UNIVERZA V MARIBORU

universität wien

Igor Pernek

LOWERING THE COMMUNICATION BARRIER WITH THE HELP OF MOBILE SOCIAL NETWORKS

Diploma thesis

Maribor, August 2009

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJSKEGA PROGRAMA

LOWERING THE COMMUNICATION BARRIER WITH THE HELP OF MOBILE SOCIAL NETWORKS

Študent:	Igor Pernek
Študijski program:	Univerzitetni, Računalništvo in informatika
Smer:	Programska oprema
Mentor:	red. prof. dr. Peter Kokol
Somentorica:	dr. Karin Anna Hummel

Maribor, August 2009

Acknowledgements

I would like to thank the supervisors, Dr. Karin Anna Hummel and Dr. Peter Kokol, for the help and support during the diploma work.

I would also like to thank my parents for allowing me to realize my full potential.

NIŽANJE KOMUNIKACIJSKIH PREGRAD S POMOČJO MOBILNIH SOCIALNIH OMREŽIJ

Ključne besede: mobilne naprave, mobilna socialna omrežja, nižanje komunikacijskih pregrad

UDK: 004.7(043.2)

Povzetek

Porast števila socialnih omrežij nakazuje, da lahko tehnologija služi kot sredstvo za vzpodbujanje socialne interakcije. Ob množici spletnih socialnih omrežij so se v zadnjih letih začele pojavljati tudi mobilne različice, ki z vseprisotnostjo mobilnih naprav omogočajo fleksibilnejšo izrabo tehnologije za socializacijska opravila. V okviru te raziskave smo se posvetili nižanju komunikacijskih pregrad v okoljih, kjer je fizičen stik med subjekti še zmeraj prisoten. Razvili smo SocioNet, prototip mobilnega socialnega omrežja, ki omogoča iskanje optimalnih komunikacijskih partnerjev v urbanih središčih.

V skladu z aktualnimi trendi mobilnih aplikacij, ki že v letu 2009 nakazujejo izenačenje mobilnih dostopov do interneta s stacionarnimi, do leta 2020 pa napovedujejo dominanco mobilnih dostopov do medmrežja, smo SocioNet razvili kot platformo, ki omogoča uporabo različnih načinov brezžične in IP komunikacije. Preučili smo obstoječa mobilna in stacionarna socialna omrežja, njihove zmogljivosti, monetizacijski potencial, arhitekture, uporabniške vmesnike in skrb za varnost ter zasebnost.

Ob pregledu obstoječih socialnih omrežij smo ugotovili, da so različna socialna omrežja skozi zgodovino ponujala nabor podobnih zmogljivosti. Tako le-te niso predstavljale ključnega dejavnika uspeha, temveč zgolj dopolnitev k izredno pomembni časovni umestitvi. Friendster, ki velja za eno iz med največjih razočaranj na področju socialnih omrežij, je namreč ponujal zelo podoben nabor zmogljivosti kot dandanes

izredno aktualen Facebook. Ob omrežjih, ki so uspela s pridom izrabiti ugodno časovno umestitev, je preboj uspel še nekaterim domensko usmerjenim omrežjem, med drugimi omrežju LinkedIn, ki omogoča vzdrževanje poslovnih kontaktov.

Pomembna aspekta socialnih omrežij sta prav gotovo zasebnost in zaupanje v omrežje samo. Slednje je posebej pomembno v mobilnih socialnih omrežjih, kjer vseprisotnost mobilnih naprav omogoča pridobivanje različnih osebnih podatkov, ki lahko ob nepravilni uporabi predstavljajo vir najrazličnejših zlorab in kraj identitete. Analiza zasebnosti mobilnih socialnih omrežij je pokazala, da obstoječa mobilna omrežja slabo skrbijo za uporabnikov nadzor nad lastnimi osebnimi podatki, prav tako pa ponujajo skope povratne informacije o uporabi in obdelavi uporabnikovih osebnih podatkov.

Skrb za zasebnost podatkov v mobilnih socialnih omrežjih je tesno povezana z arhitekturo omrežja. Dandanes se v mobilnih omrežjih najpogosteje uporabljata:

- ✓ *P2P arhitektura in*
- ✓ *arhitektura oserednjega strežnika.*

Tako ena kot druga ponujata različne prednosti in slabosti. Medtem ko uporaba arhitekture osrednjega strežnika drastično niža stopnjo zaupanja v omrežje na eni strani (množica osebnih podatkov je namreč centralizirana na osrednjem strežniku) ter izboljšuje prepustnost in hitrost omrežja na drugi, so efekti P2P arhitekture skorajda diametralno nasprotni. Ob uporabi P2P arhitekture so osebni podatki porazdeljeni po odjemalcih uporabnikov, kar zmanjšuje tveganje njihove odtujitve in zlorabe, a dodatno obremenjuje omrežje, saj so za obdelavo in uporabo podatkov potrebne neprestane migracije podatkov po omrežju.

Arhitekturo omrežja SocioNet smo zasnovali z mislijo o prednostih in slabostih omenjenih arhitektur. Odločili smo se za uporabo mešanice P2P arhitekture in arhitekture osrednjega strežnika. Osebni podatki uporabnikov se tako hranijo bodisi na uporabnikovi mobilni napravi, bodisi na osrednjem strežniku. Mesto hranjenja osebnih podatkov lahko, glede na kritičnost in željo po stopnji zasebnosti, določa uporabnik sam. Dodatno smo omrežje SocioNet obogatili še s sistemom pravil, ki omogočajo *dinamično prestavljanje osebnih podatkov glede na pomembnost dobljenih rezultatov in uporabnikove preference zasebnosti.*

Pred zasnovo odjemalčevega uporabniškega vmesnika smo preučili smernice zasnove mobilnih uporabniških vmesnikov. V skladu s smernicami smo izvedli interno analizo željenih zmogljivosti ter preučili različne zaslonske maske aplikacije. Odločili smo se, da delo z odjemalcem integriramo v namizje mobilne naprave ter tako znižamo krivuljo učenja potrebno za učinkovito uporabo aplikacije. Z željo po izdelavi intuitivnega uporabniškega vmesnika smo kot vir za določanje relacij med uporabniki izbrali kontekstualne podatke uporabnikov, s čimer smo izločili potrebo po inicializacijski izdelavi lastnega profila uporabnika. Kot poglavitne sestavine profila uporabnika smo uporabili:

- ✓ *kontaktni imenik,*
- ✓ *seznam opravil,*
- ✓ *lokacijo uporabnika in*
- ✓ *organizator uporabnika.*

Seveda smo uporabniku dopustili možnost izbire entitet, ki jih želi uporabljati za definicijo relacij z ostalimi uporabniki, ter mu omogočili popoln nadzor nad lokacijo hrambe podatkov. Stopnjo zasebnosti uporabnikov smo poskusili dodatno dvigniti z obfuskacijo podatkov, ki se hranijo na osrednjem strežniku. Tako smo podatke pred centraliziranim hranjenjem obdelali s pomočjo enosmerne funkcije in jih pretvorili v človeku neberljivo, a vsebinsko še zmeraj dovolj polno obliko.

Za definicijo relacij med uporabniki smo znotraj omrežja SocioNet uvedli naslednje vloge:

- ✓ *prijatelj,*
- ✓ *prijateljev prijatelj,*
- ✓ *interesno ujemanje in*

- ✓ *tujec.*

Kot poglavitni kriterij za določanje relacij med uporabniki omrežja SocioNet smo uporabili trenutno oddaljenost uporabnikov v evklidskem metričnem sistemu, saj smo želeli maksimizirati možnost kasnejšega fizičnega kontakta udeleženih subjektov. Dodatna pravila smo definirali na osnovni različnih virov podatkov. Tako je za definicijo prijateljstva potrebna udeleženost kontaktnih naslovov subjektov v kontaktnem imeniku uporabnikov, za definicijo interesnega ujemanja po sovpadanje opravil v mobilnih organizatorjih uporabnikov. Z vlogo tujec smo označili uporabnike, ki se nahajajo v neposredni bližini uporabnika, vendar si z njim ne delijo kontaktnih naslovov, niti se ne ujemajo v interesnih aktivnostih. Uporabnikom smo tako s pomočjo predefiniranih vlog omogočili izbiro kontaktne osebe, ki se v danem trenutku najbolje prilega interesom, željam in razpoloženju posameznika.

Prototip mobilnega socialnega omrežja SocioNet sestoji iz dveh jedernih delov:

- ✓ *odjemalec in*
- ✓ *strežnik.*

Prototip odjemalca je implementiran v ogrodju .NET Compact Framework 3.5, medtem ko je za implementacijo strežnika uporabljena namizna različica ogrodja .NET 3.5. Z željo po modularnosti smo strežnik implementirali kot množico XML spletnih storitev, ki omogoča enostavno razširanje domene komunikacije in posodabljanje storitev. Dvig razvojne produktivnosti smo zagotovili z uporabo dodatnih .NET CF knjižnic (OpenNETCF), ki so omogočale enostavnejšo integracijo odjemalca v namizje mobilnih naprav in učinkovitejšo več-nitno delovanje.

Na koncu smo opravili tehnično in uporabniško ocenitev prototipa mobilnega socialnega omrežja SocioNet. V sklopu analitične analize smo ugotovili, da uporaba P2P arhitekture znatno poveča število prenešenih podatkov znotraj omrežja v primeru uporabe večje količine uporabniških podatkov. Prav tako smo ugotovili, da predstavlja deljen pristop hranjenja podatkov odličen kompromis med zagotavljanjem visoke stopnje zaupanja v omrežje in visoko propustnostjo omrežja.

V sklopu uporabniške analize, ki smo jo izvedli s pomočjo spletnega vprašalnika, smo ugotovili, da več kot 70 % sodelujočih meni, da SocioNet odpira nove možnosti komunikacije. Medtem ko bi 60 % sodelujočih poseglo po uporabi omrežja SocioNet za vzpostavljanje kontaktnih povezav s prijatelji ali osebami, s katerimi se interesno ujemajo, bi jih 45 % stopilo v kontakt s prijateljevimi prijatelji in le 20 % s tujci. Kar 93 % anketirancev z nižjo stopnjo informacijske pismenosti meni, da hranjenje podatkov lokalno dviguje stopnjo zaupanja v omrežje in pozitivno vpliva na zasebnost, medtem ko je odstotek anketirancev z visoko stopnjo računalniške pismenosti nižji (72 %). Kljub dodatnim ukrepom za varovanje zasebnosti podatkov pa le 20 % uporabnikov z nižjo in 56 % uporabnikov z višjo stopnjo računalniške pismenosti meni, da imajo dovoljšen nadzor nad lastnimi podatki.

LOWERING THE COMMUNICATION BARRIER WITH THE HELP OF MOBILE SOCIAL NETWORKS

Key words: mobile devices, mobile social networks, lowering communication barriers

UDK: 004.7(043.2)

Abstract

As can be seen by the booming social networking platforms in the Internet, such as Facebook and Twitter, technology can foster social interaction. In this paper, we aim at lowering the communication barrier in scenarios where the physical presence is given and should be exploited. We therefore introduce SocioNet – a context-aware and rule-based system that provides the best matches for communication in urban areas. The system aims at finding matches between persons, e.g., in proximity, and therefore establishes contacts. The design principles of SocioNet are: (i) use of existing personal information data to find matches between persons, (ii) preserve privacy up to a high degree by not storing personal data on the SocioNet server. The system architecture is a hybrid central server and P2P architecture supporting matchmaking. We demonstrate the feasibility of the concept by a prototypical implementation. Finally, we have evaluated the approach with respect to user satisfaction by carrying out a questionnaire. The results show, that even with increased privacy preservation, privacy is still an issue not to use such matchmaking systems when physical presence is involved.

INDEX OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

ABRREVIATIONS

AAL – Ambient Assisted Living

GPS – Global Positioning System

GSM – Global System for Mobile communication

IP – Internet Protocol

IR – Infrared Radiation

J2ME – Java 2 Micro Edition

LBS – Location-Based Service

MSNS – Mobile Social Network System

P2P – Peer-to-Peer

RFID – Radio-Frequency IDentification

SMS – Short Messaging Service

SNS – Social Network Site

WiMax – Worldwide Interoperability for Microwave Access

WMN – Wireless Mesh Network

1 INTRODUCTION

With the tempo of living getting faster in modern societies people have less time for sometimes much more valuable socializing tasks. While large amount of free time in the late afternoon, which was usually used for socializing with friends and family, is divided into smaller fragments during the whole day, this gives us an opportunity of meeting new people and widening our social network but also makes contacts more superficial and short-timed.

People could be intuitively divided into two groups:

- ✓ *people having time to communicate, and*
- ✓ *people being in a hurry and not having time to communicate.*

It is hard to estimate, without some direct feedback from the person, if a person is available for communication or not. Of course one possibility could be a direct question or establishment of small talk conversation, which is an explicit method for getting the answer to this question. While there is certainly a lot of people being communicative and having no problems with such an approach, there is even a larger group of people being shy and rather not willing to communicate than trying to establish communication and getting rejected. Although starting the communication is usually the hardest step, it is most certainly not the last one. The question, how to find the most appropriate partner for communication or help, is not easy to answer, as only some assumptions can be based on visual judgment.

With the ubiquity of mobile devices and the development of mobile networks the process of finding a suitable communication partner is much easier to perform. As a solution for lowering the communication barrier a Mobile social network system (MSNS) SocioNet is being proposed, which could be used by various age groups and profiles of end users. With the main goals of ubiquity, unobtrusiveness, network efficiency and privacy consideration, the solution designed is a hybrid between P2P and client-server solutions, an application seamlessly integrated into a mobile device environment, allowing users to easily accept and request communication partners. In contrast to a plethora of existing MSNSs where users have to fill-in multiple profile pages, we have decided to extract attributes to match from the user’s mobile context. Our goal was not to foster virtual communication, but simply supporting human to human communication to be as easy as possible, by lowering the barrier of establishing communication with a suitable partner.

In Section 2 the areas of interest for application are described, starting with existing social networks descriptions, privacy implications, architectures and context-awareness in mobile social networks. Later on a look at location-based services and mobile user interface design rules and concepts is taken. In the following Wireless mesh networks (WMNs) as a possible network architecture for deploying such kind of services in urban areas are being examined. Section is concluded with works on ambient assisted living as a possible area where application could be of a great use.

In Section 3 the technology used in prototype implementation of a MSNS for lowering the communication barrier is described, concepts of profile obfuscation and cumulative location tracking are explained and the idea of a dynamic rule-based mechanism for storing profiles is presented.

Section 4 begins with an explanation of the overall architecture of the prototype MSNS implementation. Later on the bottom layer network architecture, application use cases, sequence diagrams, and user interface screen masks are described.

In Section 5 some of the client and server implementation aspects are described and a two-way communication protocol to simulate server-client communication without the use of XML WebServices on the client side is proposed.

In Section 6 measurement based performance evaluations as well as the results of a questionnaire about usability and trust in the SocioNet application is presented. Section 7 consists of an overall conclusion.

2 RELATED WORK

In the following subsections we are presenting existing work in the areas related to the SocioNet application. We are examining current trends in mobile applications, history of social networks, mobile social networks, location based services, mobile interface design, wireless mesh networks, and ambient assistant living. We are also taking a deeper look in mobile social networks privacy, architectures and context-awareness.

2.1 Trends in mobile applications

With over 4 billion mobile subscribers [1] at the last count, the worldwide phone penetration is high and still climbing. Although voice was the main communication channel in the past, it seems that the mobile usage is moving to data faster and faster. Currently, 15.6% [2] of mobile subscribers are using mobile access to the Internet and the number will rise as users consolidate pricey communication services into cost-effective all-in-one mobile devices. According to the last forecasts half of the new connections to the Internet in 2009 will come from phone and the mobile device will become the primary Internet connection tool for most of the people in 2020 [3].

One of the main reasons for the slow growth of non-voice mobile services is costly data communications. Mobile Internet users were too often treated unfair as they had to pay for every byte transferred. It has become hard to imagine to pay your at-home Internet provider for every page load and to need to know the sizes of the websites visited before visiting them. A lot of mobile providers came to the conclusion that their initial solution was simply not practical and they started to offer flat-rate and unlimited data transfer Web access services to define flat as a new path in mobile data communications.

Another reason for low mobile data services usage statistics were too unfamiliar mobile applications. It is obvious that today’s killer applications are Amazons and Facebooks of the world and not just some software that can be downloaded to your machine. The new generation mobile phone usage statistics are quite self explanatory, as the iPhone and Android users, who represent only 13% of the high-end mobile market, create nearly 50% of all Google’s smart phone traffic worldwide [4]. A simple fact is that mobile users have wanted fast, intuitive and full web access all along and that the trends will continue to evolve to more efficient and sophisticated applications.

It is also worth mentioning that past mobile phones and operating systems designs, which were too difficult to use for anything beyond voice calls. Inexperienced mobile users were often unable to find their path through unintuitive device menus and usually needed more than 20+ mind-numbing clicks just to access or locate portal sites [4]. With the new smart phones generation the trend is moving to preinstalled or easily installable applications, only a tap away from the simple everyday use. The fact that stresses the importance of simple and intuitive user interface design is the statistics of Google Earth activations. In February 2009, on the day of its iPhone launch, Google Earth saw more activations than any other day in the history of the product [4]. The main reason for such rapid activation growth was that the application was available through iPhone’s App Store and the on-screen layout made it easy to find and use. However, the proliferation of app stores is just a small step towards the mobile applications popularity growth, as the ease of use and on-device navigability is still critical preconditions for mainstream usage.

The next generation mobile networks will evolve with the growth of mobile Internet user database. According to Siemens NGMN vision the future mobile network architecture will be able to deliver services at fixed line quality with the cost of IP technologies. Mobile networks ownership cost reduction will lead to simplified architectures, which will be able to serve as a base for all types of different communications services. It is expected that future mobile networks will converge to full mobility, security and quality of service support and will be following the main goals of providing communications with improved throughput, latency and cost per bit [5]. Beyond existing mobile telecommunication networks, there are already other networks important for mobile data and service provisioning. While Wireless Mesh Networks (WMNs) on one side are often used in the context of future urban wireless network architecture, WiMax one the other side is still the leading wireless technology for rural areas.

With mobile networks evolving there is also a parallel evolution of mobile devices. With this evolution they are able to provide us with more detailed and useful information due to increase in number and quality of advanced functionalities. The new age phones are rich with context information from advanced calendars, task lists and are even location aware and able to track user’s geographical position, which is a new type of user context information. Although a lot of current phones are already context-aware it is to expect rapid growth of innovative context interfaces in the future, like motion sensors, temperature sensors, etc.

Today, the main non-voice mobile phone users are still the young generations, but it has to be stressed they are not even nearly the only group. It was found out that more than 50% of adult Americans have used a cell phone or PDA to do at least one of mobile non-voice activities, such as texting, emailing, taking a photo, recording a video or even looking for maps and directions. As the nature of mobile access is changing from slow and stationary to fast and mobile there has been a transformation in how people value their media tools. According to a US survey, when asked how hard it would be to give up a specific technology, the respondents were most likely to say the cell phone would be most difficult to do without [6].

It is obvious that the world is moving into a new era of mobile computing, one that could be described with a general rule: lighter, faster, and smarter. The new era promises greater variety in applications, highly improved usability and faster broadband networks [7].

2.2 Social networks overview

The rise of social networks started with the evolution of the Internet into a robust and widely available technological platform. In the early beginnings mailing lists, web forums and online groups were the most used social networking tools. The main reason for the use of social networks at that time was easier online discussing. The focus moved from discussion through wikis for collaboration and blogs for sharing to rich, embedded services integrating the virtual and physical world. The proposed tool helping people to stay in touch with their friends from non-virtual environment is an example of such an enhanced context-aware service.

Although discussion, collaboration, and blogging are used for online digital interaction, usually a social network is defined [8] as a service that allows individuals to:

- ✓ *construct a public or semi-public profile within a bounded system,*

✓ *articulate a list of other users with whom they share a connection, and*

✓ *view and traverse their list of connections and those made by others within the system.*

The timeline of launch dates of many major SNSs is presented in Figure 1. According to the definition above, the first recognizable SNS was launched in 1997.

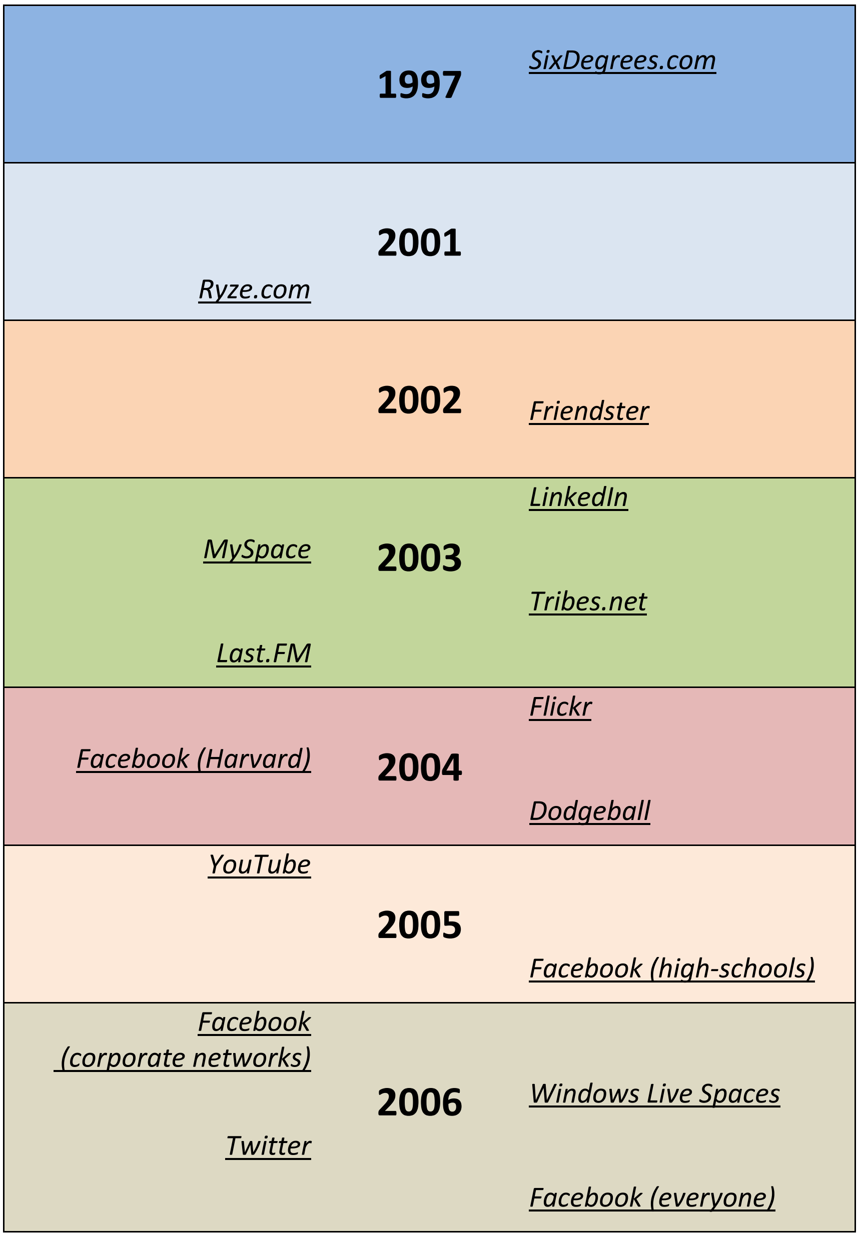


Figure 1 – Timeline of launch dates of many major SNSs [8]

SixDegrees.com[1] was the first Social Network Site (SNS) which served as a tool to help people to connect with others. Although it attracted millions of users, it failed to become a sustainable business and was closed in the year 2000. According to the founders it was shut down, because it was simply ahead of its time. As people mostly did not extend networks of their friends who were online, there was not a lot to do after accepting a friend and most users were not interested in meeting random strangers.

The next wave of SNSs was oriented towards business networks and began with the launch of Ryze.com[2] in 2001. Ryze.com was followed by Friendster[3], LinkedIn[4], and Tribe.net[5]. People behind those networks were tightly personally and professionally connected and believed that they could support each other without competing. While LinkedIn become one of the most successful business networks, Friendster on the opposite was declared as one of the biggest disappointments in Internet history. Friendster was another unfortunate example of being too early in a developing market. Despite having all of the features that are expected from social networks today, it was left behind by its later competitors, which were all highly influenced by it.

After the year 2003, many new SNSs were launched, most of them taking the form of profile-centric sites, targeting their own demographics niche. The biggest success was achieved by Facebook, which began in early 2004 as a Harvard-only SNS. While later expanding to more and more universities and going public in 2005, the open signup did not mean that new users could access users in closed networks. Gaining access to closed networks still required the appropriate email address or administrator approval. Another feature that helped Facebook to succeed was the ability for outside developers to build custom applications, which could be later on used by users for the personalization of their profiles [8].

[1] http://en.wikipedia.org/wiki/SixDegrees.com, visited: April 2009

[2] http://www.ryze.com/, visited: April 2009

[3] http://www.friendster.com/, visited: April 2009

[4] http://www.linkedin.com/, visited: April 2009

[5] http://www.tribe.net/, visited: April 2009

While first of the social networks were mostly web based, today they could be divided into three main groups:

- ✓ *pureplay web-based social networks,*
- ✓ *hybrid web-based and mobile-based social networks, and*
- ✓ *mobile-based social networks [9].*

In the chapters to follow the focus is on the last social networks group – mobile-based social networks.

2.3 Mobile social networks

As pointed out in Section 2.1, mobile phone is becoming the primary communication tool for a wide group of people. Considering this fact, it is not uncommon that social networks are migrating to mobile devices and forming MSNSs. MSNSs allow groups of friends to be accessed from a mobile phone and are extending the concept of desktop pureplay web-based social networks to a mobile phone. The user is being able to be in contact with a social network of his friends while still being mobile, which is the main added value. Being in contact can take different forms, including:

- ✓ *media upload,*
- ✓ *text messaging,*
- ✓ *status updates,*
- ✓ *locating friends, and*
- ✓ *sharing items [10].*

Different MSNSs offer different interaction methods to achieve the same goal. The following is clearly visible taking a look at two of the most popular microblogging networks, Twitter[1] and Jaiku[1]. While Twitter users were initially using SMS

[1] http://twitter.com/, visited: April 2009

communication channel, instant messaging or Twitter's web site to send their status updates to the server, Jaiku users were able to update their status using an application preinstalled on their mobile device, which is usually cheaper than sending SMS and for sure easier to use, but on the other hand only compatible with a limited set of mobile devices.

Considering the relationship of the social network provider with the wireless phone carrier, mobile social networks are divided into two basic groups. While on-deck mobile social networks, which rely on partnership between the social network provider and the carrier, are deeply integrated in mobile phones, off-deck social networks do not have such carrier relationship and have to attract users by means of other methods. A lot of off-deck social networks have evolved from web-based SNSs and are serving as a ubiquitous complement to stationary social interaction. The most well known examples are MySpace[2] and Facebook[3] going mobile after being present in the stationary community for quite some time.

Despite numerous MSNSs emerging, MySpace and Facebook are leading also in the mobile segment. According to a release from ABI Research 70 % of mobile social networking users have visited MySpace, 67 % Facebook and only 15 % have visited other MSNSs, including those specialized for mobile devices [11]. Although those numbers don't look promising at first glance for the other MSNSs, the breakthrough is still possible addressing various niche areas and focusing on the use of MSNSs as an everyday assistance tool. Instead of following the goal of enabling mobile access to social networks, it should be all about making social connections with your mobile phone. Of course important aspects like consumer demands and needs have to be satisfied and nevertheless appropriate monetization potential has to be found.

With respect to mainstream consumer expectations, including freedom of choice, care about quality and ease of use, main consumer needs for mobile social networking could be derived. One of the most important aspects is convenience, as the same look and

[1] http://www.jaiku.com/, visited: April 2009

[2] http://www.myspace.com/, visited: April 2009

[3] http://www.facebook.com/, visited: April 2009

feel on various devices leads to a higher acceptance and usage of a MSNS. A high independence level allows users access to the social networking services on any device, at any location, and most importantly, at any time. According to a Social network study [12] high control and effective management is also important, as consumers have a strong need for direct integration of Social Networks with their phone's address book. 87 % of a survey participants agreed that the combination of Social Network services in combination with the address book would be appealing to mainstream consumers. Considering monetization of MSNSs different approaches could be acceptable according to the subcategory of the consumers.

Different monetization approaches of MSNSs could be acceptable according to the subcategory of the consumers addressed:

- ✓ *using fee,*
- ✓ *carrier revenue share,*
- ✓ *advertising.*

With the rise of MSNSs new questions are opening all the time, mainly considering privacy, scalability, context-awareness, and networks architecture design. Current work, related to the aspects mentioned, will be covered in the subsections to follow.

2.3.1 Privacy

While security in MSNSs can be addressed by using the well-known network traffic encryption methods, standard authorization and authentication patterns, privacy still stays a very important open aspect. Because MSNSs relay on a large enough user databases to become usable, it is almost impossible to avoid the chicken or the egg dilemma[1] without detailed network privacy implications consideration. Two important privacy aspects are:

- ✓ *control, and*
- ✓ *trust.*

[1] In social networking The chicken or the egg causality dilemma implies that new users will join a social network only when a database of existing users will be large enough.

Privacy control allows users to define which information they would like to disclose and to whom. Public and private data distinction can be fine- or coarse-grained, depending on the smallest unit users are able to set as public or private. Facebook, for instance, allows members to set access control for virtually every profile data entry, while members of LinkedIn can only differentiate between different profile views[1]. Usually access control could be applied to individuals as well as to a group of people.

While it is relatively easy to set up a good privacy control, establishing a trusted network is much harder. There is a reasonable threat attached to storing a lot of personal information in the hands of profitable corporations, as the data could be misused to produce a persons' decision profile. Furthermore, data control presents another issue as MSMSs users can not be absolutely positive their profile cannot be retained after deletion or even sold out to the 3rd parties. It is also worth mentioning privacy in social networks in the terms of law is still quite ill-defined, as the data collected from SNSs is descriptive enough to be used for various medical and scientific researches. Gathering the same data from traditional sources is normally strictly inspected by appropriate review boards to ensure adolescent's parents have informed consent about the research, while in the case of SNSs it is not clear if the same rules apply. [13]

A privacy design analyze of a group of MSNSs [14] was carried out to reveal the privacy protection actions performed in some of the existing mobile social applications. 31 mobile social networking applications from Apple App Store[2] were analyzed with the help of Belloti and Sellen's feedback and control framework [15], considering four information flow regarding components:

- ✓ *capture,*
- ✓ *construction,*
- ✓ *accessibility, and*
- ✓ *purpose.*

[1] LinkedIn allows users to switch between public and insider view of their profile.

[2] http://www.apple.com/iphone/apps-for-iphone/, visited: April 2009

Capture defines what kind of information is being collected; construction is struggling with the question what happens to user's information once it is collected; accessibility lets us know who can access the collected information; and purposes defines how the information is used by the other people. Basically, the framework allows users to inspect the feedback the application provides, and at the same time specify the level of control along these four aspects. Analyze performed is focusing on user's location as one of the most important features of MSNSs.

The majority of applications surveyed use a popup dialog to acquire permission for the use of location, while some of them automatically acquire location on start and notify the user with a short message on the status bar, giving the user sufficient feedback about location using, but no control over it. Some applications acquire the location information continuously without the user being able to control how frequently, when, and where the location should be acquired. On the other hand some applications require users to take explicit action each time the location is to be used or even allow the user to manually provide the location. As it appears mobile social networking applications have various feedback and control mechanisms over location capturing, though they are usually quite simple and do mostly not allow users to change their location granularity. It is suggested that better feedback on automatic location collection and finer granularity should be considered as an improvement to the existing approaches. Besides location, there is usually no feedback on, which other profile data, like e.g. identity, phone number, calendar list and call history, is being used by the MSNS. As it is true users location is one of the most revealing attributes, it is not to forget very accurate personal profile models could be created with the access to sufficient other personal information. This is particularly worrisome as the applications inspected were gathered from Apples App Store and could be developed by anonymous and independent developers.

Considering third-party application development sufficient feedback and control of construction is very important for MSNSs. What happens to the information after it is gathered? Will it be sent to a central server? Will the server communication be performed using encrypted connections? Will the information be permanently stored or even shared with third-parties? According to the analysis performed, for most applications, there is almost no feedback and control over the personal information, once it leaves the users device. Information flow becomes even more complicated as MSNSs are usually meshed up together. A lot of MSNSs are using Google Maps[1] for visualizing location of users. It is not clear enough, what happens, when the location is removed from the primary MSNS – will it be also deleted from Google Maps, is it even stored in Google Maps, etc. All in all, construction feedback and control mechanisms in MSNSs are mainly week and are becoming even worse with the number of mash ups increasing. Better feedback and control should be performed to improve the construction aspect of MSNSs.

The most common accessibility model in MSNSs is the model where the location information is shared only with friends. This model is easily understandable by the users, though no explicit feedback is provided by most of the applications. But like in case of construction, more and more mesh ups are making harder to provide sufficient feedback and control over the accessibility aspects as inconsistent policies of linked MSNS make accessibility harder to track and control.

At the end the aspect of purpose within MSNSs is struggling with the same problems and leaks as the aspects of accessibility and construction. While it is not guaranteed that it is possible to infer purpose from construction and access patterns, it is not hard to define possible information misuse scenarios. Users may unwillingly reveal their anonymous profiles through a mistaken action[2] and destroy their virtual profile, which was purposed to stay anonymous.

Insufficient privacy feedback and control decreases the users trust in MSNSs. In domain specific MSNSs reasonable data obfuscation could be used to make the personal data misuse harder, but still preserve enough information to follow the MSNS purposes. Of course the level of privacy is strongly impacted by the underlying network architecture of a MNSN. The following two architectures will be closely examined:

- ✓ *P2P architecture, and*
- ✓ *central server architecture.*

[1] http://maps.google.com/, visited: April 2009

[2] For example by adding a non-personal photo to their anonymous profile, which could be recognized by their peers, because it was shown to them by the user before.

2.3.2 Peer-to-Peer architecture

According to the Utah Education Network[1] a Peer-to-Peer (P2P) network is a communication network that enables users to connect their computers and share resources directly with other users without having to go through a centralized server. In the mobile social networking context the definition could be translated as a network ability to exist and fully function without a centralized device with special tasks to perform. Usually all the clients are sharing the same responsibility and are each accountable for their execution. A simple mobile P2P network architecture is presented in the Figure 2.

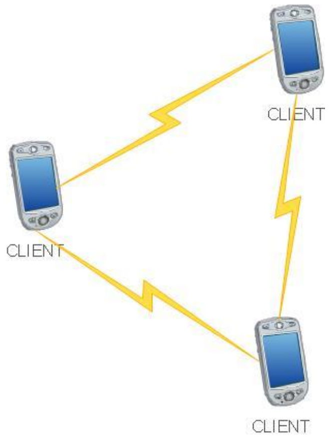


Figure 2 – Simple mobile P2P architecture

In Section 2.3 we were speaking about different objectives of MSNSs. The one we will be focusing on is locating friends and mainly consists of two important aspects:

- ✓ *profile storage, and*
- ✓ *matching*[2].

In a P2P MSNS, with the main purpose of locating friends, there is no centralized storage where all the profiles are stored, as the profile of each user is stored on each own mobile device. While such a profile storing concept cheapens the costs of network's infrastructure and increases users' level of trust in a MSNS, as they feel in-control of their profile, it raises some problems worth mentioning. When data is stored on a mobile device,

[1] http://www.uen.org/core/edtech/glossary.shtml, visited: April 2009

[2] To search for persons with similar characteristics.

matching has to be done on a mobile device, as there is no other authority to perform this task. Usually mobile devices are much more limited with processing power as specialized servers and are not able to scale with the rapid growth of users and profile data. Also, when matching two profiles, one profile has to leave its hosting device to be compared to another profile. These results in asymmetric privacy disclosure, as the device matching both profiles could be spoofed to permanently collect profile data without the knowledge of the other party. As another drawback such kind of data transmission usually has to be done for each matching request, creating a lot of unnecessary network traffic.

All in all, the use of the P2P concept in MSNSs, with the primary goal of looking for friends, has some strong advantages and disadvantages (Table 1).

Advantages (+)	Disadvantages (-)
✓ ***Cheaper infrastructure***	✓ ***Processing limitations***
✓ ***Higher level of trust***	✓ ***Asymmetric privacy disclosure***
	✓ ***High network traffic***

Table 1 – Advantages and disadvanteages of P2P in MSNSs

2.3.3 Central server architecture

While P2P MSNS architecture consists only of equivalent peer mobile devices, central server approach contains and additional central authority. In context of MSNSs with the primary purpose of searching for friends, central authority can be responsible for various tasks, like performing the matching routine, profile storage and even some additional processor consuming tasks, since the price/processing power ratio is much lower as in the case of mobile devices. Simple central server architecture is presented in Figure 3.

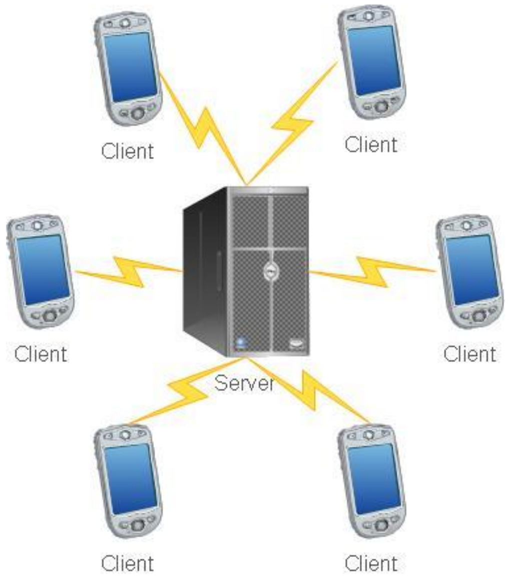


Figure 3 – Simple mobile central architecture

As mentioned before, the matching routine can be a lot simpler and more effective with the use of the central server approach, as it is performed on a stronger device with much easily affordable processing capacity. With the growth of the number of users and their personal profiles expansion the solution is a lot easier to scale on a hardware level, as the central sever authority could be expanded with some additional devices, connected in to a cluster. With the central server performing the matching routine, asynchronous profile disclosure is eliminated, because client devices do not need to communicate with each other. If the data is also saved or cached on the server, network traffic could be drastically reduced, as the network profile transmission is reduced only to profile synchronization and notifications. But on the other hand, a central server approach requires a higher hardware investment and has usually a lower level of trust, mainly because users are aware that their profiles will be somewhere centrally accumulated and could be easily misused or even passed to third-parties.

Advantages (+)	Disadvantages (-)
✓ ***Less processing limitations***	✓ ***Expensive infrastructure***
✓ ***Low network traffic***	✓ ***Lower level of trust***
✓ ***Symmetric privacy disclosure***	

Table 2 - Advantages and disadvanteages of central server in MSNSs

It is obvious that neither the pure P2P nor the central server architecture approach is superior. In MSNSs with the primary purpose of locating friends, a hybrid approach could be used, with the main aims of:

- ✓ *fast and scalable matching,*
- ✓ *high level of trust, and*
- ✓ *low network traffic.*

2.3.4 Context-awareness

In terms of MSNSs, context-awareness is an area of mobile computing, which deals with the adaptation of mobile devices to the user's context. The most recognizable definition of the term context defines context as

... any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves [16].

With respect to the widely used context definition, a context-aware system is a system that is able to make deductions and decisions about the current situation from the context information available.

Most of the MSNSs, mentioned in the Section 2.3, are based on the predefined users' profiles and not aware of the current state of the user. As stated in section 2.1, trends in mobile computing are converging to a faster, simpler and smarter approach in which context-awareness will be playing one of the main roles, as more intelligent and quicker actions will be possible.

The following context could be a great use in MSNSs with the primary purposes of looking for friends:

- ✓ *calendar,*
- ✓ *tasks,*

- ✓ *communication channels,*
- ✓ *location,*
- ✓ *etc.*

A mobile calendar could be a great source of information for matching people with the same interests. Usually people, being present at the same meetings or having the same daily routines, are a lot better matches and have a lot more conversation material as random pickups. A lot of people are already strongly attached to the use of a digital calendar for everyday tasks and with the ability of different services meshed up together the number is expected to additionally increase.

User’s tasks could provide us with information about the short term tasks the user has to perform. Two persons, with a task to pick up their children from school, could speak about the burdens of adolescence and are most probably a better conversation match than two random picked up persons.

People use mobile devices for different types of communication, opening various communications channels that can be observed. Some of the most ubiquitous mobile communication channels are:

- ✓ *voice calls,*
- ✓ *SMS, and*
- ✓ *e-mail.*

Communication channels observance could give us information about the common communication partner of a person, if a person is usually communicating with the same people, if the person is more of a voice call, SMS or e-mail user; or even if there is a match in the network, with whom the person was communicating recently.

As location is one of the more important and widely used context information, the next chapter will be devoted to location based services. All of the context data, mentioned in this chapter, is easily accessible from most of the mobile platforms. On J2ME supported mobile devices PIM API[1] could be used, while on devices, supporting .NET Compact Framework development, PocketOutlook[2] namespace contains managed classes with calendar, tasks and communication channels retrieval functionality.

2.4 Location-based services

LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device. [17] As much as it looks like LBSs started to emerge after the invention of mobile phones, the concept was also widely used before. Coffee shops located on train or metro stations are good example of a service with a meaningful location, as a lot of people are starting their day with a cup of coffee. With the innovation of mobile devices, LBS only started using more dynamic locating methods and were able to be used in more sophisticated scenarios. In order to provide a successful LBS the following points should be followed:

- ✓ *acquired coordinates should be as accurate as requested by the service,*
- ✓ *the price of establishment and service use should be as low as possible, and*
- ✓ *the impact of the service on its surroundings should be minimal.*

In order to acquire user’s location, several popular methods could be used:

- ✓ *GPS trilateration,*
- ✓ *GSM multilateration,*
- ✓ *sensor positioning.*

Although GPS cannot be used indoors and is quite expensive for the end-user, because the user has to invest in a GPS-equipped handset; it produces good results outside, using the earth-surrounding satellites. GSM does not require any additional equipment and can

[1] Specified in JSR 75 (http://java.sun.com/javame/technology/msa/jsr75.jsp, visited: April 2009)

[2] Microsoft.WindowsMobile.PocketOutlook (http://msdn.microsoft.com/en-us/library/microsoft.windowsmobile.pocketoutlook.aspx; visited: April 2009)

be used for positioning indoors and outdoors. However acquired coordinates accuracy strongly depends on the number of GSM base-stations in range. Sensor positioning could be implemented with the use of various technologies like IR, Bluetooth or RFID. Subjects being located are usually wearing badges that transmit signals providing information about their location to a centralized location service through a network of sensors [18]. The accuracy and price of such system depends on the number of sensors used for positioning.

LBSs could be classified into different classes (Table 3).

Class	Description
Location based information system	Mobile user requests information about current location.
Points of interest	Mobile user searches for local objects or entities.
Search for other users	Mobile user searches for other mobile users.
Tracking services	User tracks the position of another mobile user or object.
Assisting services	Service center receives position of a user who needs assistance.
News and announcement services	Mobile user receives advertisements.
Trigger services	Mobile user is notified when entering a certain area.
Location based fees	Fees are calculated depending on the position of the user.

Table 3 – Classification of LBS [19]

In the terms of looking for friends, the most important is the *search for other users* class. Location information could be used to identify users who are situated nearby, as they could represent a better match as someone too far away. Location information could also be used in terms of *tracking service*, as users, visiting the same places often, could be a better match.

When speaking about LBS, privacy has to be seriously taken into account. While on one side strong feedback and control mechanisms should be deployed, on the other side location data could also be stored cumulative. If the user’s location history is not needed in the raw form, there is no real reason to store it in such way. In terms of a MSNS for lowering the communication barrier, the geographical surface could be divided into a grid like area and only the user information about the relative time spent in a specific cell could be stored. Losing the time information makes it impossible to track the user and still possible to perform location-aware matching.

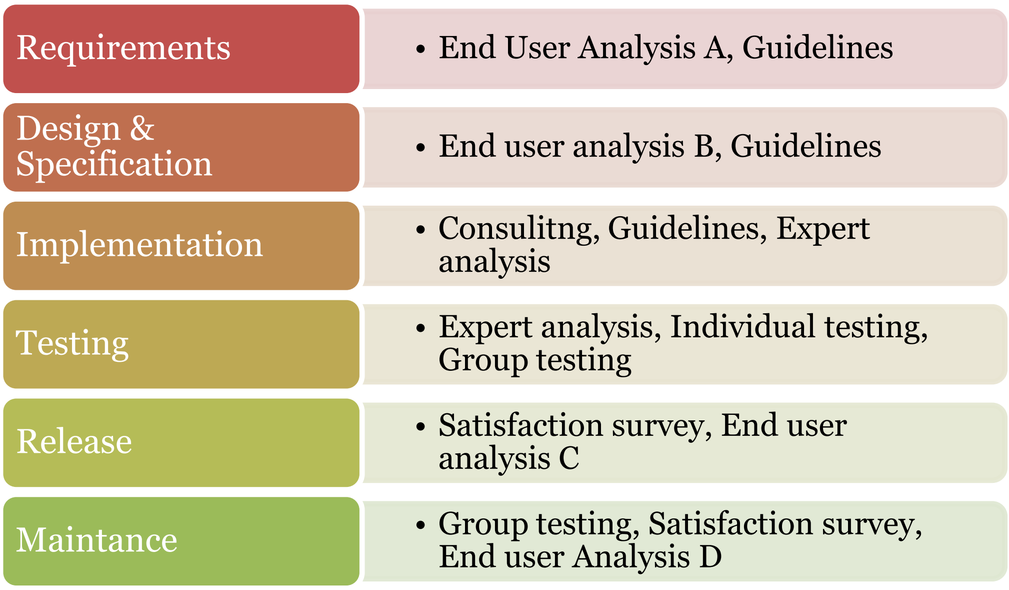
2.5 Mobile interface design

User interfaces combining computation with communication features are enabling mobile products and services to penetrate environments for work, play, and on the way [20]. Usually usability is one of the most important factors to make or break an application. This is especially important when designing ubiquitous mobile applications, as they should be unobtrusive and designed for smaller and touch capable screens.

With the growth of competition in the mobile applications market the urgency of meeting user needs has drastically increased. Usability engineering, an iterative process that has several stages from getting to know the user to collecting user’s feedback, is usually used to design a usable user interface. When applied to mobile application development, usability engineering is divided into six phases presented in Figure 4.

In the *requirements phase*, the main concept is defined and the idea behind is tested. To validate the concept usability the concept should be evaluated against the predefined usability requirements and tested with a group of end users. In this stage the concept could still be abandoned without a huge financial impact.

After the concept validation, in *design and specification phase*, technical and usability specifications are created. The entire navigation structure of the application, as well as user interaction with the application is defined. Again the output is evaluated against the usability requirements and verified that it does not violate existing standards. At the end of this stage another user analysis should be made, to check that the product design fits the end user group.



Requirements	• End User Analysis A, Guidelines
Design & Specification	• End user analysis B, Guidelines
Implementation	• Consulitng, Guidelines, Expert analysis
Testing	• Expert analysis, Individual testing, Group testing
Release	• Satisfaction survey, End user analysis C
Maintance	• Group testing, Satisfaction survey, End user Analysis D

Figure 4 – Six simple usability steps

In the phase of *implementation* the first prototype should be programmed. Usually in this phase the product is still not good enough to be tested by real end users, but can be evaluated by a usability specialist. Avoiding delays in the development process could be done with usability expert interaction and explanation of the implementation of certain features. Costly errors can still be avoided with minimal cost.

In the *testing phase* the application is thoroughly tested to make sure it still meets the requirements set. In the case of failure the product is taken back to the implementation phase to be fixed. If some new requirements have arisen during the development process they should be verified against the original requirements. This is the first phase where the product is sufficiently complete to be shown to final end users.

After passing the testing phase the first release is to be launched. The phases of *release* and *maintenance* are closely connected, as the feedback gathered through satisfaction surveys and end user analyzes is used for additional application improvements. [21]

With respect to user interface design steps we have designed an application seamlessly integrated into the mobile device desktop. With the minimization of additional application screens users are able to use the application without a lot of additional adaptations straight between everyday tasks they are performing on their mobile devices. As the application is desktop integrated it can be easily used by various age groups of end users with the assumption they are already acquainted with their mobile devices.

2.6 Wireless mesh networks

WMNs can provide an excellent framework for providing a cheap way of broadband connectivity in different community networks, as they create a robust infrastructure using the combination of ad-hoc routing protocols and wireless network technology. A WMN is a self-managing network in which all nodes act as routers that can route traffic either directly or via a multi-hop path [22].

The architecture of WMNs can be classified into three main groups:

- ✓ *infrastructure/backbone,*
- ✓ *client WMNs, and*
- ✓ *hybrid WMNs.*

Infrastructure WMNs include mesh routers forming an infrastructure for clients that connect to them. Client meshing provides peer-to-peer networks among client devices. A mesh router is not required for these types of networks, as client nodes constitute actual networks to perform routing and providing applications to users. Hybrid WMNs are simply a combination of infrastructure and client meshing. [22]

In MSNSs with the primary goal of looking for friends client meshing would be an optimal way to provide networking for a large set of geographically connected users. No additional network infrastructure would be needed as the clients would also be serving as network traffic routers. The central server equipped with a wireless interface could be easily integrated into client meshing exposing the functionalities on higher network layers. A network architecture proposal is presented in Figure 5. Of course for a pure client meshing implementation a high density of mobile clients is needed otherwise the communication could be broken. As a solution a hybrid WMN could be used with a main backbone connecting the critical parts of the network.

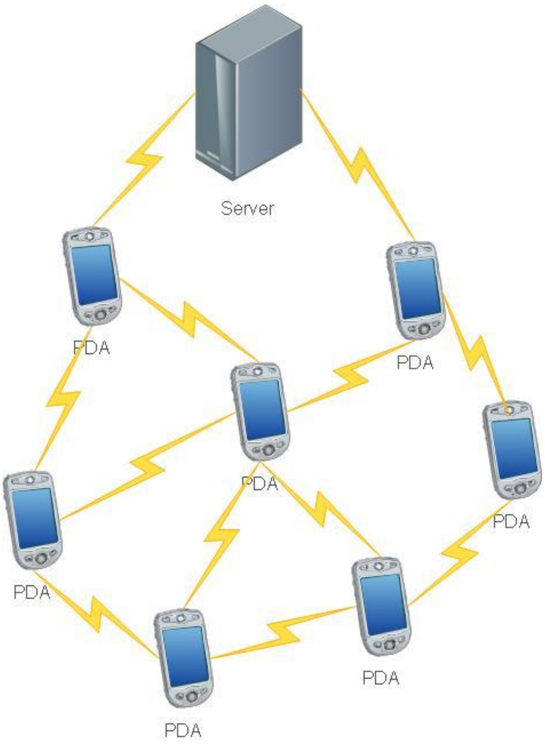


Figure 5 – A simple Client WMN architecture

While the solution proposed would be optimal for crowded areas, as with the use of client meshing no investment in infrastructure would be needed, it could be also useful in not widely populated areas, where people have usually problems with fitting in to community, for example residential facilities.

2.7 Ambient assisted living

AAL includes methods, concepts, systems, devices as well as services that are providing unobtrusive support for daily life based on context and the situation of the assisted person. With population ageing rapidly and the rising number of single person households in combination with higher expectations towards the quality of life, AAL technologies are emerging faster than ever. Besides the increasing demand of safety and comfortable living environments, AAL technologies also cater towards the increasing demand for communication and stronger social interaction. [23]

MSNSs with the primary purpose of looking for friends could be used in the context of AAL to prevent social exclusion in residential and nursing facilities. As it is usually hard to adapt to a change of environment, especially for older people due to their lack of mobility, a mobile based assistant could be used to overcome the communication establishment problems. With the use of context information, e.g., favorite locations overlapping between different users and their friends' similarity, the MSNS we are proposing could be used to find the most suitable person being able to introduce a specific newcomer in a residential facility.

3 TECHNOLOGY

In this section we will take a closer look at the idea of using a mobile social network for lowering the communication barrier. We will present SocioNet, a prototypical implementation of a MSNS proposed for lowering the communication barrier, and some of its main functionalities and goals. We will also describe the concepts we are using for storing the profiles of SocioNet users and some security upgrades we have decided to apply, to minimize the possibilities to abuse the profiles data stored.

3.1 SocioNet

As stated out in the introductory paragraph (Section 1) an MSNS could be used to enhance the real life person-to-person communication. Two of the main SocioNet goals are:

- ✓ *locating a communication partner available for chat or to help, and*
- ✓ *finding the most suitable partner for communicating.*

We are trying to achieve those goals with the use of context information obtained from the users' mobile device. This simply means that users don't have to fill out any profile information besides setting their username and selecting which mobile device data should be used and if they would like to allow the data to be stored on central server. The following mobile device data is used:

- ✓ *address book,*
- ✓ *calendar,*
- ✓ *tasks, and*
- ✓ *location.*

The address book information is used to compare the quantity of common contacts between users. With a high number of matching contacts we are assuming that two users are communicating with almost the same persons and could be a better match as the persons with fewer contacts in common.

The calendar and tasks information is used to reason about a users activities. Users with higher number of activities in common could be a better match then the ones with few or no activities in common.

At least we are using location data to relate users spending more time in the same areas. If two users are mostly visiting the same places they could present a good social match. We are also using location information to perform matchmaking only between network users in proximity as they could get together faster and easier and could present a better geographical match.

With respect to mobile information used for matchmaking roles described in Table 4 are used to provide users with the possibility to pick a person matching his personality or current mood. If the user would like to get in contact with the person he or she knows or is partly familiar with, the user can pick a result annotated with the friend or a friend-of-friend tag. If the person would like to socialize with an unknown person who shared interests with the user or even with a total stranger nearby, he or she can easily pick a match tagged with the interests match or a stranger tag and possibly expand his or her social network.

Role	Description
Friend	A person with whom the user is acquainted.
Friend-of-friend	A person with whom a friend of the user is acquainted.
Interest match	A person with same interests, taking part in the same social activities.
Stranger	A person in proximity, which is not a friend, neither a friend-of-friend nor an interest match.

Table 4 – Description of the main SocioNet roles

All of the roles used by the SocioNet application are based on some simple classification rules explained in Table 5. With respect to the described classification rules, different roles are not necessarily mutually exclusive, as the friend or a friend-of-friend could also have interests in common with the user (interest match).

Role	Profile section	Classification rule
Friend	Contact-book	A person whose telephone number is stored in the user's contact-book.
Friend-of-friend	Contact-book	A person with whom the user is sharing one or more friends.
Interest match	Calendar	A person with whom the user is sharing social events in the calendar.
Stranger	Contact-book, calendar	A person having nothing in common with the user, but residing nearby.

Table 5 – Description of the main SocioNet rules

With the use of the user's address book, calendar, tasks and location information, there is a lot of opportunities for personal data abuse. While address book information could be used for spamming purposes, the location and calendar data could reveal a lot about user's interest to people he is not willing to share this information with. In SocioNet we are using the techniques of *profile obfuscation* and *cumulative location tracking* to minimize the possibilities of profile abuses.

3.2 Profile obfuscation

The most straightforward way to provide security in distributed applications is with the encryption of the communication channel [24]. While this is certainly a good measure to prevent leaking data from the communication channel, the data is still unprotected when saved in a central environment. As basically anyone with credentials to access the central storage could read the data, the P2P solution without a central authority seems much more appealing, as all the data is stored on the user's device. In particular for the central server solution, additional methods should be applied to increase the level of security

By designing SocioNet we were focusing on two main goals considering profile data:

- ✓ *profiles should be as meaningful for the system as possible to allow creating of correlations between persons, and*
- ✓ *profiles should be as hard as possible to read by humans.*

We have decided to explore the common nature of mobile data, its atomicity. Taking a look at the phone numbers stored on a mobile device – it is basically not important if two phone numbers are matching in its original form or in some other transformed form. The same applies for the communication channel log. In the case of checking the two most frequent email communication partners used by two persons we don't need the address in a readable form, it is enough we are able to say, those two emails were the same before transformation. To make the profile data unreadable to humans, but still rich enough with information for the comparison to take place, we have decided to apply a one-way function[1] on the original data.

Although the same could be done with all of the calendar data, we have decided to preserve some of the calendar data in its original form. The main reasons for that were:

- ✓ *crucial additional information such data could provide us, and*
- ✓ *a relatively small impact on privacy.*

We have decided to preserve some numerical information like numerical parts of the address the event is taking place at. While it wouldn't have any sense storing a phone number its original form, as it is impossible to say two persons could be a good match if their phone numbers are differing only in the last digit, we could certainly say two persons could be a better match if they are regularly visiting two nearby buildings in a street. As preserving this data could be useful for getting better results, the privacy of the user is staying basically on the same level, as the information the user is visiting a unknown street with the known street number is not revealing a lot of information.

Of course this concept is working only on the top of the data entered correctly. While it is on the user to enter the name or the address of an event he is attending correctly, it is also possible for the client application to unify the data before transforming it into an unreadable form. Depending on the user's mobile provider settings, the country code could be appended to transform all of the numbers into a matching format. It is also possible to

[1] In computer science, a one-way function is a function that is easy to compute on every input, but hard to invert given the image of a random input. [25]

transform the entire calendar and tasks data into one case strings[1]. The flow of the profile data preparation is presented in the Figure 6.

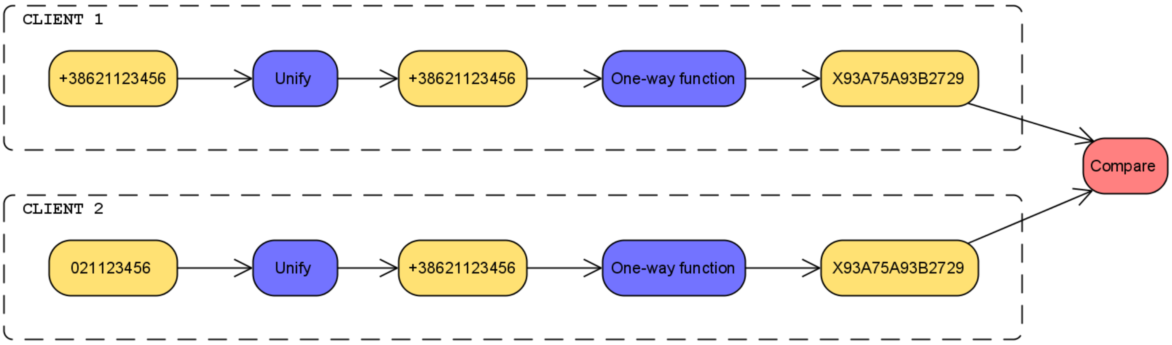


Figure 6 – Data unification and transformation.

While this method could be used for the obfuscation of address book, calendar, tasks and communication channels data a different approach has to be used for privacy preservation in the case of location use.

3.3 Cumulative location tracking

Location data obfuscation is not possible because with a one-way transformation we would be losing important information about the Euclidean distance[2] between specific locations. To preserve as much information as possible and avoid tracking the user, a different approach could be taken.

For finding a good match, it is important to know the locations a person is spending a lot of time at. Although having the information when a user was at a specific location could be a great value to match the persons visiting the same places at the same time, with ditching the time information we are drastically improving the privacy factor and still knowing which places the user is spending the most time at. For example if two users are spending a lot of time in a specific shop, it could be deduced they both like the things the

[1] For example only lower case strings could be used, as it is only important for the strings comparison that they are in the same case.

[2] http://planetmath.org/encyclopedia/EuclideanDistance.html, visited: May 2009

shop is selling, although one is visiting the shop in the morning, while the other is mostly there in the afternoon.

As the MSNS is a geographically limited network, used to help connecting the people nearby, there are two possible approaches for collecting location information:

- ✓ *location information is only collected inside the MSNS area, or*
- ✓ *location is collected all the time.*

While the second approach could give us a lot better results, as it would reveal location information in a broader range, the first one allows us some additional privacy enhancements.

The MSNS area could be divided into a grid like surface of sub-areas and only cumulative results for each sub-area could be saved. With an interval-temped sampling procedure only relative portions of time spent in each area are saved, which additionally enhances the privacy aspect and simplifies the matching routine, as only the difference of time spent in each sub-area is compared between users.

In SocioNet we have decided to use a homogenous grid with a predefined resolution.

3.4 Rule-based profile storing

As mentioned in Sections 2.3.2 and 2.3.3 P2P and central server approaches have various advantages and disadvantages. While the P2P approach is usually the much more trusted one as there is no central database with a huge amount of personal data, it comes with a disadvantage in terms of large network traffic. As all of the matching data is stored on the user’s mobile device it has to be transferred between devices on matchmaking taking part.

In SocioNet we tried to address the problem of lowering the network overhead and in the same time still preserving high level of trust in the network. We proposed a solution where users are able to decide what data could be stored on the server to speed up the matchmaking process. Additionally we introduced a rule-based profile storing approach and made it possible for the network to reorganize in such way the data producing the best matches is, with respect to user's privacy settings, always stored on the central server.

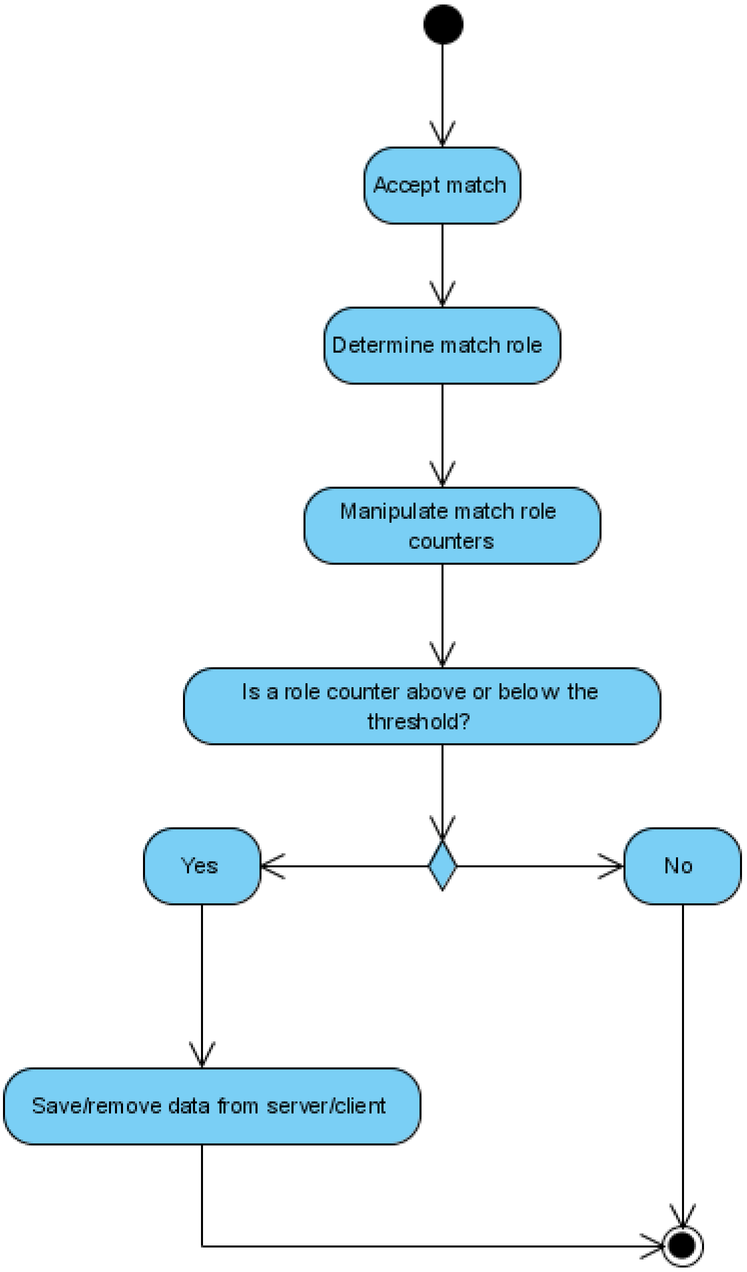

Accept match
Determine match role
Manipulate match role counters
Is a role counter above or below the threshold?
Yes
No
Save/remove data from server/client

Figure 7 – A simplified profile reorganization workflow.

Figure 7 presents a simplified workflow diagram of role processing and profile reorganization. After a match is accepted, its role is determined and specific user-role match counters are increased. If a match counter for a role meets a pre-configurable threshold, profile reorganization is performed. E.g., if a user regularly accepts interest matches it is possible to make a high probability prediction that the next match accepted will be also an interest match. Therefore, if allowed by the user’s privacy settings, profile data used to recognize interest matches (e.g., calendar data) may be stored on the server to decrease network traffic.

4 ARCHITECTURE

4.1 Network design

SocioNet is using a hybrid between the central server and P2P network architectures to overcome their disadvantages and exploit advantages they are offering. While P2P architecture offers a higher level of network trust, the central server architecture combined with the rule-based profile storing approach lowers the network traffic and allows SocioNet to be deployed on top of networks with slower throughput. Although Wi-Fi network access and IP is used in the prototype implementation, the network layer is abstracted and could easily be switched to, e.g., GPRS or a 3G network.

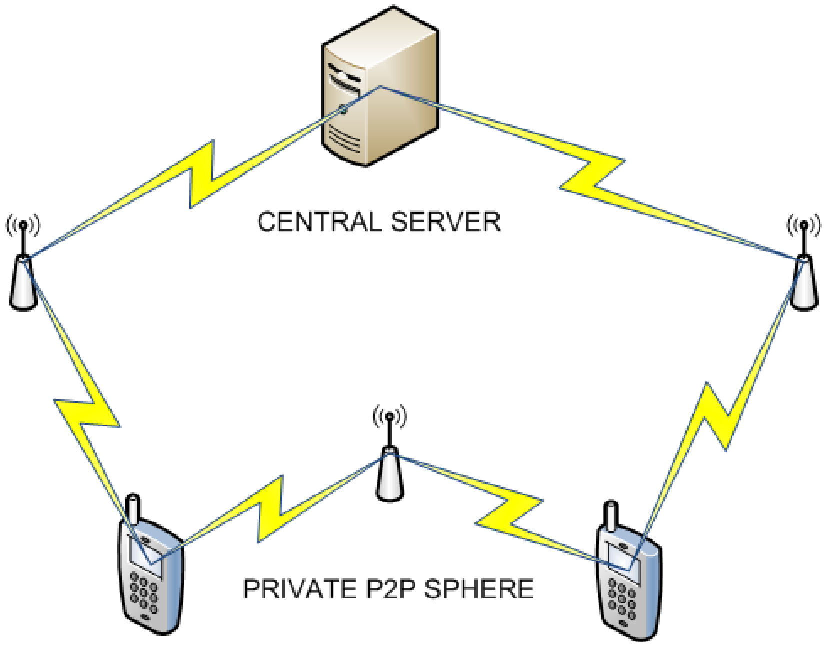


Figure 8 – Hybrid network architecture

The base components of SocioNet network architecture, as stated in Figure 8, are:

- ✓ *central server, and*
- ✓ *private P2P sphere.*

As described in section 3.4 central server is used to store matchmaking data producing the best match results. Although personal data is obfuscated and transformed into a not readable form, it could still be stolen by a network administrator and transformed back into its original form using a simple rainbow attack [26]. Therefore additional P2P private sphere is introduced where data is exchanged directly between clients and matched on mobile devices. As matchmaking data is not entering the central server at any point the risk of personal information disclosure and theft is additionally lowered.

4.2 Use cases description

The use case diagram presented in Figure 9 covers main client, server, and network administrator functionalities implemented in the SocioNet prototype solution.

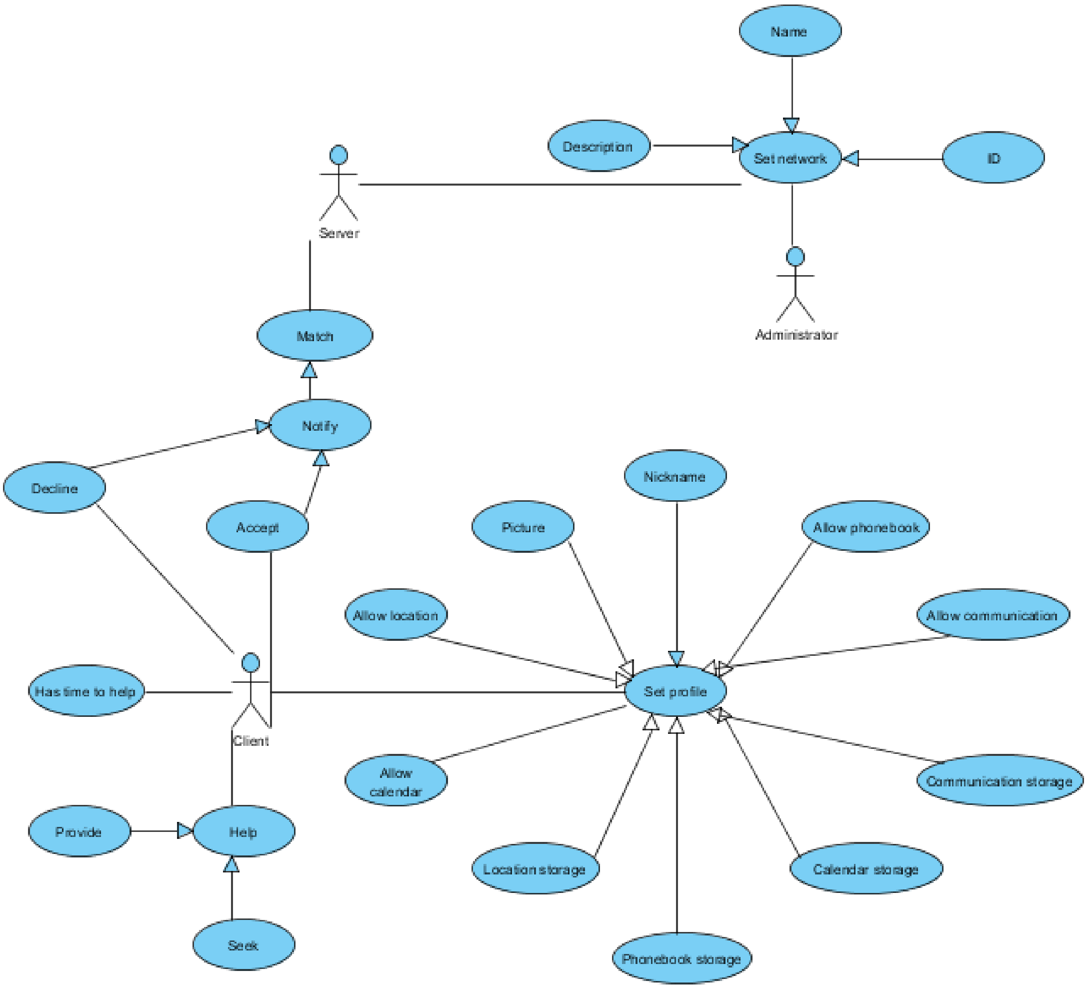


Figure 9 – SocioNet use case specification

Use case	Description
Has time to help	Used to set availability of a user.
Seek help	Used to start the matchmaking process and find matches available to help
Provide help	Used to provide help to a help seeking user.
Accept match	Used to accept a proposed match.
Decline match	Used to decline a proposed match.
Set picture	Used to set a profile picture for the user.
Set nickname	Used to set a nickname for the user.
Allow phonebook	Used to allow or deny phonebook information to be used for the matchmaking process.
Phonebook storage	Used to allow phonebook information to be stored on the central server after profile reorganization.
Allow communication	Used to allow or deny communication channels information to be used for the matchmaking process.
Communication storage	Used to allow communication channels information to be stored on the central server after profile reorganization.
Allow calendar	Used to allow or deny calendar information to be used for the matchmaking process.
Calendar storage	Used to allow calendar information to be stored on the central server after profile

	reorganization.
Allow location	Used to allow or deny location information to be used for the matchmaking process.
Location storage	Used to allow location information to be stored on the central server after profile reorganization.

Table 6 – Client use cases descriptions

Table 6 describes different use cases for Client SocioNet actor. While the main server use cases are performing the matchmaking routine and notifying users about the matchmaking process, administrator's main tasks are setting the network properties like an ID of the network, network name and a network description.

4.3 Sequence flow description

After use case diagrams modeling sequence diagrams were created. Sequence diagram in Figure 10 is describing top overview of different client and server actions and proposes a positive scenario application flow.

Initially the client provisions to the network with a login message call. An appropriate status response is issued from the server to signalize the login status. Afterwards synchronization of the data stored on the server is carried out – e.g. if phonebook information is stored on the server and user's phonebook entries have changed, server data has to be synchronized to the valid state. Server and client side checksums are used to check if the server data has to be synchronized and a status message is returned from the server to notify the client about the success of the synchronization. To start the matching procedure a *FindMatches* message is passed to the server, which starts an independent matching thread and notifies the client that matching is in progress. Matchmaking is performed on the server or on the client devices, with the respect to the user's privacy settings. After the matching thread executes a notification is sent to the client signalizing the matchmaking results are ready to be pulled from the server. The client obtains the matches with a *GetMatches* call and visualizes the matchmaking results for the user.

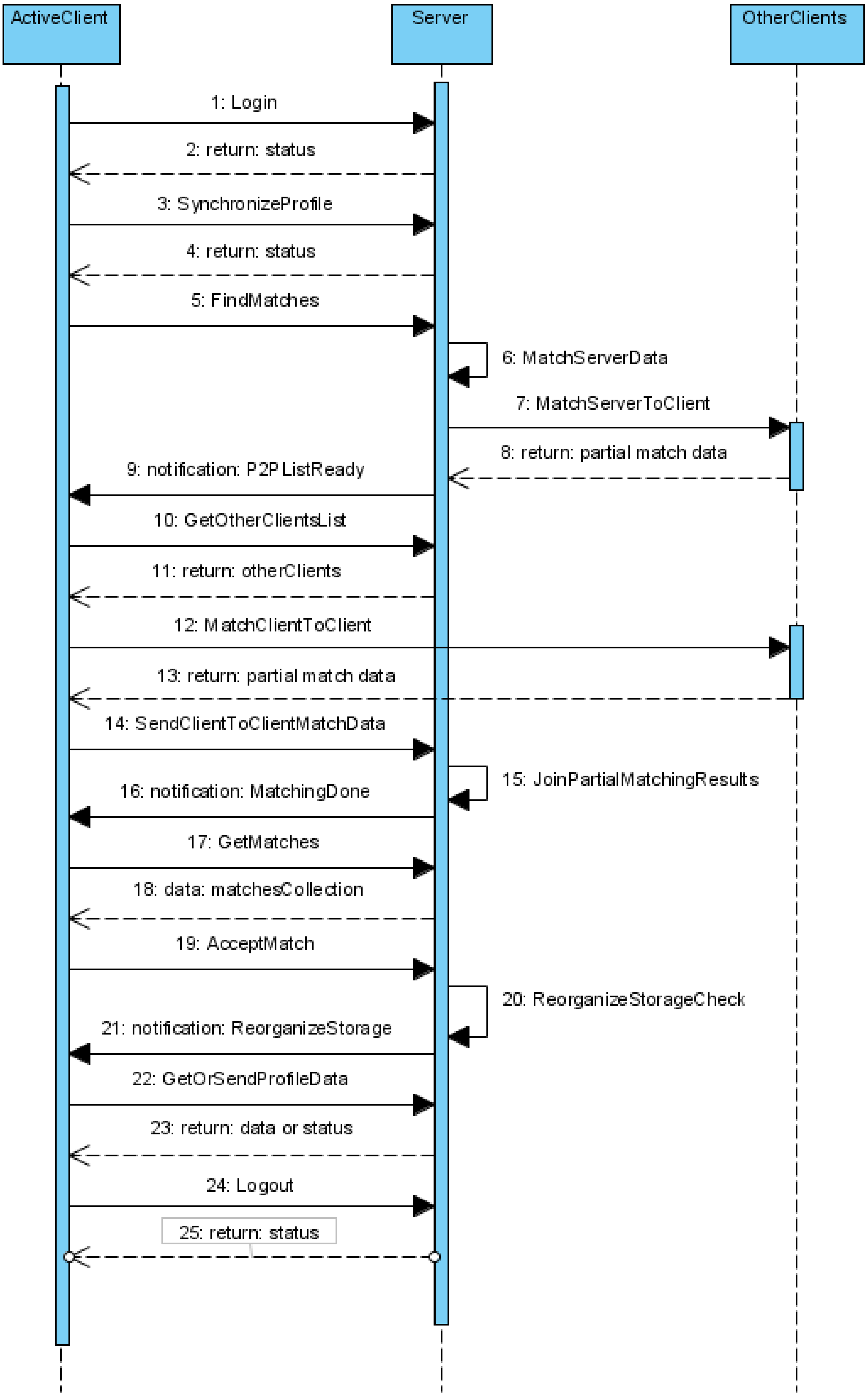


Figure 10 – A simplified client-server workflow

In the scenario explained an *AcceptMatch* message is passed to the server and an appropriate user-role counter is increased to register match preferences for the active user. Additionally, a storage reorganization check is carried out and a *ReorganizeStorage* notification is passed to the client if a user-role counter has exceeded the pre-defined threshold. With respect to the reorganization message, client and server profile reorganization is carried out. At the end a logout message is passed to the server and the client is removed from the active members queue.

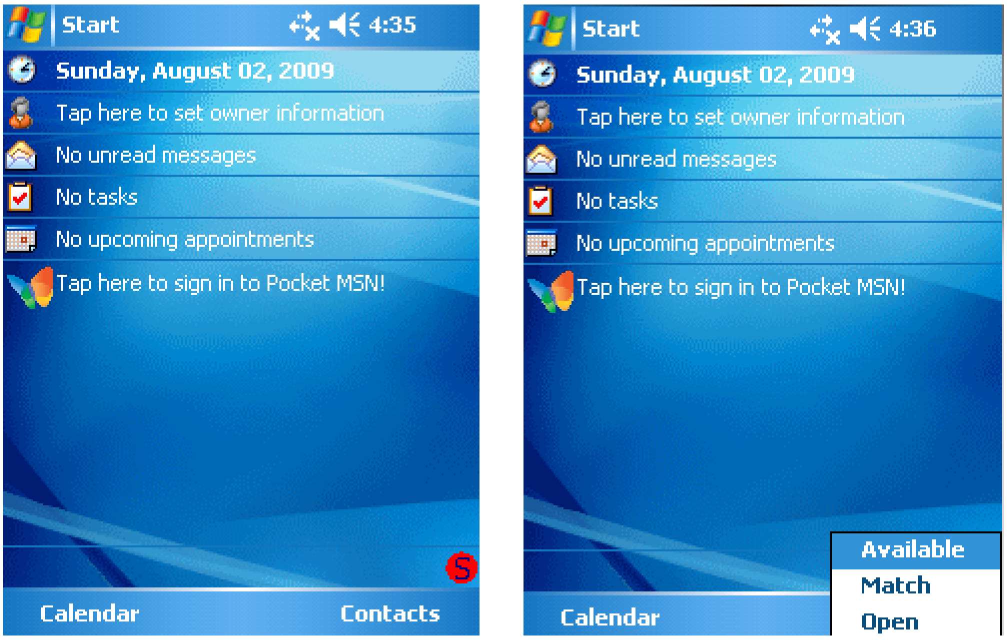
4.4 User interface design

After modeling the functionality specifications client application screen masks were designed to make the application more intuitive and easy to use. Following the guidelines from Section 2.5, we have decided to integrate the client application into the mobile device desktop and minimize the number of application's own screens to make its learning curve as steep as possible.

As presented in Figure 11, main application's functionalities were added to the mobile device tray toolbar and were made controllable through an action menu with the commands most frequently used:

- ✓ *change availability,*
- ✓ *find matches, and*
- ✓ *open application and show matchmaking results.*

Additionally, a colored application icon was used to signalize the availability of the user without entering the application or opening any additional menus.



Sunday, August 02, 2009

Tap here to set owner information

No unread messages

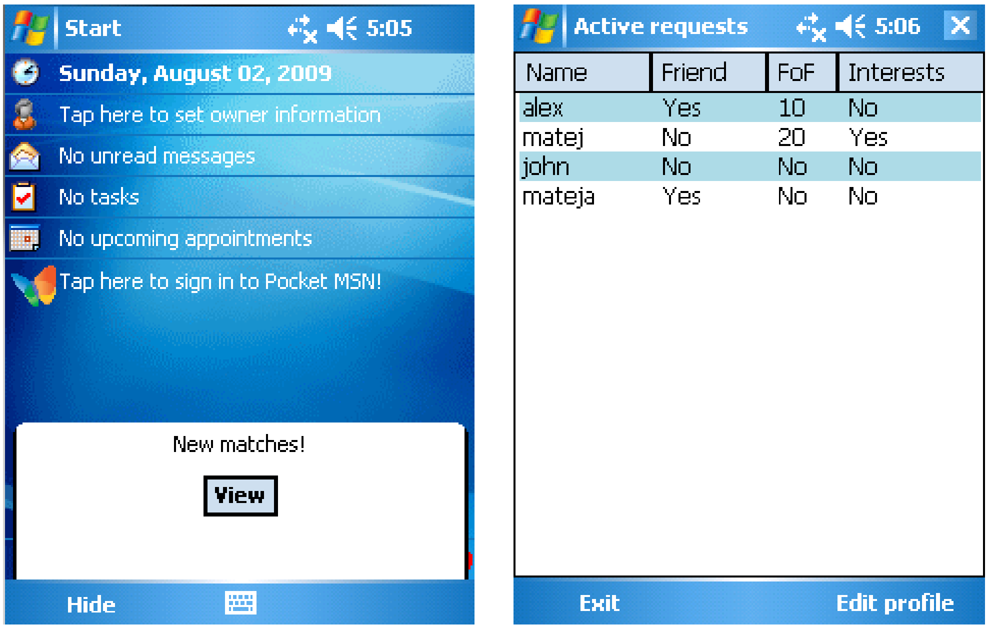
No tasks

No upcoming appointments

Tap here to sign in to Pocket MSN!

Figure 11 – Desktop integration screen masks

While the *Available* button in command menu was used to change the current availability of the user, the *Match* button immediately issues a matching request and asynchronically notifies the user about the matchmaking results. The *Open* button is simply serving as a shortcut for opening the application central screen, viewing past matches or changing different profile settings. Figure 12 presents new matches notification and the active requests screens.



Sunday, August 02, 2009

Tap here to set owner information

No unread messages

No tasks

No upcoming appointments

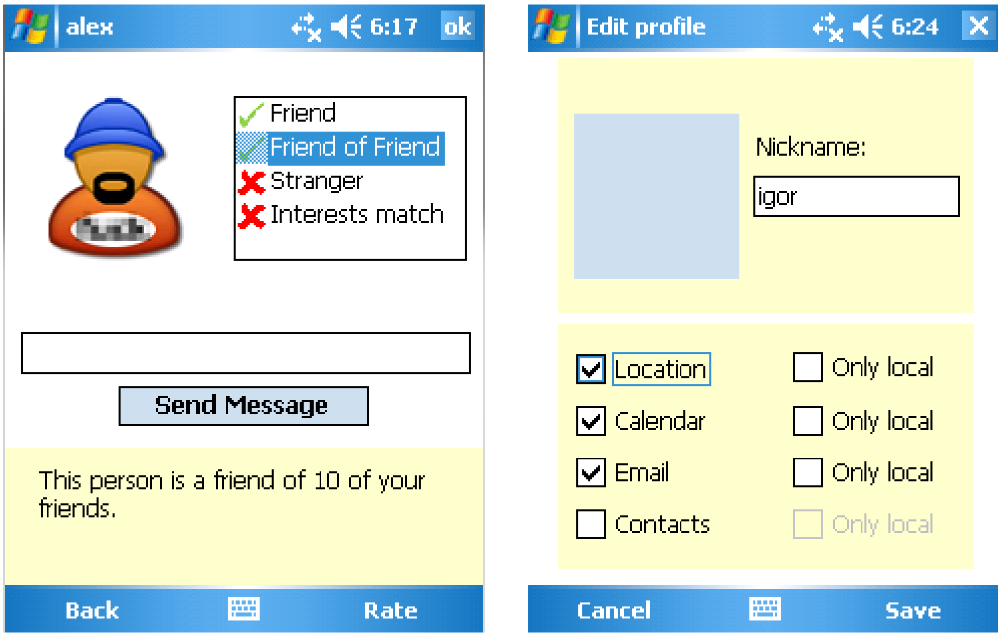
Tap here to sign in to Pocket MSN!

New matches!

Name	Friend	FoF	Interests
alex	Yes	10	No
matej	No	20	Yes
john	No	No	No
mateja	Yes	No	No

Figure 12 – New matches notification and active requests screen

From the active requests screen a user is able to view a match and initialize a short text communication with a match accepted. Finally, a user is also able to change his profile settings, e.g., which data should be used for matchmaking and where the data should be stored (Figure 13).



✓ Friend
Friend of Friend
✗ Stranger
✗ Interests match

This person is a friend of 10 of your friends.

Nickname: igor

☑ Location ☐ Only local
☑ Calendar ☐ Only local
☑ Email ☐ Only local
☐ Contacts ☐ Only local

Figure 13 – View match and edit profile screens

5 IMPLEMENTATION

The SocioNet mobile social network prototype is built as a client-server solution with P2P extension. While the server is implemented in the .NET Framework, the client prototype is based on the Pocket PC .NET Compact Framework 3.5 platform. Although Wi-Fi network access and IP are used in the prototype implementation, the network layer is abstracted and could easily be switched to, e.g., GPRS or a 3G network. In the subsections to follow client and server implementations will be presented briefly. Additionally we will also describe how the two-way communication channel was established.

5.1 Client implementation

As mentioned before, the .NET Compact Framework 3.5 platform was used to implement the SocioNet mobile client application. A Pocket PC platform was chosen for the prototypical implementation, as the touch functionality could be of a great use and could made the application more intuitive as one may experience by using new smart phones as iPhone or HTC magic as well.

Additionally, a non-standard .NET CF library (OpenNETCF[1]) was used, mainly because of a more intuitive approach to multithreading and to avoid using unmanaged code to access Windows Mobile notification icons. The classes we used were:

- ✓ *BackgroundWorker, and*
- ✓ *NotifyIcon.*

The main classes and interfaces presented in Table 7 were used in the prototype implementation of the SocioNet client.

[1] http://www.opennetcf.com/, visited: April 2009

Name	Type	Description
ApplicationContext	Class	Contains current state of the application.
HealthRoutine	Class	A class used to respond to a health ping from server.
MatchResult	Class	Entity structure for storing matchmaking results.
ServerSynchronizationForm	Class	A form used to visualize synchronization progress.
NotificationServer	Class	Main server used to establish client-client and server-client communication.
ViewMatchForm	Class	A form used to view a match.
CreateEditProfileForm	Class	A form used for creating and editing profiles.
ProfileData	Class	A class used for managing profile data.
GeneralHelper	Class	An utility class with commonly used methods.
MainForm	Class	A form used to present the matchmaking results and provide main application actions.
MatchesRoutine	Class	A class used to respond to matchmaking actions.
INotificationRoutine	Interface	An interface used to provide extensibility for additional message

	intercepting routines.

Table 7 - Classes and interfaces used in SocioNet client solution.

5.2 Server implementation

While .NET Compact Framework was used in SocioNet client solution, SocioNet server was implemented on the top of the .NET Framework 3.5 platform. To achieve a higher level of modularity, the server solution is implemented as host application and a library of XML Web services, which could be upgraded or replaced without much effort.

SocioNet server solution contains the following projects:

- ✓ *SocioNet.Server,*
- ✓ *SocioNet.Services, and*
- ✓ *SocioNet.Database.*

SocioNet.Server project is responsible for hosting various services implemented in SocioNet.Services namespace. To achieve database extensibility SocioNet.Database namespace is proposed to abstract the database used by SocioNet server-side solution.

Some of the main SocioNet.Services are:

- ✓ *ManagerService, and*
- ✓ *AdministratorService.*

While ManagerService is responsible for mainstream client-server communication AdministratorService is available to be used by a network administrator and exposes network management methods.

ManagerService exposes the following methods:

Method name	Description
SynchronizeProfile	Used to synchronize server profile data with client profile data.
GetChecksums	Used to obtain server profile data checksums.
ChangeAvailability	Used to change the current availability of a user.
FindMatches	Used to start the matchmaking procedure.
GetMatches	Used to obtain matches results.
GetProfileData	Used to obtain profile data from the server.
SetProfileData	Used to store profile data on the server.
SendHealthResponse	Used to respond to a server health ping.

Table 8 – ManagerService methods.

5.3 Two-way communication protocol

As there was no official support for hosting WebServices on a mobile device using .NET Compact Framework, a two-way communication protocol was designed in order to simulate server-client communication (Figure 14).

We have decided each client will be listening on a predefined port for TCP messages. Each message will contain a 1 byte long notification code, which will be used to execute the appropriate client routine. With respect to the notification code a client routine is executed and a server WebService is optionally called.

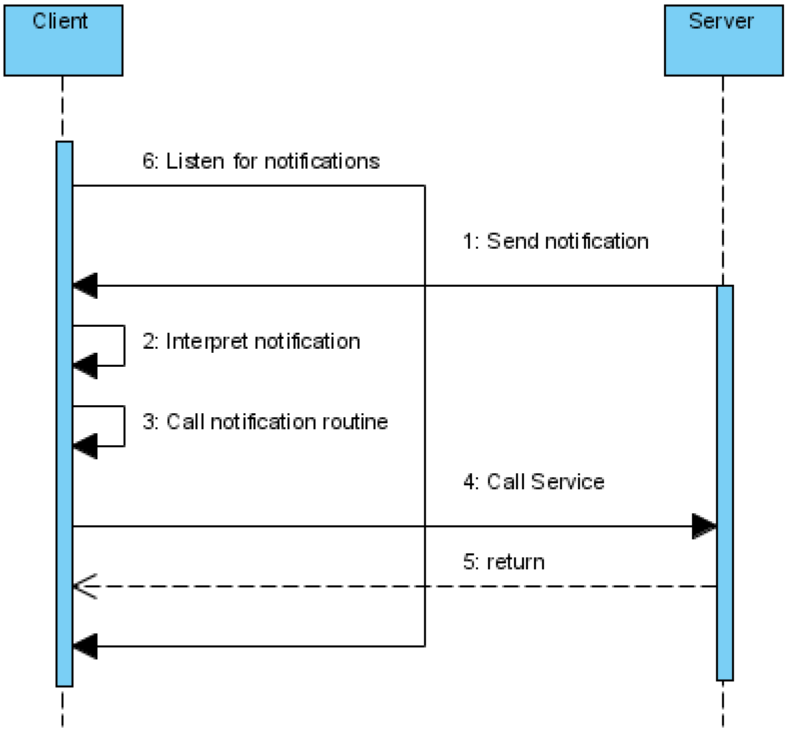

Client
Server
6: Listen for notifications
1: Send notification
2: Interpret notification
3: Call notification routine
4: Call Service
5: return

Figure 14 – Two-way communication protocol

In the prototype implementation SocioNet client is using the following messages:

- ✓ *matchmaking finished message,*
- ✓ *health check message, and*
- ✓ *cache message.*

6 EXPERIMENTS AND RESULTS

The prototype has been evaluated with respect to performance and user acceptance. In this section, we will briefly summarize the results of these investigations.

6.1 Performance experiments and considerations

We have evaluated the impact of the central server, client-side, and rule-based approaches in terms of network traffic (bytes transferred) by analytically deriving the relationships and subsequent measurements for plausibility reasons. The metric used is the overall traffic size in bytes.

With respect to simplicity, the experiments were performed with the use of contact-book information only. 100 users nearby were simulated, each one with 100 contact-book entries of the size of 32 bytes per record after obfuscation. Each user requests matchmaking to take place 10 times in the simulation period. The use of the pure P2P architecture (client-side storing profile approach) resulted in 3200 bytes transferred 99 times by 100 users, as for each matchmaking request the whole profile had to be transferred to the match-requesting mobile device. After 10 matchmaking requests, 316.800.000 bytes have been transferred over the network.

The pure P2P solution is most beneficial in case small amounts of data have to be exchanged. With the use of a central server, there is an initial data transfer, where the whole data is transferred. Because matchmaking is done on the server there is no need for additional data transfer with the exception of profile updates as a result of device data changes (e.g., a new contact is added to the address book).

For the pure server solution, the size of the initial data transfer of 100 clients is 320.000 bytes, while the incremental synchronization weight depends on the frequency of contact-book changes. When using the rule-based profile, storing approach the traffic size numbers are higher as the numbers of the pure central server storing approach, but still lower as the pure P2P approach numbers, depending on the quantity of the profiles stored on the central server. If, e.g., 50% of the profile is stored on the server, the expected traffic load should be at least half the traffic load produced by the peer-to-peer solution (here, without considering profile changes).

6.2 Questionnaire

After a short demonstration of the features of SocioNet, we conducted a user questionnaire to find out the answers to the following crucial questions:

- ✓ *Do the participants think that SocioNet will open new communication options and which roles do they like to interact with (friends, friends-of-friends, persons sharing interests, strangers)?*
- ✓ *Do the participants think that privacy is assured by keeping data local?*
- ✓ *Do the participants think that their mobile personal information is rich enough to lead to good matches?*

We evaluated the answers of 40 participants in the age ranging from 18 to 48 years. The *Computer Proficiency (CP)* self-estimate of the participants was either *high* (*CP*=3), *medium* (*CP*=2), or *low* (*CP*=1). The average proficiency CP was calculated as 2.6 and is rather high. Table 3 summarizes the results of the questionnaire.

Classes [# of participants]	CP < 3 [15]	CP = 3 [25]	All [40]
Opens new communication options	73	76	75
Use SocioNet to contact a *friend/ friend-of-friend/stranger/interest match*	53/33/13/46	64/52/24/68	60/45/20/60
Privacy: *in control/local storage*	20/93	56/72	43/80
PI-data sufficient	53	52	53

Table 9 – Results of the User Questionnaire [in %]

We can state some observation based on the results of the questionnaire:

- ✓ *More than 70% of both computer proficient participants and participants not that familiar with computing technology think that SocioNet opens new communication possibilities.*

- ✓ *When asked about the type of contacts which should be fostered via SocioNet, it is remarkable that higher proficiency generally comes with a higher likelihood of using SocioNet for establishing (social) contacts.*

- ✓ *Generally, 60% of the participants would use SocioNet to contact their friends and persons with similar interests, while only 45% would contact friends-of-friends, and 20% of the participants would also like to use SocioNet to contact strangers. These first results show that the participants base their contacts on existing close relationships or on existing similar interests.*

- ✓ *It is remarkable, that only 20% of low and medium proficient participants think they are in control of their privacy, while 56% of the highly proficient participants feel in control. Here, we see a general lack of mental models of where digital personal data is best stored and how unauthorized access can be prevented which may be a cause of this feeling.*

- ✓ *On the other hand, 93% of non computer proficient participants felt that storing the data not on a central server helps to increase privacy, while highly computer proficient participants were more reluctant (72%). Here again, computer proficient participants have most likely considered several remaining threats to privacy (like, men-in-the-middle, not trustworthy P2P system, etc.).*

- ✓ *The answers resulted in a rather low rating of the sufficiency of personal information, which is used by SocioNet to avoid entering of boring user profiles. Many participants think that personal information stored on their mobile devices will not be sufficient to identify persons to contact. Here, we see that a longer field study using real personal information data would be required to come up with results proving the usefulness – but also the limits – of our approach.*

Finally, the participants were asked why they would *not* use SocioNet to contact others. The major considerations were the following: Some participants thought that they do not need technology to contact friends or friends-of-friends. Additionally, trust in the system is a major, open issue [27].

7 CONCLUSION

In this work, we have introduced SocioNet, a social networking application dedicated to mobile smart devices to support social interactions among persons in proximity. While addressing performance and trust issues, SocioNet should provide pointers to others willing to interact and help in particular every-day situations. SocioNet follows the vision that human-to-human interaction allows for better problem solutions due to the richness of human social interactions when compared to a solely database based recommender system. To find best matching persons, we introduced an extended role model representing social relationships.

Due to privacy issues, we provided a solution where personal information does not have to be stored on a central server but may be exchanged between the participating mobile devices directly. The prototypical implementation showed, that this peer-to-peer solutions results in significant messaging overhead, which could be reduced by adding rules resulting in only partial central storage of personal data.

To evaluate the usefulness of SocioNet and the privacy enhancing solutions, we performed a questionnaire. 75% of the 40 participants (computer proficiency high) rated the prototype as capable of establishing new communication options, which was our general aim. SocioNet would be more likely used to establish communication channels with friends and persons of similar interest than with unknown persons (even if they are friends-of-friends). When it comes to privacy, we investigated that the loss of control over private data is felt more intensive by low and medium computer proficient participants. A mental model for digital data and data privacy would help to improve this situation.

The questionnaire also pointed out that it is not yet clear to the user whether personal information retrieved from the databases available on mobile smart devices is sufficient to perform good matchmaking [27].

8 LITERATURE

[1] J.Q. Pearce, »4.1 Billion Mobile Subscribers Worldwide Helps Reduce Digital Divide«, http://www.washingtonpost.com/wp-dyn/content/article/2009/03/02/AR2009030201931.html, visited: April 2009

[2] Worldwide State of the Mobile Web 2008, Nielsen Mobile, July 2008

[3] J.Q. Anderson, L. Rainie, Future of the Internet III, Pew Internet & American Life, December 2008

[4] V. Gundotra, »Follow the Mobile User«, http://www.techcrunchit.com/2009/03/29/follow-the-mobile-user/, visited: April 2009

[5] M. Schopp, Trends in Mobile Network Architectures, Siemens Networks, November 2006

[6] J. Horrigan, Mobile access to data and information, Pew Internet & American Life, March 2008

[7] R. Godwin-Jones, Emerging technologies mobile-computing trends: Lighter, Faster, Smarter, Virginia Commonwealth University, October 2008

[8] D.M. Boyd, N.B. Ellison, Social Network Sites: Definition, Histroy, and Scholarship, Journal of Computer-Mediated Communication, 2007

[9] N. Ziv, B. Mulloth, The Social Networking Arena: A Platform for Innovation, Institute for Technology & Enterprise, Polytechnic University, September 2006

[10] O. Mäkinen, Mobile social networking: T-109.4300 Network Services Business Models, April 2009

[11] MySpace and Facebook Fast Becoming the Leading Mobile Social Networks, »http://www.abiresearch.com/press/1253-MySpace+and+Facebook+Fast+Becoming+the+Leading+Mobile+Social+Networks«, visited: April 2009

[12] L. Dajani, Social Network Study – Phases I&II, Stratemerge Inc. October 2008

[13] M.A. Moreno, N.C. Fost, D.A. Christakis: Research etics in the MySpace Era, Pediatrics, January 2008

[14] G. Chen, F. Rahman, Analyzing Privacy Designs of Mobile Social Networking Applications, In Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, December 2008

[15] V. Bellotti, A. Sellen, Design for privacy in ubiquitous computing environments, In Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work, 1993.

[16] A.K. Dey, Understanding and Using Context, Personal and Ubiquitous Computing, 2001

[17] S. Steiniger, M. Neun, A. Edwardes, Foundations of Location Based Services, University of Zurich, 2006s

[18] R. Want, A. Hopper, V. Falcão, J. Gibbons, The active badge location system, ACM Transactions on Information Systems (TOIS), 1992

[19] J. Roth, Mobile Computing, 2005, Chapter 8

[20] A. Marcus, Mobile User-Interface Design: For Work, Home, and On the Way, CHI Tutorial, 2004

[21] »Usability Is The Heart Of Development v1.0«, http://www.forum.nokia.com/main/html_readers/usability_is_the_heart_of_development.html, visited: April 2009

[22] I.F. Akylidiz, X. Wang, W. Wang, Wireless Mesh Networks: A Survey, Computer Networks, March 2005

[23] Wikipedia, »Ambient Assisted Living«, http://en.wikipedia.org/wiki/Ambient_Assisted_Living, visited: April 2009

[24] A. Josang, G. Sanderud, Security in Mobile Communications: Challenges and Opportunities, Conferences in Research and Practice in Information Technology Series, 2003

[25] Wikipedia, »One-way function«, http://en.wikipedia.org/wiki/One-way_function, visited: May 2009

[26] Oechslin, P.: Making a Faster Cryptanalytic Time-Memory Trade-Off, In: Advances in Cryptology - CRYPTO 2003, pp. 617-630 (2003)

[27] I. Pernek, K.A. Hummel, SocioNet: A Context-aware Approach for Lowering the Communication Barrier, accepted for publication at MONET'09, November 2009