationship status to data of ClientA
Do Nothing
(g)
Send data of ClientA to ClientN
(h)
Receive Data from backend
Display ClientA
Client A
Backend
Client N

Figure 12: Sending and receiving geolocations This figure shows the process of sending and receiving locations. Every five seconds, a client gets the GPS data from the iPhone's GPS module **(a)**. As the reverse geocoding functionality is costly and, according to Apple, might be blocked if it is performed too often, the client checks if the new position differs significantly from the last reverse geocoded position **(b)**. The reverse geocoding provides human-readable information about the location, such as the street name, the city name et cetera. These are added to the new GPS data **(c)** and then send to the backend **(d)**. The backend uses the user settings to decide whether the client is visible to any other logged-in client **(e)** If it is, the backend adds the correct relationship status (Friend, Acquaintance or Foreigner) to the positional data **(f)** before those are sent to anyone who is able to see the client **(g)**. After receiving positional data from the backend, the sender is displayed on the map with an icon that matches the relationship status **(h)**

many of his friends like his university - or even who simply likes a certain place. With the concepts introduced earlier, such a poll might have a different size for friends (who, for example, may be able to see it no matter where they are) and for foreigners (who might in this case only participate if they are exactly at the place to which the poll refers).


(a)
BOOL aFunction(int anArgument);
(b)
localBool = aRemoteFunction(5)
Parse to XML
(c)
Send to backend
Parse to function name and argument
(d)
return   aRemoteFunction(5);
Parse to XML
(e)
Send to client
Parse result and set value of local bool
(f)
localBool == TRUE;

```
<iq type='set'
    from='hamlet@maraudersserver.com/iphone'
    to='back.maraudersserver.com
    id='rpc1'>
  <query xmlns='jabber:iq:rpc'>
    <methodCall>
      <methodName>aRemoteFunction</methodName>
      <params>
        <param>
          <value><i64>5</i64></value>
        </param>
      </params>
    </methodCall>
  </query>
</iq>
```

```
<iq type='result'
    from='back.maraudersserver.com'
    to='hamlet@maraudersserver.com/iphone'
    id='rpc1'>
  <query xmlns='jabber:iq:rpc'>
    <methodResponse>
      <params>
        <param>
          <value><boolean>1</boolean></value>
        </param>
      </params>
    </methodResponse>
  </query>
</iq>
```

Figure 13: The Jabber-RPC This figure shows the overhead produced by using the Jabber RPC. Suppose there is a function called aFunction, which takes an integer as an argument and returns a bool **(a)**. This method wants to evoke a remote function on the backend to set the local bool **(b)**. The function and the argument have to be parsed to an XML stanza shown in **(c)** The XMPP-Framework sends the stanza via the server to the backend, which then has to map the provided function name to a local function, as well as the string representation of the argument to an integer. The function is then executed **(d)**. The result also has to be parsed to an XML stanza **(e)**. To distinguish between multiple results, unique IDs are used. The client receives the result, parses it and finally assigns the correct value to the local bool **(f)**. For illustration reasons, this process has been slightly simplified.

Sharing information and spread visualization

Another interesting feature of current social networks is the ability to share contents with other users. Some social networks such as Twitter focus on this aspect as a key feature, others, such as Facebook, provide it among other components. Every user maintains an output stream, to which he can publish information such as links, photos or raw text. Depending on the social networks as well as the settings of the user, other users can subscribe to that output stream and receive the previously published information in their inbox. Some social networks enable their users to publish media from the inbox directly to their output streams. This feature has different names: Tumblr calls it "repost", Facebook users "share" the content, and users of Twitter "Retweet". The idea remains the same: If a user chooses to share the information of another user, his name as well as the original user's name is attached to the information, which is then send to his outbox. Interesting information may cause a snowball-effect, which results in a fast distribution of information along the network. Using spatial information of the users who generate or share information in social media to visualize the spread of information is a promising idea of future work, as well as the combination with existing approaches to visualize social graphs (for example in [14])

Integration into other location based services

The introduced concept can easily be integrated in other location based services. This enables users to share points of interests in their social media and facilitates the filtering of the content generated by other users. Future work might include developing heuristics that highlight points of interests according to the interests of the user (derived from social services) as well as interests of his friends.

Data security and subjective privacy perception

With the increasing influence of social networks, data security and privacy becomes more and more important. In this work, an application is proposed that constantly sends highly sensitive data from the user to a server, which then processes them in a way unknown to the user. Nearly every tester of the application expressed concerns about his privacy, even those who where familiar with the technological details of the application. This high awareness of the security of personal data makes the development of any location-based application impossible without considering safety issues. Moreover, there seems to be a difference between subjective security perception, meaning how a user perceives the security of his data, and the existing safety precautions.

One step towards the semantic web

With the ongoing personalization of web content, the perception of the Internet as a "read/write-web" [16] shifts towards a web that automatically presents filtered and highly personalized contents. The user no longer searches actively for information, instead, his device automatically tries to guess what the user might be looking for - very much like in Tim Berners-Lee's vision about a semantic web [2]. This idea has already found its way to modern search engines: By typing a part of the keyword, the search engine tries to guess the whole search expression by comparing the input with previous queries from other users. By combining social services with current smartphones, as proposed in this work, even more information can be considered: For example, the combination of personal interests such as favorite food with the time of the day, recommendations from friends and, of course, the current position enable the device to automatically propose a nearby restaurant to the user which matches his needs. Additionally, personalization of the web facilitates the assignment of meaning to a search query: A user who searches for "drink" is more likely to look for a place where he can simply buy beverages if it's about noon, hot and he is located in another city than when it's late, his interests are parties with friends and he's located in his home town. As current mobile devices typically lack a full-featured, full-sized keyboard, deriving the right semantics from small keywords ("drink" instead of "beverages") becomes an attractive feature for users. However, adding correct semantics to keywords highly depends on the information available about the user, and as mentioned before, he might not be willing to share those due to security reasons.

Bibliography

[1] *Apple Human Interface Guidelines: The Apple Desktop Interface.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1987.

[2] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[3] C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein. The impact of semantic web technologies on job recruitment processes. *Wirtschaftsinformatik 2005*, pages 1367–1381, 2005.

[4] Gabriella Castelli, Marco Mamei, and Alberto Rosi. The whereabouts diary. In *Proceedings of the 3rd international conference on Location- and context-awareness*, LoCA'07, pages 175–192, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Robby Hanson. Weblink xmppframework. http://code.google.com/p/xmppframework/.

[6] J. Kjeldskov and J. Paay. Just-for-us: a context-aware mobile information system facilitating sociality. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, page 30. ACM, 2005.

[7] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24, New York, NY, USA, 2008. ACM.

[8] Twisted Matrix Labs. Weblink twisted. http://twistedmatrix.com/trac/.

[9] Glyph Lefkowitz. Abstract network programming for the rest of us.

[10] M. Levene. *An Introduction to Search Engines and Web Navigation.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[11] P. Mika. Social networks and the semantic web. In *IEEE/WIC/ACM International Conference on Web Intelligence, 2004. WI 2004. Proceedings*, pages 285–291, 2004.

[12] M. Niepert. Towards Collaboratively Learning and Populating Ontologies for the Social-Semantic Web by Mathias Niepert, with Jessica Rubart as coordinator. *ACM SIGWEB Newsletter*, (Spring):1–4, 2010.

[13] Peter Saint-Andre. *RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core.* Internet Engineering Task Force, October 2004.

[14] J.S. Saltz, S.R. Hiltz, and M. Turoff. Student social graphs: visualizing a student's online social network. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 596–599. ACM, 2004.

[15] P. Singla and M. Richardson. Yes, there is a correlation:-from social networks to personal behavior on the web. In *Proceeding of the 17th international conference on World Wide Web*, pages 655–664. ACM, 2008.

[16] Gottfried Vossen and Stephan Hagemann. *Unleashing Web 2.0 - From Concepts To Creativity.* Morgan Kaufman, 2007.

[17] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th international conference on World Wide Web*, page 426. ACM, 2006.

[18] D. Yamamoto, I. Takumi, and H. Matsuo. Location-based social network services employing student cards for university. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pages 21–24. ACM, 2009.