Trust and Security for Spatial Messaging

Michel Deriaz

Abstract. Spatial messaging is a term that defines the virtual publication of data in physical places. Anyone in the neighborhood of such a publication point gets the message. Architectures allowing users to publish freely spatial messages already exist. However, experiences realized with volunteers showed that there is only little interest in posting such notes. To our view, the main reason is that there is currently no trust mechanism which informs about the reliability of the messages, thus preventing any serious application. Filling this gap will promote the success of spatial messaging, and the growing success of localization and mobile techniques will provide a good support for this concept. After a general presentation conducted by hypothetical scenarios to show the potential impact of spatial messaging, we will discuss how to build a trust model, and propose a first move to a concrete architecture.

1 Introduction

Spatial messaging, also called digital graffiti, air graffiti, or splash messaging, allows a user to publish a geo-referenced note so that any other user that attends the same place can get the message. For example, let us consider the community of the Mt-Blanc mountain guides. The members would like to inform their colleagues about dangers in specific places or about vacancies in refuges. One guide can publish a geo-referenced message that informs about a high risk of avalanches, and any other guide that attends the same place will get the warning, and comment it if necessary. Spatial messaging is a kind of blog, in which editors and readers share the same physical place.

There are many reasons to believe that spatial messaging will become a wide spread concept in a nearby future. Today, people use the connection capabilities of their mobile phone mostly in one way, to download information. But in the same way that people passed from television to Internet, the next generation of user will probably become "active" and publish information. If we remember how fast the computer power and the communication capabilities of these devices improve, and the fact that there are today more modern mobile phones (with Internet connection) than desktop computers in the world, we can easily paint a glorious future for mobile technology. This assertion can be confirmed by the growing interest for location awareness. The success of Mobile Mappy [1], a service that allows you to download maps on you mobile phone as well as POIs (Points Of Interest) wherever you are, is an indicator of this growing interest. And Mobile Mappy is not alone. There are more and more applications or Internet services for mobile users that provide maps and other information related to your current position.

What interest us specifically in spatial messaging are the trust and security aspects. In our Mt-Blanc mountain guides example, the security aspect will ensure that the posted messages are really posted by the mentioned author, that no modifications of the original text can be made afterwards, and that the service is available for everyone that is authorized. But this is not sufficient. Even if you can be sure about the identity of an author, it is useless if you do not know him and therefore cannot trust the content of his message. That is why we are specifically interested in the trust aspect. There are already some implementations of the spatial messaging concept, which will be discussed in 3 - "Related work". But, to our knowledge, there is no architecture that takes into account the trust and security aspects. This paper will therefore try to fill this gap.

2 Scenarios

This section describes different hypothetical scenarios using spatial messaging. They help the reader to better understand the contribution that this concept will bring in the improvement of everyday life, and some of them will be cited as examples from other sections.

2.1 Posting a picture

John is enjoying his holiday at Sunny Beach. It takes some pictures, adds comment about his tourist experience, then defines that all the people in his address book marked as "friends" have access to it, and finally post the whole at his current position. One year later a friend of John that chose the same place for his holiday finds the pictures and can read John's notes; he learns among other things that the expensive restaurant so recommended by the tourist guides actually does not worth it.

2.2 A virtual path for scuba-divers

"Do you already get lost during a diving tour?" Experienced divers like to say that either people answer yes, or they lie. Finding your way is very difficult in an environment with such a limited visibility, and a compass is only useful if you know where you are. We could of course paint big yellow arrows on the reef to define the diving tour, but Black Shark Diving Club found a better solution; they posted virtual messages along the path to follow. These messages can be empty, if their role is just to reorient the divers, or can also provide information about what has to be seen here. The diving computer of each diver indicates where the closest messages are, and allows also the diver to publish his own notes, intended for himself (for example to find his way back), or for his friends.

2.3 Visually impaired people: A specific community

Blind and visually impaired people do not perceive their current environment like us. Sally, blind since her birth, knows very well the paths she follows every day to reach her school, an institute specifically build for blind and visually impaired people. Some new dangers, like road works, cannot be perceived in time by Sally. That's why her school decided to give the opportunity for each student to get and publish virtual messages (voice and/or vibrations), so that new dangers can be communicated in time to each other.

2.4 Road traffic management

Your life expectancy when you walk on the emergency lane of a highway is about 20 minutes, according to the French national police [2]. There are not such statistics for other kinds of roads, but we can easily believe that placing a warning triangle to signal a breakdown or an accident is a dangerous operation. The road traffic management system of Bob's car uses spatial messaging in different situations. In case of an accident, a virtual warning triangle is placed a few hundred meters behind the vehicle. If Bob gets stopped in a traffic jam on a highway, a message is placed before the last exit so that other drivers can reconsider their path. An unusual behavior that potentially represents a danger to others is also signalized by a spatial message. In all these cases, the spatial messages can either be confirmed by succeeding readers if they agree with the content, or repudiated if they think the message is misleading or outdated. The message is therefore automatically destroyed when too many users disown its content.

2.5 Follow me!

Samir, an Egyptian tourist guide, organizes every Wednesday a trip to the Bedouins. Since they are nomad, they post regularly a spatial message at their current position so that they can easily be found. During the trip in the desert, Samir always drives the first car and uses its strong driver experience to find a good path to the Bedouins, avoiding all the dangers like boulders or slopes. Every second a spatial message is automatically sent by his vehicle, thus drawing a virtual road. The tourist cars can then easily follows Samir's track, even if they have to leave a few hundreds meters between each car to avoid the sand moved by the previous one. After the dinner at the Bedouin's, our procession can effortlessly find its way back thanks to the posted messages. And even Samir uses them, since driving at night in the desert is difficult and dangerous.

2.6 Reputation of restaurants

The vTags server http://vtags.restaurant-reputation.com encourages people that are either very happy or very disappointed about a restaurant to post a spatial message. The server keeps the last 20 messages and ensures that the same user can publish only

one message every six months. This restriction avoids that the employees send systematically good comments or that a malevolent user harms the good reputation of a restaurant.

2.7 Virtual meeting points

It is not always easy to find friends in crowded or big zones, like an airport or a ski resort. To help people, we find more and more so called meeting points, meant to be easy to be found. But they are always limited in number (an airport has not hundreds of such points) or even absent (ski resorts). On ski holiday, Jack likes difficult runs when the rest of its family prefers easy ones. To satisfy every wish, they often split up and meet again a few hundreds meters below. But sometimes they miss each other. In this case, Jack's wife creates a virtual meeting point at her current position. Jack can then easily rejoin his family.

2.8 New games

Game designers will also benefit from the emergence of spatial messaging. We can imagine lots of outdoors games, in which player have to move around physical places, post messages for other players, act according to the context in which they are, and so on.

2.9 A virtual medical record

Clara is a nurse that visits old patients at their home. Because it is not always the same nurse that visits the same patients, Clara writes any observation she makes or any medication she gives in the virtual medical record of the person. The data is then encrypted. Thus, any nurse with the required authorizations and that enters a patient's home can access its medical record and complete it.

2.10 Where is the nearest Italian restaurant?

The Boccalino, an Italian restaurant, posted a virtual message that covers a radius of 500 meters around the building. It is an advertisement for this restaurant which provides, among others, the price list and the number of available places. Mark, standing nearby and looking for an Italian restaurant, can now search via keywords all the advertisements about Italian restaurants in the neighborhood.

3 Related work

Before starting this section, we would like to precise the difference between spatial messaging and LBS (Location Based Services). In short, LBS is a kind of spatial messaging in which the user can only **get** data, and **not post** them. Lots of LBS applications for augmented cities (tourists get information, in their mother tongue, about their current place) or augmented museums (visitors get information about what they are looking) have already been implemented. We are clearly interested in spatial messaging in general, where users also post information. Three academic projects, E-Graffiti [3], GeoNotes [4] and ActiveCampus Explorer [5], as well as a commercial application, SmartSpeed [6], would deserve to be described here. But since this is already done in another paper (in this collection of papers) we direct the interested reader to [15]. We see there that spatial messaging is clearly not a new concept. Nevertheless, none of the described systems take the trust and security aspects into account. In E-Graffiti users reveal their real identity. In GeoNotes people may stay anonymous, but we see that user then usurped others' identities; it is therefore not possible to trust a message. And in SmartSpeed, even if you are sure that the one that posted a message is called John Smith, how can you trust the content if you do not know the author? The aim of this paper is to fill this gap by proposing a trust model that informs the user about the reliability of a message.

4 Positioning techniques

There are many ways to obtain one's current position. The most well-know is the GPS (Global Positioning System), an American system made of 24 satellites equipped with a very precise clock and that send the current time to the Earth. A GPS receptor compares the different signals it catches and determines its current position. This system works everywhere on Earth with a precision of about 10 meters, but degrades in difficult situations (for example a place surrounded by high buildings), and becomes useless in many indoors places. The GPS system will in a few years be completed by Gallileo, a European system using 30 satellites and working in a similar way.

But there are also local solutions, like wireless triangulation. It consists in analyzing the different signals strengths, and according to the positions of the antennas, determining the position of the device.

We observe that "positioning" is an active research topic, and that mobile phone operators stay in the run by preparing (and already providing for some of them) the aGPS technique. Assisted GPS (aGPS) is a signal sent from the antennas of the mobile phone operator that helps to localize the satellites. The user gets then a position much faster.

Anyway, we are not interested in these techniques for our work. We just assume to have a technical manner to get a position, expressed in latitude and longitude, and that the precision as well as the availability of this information can vary according to time and position. We saw that positioning techniques are improving and becoming omnipresent; it is therefore reasonable to suppose in our work that, in a way or

another, we know our current position. We can therefore completely abstract from the techniques underneath.

5 Securing spatial messaging

In a secured spatial messaging system, a user can be sure that the message he is reading is really written by the mentioned author, that nobody has modified the content of the original message, and that all other available messages at this place are available. More precisely, a secured spatial messaging system has to respect the "traditional" security services that are:

- Confidentiality. Protection of the information against divulgations.
- Integrity: Protection of the information against modifications.
- Availability: Information is always available.
- Entity authentication: The author can be identified.
- Data origin authentication: Information can be linked to its author.
- Non-repudiation: The author cannot repudiate a message.
- Non-duplication: Protection against copying the information.
- Anonymity: The real-life identity of the users must be preserved.

These security services are well-known [7] and won't be discussed here. There are many implementations that already proved their efficiency. Our aim is to focus on specific security services, the ones that are required for spatial messaging (in addition to the "traditional" ones). These are centered on the pseudonym concept. What we would like is a system in which an author can be identified, but at the same time we would like to prevent any link with his real-life identity. A new user is therefore able to get a pseudonym in an anonymous way, but only one. If the person can obtain an unlimited number of pseudonyms, then the system can be victim of a Sybil attack [8]. The user must also be able to change its pseudonym. Again, this must be done in an anonymous manner and it must be impossible to link a former pseudonym with the new one.

A secured spatial massaging system must therefore respect, in addition to the "traditional" security services, the following "specific" ones:

- A user has only one pseudonym at a time.
- A user must be able to change its pseudonym.
- It is impossible to link a pseudonym to a real-life identity.
- It is impossible to link two pseudonyms of the same real-life identity (an old one with a new one).
- Each pseudonym is unique, it is impossible that two different real-life identities share the same pseudonym. This is even true during time; if a user changes its pseudonym, the old one is locked and can never be used again.

In the scenario 2.1 - "Posting a picture", the author chooses who will be able to read the posted message. We are not going to discuss **if** it is a good idea to allow

publications for a restricted audience, since anyway we cannot avoid it. Indeed, an author can always encrypt the content of his message with a key *K* and then encrypt *K* with the public keys of each addressee. The question is more **how** we are going handle this issue, or what tools a spatial messaging system should provide for that. With no tools then encrypted messages will be visible for everyone (even if the content itself cannot be understood), so everyone can see who published this message and can add comments to it. With supplied tools the system could hide the messages for unwanted addressees. Which solution is better? Do the facilities introduced by these tools compensate the fact that the users loose control over the system? This is still an open question.

6 Trusting spatial messaging

Section 5 discussed the security aspects of spatial messaging. A reader can be sure that a given message is really posted by its signer and that the content has not been modified since. But even if the reader can be sure about the author's identity, it is useless if they do not know each other. This section discusses how to add trust information on spatial messages so that the reader can evaluate the reliability of a message.

Trust is a very complex concept. Even if it is part of everyday life, different people give also different definitions of what trust is. This observation is even strongly accentuated when we try to explain how to build a trust relation between machines, or between humans and machines. The author's own point of view of trust can be found in [9]. This paper presents some global definitions, presents some related work, and particularly points out the fact that there is no existing trust model that could be applied to spatial messaging. One reason is that most models are only designed and specialized for peer-to-peer files sharing systems. For example, these models do not take time into account. In spatial messaging time is very important. For example a message indicating a high risk of avalanches posted yesterday has to be taken more seriously than the same message posted six months ago.

Spatial messaging needs a specific trust model that takes time into account, as discussed previously, and that is sufficiently flexible to be adapted to different situations. For example, in our Mt-Blanc mountain guides example, we suppose that the community of users is quite small and that a Web-Of-Trust trust model [10] will be sufficient. If Alice trusts Bob at 0.8 (out of 1), and Bob trusts Charlie at 0.5, then Charlie's rating (in Alice's eyes) will only count for 0.8 * 0.5 = 0.4. This does not mean that Alice's trust in Charlie is only 0.4. It is only the number by which Charlie's rating will be multiplied. This easy formula is sufficient to give more importance to close friends, and of course also more importance to reputable ones.

However this model does not work for large communities, like the one described in the scenario 2.4 - "Road traffic management". In this case we need to know the global reputation of the author. We could of course provide two different models, but what if the community is middle-sized? Anyway this is not a good solution since we can also have a big community that contains smaller ones, in which the people knows each other. In the scenario 2.7 - "Virtual meeting points", we can easily imagine that lots of

unknown people are putting such kind of messages, but we are only interested in the ones that are posted by friends.

And there is a third trust model, the one that informs about the reliability of the message itself, without taking care of the author's reputation. Even a very reputable author can make a mistake and publish wrong information. Or, even more likely, a message signed by a reputable editor can contain outdated information.

The fourth and last trust model we present here is actually the model that will combine the former ones. His role is to answer the "How to trust the different trust models" question. The three previous models will give us three different trust values, and the forth model's role is to determine, according the current situation, how much weight to give to each value. It is this model that will be used to make automatic decisions.

7 Relation between trust and security

Are trust and security two completely different topics, which can be handled independently? The answer can be yes... ...or no!

Instinctively, as a first approach, we can consider that the two concepts are independent. Security deals with problem like integrity, authentication, non-repudiation and so on, in other words with what everyone would except from a "secure" system. The security can be "measured" with metrics like the length of the key for a given algorithm or the quality of the algorithm itself. Trust, like interpreted by human communities, can be defined as the amount of risk that one is ready to take for a given action. In this sense trust and security are two different topics. In the example of e-banking, the security part will ensure that no third party is able to read or modify the exchanged messages, or to prevent any transaction from a denial of service attack. The trust part concerns the trust I have in the bank. It is a good point that my money arrives safely in the bank, but it is useless if the bank is not honest and makes me loose all my money. But, again, there are clearly two independent concepts. Changing a security component of the bank should not modify the trust I have in it for managing my money. As time passes, the trust I have in the bank can increase even if the security components do not change. So, how to build a new system? We could first create a system without taking care about security and trust aspects, then we secure all the part that need to be secured, and finally we add a trust mechanism. But this approach is wrong. And dangerous. Among the most famous example, we could cite Internet. It was designed to be very open, and at its creation security aspects were not taken into account. Neither trust, of course. And everyone knows the current situation. We create more and more security tools, that we try to merge as well as possible with the rest of the system, but paradoxically the number of attacks and other annoyances, like spam, is increasing. Today, to surf the Internet, you need at least a firewall, an anti-virus, an anti-spywares, a spam filter, and be sure that all you software is up-to-date; tomorrow you will also need an anti-phishing, an anti-hoax, a scanner detector, a key-logger finder, and a cleaner that erases all sensitive information that can be found in your computer. Past experiences teach us security is

not a tool. Security is a process. And this process must start at the beginning of the design time.

That's why special care must be taken in order to avoid to doing the same mistakes with the trust concept. We believe that trust is also a concept that must be included at the beginning, during design time. If we want to go beyond the simplistic e-banking former example, if we want to design a secured and trusted system, then trust and security have to be studied from the very beginning. It is important to note that in the e-banking example, the trust is not physically included in the system. The security concepts are implemented by algorithms, but the trust is only present in brains. It is only computed and managed by humans. But future applications will have to deal with trust in an automatic way. Trust must be interoperable, computable, and manageable by humans as well as by machines. Then, trust and security are in fact a single topic, or better said, a single process; a single process responsible to keep reliable the environment for which it was designed.

8 Towards a trusted spatial messaging architecture

GeoVTag is an architecture that supports trusted and secured spatial messaging. In GeoVTag we call "vTag" (virtual tag) a spatial message. Our architecture is centralized. Each server manages vTags of a specific subject and each of them is identified by a different URL and works independently from the others. Any user can obtain anonymously all the vTags in his neighborhood just by queering the server. To become a member, and therefore be able to review vTags (add comments to an existing vTag) or be able to create new vTags, a user has to register. The registration process allows you to choose a pseudonym and returns a key pair that will be used each time to reconnect to the server. The registration process is done so that it is impossible, even for the server, to make a link between a pseudonym and the corresponding real-life identity. The process guarantees also that each pseudonym is unique, so that it can be used as a unique identifier.

The rest of this section focuses on specific challenges in the GeoVTag architecture, like how to get anonymously a pseudonym, how to change it, and how to deal with the trust information.

8.1 Getting a pseudonym

The process must respect the following rules:

- **Rule 1**: A user must stay anonymous from other users and from the server.
- **Rule 2**: A user can have only one pseudonym at a time.

The first rule explains by itself why we need pseudonyms. It is simply the only way to stay anonymous (there is no way to link a message to a real-life identity) and still have a way to be uniquely identified. The second rule avoids a Sybil attack [8], where a user would create many different virtual identities in order to subvert the system.

To get a pseudonym, a user must own a digital certificate, like the ones supplied by Verisign or by some country ID cards (for example Belgium). We propose the following algorithm, which respects the previous rules, for a new user to get a pseudonym:

- Alice chooses a pseudonym, creates a pair of asymmetric keys, and makes a member certificate with the pseudonym and the public key.
- Alice connects to the server and sends her real-life certificate and her member certificate hidden in an envelope (this will be explained later).
- The server checks the validity of the real-life certificate and checks that Alice connects for the first time (or that Alice doesn't have already a pseudonym). If OK, the server signs the envelope (blind signature).
- Alice opens the envelope and gets its new member certificate, signed by the server.
- Alice waits a while before using the system to avoid that the server links the new pseudonym that will appear on vTags to her.

The blind signature algorithm [11] allows a signer to digitally sign information without seeing it. Let's say Alice wants that a server signs blindly her certificate *c*. She chooses a random number *k*, and using the server's public key *e*, she sends the envelope:

$$(c \cdot k^e) \bmod n$$

The bank signs the envelope using its private key *d* and returns:

$$(c \cdot k^e)^d \bmod n$$

Alice opens the envelope and gets:

$$\frac{(c \cdot k^e)^d}{k} \bmod n = \frac{c^d \cdot k^{ed}}{k} \bmod n = \frac{c^d \cdot k}{k} \bmod n = c^d \bmod n$$

Alice has now a certificate signed by the server, even if the later never saw it.

8.2 Changing a pseudonym

A member must be able to change its pseudonym. First for privacy reasons, if a member thinks that his pseudonym can be linked with his real-life identity, he will want to change it, and second to respect the rule that says that each pseudonym has to be unique. If two members choose the same pseudonym at the same time, one of them must be able to change it. And remember, the members certificates are blindly signed, so it is impossible for the server to avoid to sign twice the same pseudonym. The server discovers a new pseudonym only the first time that the corresponding member reviews a vTag or creates a new one. Publishing a list of all the existing pseudonyms is not a solution either, since the laps of time between the creation of a new certificate and its first utilization can be arbitrarily long. Note that the fact that the server signs two certificates with the same pseudonyms is not really a security problem, since each pseudonym is linked to a public key, and that during a connection initialization the user will have to prove that he is the owner of the corresponding private key. The process of changing a pseudonym must respect the following rules:

- **Rule 1**: A user must stay anonymous from other users and from the server.
- **Rule 2**: It is impossible to link a new pseudonym with a former one.

The first rule is the same as for getting a pseudonym for the first time. We insist here that changing its pseudonym must in no way affect the privacy of the seeker. We propose the following algorithm, which respects the previous rules, for a user to change it pseudonym:

- Alice connects to the server and identifies herself with her member certificate.
- The server challenges Alice to verify that she is really the owner of the old pseudonym. The challenge is done by a ZKP algorithm. If the challenge is OK, Alice chooses a new pseudonym, creates a pair of asymmetric keys, and makes a member certificate with the pseudonym and the public key.
- Alice sends her new member certificate hidden in an envelope (see 8.1 - "Getting a pseudonym").
- The server signs the envelope and sends it back to Alice.
- Alice opens the envelope and gets its new member certificate, signed by the server.
- Alice waits a while before using the system to avoid that the server links the new pseudonym that will appear on vTags to the former pseudonym.

A ZKP (Zero Knowledge Proof) algorithm allows a proover (the entity that wants to prove something) to prove to a verifier (the entity that challenges the proover) that he owns a secret key, without giving any information about the key itself. This last property is typically not fulfilled in the basic scenario in which entity A chooses a random number and asks entity B to digitally sign in order to prove that he own the corresponding private key, since entity A could carefully choose the "random" number in order to prepare what is called a chosen plaintext attack. A simple ZKP algorithm was proposed by Fiat and Shamir [12]:

Let p and q be two different huge and secret prime numbers and the public value $n = p \cdot q$. The proover P chooses its private key s (primitive to n) et publishes its public key $t = s^2 \bmod n$. P proves to the verifier V that he knows s without giving any information:

P chooses a random number r and sends the witness $x = r^2 \bmod n$.

V chooses the challenge $c = 0$ or $c = 1$.

P sends $y = r \cdot s^c \bmod n$.

V checks that $y^2 \bmod n = x \cdot t^c \bmod n$.

Example with $c = 0$ (no modulo):	Example with $c = 1$ (no modulo):
P chooses r and sends the witness $x = r^2$. V chooses the challenge $c = 0$. P sends $y = r$. V checks that $y^2 = x$.	P chooses r and sends the witness $x = r^2$. V chooses the challenge $c = 1$. P sends $y = r \cdot s$. V checks that $y^2 = x \cdot t$.

8.3 Trust management

This part discusses how to manage the trust information. We chose a completely centralized approach. We could of course have chosen a decentralized approach, like proposed by the Eigentrust [13] algorithm, but these systems are often designed with a file-sharing application in mind and suggest that the different peers are online a significant amount of time. If we take our mountain guides example, we can clearly imagine that only a reduced number of users are online at the same time, if there is not only one. So, to have sufficient trust information the only solution consists in storing it in a place that is always available; a centralized server fulfils quite well this requirement.

In section 6 - "Trusting spatial messaging", we described four trust values. One concerned the reputation of the message itself (without taking care of the author's reputation), one about the local reputation of the author (what a user and his friends think about him), one about the global reputation of the author (what people in general think about him), and finally one that combined the previous one according to the context and that can be used to take decisions in an automatic way. These four values are included in all the requested vTags, so that the reader can make himself an opinion about the reliability of the message. We will now skim over how the server could compute these values. Future work on this project will precisely heavily focus on this part.

8.3.1 Tag reputation

This value indicates how reliable a vTag is, according only to the marks given by the reviewers (we do not take into account the reputation of its author). A simple solution consists in computing the average of all the reviewers' marks. Other solutions have to be studied, like giving more importance to reputable reviewers' marks, or giving more importance to recent reviews.

8.3.2 Global reputation

The global reputation of an author is computed according to the marks of the reviewers on all the vTags he authored. We could imagine that this value is simply computed by doing an average on the marks, or even by giving more importance to recent ratings, but we are not convinced that such an approach will work. First, there is no motivation to rate other's vTags. Second, a malevolent user could systematically and automatically rate badly other's vTags. Thus, the quality of the rating itself must

also be taken into account, and have an incidence on the reputation of the reviewer. But how can we judge the quality of the review?

Proposition 1: Compare to other reviewers and increase the reputation if the mark is similar (and decrease if it is different). Problem: A user can rate automatically all the vTags like the others in order to improve its trust value.

Proposition 2: Same than proposition 1, but compare the user's rating only to marks that are done afterward. Problem: It is better, but the attack remains the same: If a vTag owns good ratings, it is also likely that the ensuing marks will be good.

Proposition 3: Using pitfalls. Same than proposition 1, but the server sends time to time either good vTags that are badly rated, or bad vTags that are good rated. If the user cheats by using an automatic rating system, he will fall in such pitfalls. He becomes then suspect and will be observed more carefully. For example the system could check if this user rates two vTags that are too distant from each other to be reviewed within a given amount of time, or if this user tries to rate fake vTags that are posted in inaccessible zones. If it becomes clear that a user is cheating, the server can simply revoke its pseudonym. This is a big problem for the cheater since he won't be able to get a new one without revealing is real-life identity. Otherwise there is no other choice than to abandon definitively the system. Since it is quite difficult to isolate cheaters, it seems important to have a strong sanction against them. In this way we differ from many other reputation systems in which cheaters can usually simply change, in an anonymous way, their pseudonym and start again with the same chances than any new user.

8.3.3 Local reputation

The local reputation is the reputation of the author in the eyes of the reader and its friends. Each user holds a list of his friends as well as the trust value he grants to each of them. This list is only modified on the user side but is stored on the server. These lists are not available from the other users. The trust mechanism is analogous to the PGP web of trust [10] for human entities. For example, if Alice trusts Bob at 0.8 (out of 1), and Bob trusts Charlie at 0.5, then Charlie's rating (in Alice's eyes) will only count for 0.8 * 0.5 = 0.4. This does not mean that Alice's trust in Charlie is only 0.4. It is only the number by which Charlie's rating will be multiplied. This easy formula is sufficient to give more importance to close friends, and of course also more importance to reputable ones.

It is not conceivable to keep these friends list locally because they are used to make "friend of a friend" relationships and because in this kind of application we can easily imagine that the users are offline most of the time.

8.3.4 Server's recommendation

This value is computed according to the previous ones, to the context, and to the kind of service that is provided by this server.

9 Implementation of GeoVTag

This section presents some implementation details of the GeoVTag application, and aims to help the reader to better understand the previous concepts. A vTag is divided into three parts. The first part is written by the server. It contains trust information that can be read by the user in order to determine how reliable this vTag is. The second part is the one written by the author of the vTag. It contains geographical coordinates and the content of the vTag itself. The third part is written by reviewers. Every member can indicates how much he agrees with the content of the vTag and add information if needed.

We propose a standard way to represent vTags, based on XML. One reason is to make the system interoperable. For example, we can imagine a tourist that comes to our country. He sees on a flyer the URL of a vTag server supplying useful information in his mother tongue, like information about places he is visiting, or what people with the same cultural background than him think about the different neighboring restaurants. If the potential user seems interested by this service, he will probably accept to add the URL at his server list, but it is much more unlikely that he accepts to add new software on his mobile phone for each new service. That's way we propose a standard way to represent vTags so that a single application can display all the vTags around.

A typical vTag looks like this:

```
<vtag>
  <server>
    <url>vtag.unige.ch</url>
    <no>184467440737095519999</no>
    <reputation>
      <this_avg>1.0</this_avg>
      <this_conf>0.3</this_conf>
      <global_avg>0.74</global_avg>
      <global_conf>0.5</global_conf>
      <local_avg>0.87</local_avg>
      <local_conf>0.3</local_conf>
      <recommanded_value>0.75</recommanded_value>
      <recommanded_conf>0.7</recommanded_conf>
    </reputation>
  </server>
  <author>
    <pseudo>Alice</pseudo>
    <utc>2005-11-29 12:34:56</utc>
    <lat>46.330422</lat>
    <lon>6.343443</lon>
    <title>Danger - Avalanches</title>
    <content>
      Measures taken recently suspect a high risk of avalanches here
    </content>
    <exp>8640000</exp>
    <radius>1000</radius>
  </author>
  <reviewers>
    <note>
      <pseudo>Bob</pseudo>
      <utc>2005-11-29 14:54:55</utc>
```

```
        <agree>1.0</agree>
        <content>I have seen the measures</content>
      </note>
    </reviewers>
  </vtag>
```

9.1 XML tags in <server>

XML tag	Required	Description
<url>	Yes	The URL of the service that hosts the vTag.
<no>	Yes	The number of the vTag in 64 bits.
<reputation>	Yes	Trustworthiness of this vTag.
<this_avg>	No	The average reputation of this vTag, based only on the <agree> XML tags, [0..1]
<this_conf>	No	Confidence of <this_avg>.
<global_avg>	No	The global reputation of the editor, [0..1].
<global_conf>	No	Confidence of <global_avg>.
<local_avg>	No	The local reputation of the editor, [0..1].
<local_conf>	No	Confidence of <local_avg>.
<recommanded_value>	No	A combination of <this_avg>, <global_avg> and <local_avg>, made through heuristics, which indicates how trustful this vTag is.
<recommanded_conf>	No	Confidence of <recommanded_value>.

9.2 XML tags in <author>

XML tag	Required	Description
<pseudo>	Yes	The pseudonym of the author. Each pseudonym has to be unique.
<utc>	Yes	The current UTC date and time. This information is important since it is possible to revoke pseudonyms.
<lat>	Yes	The latitude, expressed in decimal degrees.
<lon>	Yes	The longitude, expressed in decimal degrees.
<title>	No	The title of the vTag.
<content>	No	The content of the vTag.
<exp>	No	Expiration of the vTag, expressed in the number of seconds since the creation of the vTag.
<radius>	No	The radius of the circle, in meters, in which the vTag is visible. The default value equals 20.

9.3 XML tags in <reviewers>

XML tag	Required	Description
<note>	Yes	Every review is contained in a note.
<pseudo>	Yes	The pseudonym of the reviewer. Each pseudonym has to be unique.
<utc>	Yes	The current UTC date and time. This information is important since it is possible to revoke pseudonyms.
<agree>	Yes	A number in the range [-1..1] that indicates how much the reviewer agrees with the vTag. -1 means a full disagreement, 0 a neutral opinion, and 1 a full agreement. Applications can either stick to these three values, or use all the floating point number in-between. In case there is no opinion, this XML tag as to be left empty. Note that for a reputation computation algorithm there is a big difference between a neutral opinion and no opinion.
<content>	No	The content of the note. This optional field can be used to justify the value in the <agree> XML tag or to give a complement of information.

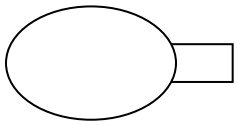
9.4 Extensions

The XML tags described above are standard, but every service can add its own ones. For example the <radius> XML tag, which is not required, could be replaced by:

```
<visibility>
  <rectangle>
    <n_w_lat>46.330423</n_w_lat>
    <n_w_lon>6.343442</n_w_lon>
    <s_e_lat>46.330200</s_e_lat>
    <s_e_lon>6.343662</s_e_lon>
  </rectangle>
  <oval>
    <n_w_lat>46.330600</n_w_lat>
    <n_w_lon>6.343300</n_w_lon>
    <s_e_lat>46.330100</s_e_lat>
    <s_e_lon>6.343500</s_e_lon>
  </oval>
</visibility>
```

"n_w" means North-West, or the top left corner (of a rectangle), "s_e" means South-East, or bottom right corner. For ovals, this coordinates correspond to an imaginary rectangle that just contains the oval.

The example above will describe a visibility zone that looks like:



A combination of basic shapes allows to defining every random visibility zone. In a future work we will provide a list of "suggested" XML tags, in order to decrease the interoperability problems.

10 Open issues and other remarks in bulk

Currently, we have one pseudonym at a time, which is different for each server (to avoid links with the real-life identity). If they are several pseudonyms for a single server, then we have to find an algorithm to allows to link them together if, and only if, the same person uses two different pseudonyms to sign the same vTag (for example one signature for the vTag and another for a positive rating).

All the trust computation is done publicly. The only possible cheater is the server itself because he can create new identities and use them to post funny vTags or rate badly other ones. But it is like shooting oneself in the foot, since there is not really an interest in doing that. People that are not happy with the vTags they find or the way that their own vTags are commented will leave the system. We imagine also that an external and neutral trust system, like Epinions.com [14], will probably appear and be specialized in managing reputations of vTags servers.

To make it more difficult to make links between real-life identities and pseudonyms, or between old pseudonyms and new ones, servers are meant to publish dates when users are encouraged to join the system or change their pseudonym.

An author cannot review its own vTag and a reviewer can review only once a given vTag. Reviews must be independent from each other. Reviewer mustn't comment the reviews of other reviewers.

Deleting a vTag. The procedure is application dependent. A typical solution consists in deleting the vTag if the sum of the <agree> contents plus one, divided by the number of reviews plus one, is less than 0. The "plus one" stands for the author who rates its own vTag with "1". So, a vTag is NOT deleted if we have (-1), or (-0.5, -0.5), or (0.5, -0.5, -0.5, -0.5), but the vTag is deleted if we have (-1, -0.1). Note that an author cannot delete his own tags. Note also that a vTag contains a <exp> field, and when the given time is reached the vTag is automatically deleted.

Modifying a user's reputation. We saw previously how to interpret the trust values that we find in the vTags. This part discusses how to compute the reputation of a user. Every new member, or each time that a member changes its pseudonym, starts with a trust value of 0. In case of "good" behavior this value increases, to a maximum of 1, and in case of "bad" behavior this value decreases, to a minimum of 0. The question is what is a good or a bad behavior, and how much to increase or decrease the trust

value accordingly? In the case of local reputation, we send the reader to 8.3.3 - "Local reputation". What interest us here, is how to compute the global reputation of a user. If the user acts as an author, his reputation varies according to the marks of the reviewers. Less importance is given to the marks that are done close to the expiration date. If there is no expiration date, the same importance is given to each mark. If the user acts as a reviewer, then each mark participates in increasing its trust value, unless he rates in a malevolent manner (like systematically bad in order to decrease the author's trust, or if he rates automatically all the vTags without reading them).

11 Conclusion

This paper started by giving a global presentation of spatial messaging. In the related work, we saw that this concept is currently far from widely accepted, and that users are not very interested in publishing virtual messages attached to physical places. We believe however that adding trust and security will completely change the deal. As mentioned earlier, the problem with applications like E-Graffiti or GeoNotes is that they are not very useful. Spatial messaging will be more successful if it can address specific communities with specific needs. But then, the system must be reliable and the different messages must be trustworthy, even if the real-life identity of its author is hidden.

We presented the trust and security aspects for spatial messaging. Even if the literature proposes a lot of trust management systems, most of them are designed for peer-to-peer architectures and no one seems to suit to our needs. Our future work will therefore heavily focus on building a new trust model, specifically designed for spatial messaging.

12 References

[1] Mappy. http://www.mappy.com/
[2] French constabulary. http://www.defense.gouv.fr/gendarmerie/
[3] Burrell, Jenna, Gay, Geri K. (2002): E-graffiti: evaluating real-world use of a context-aware system. In Interacting with Computers, 14 (4) p. 301-312
[4] Persson, P., Espinoza, F., Fagerberg, P., Sandin, A., and Cöster, R. GeoNotes: A Location-based Information System for Public Spaces, in Höök, Benyon, and Munro (eds.) Readings in Social Navigation of Information Space, Springer (2000)
[5] William G. Griswold, Patricia Shanahan, Steven W. Brown, Robert S. Boyer, Matt Ratto, R. Benjamin Shapiro, Tan Minh Truong: ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing. IEEE Computer 37(10): 73-81 (2004)
[6] SmartSpeed website: http://www.smartspeed.fr/
[7] Book "Hacker's Guide, Edition DeLuxe"
[8] John R. Douceur. The sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.
[9] Michel Deriaz. What is Trust? My Own Point of View. In this collection of papers
[10] P. Zimmerman. PGP User's Guide, MIT, 1994.
[11] Website: http://www.schneier.com/book-applied-toc.html
[12] Fiat-Shamir protocol. http://www.cse.scu.edu/~tschwarz/coen350/zkp.html

[13] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigen-Trust Algorithm for Reputation Management in P2P Networks. 2003.
[14] Epinion.com. http://www.epinions.com/
[15] Michel Deriaz and Jean-Marc Seigneur. FoxyTag. In this collection of papers