Air-Writing: A Platform for Scalable, Privacy-Preserving, Spatial Group Messaging

Rene Mayrhofer
Upper Austria University of Applied Sciences
rene.mayrhofer@fh-hagenberg.at

Alexander Sommer
University of Vienna
alexander.sommer@gmail.com

Sinan Saral
University of Vienna
sinan.saral@gmail.com

ABSTRACT

Spatial messaging is a direct extension to text and other multi-media messaging services that have become highly popular with the current pervasiveness of mobile communication. It offers benefits especially to mobile computing, providing localised and therefore potentially more appropriate delivery of nearly arbitrary content. Location is one of the most interesting attributes that can be added to messages in current applications, including gaming, social networking, or advertising services. However, location is also highly critical in terms of privacy. If a spatial messaging platform could collect the location traces of all its users, detailed profiling would be possible – and, considering commercial value of such profiles, likely. In this paper, we present *Air-Writing*, an approach to spatial messaging that fully preserves user privacy while offering global scalability, different client interface options, and flexibility in terms of application areas. We contribute both an architecture and a specific implementation of an attribute based messaging platform with special support for spatial messaging and rich clients for J2ME, Google Android, and Apple iPhone. The centralised client/server approach utilises groups for anonymous message retrieval and client caching and filtering as well as randomised queries for obscuring traces. An initial user study with 20 users shows that the overall concept is easily understandable and that it seems useful to end-users. An analysis of real-world and simulated location traces shows that user privacy can be ensured, but with a trade-off between privacy protection and consumed network resources.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; H.4.3 [**Information Systems Applications**]: Communications Applications

Keywords

spatial group communication, location privacy

iiWAS2010 8-10 November, 2010, Paris, France

1. INTRODUCTION

Textual messaging is one of the most popular applications on mobile phones [2] and still more widely used than the "mobile web" [1]. Potential reasons are a lack of cheap data flat rates in cellular networks until very recently and – maybe more importantly – limited user interface capabilities on current mobile phones. Small screens and limited text input are sufficient for the short messaging service (SMS), and both users and services have become accustomed to the surrounding limitations (limited text size, non-real-time, no guaranteed delivery, partial service outages due to network overload, etc.). Thus, textual messaging proves effective and popular in many use cases.

We suggest that textual messaging can be significantly improved by adding various mixture-sets of attributes like the spatial or group [6] component. Previous prototypes have already shown encouraging results in gaming [3], meeting scenarios [11], and grass-roots group communication [19]. In this paper, we contribute a specific platform design and implementation for *attribute based messaging* using mobile devices. Our design explicitly aims at global scale deployment, guaranteeing user privacy even though messages can be localised with full accuracy, and dealing with the inherently limited user interfaces on mobile phones. Potential application areas are manifold; we initially aim at mobile gaming, building local communities and localised, personalised advertisement.

Our approach is to use centralised storage for actual messages and rich clients capable of safeguarding their users' privacy in addition to handling the complete user interface. Web clients are supported for interaction with "remote" places, but may not provide the same privacy guarantees. There are several advantages of such a centralised approach — the most important is the pragmatic reason that web server based architectures are well understood and supported by a wide range of potential clients and hosting platforms. Although peer-to-peer (P2P) based approaches seem more compelling in many scenarios (e.g. infrastructureless interaction) and may potentially offer better scalability, practical experience shows that client/server architectures (currently) often provide better performance and user experience. Global scalability can be achieved using already established content distribution networks (CDNs), and we therefore design our architecture to be applicable to such static content distribution. Our suggested client/server protocol is based on standard HTTP connections (e.g. over UMTS) and has been explicitly designed to withstand attacks against user privacy on the level of telecommunication

and Internet providers. We reach this objective with three design principles: a) Clients never transmit their exact location to the server. Instead, they query and cache group messages for a larger area and filter locally. b) Queries do not contain any unique identifiers (besides the source IP address, which may optionally be obscured). Instead, the underlying messaging concept uses groups comparable to virtual blackboards. Users are by default anonymous and may only optionally identify themselves for posting messages or convenience features. c) Queries are subject to randomisation in time and space to impede statistical attacks.

The present paper makes two contributions: to present this architecture, and to study its usability and privacy on a specific implementation and a first mobile game. 20 users participated in a competitive scavenger hunt in the inner districts of Vienna. Their exact location traces and interactions with their mobile clients were recorded and used to suggest improvements for the prototypical user interface and to simulate how effective attacks on the communication link would be in violating user privacy.

In sections 2 and 3, we start with a brief discussion of related work and how we use the two terms "attribute based messaging" and "spatial messaging" in the scope of this paper. Section 4 then defines the overall architecture and protocol between a central server and distributed, mobile clients and therefore the main scientific contribution. Both the protocol and filtering and caching methods implemented by mobile clients are responsible for safeguarding user privacy and message security, which are discussed in more detail in sections 4.1 and 4.2. A specific implementation of this architecture, *Air-Writing*, is briefly presented in section 5 and is online at `http://airwriting.com` for global public use. In section 6, we present an initial analysis of its usability (section 6.2) and the effectiveness of our privacy safeguards against some statistical attacks (section 6.3).

2. RELATED WORK

In an earlier system, interest in spatial messaging was low in practical experiments [5], but increasing[1] [2]. Most of the location based platforms like GeoNotes [18], Plazes[3] and JotYou[4] have similar functionalities. One can read and write location based messages and/or see (meet) friends who are nearby. Air-Writing attempts to go beyond these basic applications and provide a sustainable [8] and privacy aware system (as explained in section 3).

Privacy.

Various approaches have so far been presented for protecting user privacy in spatial messaging systems. Kido, Yanagisawa and Satoh [13] describe a solution using dummies for hiding the current location, but their approach leads to additional network traffic. In Air-Writing, we intend to provide privacy protection without narrowing the possible use cases and with lower overhead.

Gedik and Liu [9] have presented "k-anonymity", which tries to improve the solution proposed by Grunwald and Grueteser [10]. K-anonymity relies on indistinguishability of k persons in a region. Adversaries shall be unable to distinguish who is who and thus the privacy of each individual is protected. In k-anonymity, the level of privacy is defined by the number of (for the adversary) indistinguishable subjects in a given region – the higher the better. With its global scope, Air-Writing cannot guarantee a minimum number of users for any given region and time. Therefore, although k-anonymity is a general way of measuring privacy, we cannot directly apply it to Air-Writing in general.

Bowen, Raymond, and Martin [12] described a way of cloaking for hiding the actual location of individuals. Although similar in aim to the privacy concept used in Air-Writing, our approach consumes less resources by using client caching and filtering and therefore requires only one instead of many server queries. John Krumm [14] compares the effectiveness and applicability of such spatial cloaking with different privacy algorithms based on inference attacks, showing that cloaking, among others, is in principle worthwhile for improving privacy. Our selection of privacy measures (client caching and filtering as an extension to cloaking and randomisation in time and space) were inspired by this work.

Scalability.

Another major problem of location based services is the global scope and its impacts on system performance. Nartz, Pomberger, Ferscha, Kolb, Mueller, Haertner, and Haring [16] suggest the use of groups for distributing the computational workload over different computers and a "direct addressing" strategy. Air-Writing also supports groups; they are primarily used for building communities, but are also central to preserving privacy. A variation of direct addressing can be achieved by using Content Distribution Networks [17] and thus providing global scalability.

3. ATTRIBUTE BASED MESSAGING WITH MOBILE DEVICES

Attributes are the main concept for composing messages in Air-Writing. In all messaging systems, the most important attribute is the *text* itself. Also, the *group membership* can be seen as an attribute (for group based systems like newsgroups). Another and currently popular attribute is the *location* (for location based messages). One of the initial aims of Air-Writing is to explore and provide as many different attributes as imaginable regardless of their significance. This is a basis for combinations of attributes, flexible application design, and general exploratory research on messaging applications.

Some of the message attributes currently implemented in Air-Writing are:

group-id (mandatory, user-defined, set on client): ID of the group where messages are sent to or received from. E.g., Judas sends to the "love" group.

text (mandatory, user-defined, set on client): The text of the message. E.g., Judas sends Maria the text "I want to kiss you!".

longitude, latitude (mandatory, sensor-derived, set on client): A message is linked to a specific place with the help of GPS, and therefore is only visible (ready to receive) for

[1] http://www.bdnooz.com/lbsn-location-based-social-networking-links (2009)
[2] ZYB, a social platform, was recently bought by Vodafone, www.gomonews.com/vodafone-acquires-100-of-zyb/ (2008)
[3] http://www.plazes.com/ (2009)
[4] http://www.jotyou.com/ (2009)

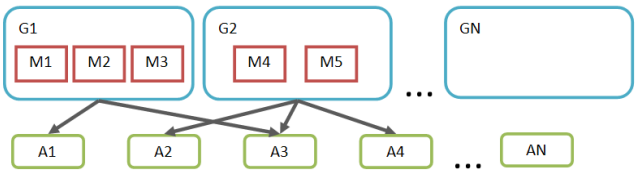


Figure 1: Service Model from an user's point of view

another user where the message was sent from. E.g, user Judas writes a message "I want to kiss you" at "Stephansplatz". If his girlfriend Maria is coming to Stephansplatz, she will receive it. Location is determined by GPS and can not be directly defined in the user interface (note that Judas cannot "cheat" his current position using normal clients, but could fake it with a modified one).

radius (optional, user-defined, set on client): The scope (radius in meters) of a message the client is sending. E.g., Judas wants his message to be visible not only at Stephansplatz Church but really at Stephansplatz location. Therefore he sets the location radius to 30 meters (instead of the default value of 10 meters).

start-date, end-date (optional, user-defined, set on server): Messages will be available a specific time. E.g., Judas wants to send Maria a virtual kiss message on valentine's day. Therefore, he sets the time scope of the message to the 14. February.

amount (optional, user-defined, set on server): Messages can be grabbed from a place n times. After the nth "pick" action (i.e., reading a message), the message will disappear. E.g., Judas only wants to send one virtual kiss. He sets the pick value to 1 of his message. When Maria receives the kiss message, it will be the only one.

encrypted (optional, user-defined, set on client): Messages can be locked by textual passwords. E.g., Judas adds a special password to the virtual kiss message that only Maria knows. Nobody besides Maria will be able to read the kiss message.

Any non-mandatory attributes can be freely combined as shown in the examples below as long as they are not mutually exclusive. If the client assigns the value, cheating is possible; if the server enforces certain attribute values, it is not. E.g., the location attribute could be faked using a modified client. Possible strategies to avoid or defuse cheating are mentioned in section 4.3. Because single attributes are often not enough or even not useful enough for creating a meaningful message, we also explore potentially useful combinations of such attributes. *Groups* are an intuitive means to define usable attributes and attribute combinations. Different groups can re-use such combinations in different ways and therefore increase the value of a single combination.

Air-Writing follows a top-down approach for realising an attribute based messaging service as shown on figure 1: A combination of attributes forms a group, and such groups hold many messages. The purpose is firstly an attempt for flexibility. If attributes or attribute combinations (and therefore groups which are using them) turn out to be useless, they can simply be ignored without weakening the architecture or implementation. Secondly, it is a call for comparative studies to determine why some combinations fail or are successful. And finally, it provides the capability for managing a vast amount of new message types and messages themselves.

Groups.

For efficiency, flexibility and user-experience reasons, the visibility of messages is regulated by their membership to *groups*. A group consists of the group name (freely definable by the creator, which can be any registered user), the creation date, the attribute mixture set (and if all of these attributes are mandatory for messages to this group), a list of members, messages from users, and membership settings (private or public). Examples of groups are:

Aidtips (attributes: group-id, text, longitude, latitude): Provide additional information for people with disabilities - A handicapped person visits a tourist attraction and suddenly needs a wheelchair accessible toilette.

Amor (attributes: group-id, user-name, text, longitude, latitude, radius, start-date, end-date): A location based contact service - A single is looking for a partner within a radius of 50km on valentine's day.

Scavenger Hunt (attributes: group-id, text, longitude, latitude, encrypted): A location based game - A group of students is playing "scavenger hunt" - to win, they need to unravel some mysteries.

The group "Scavenger Hunt" has been implemented for a first game and is described in more detail in section 6.1. This particular group does not require to set all attributes; e.g. the "encrypted" attribute in this group is not mandatory and therefore, messages to this group can consist of locked (encrypted) and unlocked (plain text) messages.

As a final aim, an attribute based architecture should also be able to communicate with other attribute based architectures, whereby those architectures can have different sets of attributes and communication protocols. Therefore, they should be able to handle *incomplete* sets of attributes, also for further extensions. Third parties may use subsets of groups on their own clients or online platforms from our architecture, instead of building their own.

4. MESSAGING ARCHITECTURE AND PROTOCOL

The implementation of Air-Writing consists of an Internet platform and mobile clients. A message in Air-Writing consists, at the minimum, of the following properties: text (content), longitude, latitude, radius and group-id. Optional attributes which each message may currently include (but are not limited to): subject, user-name, start-date, end-date, encrypted, amount. Both the protocol and the reference implementations for server and client parts are extensible and can easily support arbitrary new attributes.

There are several types of request and response messages in our current Air-Writing protocol. The most important ones are for polling the server *Get Messages* and for writing a new message *Send Message*. Figure 2(a) shows a poll request in which the rich mobile client queries the Server for messages in a specific group and a given location (defined by

(a) Get Messages (b) Send Message

Figure 2: Basic Air-Writing protocol for receiving and sending spatial group messages

longitude and latitude). The server responds with all messages from the specified group for this area. Note that there is no user name required for any client to receive messages.

Figure 2(b) illustrates a send message session. The user constructs a message on their mobile device and sends it to the server via a *Send Message* request. In turn, the server responds with an acknowledgement or failure message; the operation can fail if the user does not have sufficient access rights to post to the specified group (if it is private and restricted to its members) or if the user did not provide a user name to a non-anonymous group (the group creator can specify if anonymous postings are allowed). Consequently, the *Send Message* request consists of group-id, text, longitude, latitude, an optional user name and all further optional attributes like pick amount, encryption status, and others.

To summarise, our architecture is defined by:

- Messages are organised in "groups", which are arbitrary strings.
- Groups also act as pseudonyms for querying messages.
- Clients request messages based on group name and their location. For each group there is a distinct query (with randomised delays) to prevent tracing based on the specific set of groups a client is interested in.
- Posting new spatial messages can be done anonymously or identified by user nickname.
- Users don't have to login; by logging in, they optionally get a "profile" with their nick and the set of groups they are interested in and can also use a web portal instead of their local device; also, when starting to use a new mobile device, the profile can be used to configure it with the groups it should use.

4.1 Safeguarding User Privacy

User privacy is, as mentioned before, a critical issue for all spatial messaging services. Air-Writing depends on strong privacy protection as a design principle if users should be expected to use it on a daily basis. The following principles are applied to all parts of the architecture to provide a reasonable compromise between strong privacy guarantees, flexibility, and scalability:

- Queries do not contain any form of identification besides the querying IP address, which can optionally be obscured using standard methods such as tunnelling through the Tor network [7].
- User-configurable query intervals are randomised by up to 50% to make correlation attacks based on exact query times harder.
- Mobile devices do not transmit their exact location but only an area (as defined in a hierarchical lattice for user-configurable query "width") and receive messages for this scope; clients then *locally* filter messages to only show those for the exact location; this not only minimises the number of transmitted messages but also prevents recording exact location traces on the server side; additionally, clients can also randomise these queries by also (randomly) querying neighbouring areas in addition to the one they are currently in.

4.2 Secure Messaging

Besides privacy, Air-Writing also provides secure messaging capabilities. Users may encrypt their messages before sending them to the server when the client platforms support the required encryption algorithms. Any recipient can only see the real content if they know the required password for decryption. Because the messages are encrypted and decrypted on the client platforms, neither potential eavesdroppers (even on the level of Internet service providers) nor Air-Writing server administrators can possibly read the content of secured messages. The decryption key can be generated locally at the client using different means, e.g. from pictures takes with the mobile phone camera (cf. [4]) or from 2D barcodes attached to the location — keys can of course also be privately agreed using any other out-of-band scheme (a number of these are provided, e.g. by OpenUAT [15]).

4.3 Protecting against Cheating

Users or third-party clients may manipulate the value of attributes before communicating with the server. If an attribute value is set by and consequently checked solely on the client, the value can easily be faked because the server has no control over its integrity. In order to avoid cheating, two generic strategies exist: The first is to make cheating useless. If the scope of a message can be set freely, it doesn't make sense to cheat. The second strategy is to restrict the possibilities to cheat. In order to ensure that messages are accessed only by users on a specific location, users have to physically place passwords or barcodes at that location and encrypt all messages with those passwords or barcodes before virtually placing them in Air-Writing on that location.

4.4 Protecting against Denial-of-Service

By allowing anonymous posting and queries based only on group names, denial of service attacks become (slightly) easier than when forcing users to log in. This can be mitigated by making queries cacheable and therefore open to using a content distribution network (such as Akamai). In the current implementation, clients only query "static" HTTP URLs that encode their location area and the group they are interested in. Responses can be regularly pre-generated and pushed into a world-wide distribution network that would be resilient against DoS.

5. IMPLEMENTATION

We have developed a specific implementation of our attribute based messaging architecture "Air-Writing". It is a private, mobile, group based, spatial messaging service and currently implements all attributes listed above. Additionally, the first mobile game application "Scavenger Hunt" has been implemented in the form a of group as explained in section 6.1.

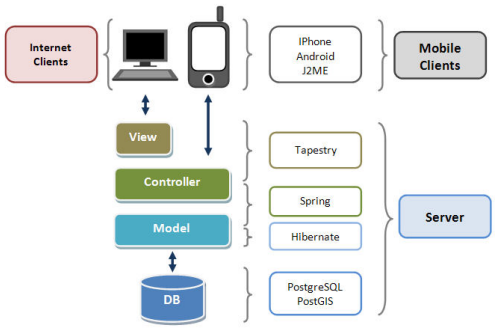


Figure 3: Infrastructure of Air-Writing

5.1 Server Backend

The server backend of Air-Writing is responsible for the following operations:

- The backend acts as a communication port for mobile clients. Mobile clients query the server for messages at their location. They may also send messages to the server. All the information such as user accounts, groups, and messages are managed by the server.
- It provides a portal for web-based access.
- It offers services and aggregated statistical data to third parties (note that even if desirable by some third parties, violation of user privacy is impossible even by server operators due to the design principles explained above).

As shown in figure 3, several open source frameworks were combined to realise the server backend: PostgreSQL[5] is an open source relational database system. Air-Writing makes use of PostgreSQL on the *database layer* with the PostGIS[6] extension. PostGIS enables PostgreSQL to operate on geographic objects. Air-Writing saves the scope of messages as geometries in the database. Hibernate[7] is used to map application objects to relational database tables, and business logic resides in service classes which are managed by the Spring Framework[8]. These service classes form the *controller layer* of the server application and use Hibernate to interact with the database layer. The *view layer* utilises the Tapestry5 Framework[9] which is a new component based Web Framework. Tapestry5 builds upon the Java Servlet API, and thus runs within any servlet container. For Air-Writing, we currently use Tomcat as the servlet container.

5.2 Clients

The client software has been tested on Nokia's E50, N73, N95, Apple's IPhone G3, and on the Android Dev Phone 1. Screenshots of all platforms are shown in figures 4(a), 4(b), and 4(c) in different application states to highlight comparability of all rich client implementations.

[5]http://www.postgresql.org/ (2008)
[6]http://postgis.refractions.net/ (2008)
[7]http://www.hibernate.org/ (2008)
[8]http://www.springframework.org/ (2008)
[9]http://tapestry.apache.org/tapestry5/ (2008)

6. ANALYSIS

For analysing the general Air-Writing architecture and our server and client implementation in particular, we have implemented a first game and evaluated it with 20 initial users. The objective of this analysis is to evaluate if the overall system seems intuitive and usable and if our intended privacy guarantees can be met based on real-world data sets.

6.1 Scavenger Hunt

The rules and challenges of the game "Scavenger Hunt" as shown in table 1 are simple; (1) All participants in the game have to unravel a mystery question as quickly as possible. (2) For that reason, the gameplay owner has to provide one locked (encrypted) and some unlocked messages to the gaming field, an area of approximately one square kilometer. (3) The unlocked messages act as pieces for solving the puzzle question. (4) Additional help messages define the gaming field borders as "border fence is reached".

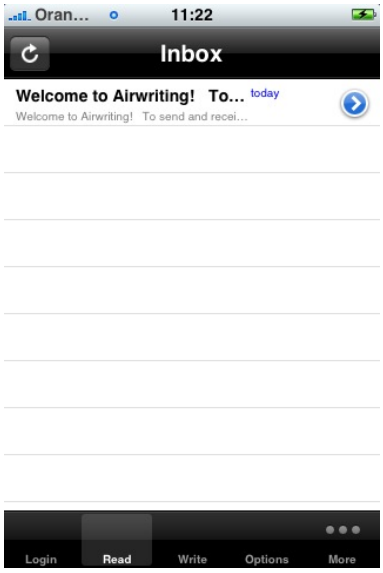
Gameplay Scenario.

Maria and Judas are students at the Technical University Vienna. They both have heard about a location based game close to their university published at Karlsplatz and are interested in playing it. They visit the page with their mobile clients, download, and install the client software. After starting the Air-Writing client software, a group "Scavenger Hunt" becomes visible. They join it and receive 3 messages. Message (1) as the welcome and unlocked message and two locked messages (6a and 6b) they cannot yet read. Both start following the "right wing of the owl" order. After a few meters, they arrive at the stone statue and receive a message (2). Maria says to Judas: "This is easy, the church is straight away!" They keep on moving and receive two messages at the church (3a and 3b). Maria decides to pray further (go to the respective location), and Judas to find the lost soul (another location). Judas is looking around and receives a warning "border fence is reached" and turns back to the church. He takes another route and finds the next message (4b) and with it the first part of the password "hell". At the same time, Maria arrives at Karlsplatz church and also receives her first part of the password "holy" (4a). Maria is told to look for a soul and starts to walk around again. Judas arrives at Naschmarkt and receives his last order and part of the password "fire" (5b). At the same time, Maria finds the soul (5a). Both start running back to the owl. Maria is first, unlocks one of the two locked messages and wins.

6.2 Usability Analysis

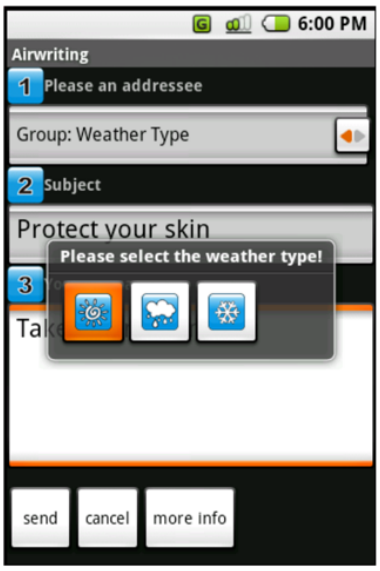
Procedure.

The game Scavenger Hunt was tested with 20 users within a time period of 10 ten days. Each test was designed to take one hour at maximum. At the beginning, the idea and application was explained (5 minutes). Afterwards, users received a pre-configured mobile device (Nokia N95, Nokia E50) with external GPS receivers and started to play at the owl. All important activities were logged on the client and server. After finishing the game, all logs from the client were sent to the server by the test instructor. Finally, users were asked to report their experiences with a short survey of about 20 questions.



(a) Reading a message with the iPhone prototype

(b) Setting the attributes with the J2ME prototype



(c) Writing a message with the Android protoype

Figure 4: Screenshots of various client platforms

Participants.

The participants in this study were sampled from a group of students aged between 20 and 36 years and two older participants aged above 50. All of them are using the Internet and their mobile phone regularly.

Results.

19 of the 20 tested users (95%) would use Air-Writing if it would be available on their mobile phone, even as a permanent background application (88%). The satisfaction factor for the usability of the alpha version is moderate to good. 80% liked the alpha prototype but 29% had sometimes problems in using it. Most testers would use it both for gaming and writing messages (62%), only 8% would use it just for gaming purposes. Another 30% would just write with it. 97% of the users would use it both on the the client and the online platform, the rest just on the client. We also asked the participants to put a monetary value on the security/privacy aspect (which Air-Writing implements). The maximum values were EUR 150 per year, EUR 2 per month, or EUR 0.3 per message. The lowest values were EUR 0.3 per message with no yearly or monthly cost. Some users seemed reluctant to pay for such a service, which suggests advertisement-based models. However, the large differences in the willingness to pay indicate different expectations and should be analysed further.

We also asked our subjects for potential new attributes, attribute mixtures or groups. The most interesting answer was a feature which realises mutually exclusive messages, (therefore, if a user is reading message A on place X it is not possible anymore to read message B on place Y). The preferred times for setting up an individual group were 5 minutes (31%), 10 minutes (54%), 20 minutes (8%), the rest decided for the optional answer that the time depends how funny it is creating a group.

An exploratory analysis of server and client logs showed several interesting points; Firstly, that more than 50% of the users have restarted the Air-Writing client at least once, with a maximum restart count of 4. The reason for restarting the client lay supposedly not in client errors, but due to the exit button being next to the option button of the N73 model, which could easily be pushed by accident. Secondly, that the average time for receiving messages after entering a location was 2.7 (standard deviation 2.7) seconds. Thirdly, that users were switching to different views about 32 times (standard deviation 22.6). Because there are only 7 views (Menu, Login, Write, Read, Options, Setup, Exit) this value is rather high and was not expected. For each view switch, approximately 4.5 clicks are needed. This means that 144 clicks were done on average for view switches by the user. We will take this value as a reference for further improvements.

6.3 Privacy Analysis

The subjects in our initial user study intentionally used different values for *poll interval* and *poll range*. For a good compromise between privacy protection, guaranteed message delivery, and transfer volume, these parameters depend on the number of messages in the area, the speed of the user and how far the messages are distributed. The primary goal is not to let users pass any messages without receiving them while still protecting their privacy.

Every poll request reveals the rough location of the user to the server (unless this query is a randomised "cloaking" query as explained above). During a request, the location information on the client is sent to the server after it has been randomised (or rounded to be able to use Content Distribution Networks[10]). Using shorter poll intervals results

[10]For achieving global scalability, we propose the use of CDNs. Messages can be stored and retrieved under standard HTTP URLs which reflect the rounded location in a lattice with hierarchically structured resolution. Rounding is necessary to keep the number of these URLs manageable.

Sequenced Instructions	
(1) Welcome to Air-Writing at Technical University Vienna! This game will take one hour at maximum and is placed on a one-square kilometer gaming field. You will receive a notification if you cross the borders. Beware, if you cross them more than 3 times, you will loose this game. To win the game, you have to unravel one of two mystery questions which are located at this starting point (so you have to return to your current location at the end) - Lets start, your first quest is simple: find the lord for your next instruction! Hint: Follow the right wing of the owl.	
(2) The Lord: Hello Airwriter, you will come to heaven if you win this game. If not, you will burn in hell - sorry... there is not enough space for everyone in heaven. But don't care, your next quest is simple: say a prayer at the next visible church!	
The Lord	**The Devil**
(3a) The Lord: Good prayer! you will be an angel soon. but first, go to the Karlsplatz church and pray again!	**(3b)** The Devil: stop praying for the lord and don't go to Karlsplatz church but find some lost souls for me. The next is... Hm... Just 3 mins away!
(4a) The Lord: Good prayer! you will win this game as an angel, I will give you the first part of your password: holy - but to win, you have to finish your last quest: find a lost wounded soul in the park!	**(4b)** The Devil: you have found a lost soul. Yeah! Bring it to Naschmarkt! This is the first part of your password: hell.
(5a) The Lord: you found the wounded soul! Bring it back to the owl! The second part of the password is soul. Run back to the owl.	**(5b)** The Devil: My well obeying servant! Thank you for this soul. Your second part of the password is fire. Hurry back to the owl!
(6a) Password: holysoul. Good prayer! You won the game. I will give you wings to fly!	**(6b)** Password: hellfire. Hahahah! you won the game! come and join me in hell!

Table 1: The Scavenger Hunting Game

in sending the location information more often and is therefore disadvantageous in terms of privacy. Longer poll intervals result in better privacy protection, but require the use of wider ranges in order to prevent users from leaving the cached areas and thus missing some messages. The problem with wider ranges is that with every request more messages will be transferred to the client. Mobile devices still have limited resources (memory, CPU, etc.); a large amount of messages may therefore cause performance problems on the client side as well as larger transfer volume on the wireless Internet connection (and may thus lead to higher cost).

In our privacy analysis, we have taken the path recorded in our user study (see figure 5(a)) and simulated multiple instances of using the Air-Writing protocol with different combinations of poll intervals and poll ranges. We also generated two extra paths (shown in figure 5(b)) to analyse indistinguishability of these paths based on the data the server would be able to record in each of the simulations. The resulting measure is comparable to k-anonymity: when all three paths are indistinguishable given the server log, then perfect privacy has been reached in this scenario. In practice, we expect many more concurrent paths and thus better user privacy with even lower poll intervals and ranges. One of these paths leads from north-east down to south-west and the other one from north-west down to south-east. Message positions as used in the game "Scavenger hunt" are also indicated in figures 5(a) and 5(b). In each simulated instance, the aim was for users to receive all of these messages as soon as they pass the respective location.

Results of the simulation are summarised in table 2 for cases in which no messages were missed by the clients. Ideally, every message should be polled only once in order to save bandwidth and client resources. However, because the ranges between two subsequent queries may overlap, some messages may be requested and transferred more than once. Column *Overhead* shows the total number of redundant message transfers during the run, while column *Largest poll* represents the amount of messages transferred during the biggest poll in that run, which is the query where the highest number of messages have been transferred and indicates the maximum resource consumption on the client. Finally, column *Privacy* shows the level of privacy achieved during the run, represented by the number of paths which an attacker can not distinguish from the actual user path, given full access to the server logs or by eavesdropping on the communication. With one real and two simulated paths, a privacy value of *3/3* indicates that looking at the communication logs between the client and the server none of the three paths can be distinguished from the others. Figures 5(c) and 5(d) show the communication log in form of the queried areas in each query and may help to more intuitively optimise parameters in practical settings.

As seen in the first two lines of table 2, using wide ranges for poll requests ensures location privacy, but causes more messages to be transferred within a single poll. The bottom three lines of the table show that we can reduce bandwidth usage by shortening the poll range, but we also lose the privacy protection. This shows a clear trade-off between the privacy level and the required bandwidth or device capacity, which can be configured by each user to account for personal preferences. In the current Air-Writing implementation, group moderators have the responsibility for keeping

the number of messages in a given region below a specified limit. This guarantees that poll responses do not exceed the limited capabilities of mobile devices while privacy can be protected using wide poll ranges.

Range[m]	Poll interval[sec]	Overhead	Largest poll	Privacy
700	1000	0	8	3/3
700	350	19	8	3/3
400	350	6	6	2/3
300	350	2	3	1/3
45	30	14	2	1/3
200	300	0	3	1/3

Table 2: The privacy level depends on poll interval and range, but these also influence client resource usage and bandwidth.

7. CONCLUSION AND FUTURE OUTLOOK

In this paper, we propose to enhance textual messaging services by introducing an abstract, attribute based messaging service architecture. We demonstrate a working implementation with various attributes for three different client device platforms (J2ME, Android, iPhone) with special support for the highly important location attribute, thus supporting spatial messaging applications. Both our architecture and the specific implementations have been designed to protect users' privacy from the start: By polling messages anonymously based on message groups and supported by client caching and filtering as well as active randomisation in time and space, strong privacy protection is enabled by default. Even potential adversaries on the level of mobile Internet service providers or server operators would be unable to distinguish locations traces from different users. The results of an initial user study indicate that the concept of our architecture seems clear and accessible, even if the analysed logs show that there is place for many improvements in terms of user experience.

8. REFERENCES

[1] Mobithinking web page. `http://mobithinking.com/sites/mobithinking.com/files/dotMobi_and_AKQA_Mobile_Usage_and_Attitudes_Study.pdf`, 2008.

[2] Nielsen web page. `http://blog.nielsen.com/nielsenwire/online_mobile/in-us-text-messaging-tops-mobile-phone-calling/`, 2008.

[3] R. Ballagas, S. G. Kratz, J. O. Borchers, E. Yu, S. P. Walz, C. O. Fuhr, L. Hovestadt, and M. Tann. REXplorer: a mobile, pervasive spell-casting game for tourists. In M. B. Rosson and D. J. Gilmore, editors, *Proc. CHI 2007*, pages 1929–1934. ACM, April 2007.

[4] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Secure ad-hoc pairing with biometric: SAfE. In *Proc. IWSSI 2007*, pages 450–456, September 2007.

[5] J. Burrell and G. Gay. E-graffiti: evaluating real-world use of a context-aware system. *Interacting with Computers*, 14(4):301–312, 2002.

[6] S. Counts. Group-based mobile messaging in support of the social side of leisure. *Comput. Supported Coop. Work*, 16(1-2):75–97, 2007.

[7] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. USENIX 2004*, pages 303–320, 2004.

[8] P. Froehlich, R. Simon, E. Muss, A. Stepan, and P. Reichl. Envisioning future mobile spatial applications. In *People and Computers XXI*. British HCI, 2007.

[9] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mob. Comput*, 7(1):1–18, 2008.

[10] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*. USENIX, 2003.

[11] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proc. MobiSys 2005*, pages 177–190. ACM Press, June 2005.

[12] C. L. B. III, D. R. Raymond, and T. L. Martin. Location privacy for users of wireless devices through cloaking. In *HICSS*, page 295. IEEE Computer Society, 2008.

[13] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services. In *ICDE Workshops*, page 1248, 2005.

[14] J. Krumm. Inference attacks on location tracks. In *Proc. Pervasive 2007*, volume 4480 of *LNCS*, pages 127–143. Springer-Verlag, May 2007.

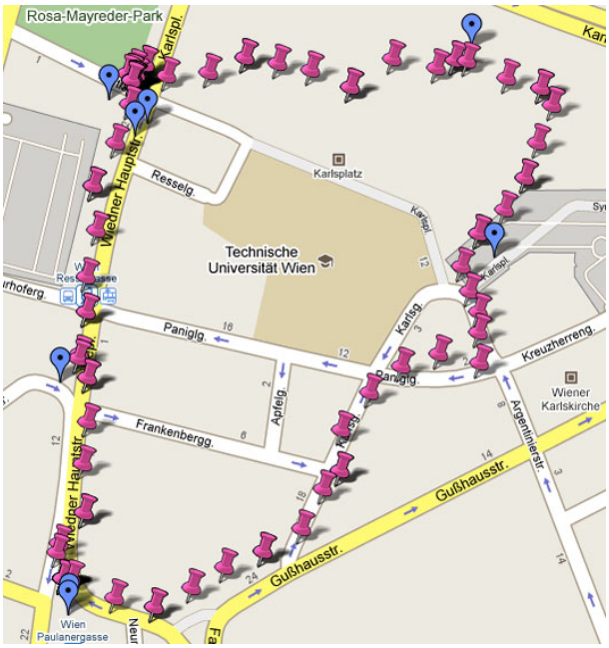
[15] R. Mayrhofer. Towards an open source toolkit for ubiquitous device authentication. In *Workshops Proc. PerCom 2007: 5th IEEE International Conference on Pervasive Computing and Communications*, pages 247–252. IEEE CS Press, March 2007. Track PerSec 2007: 4th IEEE International Workshop on Pervasive Computing and Communication Security.

[16] W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, H. Hörtner, and R. Haring. Addressing concepts for mobile location-based information services. In C. Stephanidis, editor, *Proc. UAHCI 2007*, volume 4555 of *Lecture Notes in Computer Science*, pages 507–516. Springer, July 2007.

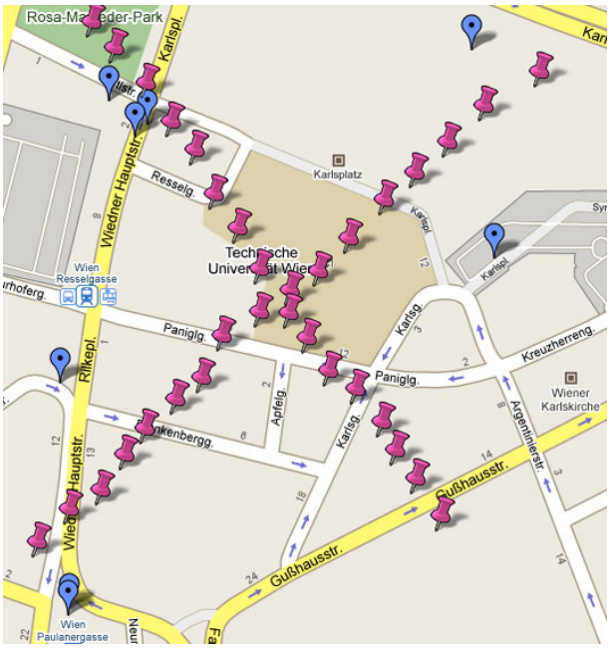
[17] G. Peng. CDN: Content distribution network. *CoRR*, cs.NI/0411069, 2004. informal publication.

[18] P. Persson, F. Espinoza, P. Fagerberg, A. Sandin, and R. Caester. Geonotes: A location-based information system for public spaces. In *Readings in Social Navigation of Information Space*. 2002.

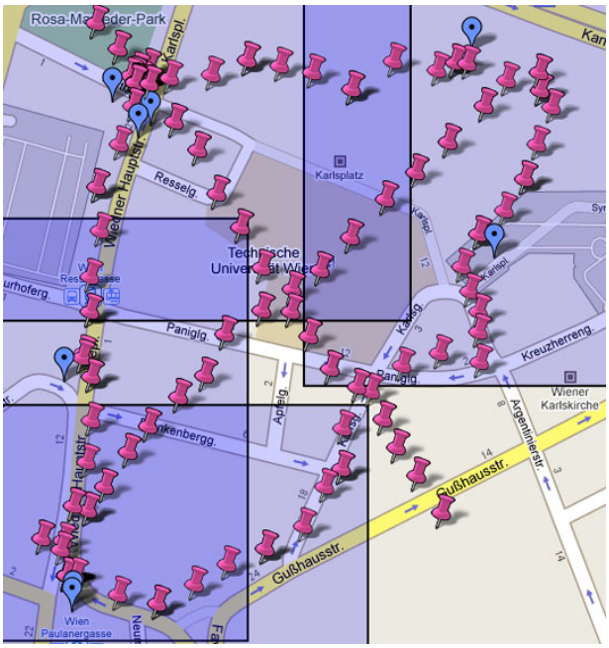
[19] H. A. Rahemtulla, M. Haklay, and P. A. Longley. A mobile spatial messaging service for a grassroots environmental network. *J. Locat. Based Serv.*, 2(2):122–152, 2008.



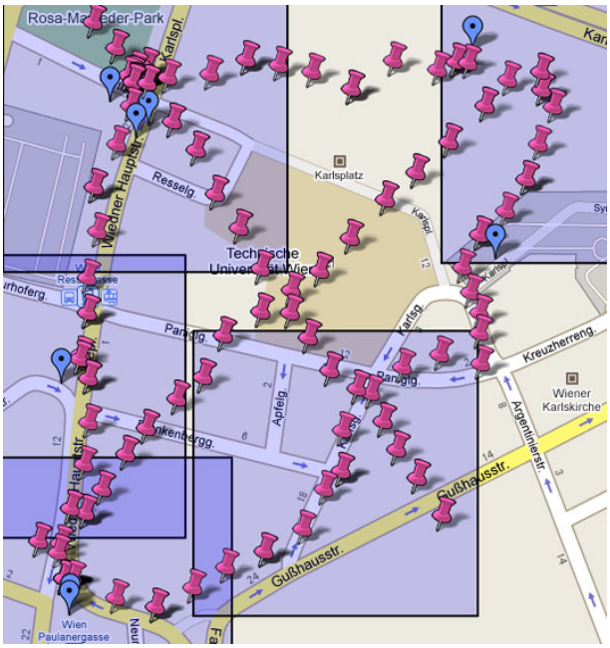
(a) Messages and real path



(b) Messages and simulated paths



(c) Polls with range 300 m and interval 350 sec



(d) Polls with range 200 m and interval 300 sec

Figure 5: Logged and simulated location traces