Enhancing Privacy in Mobile Collaborative Applications by Enabling End-User Tailoring of the Distributed Architecture

Mohamed Bourimi[1], Bernd Ueberschär[2], Marcel Heupel[1]
Dogan Kesdogan[1], Thomas Barth[1]
University of Siegen[1], Germany
University of Kiel[2], Germany

Abstract

Nowadays, supporting social interaction and multi-user requirements with mobile applications becomes indispensable. Thereby, security and privacy are of major concern due to the frequent scandals related to misusage of end-users data or various threats such as different kinds of man-in-the-middle attacks based on inferring interaction traces. Preserving the end users' privacy, especially in mobile collaborative settings, is the most often-cited point of critique of mobile and ubiquitous computing. In this paper we present an approach empowering endusers to tailor (adapt) their mobile applications according to their privacy needs by adjusting the distributed architecture decentralisation degree also at runtime (e.g. switching among various data storage and communication servers without leaving the main social setting context). The gathering of requirements is based on lab and user trials as well as derived from accumulated experiences from various projects. We show this exemplary with the help of a prototypic mobile application to support an angling community with privacy and collaboration needs related to location-based services.

1. Introduction

The use and disclosure of personal information for private and business life is a major trend in our information society. Thereby, enhancing security and preserving end-users' privacy are becoming more important than ever especially due to the increasing frequency of scandals related to data misusage and different kind of attacks on end-users' privacy. Indeed, end-users' privacy is the most often-cited point of critique of ubiquitous computing [19]. Recently mobile applications for supporting communities and social networking are experiencing an increased usage. The conceptual design of such applications is focusing on information exchange (e.g. via shared workspaces), collaboration and social interaction to fulfil multi-user requirements in different social settings. Distributed collaborative applications, groupware, and social software are used for supporting social settings. These technologies are often referred to as socio-technical systems in different research fields and have to provide contextual information (e.g. presence, group and social awareness, etc.) [10][16][9][28]. Social interaction and data sharing being essential aspects of these distributed collaborative applications typically result in conflicting goals and represent non-functional requirements (NFRs), primarily awareness vs. privacy as well as usability aspects for both. Furthermore, different users in the same community may have very different needs for privacy as well as interpretations of privacy. With increasing awareness of users about the misuse of their data, the success of existing and further emerging services supporting social interaction in the future will largely depend on the consideration of privacy. This tendency requires special attention in the light of the growing criticality of privacy primarily because of the recent scandals and the importance of privacy for the EU. Because of the wide-spread server-centric nature of these applications and the fact that users have to disclose private information and reveal their identities (partially or fully), working with those systems allows to create user profiles at the server side which could reveal more information about the user, than he wants to give. Furthermore, such environments may construct profiles about users' interaction, which may be used for man-in-the-middle attacks. Thus, the architecture of distributed platforms supporting collaborative (social interaction) has to consider NFRs such as guaranteeing a user's privacy concerns. Considering these NFRs too late in the platform's lifecycle leads to expensive costs and insufficient applications that cannot be used as intended because of the end-user dissatisfaction [4]. In this paper we address the problem of considering these trade-offs focusing primarily on privacy, usability, and awareness provisioning in mobile collaborative applications. Further, we argue that privacy design is specific to the end users interaction context and we thus analyse privacy here by considering usability and awareness based on concrete case studies. The approach presented here allows end-users to tailor the distributed architecture according to their privacy needs. This is complemented by enabling the endusers to choose the degree of decentralisation wished in the respective communities with decentralised group-centric solutions. The requirements were gathered through an analysis of users needs as well as first evaluations of prototypes. Those were built for different case studies focusing on privacy, trust, and identity management in real-life communities. We report on the outcomes of our work and show this exemplary with the help of a mobile prototype application to support an Angling Community with privacy and collaboration needs related to location-based services. The remainder of this paper is organised as follows. We first present related work. Next, we present the derived requirements and needs primarily based on the results of the some projects such as the EU project PICOS (Privacy and Identity Management for Community Services) [25]. Then we describe our approach consisting of a tailorable decentralised group-centric solutions that can b tailored to the end users' needs. Then, we show how our approach was applied by building a mobile collaborative application supporting location-based services (LBS) scenarios. Finally, we present first evaluation and our conclusions.

Copyright © 2010, Infonomics Society

2. Related Work

Palen and Dourish mention in [23] that some level of disclosure is needed to sustain social engagement. For this, collaborative settings need some degree of information disclosure (e.g. partial or full identity revelation) in order to achieve the intended goals. For instance, in the case of social networks supported by collaborative platforms such as LinkedIn or Xing, users have not only to reveal their real identity but also reveal some private and sensitive information on their profiles (e.g. telephone numbers, addresses and professional references and so forth) [15]. In general, any kind of software and/or hardware supporting collaborative applications represents a classical research topic in the Computer Supported Collaborative Work (CSCW) research field since some decades. The recently widespread adoption of mobile devices such as mobile phones represents an additional opportunity to support interaction with situated displays in different collaborative settings. Thus, mobile collaborative applications become more and more important. Especially because the complex nature of collaborative applications which is mostly reflected in the UI, human-computer interaction (HCI) also focuses nowadays on human aspects of the development of computer technology in mobile and ubiquitous collaborative settings. Many CSCW and HCI key literature studied therefore usability in (mobile) collaborative applications and the trade-offs which could arise between privacy and awareness in those applications. Privacy is important for HCI design of collaborative environment [11][7]. Thus, design for privacy is a key requirement from this standpoint.

From the privacy perspective, most available implementations of mobile collaborative applications have either a server-centric architecture or a user-centric/client-centric architecture. User-centric approaches are not sufficient and suitable for collaborative settings since the exchange of information is the central idea of such settings. Servercentric approaches imply that the server is the central point of information exchanging. In order to bypass this problem, concepts and solutions are being developed and evaluated e.g. in many EU research projects such as the work developed within the EU FP6 integrated project PRIME (”Privacy and Identity Management for Europe”) [1]. However, a K.O. criterion for server-centric architectures which could be cited every time is that existing systems and approaches do not fully eliminate accidental or intentional risks and threats which can arise through the analysis or reconstruction of personal information and interaction traces. The building of a users fully-fledged profile remains possible at least through the judicial authorities which enforce for example service providers to allow dispute resolution means in order to recognise frauds. Furthermore, including mechanisms for better end users’ privacy control is not a trivial task from the software engineering perspective when considering that distributed systems are generally hard to design, implement and maintain; the domain complexity becomes more critical for collaborative applications and systems. Researchers in CSCW generally assume that privacy issues arise due to the way systems are designed, implemented, and deployed [8].

One way of reacting on emerging changes is to allow for tailorability. Henderson defined tailoring as ”the technical and human art of modifying the functionality of technology while the technology is in use in the field” [18] and Bannon argues that ”there will always be a need for some form of tailoring in order to fit a system into any particular setting” [2]. When developing socio-technical systems and applications, satisfying user needs and requirements is even more difficult than in the context of single-user application development. To achieve the task-technology fit [29], different approaches suggest different levels of tailorability. Thereby, the tailorability level varies from supporting customization, extension or integration [22] up to tailoring the collaboration or the development process of the socio-technical itself [14][27]. Here too, tailorability is directly related to NFRs in the case of integration or indirectly such as in the case of customisation (i.e. usability concern) or extension (i.e. architectural concern). Due to the exploratory nature of socio-technical systems and applications [28], the same socio-technical system can lead to different evaluation results in different social environments [16]. Indeed, Suchman states that action is situated in time and place and users constantly adapt their (action) plans according to the circumstances [30]. As users’ experience increases with the usage of the respective system, their desire for more extensive functionality also increases [28]. To meet the socio-technical requirement that technology has to support end-users’ needs over time, socio-technical systems and applications are designed so that they can be adapted to match emerging requirements [29]. A tailorable system can be adaptive (by changing its own behavior as a reaction to given interaction) or adaptable. Latter is based on providing mechanisms (e.g. at the level of the UI) to the end-users in order to adapt the systems to their needs. Mostly, end-users have to balance functionality and their security/privacy preferences by using several mechanisms built into the system. Thus, we focus in this paper on approaches following end-user tailoring, namely, the adaptability of the system behaviour by the end-users themselves, e.g. directly from the application (UI).

In [5], Bourimi et al. present a decentralised group centric approach which empowers the users to host the environmental system needed for collaborative settings, instead of a central hosting. The users are responsible for handling the data. Communication of different groups is still possible through the main platform of the system. The user hosting a surrounding node can share data with other groups having complete control over their data. This approach builds the basis of our work in this paper and resolves in our opinion the disadvantages cited above, namely, allowing the end users to tailor the distributed architecture according to their privacy needs. Thereby, the degree of decentralisation can be adjusted also at runtime.

Copyright © 2010, Infonomics Society

3. Requirement Analysis

The privacy requirements we address in this paper by primary consideration of usability and awareness aspects were identified in the EU project PICOS [25] as well as by developing and using the CURE (Collaborative Universal Remote Education) for typical CSCL and CSCW scenarios at the FernUniversit¨at in Hagen [17] since 2004. Thus, in this section we describe briefly these case studies, followed by analysing the needs from the end users perspective.

3.1. Requirements based on the PICOS project

The EU FP7 Integrated Project PICOS [25] dealed with privacy and identify management in mobile communities. The principal objective of the PICOS is to develop an open, privacy-respecting, trust-enabling identity management platform which supports the provision of community services by mobile communication service providers. PICOS followed a user-centered and scenario-based proceeding to gather needs and requirements (such as user stories, interviews and questionnaires) for three different mobile communities (Taxi Drivers Community, Angling Community, and Gamer Community). The first prototype addressed the needs of the Angler Community. Detailed documentation and further information concerning the requirements, PICOS platform architecture and design, and the first prototype platform server-side as well client-side architecture can be found in [26]. The first prototype was implemented for Nokia's MusicExpress 5800 devices by using Java Mobile Edition for the mobile client and an extended variant of HP's open call platform for the PICOS Platform at the server side. The server-centric architecture of the first PICOS prototype is shown in the Figure 1. There some screen shots for LBS are depicted, too. The PICOS LBS scenarios are of collaborative nature which means, that they presume the interaction of the community members (i.e. entering watercourses and fishing spots, rating those spots, etc.). The first prototype implemented selected LBS scenarios such as Sharing Fishing Sites, with its two use cases Show Fishing Spots and Add Fishing Spots; as well as the Localizing Contacts around Me. A watercourse has a name, a type, a description, and geo-location information. Members of the Angling Community are able to add/show fishing spots corresponding to a given watercourse by using their GPS or basing on a global watercourse list provided from IFM-Geomar. A fishing spot has a name, geo-location information, a description, and a photo. The fishing spot is meanly concerned with the use case, where an angler is at a watercourse (e.g. lake) and at a given position in the watercourse, where he can add this position as a spot by uploading also the photo of a fish (captured there).

The prototype showing the feasibility of our approach in this work is mainly based on the results of the lab and user trials primarily for those LBS scenarios and carried out for the first prototype with the help of the Angling Community. Lab and user trials took place on the 27th/28th November 2009 in Vienna and 12th/13th December 2009 in Kiel. Further lab tests in November took place at the lab of Center for Usability Research & Engineering in Vienna and the according field trials took place at a fishing lake in Gro-Enzersdorf, near Vienna. Lab tests in December took place at the lab of Leibniz Institute of Marine Sciences in Kiel and the according field trials were conducted at two fishing lakes in Jevenstedt, close to Kiel.

In the first prototype, the PICOS platform is assumed as trustable. However, first analysis helped us to derive concrete requirements for some common points:

- Better guidance and context-sensitive help information (R1),
- Support of direct and simple navigation in the application especially for LBS scenarios on maps (R2),
- Improvement of crucial mobile HCI aspects such as entering data in general. Special cases are manual GPS entering and community related content processing such as watercourse and spots adding etc. (R3),
- Enhancement of synchronous group awareness and synchronous group as well as personal communication (R4),
- Improvement of the privacy concerns related to R1-R4 (R5), and
- Improvement of the privacy concerns related to the architectural design of the whole application (on the server and on the client sides). So the end-users are able themselves to adapt the used server components e.g. for communication, awareness or data sharing according to their privacy needs from the same application at runtime without restarting the system or client application (R6).

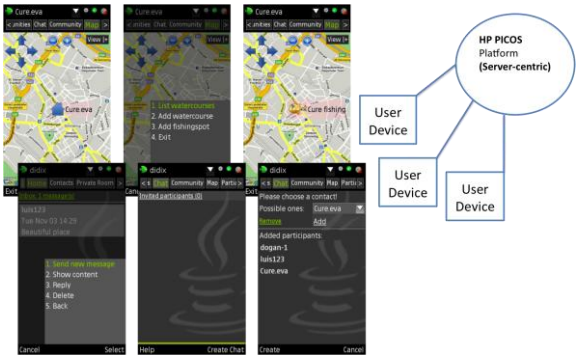


Figure 1: PICOS client screen shots (on the left) and overall architecture of the 1st Prototype (on the right)

3.2. Detailed requirements based on the derived requirements and adapted scenarios

After analysing the usage scenarios from the PICOS project, we defined more detailed sub-requirements, which are presented in the following:

- Context Sensitive Help (R1.1)

Copyright © 2010, Infonomics Society

- Better Guidance at the First Start of the Application (R1.2)
- Better Guidance when using the Application Unregistered (R1.3)
- Better Error Reporting and Feedback (R1.4)
- Better Navigation Inside the Application (R2.1)
- Better Navigation on Maps (R2.2)
- Better entering of location data (R3.1)
- Minimisation of Data Entering (R3.2)
- Awareness related to social interaction (R4.1)
- Awareness on the Map (R4.2)
- Fine grained privacy settings (R5.1)
- Blurring of fishing spots (R5.2)
- Encryption (R5.3)

3.3. Similar requirements based on CURE-related projects

The requirements gathered based on PICOS are similar to requirements and needs we identified in previous works related to the CURE platform [6][21].

In summary, the work described in [5] reports that the usage policy was a way to enable some features in CURE (such as activity indicators and collection of log data) since the University privacy supervisors followed the development process and required users consent to use of their private data. Furthermore, the work reports on the privacy needs of the CURE end users, identified while using the platform for many years in different collaborative learning scenarios.

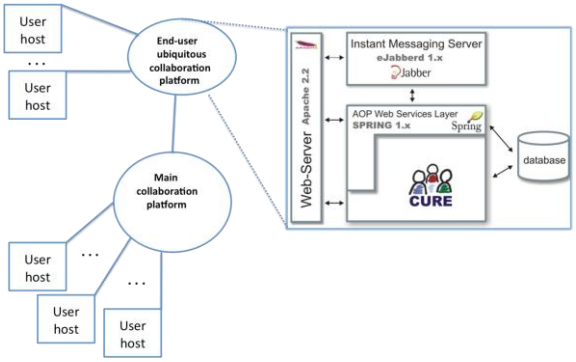


Figure 2: System architecture of the retrofitted CURE system

In general, some users did not appreciate the persistent functionality of the chat and some awareness functions. Some students communicated to their instructors, that they did not contribute because they did not wanted to be monitored. Students remained uncertain if the system provided back doors to the instructors or super users (e.g. administrators) in general. It was observed, that some students restricted their interactions in the collaborative environments to the minimum (i.e. needed interaction for deliverables or for communication triggered by the instructors). Some of them used in parallel their own collaboration tools (e.g. wikis or forums) instead of the provided collaborative environment hosted on the CURE learning platform.

However, this was observed mostly with computer science students, who seem to be more sensible to privacy and trust concerns. Some end users did not care at all. While instructors in general were interested in supervising the different learning settings and were happy about awareness in the system, they had inhibitions to design their learning environments at the beginning of the courses (i.e. because of being afraid to negatively affect the system or make mistakes etc.). For meeting these needs, a decentralised group-centric architecture was developed that is depicted in 2. It introduced an approach for tailoring collaboration according privacy needs. In contrast to traditional collaboration environments with central hosting, this approach gave each group the whole responsibility of hosting the collaboration environment by using their own technical means.

CURE was used to evaluate potentials of SocialTV in the light of privacy, too. The concrete requirements we gathered through scenario-based interviews with experienced fourteen people from the IT Security and Media Science Departments. Three main requirements categories were identified: (1) Support for synchronous watching of live streams) as well as asynchronous SocialTV-related interaction (e.g. in forums) of the users (co-located and geography distributed) with needed group and activity awareness (R1), (2) Provision of an integrated environment instead of using different ones or various devices for the different the different components such as media content component and social interaction components (e.g. chat, forums) etc. (R2), and (3) Privacy-respecting social interaction and data sharing (R3).

For meeting these requirements, we extended the CURE system to fulfil the R1-R3. The resulting system architecture and the resulting Web-based prototype UI are respectively depicted in Figure 3 and Figure 4.

In the SocialTV study in cooperation with the Media Science Faculty at our university, we enabled eighteen people of various ages and expertise to use centralised as well as decentralised (group-centric) solutions for SocialTV by means of a Web-based software prototype. The description of the experimental setting as well as of ist results can be found in [3]. The interesting thing in the context of this work is that, the same system (here CURE) was used in two different social settings: (1) eLearning settings described in [5] and (2) SocialTV (leisure) setting described in [3]. In the case of the eLearning context, the observed people were cautious in general than in the leisure context. In the SocialTV context the lab tests (observations by using usability testing software) and the field tests (observation in a real room with lean-back situation), some of the participants confirmed their willing in disclosing data for added value services and others wanted to have the choice to adapt the collection of such data disclosure at the level of the Web-based application. Since the SocialTV strictly allowed for centralisation or decentralisation, there was no flexible way in doing this with the same application. Therefore, we recognised the

Copyright © 2010, Infonomics Society

need for adapting the distributed architecture for privacy needs at the level of a main social interaction context (SocialTV here) within the same client application. This means that users wanted in separating their awareness and communication facilities without leaving watching TV with others. Other users wanted to leave watching but at the same time remain in communication with others and receive group awareness to these persons (e.g. online status). In the latter case, the main social interaction context is the communication and awareness not sharing the content.

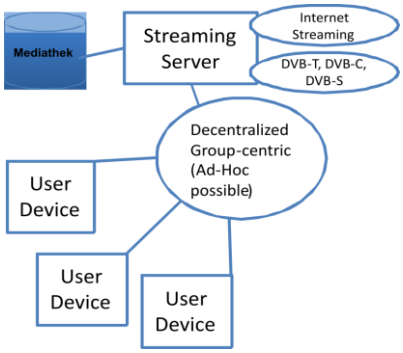


Figure 3: Overall architecture of the prototype

Based on this, we extend R6 to include the following end-user tailorability requirement: The system architecture used for supporting the social interaction allows for separating the different components to be used in a given social setting context. So the end-users are able themselves to adapt the used server components e.g. for communication, awareness or data sharing according to their privacy needs from the same application at runtime without restarting the system or client application.

4. Approach

In order to fulfil the requirements stated above, we built an iPhone-based prototype for the same angling community as the PICOS project worked with. Further, we developed an architecture which allows different levels of decentralisation. The choice to use the iPhone as the mobile device is due to the available resources at our institutions.

Figure 4: Synchronous interaction by watching live streams in CURE

Apple offers an iPhone University Program that gives students the possibility to develop iPhone applications without additional costs. We decided to join this program primarily because of Apples strong HCI guidelines. Further, according to our evaluation, the provided features in the iPhone SDK ease the development of mobile collaborative applications. However, any SDK on any other mobile platform supporting our suggested concepts would be applicable for validating our approach.

In the following section we will describe the fulfilling of the requirements we gathered in detail. Because the six requirements, we put our focus on (R1-R6), are mainly non-functional, we needed to define and implement secondary functional requirements. The use cases will also be described in the related context.

Figure 5: First start of the application

4.1.Context Sensitive Help and Better Guidance (R1)

The first requirement we derived from the results of the lab- and field-tests of the PICOS prototype was to provide a better guidance and context sensitive help(R1). In this section we will discus, what mechanisms we implemented to guide the user while using the application and how we provided a context sensitive help. By starting the iAngle client, the application recognises if it is called for the first time (R1.2). In this case, a message box will show up, asking the user to register. However, the user has the option to ignore this request and is able to navigate in the application. It is still possible to use a reduced amount of its functionality. For example the user can search for different fishes and other species in the species database or he can browse the list of shared watercourses and their spots, which had already been added by other users. If the user decides to register on the first start, he/she tabs on the button labelled Register now and the registration view will automatically show up, allowing the user to enter a desired username, a password and his email address (see Figure 5). If he/she first ignored the request but decides to register later, it is possible to get to the same form, by tabbing Create new account in the preferences tab.

4.2. Support of Direct and Simple Navigation (R2)

The second requirement we defined was to make the navigation in the whole application and especially the navigation on the map, as direct and simple as possible.

Copyright © 2010, Infonomics Society

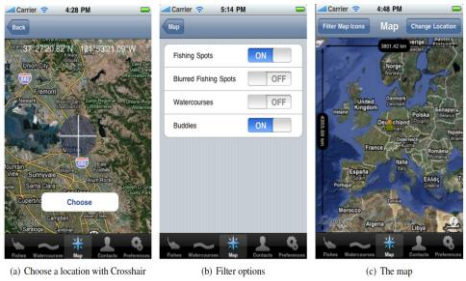
The general navigation on the map is naturally simple in iPhone applications, but some additional features needed to be added to support the LBS scenarios the application should support. Further we designed the general navigation inside of the application as simple as possible. In order to provide this, we implemented a tab bar, to access the main functionalities of the application, like Fishes, Watercourses, Map and Contacts as well as the preferences view. With this structure, it is possible to access all functionalities of the application with a maximum of three tabs. By tabbing the map tab for the first time, a prompt will show up, asking the user to enter his location. He can chose to enter this manually, entering coordinates, set a position with a crosshair, or to let the device determine the own position automatically. Once a position is chosen, the map is displayed (R2), where different points of interest are presented (i.e. fishes for spots and human figures for buddies). Which icons are shown on the map can be configured in the filter screen (see Figure 6(b)) (R2). The user can navigate on the map by simply dragging his/her finger in the desired direction. To zoom in or zoom out, the user has to perform a pinch resp. spread gesture (well-known gesture by the iPhone for zooming in and out pictures) (R2). On the top and left side of the map, the scale of the currently viewed area is shown (see Figure 6(c)) (R2). If the user is viewing the details of a user on his/her contact list, it is possible to directly navigate on the map to the position of the contact.

4.3. Improvement of Crucial HCI Aspects (R3)

The third requirement was to improve crucial HCI aspects in the application. This is of particularly importance when using mobile devices because of the limited capabilities to enter data and the small screen size. Because of this we needed to minimise the entering of data in general and if possible avoid the entering of text by the integrated software keyboard (R3.2). Using the iPhone as development device and therefore sticking to the Apple Human Interface Guidelines [20] helped us to fulfil this requirement. The following section will show how we simplified the entering of data and other HCI aspects. By tabbing on the Map tab, the user is asked for his position. He can chose to set this automatically, by allowing the application to use the integrated GPS device. Also, he can enter it manually, either by choosing a location with a crosshair (see Figure 6(a)) or by entering coordinates. If the user decides to enter the coordinates manually, he/she can chose it by utilising the picker-wheels to pick the latitude in a sexagesimal representation (R3.1)(see Figure 7(a)). Alternatively it is possible to use the built-in software keyboard to set the latitude in decimal representation (R3.1). To provide the user with a consistent interface, setting the latitude in one of the representations automatically changes the latitude in the other one (R3.1). After the user has chosen the latitude he can continue by selecting the longitude so as to describe a complete coordinate. The user is also able to cancel the whole navigation process by tabbing on the Map button and going back to the map without changing the coordinate. Once, a GPS based or manual coordinate is chosen, the Map is displayed. Another feature we implemented is the non-case-sensitive search for contacts. The search is performed in real time during the entering of the letters and will directly show all contacts, which contain the search pattern (R3.2). By doing this, it is not necessary to enter the whole name of the contact (in most cases), and buddies can be found and added to the contact list faster. If the user exits the application, all current settings will be saved (R3.2). The login data, as well as the login status will be remembered, so the user needs not to enter this data on every start of the application. Also the last position on the map will be saved, so that it is not necessary to enter it again after exiting the application. If the application was set to use automatic location, it will continue to use this location too. To optimise the entering of text using the chat feature of the application, the iPhone supports an auto-correction and auto-complete function (R3.2). Because of the fact, that this feature is not accepted by all users in the same way, it is possible to switch it on or off in the application preferences.

4.4. Synchronous Group Awareness and Communication (R4)

The fourth requirement was to enhance the synchronous group awareness and the personal communication. For this we implemented several well known mechanisms to provide message exchange and chat as well as a personal contact list using the XMPPframework [31] and eJabberd. In the contacts tab the user can see his contact list. The contacts are neatly arranged and sorted by online status and name (R4.1). Additionally the names of online contacts are written in green colour (R4.1).



(a) Choose a location with Crosshair (b) Filter options (c) The map

Figure 6: Map navigation

By clicking the + on the upper right corner the Search View will show up. The user can search for new contacts with the help of non case-sensitive search (see Figure 7(b)).

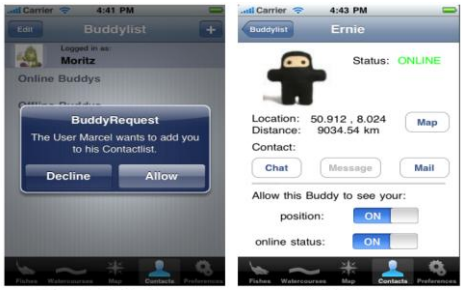
The search is performed in real time during the entering of the letters and will directly show all contacts, which contain the search pattern. After this, the user can choose a contact and add it to his contact list. This process triggers a confirmation request on the device of the added contact (see Figure 8(a)).

Copyright © 2010, Infonomics Society

(a) The Latitude Picker (b) Search for Contacts

Figure 7: Minimization of data input

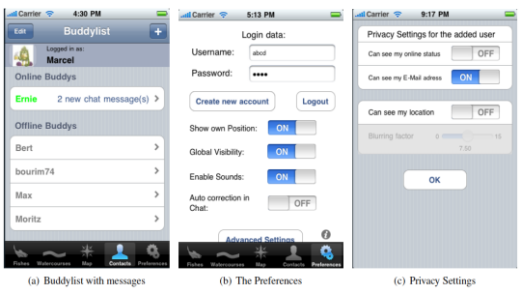
A message box will show up, asking if he/she wants to confirm the request. (R4.1). If the added contact confirms, both involved users can see each other on the buddy list, which enables them to see their presence status (R4.1). The request is sent once in order to not disrupt invitees. If the invitee is offline at the time of the request, the request will show up the next time he/she goes online. By choosing a contact from the buddy list, the detailed buddy screen is shown (R4.1) (see Figure 8(b)). There, the user can see the current location of the corresponding buddy (if he/she allowed the user to see it) and the distance to this location (R4.1, R4.2). Moreover, he/she can see the position of the buddy on the map by clicking on the Map button in this screen (R4.1, R4.2). Further communication is accessible through the Chat, Mail, and Message buttons (4.1).



(a) Received a friendrequest (b) The Buddydetail view

Figure 8: Selected GUI masks of the prototype

To provide a good awareness of incoming messages, a counter will increase if a message is received and a text in small blue letters will show up behind the username (R4.1), indicating that there are new messages from this contact (see Figure 9(a)).



(a) Buddylist with messages (b) The Preferences (c) Privacy Settings

Figure 9: Awareness and Privacy Settings

If another tab than the Contacts tab is selected a red badge with the number of unread messages will show up on the Contacts tab-icon (R4.5). Additionally it is possible to play a sound on incoming messages (R4, which can be enabled in the Preferences tab (see Figure 9(b)). When a user adds a new watercourse or fishing spot to the database, the data is automatically published all other clients, so that they are always aware of the newest watercourses and fishing spots (R4.1).

4.5. Improvement of Privacy and Security in R1-R4 (R5)

One of the most important requirements we gathered was the improvement of privacy and security in the scenarios stated above, and especially in the context of the requirements R1-R4. This section will show the subsidiary requirements we derived and how we implemented them.

4.5.1 Blurring of Location Data (R5.1). The spots on the map can either be precise GPS location data or they can be blurred locations. In the filter options there is also the option to only show/hide blurred fishing spots, if the user is only interested in precise data. If a user wants to add a blurred fishing spot he/she is able to choose how strong the spot should be blurred (in a range from 1 to 15 km). The blurred spot is represented as circle area (region) on the map, including the precise fishing spot at an aleatoric position in the region. This position is calculated according to a blurring algorithm, which increases (or decreases) the latitude and longitude of the exact position for a random value, depending on the blurring factor. These calculations are performed locally on the users' device while creating the new fishing spot. The new latitude, longitude as well as the corresponding blurring factor are then sent to the server and stored there. If another user wants to view the fishing spot on the map, his iPhone downloads the coordinates of the centre of the circle and the corresponding blurring factor from the server. The region size can then easily be calculated using the blurring factor and shown on the map.

4.5.2 Encrypted Communication (R5.2). To secure the personal communication data and prevent interception, all messages are encrypted. XMPP naturally supports TLS encryption to secure client-server communication. In cases where a small group is using a trusted server this should be sufficient. Because of the fact that the messages and also location data could be read in plaintext on the server side, and also due to the fact, that server to server communication is not encrypted by default, it is necessary to use additional encryption mechanisms. In this work we implemented an additional public key infrastructure to provide secure client-to-client communication and location publishing. Therefore the client generates and publishes a public key like described in the XMPP extension XEP-0189 [24], which enables other users to send encrypted messages or to encrypt the location data, so that only the contact who is supposed to read it can decrypt it.

Copyright © 2010, Infonomics Society

4.5.3 Individual Privacy Settings (R5.3). In order to provide privacy respecting group awareness we provide fine-grained privacy settings. These settings include hiding one's online status towards certain or all users on the "buddy list" (see Figure 9(a)) (invisible mode). This is done in order to provide a fair amount of security to the personal user data, especially the own location, which can easily be used to get clues about the real identity of a person. All personal data, besides the pseudonym is only revealed to contacts, which specially got the permission from the user to see them. For instance the own location is only published to a contact in the own contact list, if the user set the corresponding switch in the contacts detail view to on. (see Figure 8(b)) The same applies to the revelation of the own online status. By default both switches are set to off, to provide a maximum of privacy, which was favoured by the representatives of the angling community.

4.5.4. Other Mechanisms. It is possible to use some functionality of the application, even when not registered, or just not logged in. In this case, it is possible to browse the existing watercourses and fishing spots, as well as see them on the map. Showing buddies on the map is disabled by default to allow anonymous navigation (i.e. navigation without authentication). But even when logged in, it is possible to be completely invisible. To do this, there are two switches in the preferences, enabling the user to prevent location publishing and/or the online status. By setting the Global Visibility switch to off it appears to other users as if the own device logged out. If another user adds the user to the contact list, a message box will show up, asking if he/she wants to confirm the request. If the request was confirmed, an acceptance is shown to the inviting user. After confirming the request, a spezial privacy settings view will show up (See Figure 9(c). In this view, the user can determine if he will allow the new contact to see his/her online status, location and email address. The same view will show up on the invitee's device after the friend request is sent. The client supports free selection of the current location, which makes it possible lie about one's current location by sending false coordinates.

4.6. Enhancement of Privacy by Enabling Tailorability of the Distributed Architecture (R6)

We decided to use a group centric architecture with two lightweight servers. These servers can be easily installed on a home computer or other hardware from the users themselves, if the user doesnt trust the owner of the public server and wants to have full control over his user data (R6). Additionally, it allows the formation of sub-communities. The sub-communities can restrict access to their data so that only members of the community can access it. The two servers we used in iAngle are the eJabbered Server for communication and location publishing, and a retrofitted CURE Server [21], mainly containing the database of watercourses and fishing spots, but additionally involved in the registration process, to support unlinkabilty of communication data and user identities.

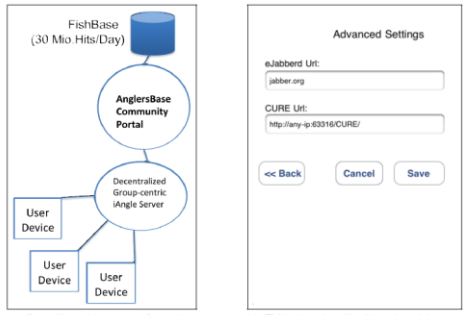
Since the retrofitted CURE supports ubiquiitousity in form of decentralised group-centric servers, we developed many decentralised solutions allowing to fulfil R6. The iAngle client can be set to use an eJabberd server locally installed by the users themselves (members building a trust- worthy sub community) [6]. However, the central community AnglersBase is still being developed in PICOS and there is at the moment no possibility to share data at a more global level as described in [6]. Currently, the iAngle server is playing the role of the AnglersBase. Another important aspect in our approach is the fact, that no sensitive user information is stored on a server.(R6) The data like location information, email etc. is stored on the own mobile device and only send to authorised contacts in an encrypted message. We might have a slightly increased communication traffic compared to other architectures, where the data is uploaded to a server, but the user possesses full control about his personal data all the time. The pseudonyms used for entering the iAngle server, where watercourses, and precise as well as blurred spots are stored, are totally different from the eJabberd accounts. With this, the observability and linkability of the users is made difficult especially by separation the communication as well as awareness functionality from the collaborative LBS scenarios (R6).

We implemented our approach for supporting a mobile Angling Community with privacy and collaboration needs related to location-based services. We built an iPhone based prototype and we developed an architecture which allows for different levels of centralisation and decentralisation. The resulting architecture is depicted in Figure 10(a). Here we describe how the end-user can adjust (adapt) the distributed architecture in order to reach his/her privacy needs related to communication and awareness on the one hand, as well as shared data expectations on the other hand. We achieve this goal by enabling the user to configure the application architecture individually on the client-side (see Figure 10(b)). Depending on the user's configuration, there are several architectures possible. In the following we will introduce three possibilities. Figure 11(a) shows client-server architecture with the server being central for all users. Communication, awareness and data sharing is global in this configuration and therefore privacy is also a global concern. The typical user of this configuration has no or just a minor interest in privacy. This model is also widely used by today's social networks.

The next possibility is presented in Figure 11(b). In this case all users share their data globally while communication and awareness aspects are handled group-centric. This model allows to share contents publicly while respecting the user's privacy.

Finally, the third configuration (see Fig. 11(c)) is totally group-centric. Data is shared on the group-level and furthermore communication and awareness are also handled on the group-level.

Copyright © 2010, Infonomics Society



(a) Overall architecture of the iAngle Application (client and server side)

(b) Tailoring the distributed architecture by setting the used server components

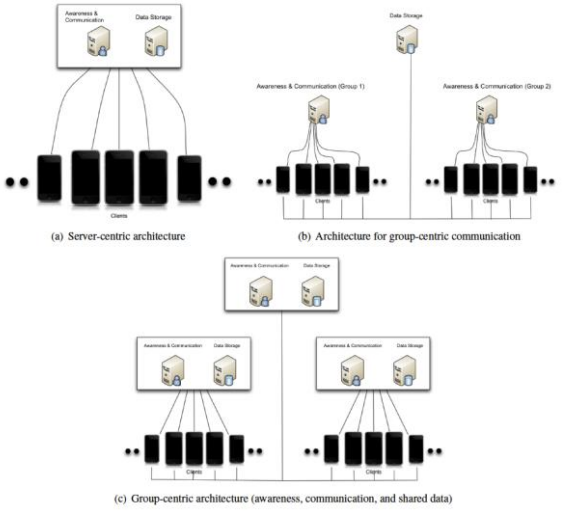
Figure 10: Flexible iAngle architecture

In our case the interoperability is granted by the usage of XMPP with the help of the eJabberd server [12] as a communication and awareness server whereas CURE is used as a shared artefacts server (all CURE instances provide an API for importing and exporting data to other CURE instances). Both the ubiquitous CURE (including an eJabberd sever) [5] as well as further ad-hoc setup eJabberd servers (ca. 12 MB) can be deployed on the the end-users machines and hosted there. The client application in our case creates temporary identities and use them if the client switch for the first time to an eJabberd instance. In the case of CURE we enforce the user to enter the correct credentials through a separate UI. Since we only retrofitted the CURE implementation, the provided implementation details correspond to those described in [5].

5. Evaluation of platform architectures on the basis of iAngle scenarios and discussion

The described approach was tested by involving different end-users from the angling community with different background. So far, we received almost only positive feedback in first evaluations in iAngle and requests for additional functionality. The end-users were able in our usage scenario to download the eJabbed server and to use it. Except from the positive feedback, we also received feedback that required additional improvements such as such easing the finding and entering URLs etc. We have shown with our approach how to adapt distributed architectures according to the user's privacy needs. Even though we implemented our approach for supporting relatively simple collaboration social settings, our approach can be used in a similar fashion with respect to the resources used by the system and which can be hosted in the cloud. This would add an additional dimension of flexibility to the socio-technical system as the user would be able not only to tailor the architecture but also the resources used in order to comply with his/her privacy and security needs. However, in Europe the consumption of cloud services is still cautious due to security and privacy concerns (e.g. industry espionage and customer data). Scandals like the practice of US authorities to subpoena records and data from Swift has lead to a substantiation of this position [13]. As a result, systems which are unalterably Cloud-based, may not be accepted by all users. With our approach however the user himself can decide where to store his data while using Cloud-based communication services or vice versa.

Our approach is regularly used by our group in prototypes we build in different research projects. During performed lab or filed tests we gather a lot of new change requirements (from the respective communities or users) which allows us to improve our approach.



(a) Server-centric architecture

(b) Architecture for group-centric communication

(c) Group-centric architecture (awareness, communication, and shared data)

Figure 11: The three different architecture approaches

6. Conclusions and Future Work

In this paper, we addressed the importance of mobile collaborative applications and the privacy problems which can results due to their frequent server-centric nature. We presented an approach allowing for tailoring the distributed architecture according to their privacy needs by the end-users themselves. The requirements gathering considered three projects from different fields (leisure life, research and professional scenarios) which ensure the generalizability of the provided approach towards an elaboration of a generic platform for arbitrary support of different social settings. We showed the feasibility of our approach by implementing the approach for supporting a mobile angling community with privacy and collaboration needs. Different architectures for global or group-centric communication and awareness as well as shared data become possible. Therefore, end-users are empowered to define their main interaction context (e.g. communication and awareness or data sharing in our case) and tailor the distributed architecture at runtime. Thereby further ad-hoc settings are also supported. Latter shows the possibility to extend our approach to cover other fields such as discussed for cloud computing. The main contribution of these paper consisted of the extension of decentralized group-

Copyright © 2010, Infonomics Society

centric approach suitable for preserving privacy in mobile collaborative settings. The presented approach represents in our opinion a good starting point for studying further research topics in the future which are related to the design of usable privacy-preserving mobile collaborative systems. This is especially true when considering the similarity of the CSCW and HCI research fields which are both based on experimentation and heuristic evaluation. Future work aim at addressing increasing the interoperability possibilities with existing services in the cloud such as using communication channels of existing providers or use personal data stored on different social networks.

7. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2011) under grant agreement n 215056. We thank all PICOS partners who supported this work. Thanks are also due to Fatih Karatas, Philipp Kropp and Julian Dax.

8. References

[1] C. Andersson, J. Camenisch, S. Crane, S. Fischer-Hubner, R. Leenes, S. Pearsorr, J. Pettersson, and D. Sommer. Trust in prime. In Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on, pages 552 –559, 2005.

[2] L. J. Bannon. Customization and tailoring of software systems:thinking about the context of tinkering and tailoring. In Customizing software systems, pages 4–8, 1992.

[3] M. Bourimi, T. Barth, J. Haake, B. Ueberschaer, E. Ganglbauer, and A. C. Garcia. Affine: A lightweight framework for facilitating acceptance of mobile collaborative applications. Submitted to the The 30th International Conference on Distributed Computing Systems 2010.

[4] M. Bourimi, T. Barth, J. M. Haake, B. Uebersch¨ar, and D. Kesdogan. Affine for enforcing earlier consideration of nfrs and human factors when building socio-technical systems following agile methodologies. In Proceedings of the 3rd Human-Centered Software Engineering Conference, Reykjavik, Iceland, 2010.

[5] M. Bourimi, F. Kühnel, J. M. Haake, D. el Diehn I. Abou-Tair, and D. Kesdogan. Tailoring collaboration according privacy needs in real-identity collaborative systems. In CRIWG, pages 110–125, 2009.

[6] M. Bourimi, S. Lukosch, and F. Kuehnel. Leveraging visual tailoring and synchronous awareness in web-based collaborative systems. In J. M. Haake, S. F. Ochoa, and A. Cechich, editors, CRIWG, volume 4715 of Lecture Notes in Computer Science, pages 40–55. Springer, 2007.

[7] M. Boyle and S. Greenberg. The language of privacy: Learning from video media space analysis and design. ACM Trans. Comput.-Hum. Interact., 12(2):328–370, 2005.

[8] M. Boyle, C. Neustaedter, and S. Greenberg. Privacy factors in video-based media spaces. In S. Harrision, editor, n Media Space: 20+ Years of Mediated Life, pages 99–124. Springer, 2008.

[9] M. Boyle, C. Neustaedter, and S. Greenberg. Privacy Factors in Video-based Media Spaces, pages 97–122. Computer Supported Cooperative Work Series. Springer, 2009.

[10] L. Cranor and S. Garfinkel. Security and Usability. O'Reilly Media, Inc., 2005.

[11] P. Dourish. Culture and control in a media space. In ECSCW'93: Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work, pages 125–137, Norwell, MA, USA, 1993. Kluwer Academic Publishers.

[12] EJabberd. eJaberd, the Erlang Jabber/XMPP daemon. http://www.ejabberd.im/, January 2010.

[13] EUROPEAN PARLIAMENT. Swift data sharing - a look at its slow legislative death. http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+IMPRESS+20100219STO69260+0+DOC+XML+V0//EN, February 2011.

[14] A. Fernandez, J. M. Haake, and A. Goldberg. Tailoring group work. In CRIWG, pages 232–244, 2002.

[15] R. Gross, A. Acquisti, and H. J. Heinz, III. Information revelation and privacy in online social networks. In WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pages 71–80, New York, NY, USA, 2005. ACM.

[16] T. Gross and M. Koch. Computer-Supported Cooperative Work (CSCW). Oldenburg, 2007.

[17] J. M. Haake, T. Sch¨ummer, A. Haake, M. Bourimi, and B. Landgraf. Supporting flexible collaborative distance learning in the cure platform. volume 1, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[18] A. Henderson. Tailoring mechanisms in three research technologies. In Workshop on Tailorable Groupware: Issues, Methods, and Architectures at the ACM Group'97 conference organized by Mørch, Anders; Stiemerling, Oliver; Wulf, Volker., 1997.

[19] J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pages 177–189, New York, NY, USA, 2004. ACM.

[20] Apple Inc. User interface guidelines. urlhttp://developer.apple.com/iphone/library/documentation/. [Online; accessed 05-February-2011].

[21] S. Lukosch and M. Bourimi. Towards an enhanced adaptability and usability of web-based collaborative systems. International Journal of Cooperative Information Systems, Special Issue on 'Design, Implementation of Groupware, pages 467–494, 2008.

[22] A. Mørch. Three levels of end-user tailoring: customization, integration, and extension. MIT Press, pages 51–76, 1997.

[23] L. Palen and P. Dourish. Unpacking "privacy" for a networked world. In CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 129–136, New York, NY, USA, 2003. ACM Press.

[24] I. P. Peter Saint-Andre, Dirk Meyer. Xmpp extension 189. http://xmpp.org/extensions/xep-0189.html, 2010. [Online; accessed 05-February-2011].

[25] PICOS EU Project Homepage. Privacy and identity management for community services. http://www.picos-project.eu, 2010. (Access date 9 February 2011).

[26] PICOS TEAM. PICOS Public Deliverables Site. http://picos-project.eu/Public-Deliverables.29.0.html, 2010. (Access date 9 April 2010).

[27] T. Schümmer. A Pattern Approach for End-User Centered Groupware Development. Schriften zu Kooperations- und Mediensystemen - Band 3. JOSEF EUL VERLAG GmbH, Lohmar - K¨oln, Aug. 2005.

[28] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Shneiderman, 5 edition, March 2009.

[29] R. Slagter. Dynamic groupware services, modular design of tailorable groupware. PhD thesis, University of Twente, http://asna.ewi.utwente.nl/research/Ph.D.

[30] L. A. Suchman. Plans and Situated Actions: The Problem of Human-Machine Communication (Learning in Doing: Social, Cognitive and Computational Perspectives). Cambridge University Press, 2 edition, December 1987.

[31] xmppframework. XMPP Framework for Cocoa. http://code.google.com/p/xmppframework/. [Online; accessed 05-Feb-2011].

Copyright © 2010, Infonomics Society