International Journal of Information Science and Intelligent System, 4(1): 105-118, 2015

Geo Signature and Its Applications

Dmitry Namiot *

Lomonosov Moscow State University, Moscow, Russia

Received: 07 November 2014; Accepted: 09 December 2014

Abstract

This paper summarizes definitions and uses cases for the sharing location information via geo messages. Geo messages let users of location based systems share location information as signatures to the standard messages (e.g., email, SMS). Rather than let some service perform the constant monoitoring for the user's location (Google Latitude) or share location info within any social circle (Facebook's check-in, etc.) Geo Messages approach lets users share location data on the peer to peer basis. Users can share own location info via any of existing messaging systems. The basic tenet of this process is the separation of the identification and location information. With this approach shared location information (geo-coded link, map, etc.) does not contain the users' identity. It uses identity information from the messaging system. It means that the whole process of location info exchange does not reveal user's privacy. For a multilateral exchange of geo-positional information, this approach proposes a scheme with replaceable user authentication where only the creator of temporary name knows the identity mapping between temporary names and true identities.

Keywords: LBS; Location; Messaging; Geo Coding.

1. Introduction

It is obvious, that the question "where are you" is one of the most often asked during the communications. 600 billion text messages per year in the US ask "where are you?" – as per Location Business Summit 2010 data. A vast amount of mobile services is actually being built

* Corresponding author.
E-mail address: dnamiot@gmail.com (Dmitry Namiot).

around this question so their main feature is a user's location exchange [1]. In the most cases it presents the ability for the mobile user (mobile phone owner) write down own location info in the some special place (special data store, supported via some mobile application). But it means, of course, that user must be registered in this service and download a priori that special application. What is even more important here – everyone who needs this location information must use the same service too.

There are several models for location information sharing in services. On the first hand, it is so-called passive location monitoring. The typical example is Google Latitude [2]. Passive location sharing model does not require specific actions from mobile users. Accumulated data become available some API. The privacy is probably the biggest issue with this approach. As a user I should be aware that some third party tool is constantly monitoring my location and saves it on the some external server. And of course, the shorted life of the handset's battery is the second biggest issue with this approach.

Another model is voluntary location sharing. The typical example is so-called check-in [3]. Check-in is a special type for the record (status) in some social network. It could be an active (e.g., Swarm), when the user directly sets his/her current location or passive (e.g., Twitter), when location information could be added as an additional attribute to the current status. Of course, for sharing location information both parties must be registered in the same network. And here we can see "all or nothing" problem with location sharing. Shared location info could be visible to all friends, but in the real cases for most of them it is just a noise. The location info could be actually interested only for the physical friends. E.g., for the majority of twitter followers my location info (Swarm status in Twitter time line) is just a noise.

The idea of the signed geo messages service (geo mail, geo SMS) is based on the ability to add user's location info to the standard messages like SMS or email [1]. Location information in such system is just a signature. So, it would be just enough with this service for telling somebody 'where I am' send him a message. And the target party does not need to use any additional service in order to get information about the sender's location. He will simply read SMS or email.

Speaking more broadly, this service separates location information and identity information. It is possible, because the message itself contains the identity. So, we do not need to introduce ant additional authentication layer. And shared location information data contain the location information only. Only the combination of both elements (message and location) lets us associate location data with identity.

Obviously, it is peer to peer sharing and it does not require any social network. The geo signature has got a form of the map (link to the map) with the marker at the shared location. And what is important here – the map itself has no information about the sender and recipient. That information exists only in the message itself. The map (marker) has no information about the creator for example. So, without the message itself there is no way to know sender and/or recipient. It is is just a map with some location. That is all about the privacy.

This model is, probably, the easiest way for sharing location information. It does not require any application downloading or registration in social networks from the potential users and provides a smooth extension for the existing communication services.

There are several services that implement Geo Messages approach. Originally, they were described in [1]. And this paper summarizes the latest development, as well as discusses the possible extensions.

2. Project Description

The main idea behind Geo Messages is how to deliver location info via the standard messaging (SMS and Email). This approach borrowed the ideas from SMS based content delivery. Typically, when the mobile users request some service via SMS it means users are actually getting some link within the text message. This link leads to the downloading service for pictures, ringtones, etc. And this approach uses the simple fact that all native SMS clients nowadays are smart enough to discover links (substrings in the format *http://something_is_here*) within the text and allow one click internet access for opening that link. So, we can use the same approach for delivering location information too.

The location info will be presented as a link, leads to the appropriate map. So, if the sender will be able to automatically add such a link to the message, the receiver will be able with just one click open the map. The map will show the sender's location. Alternatively, we can provide a link to some specially created landing page, probably again with an embedded map or any other location related info.

Our original development targeted feature phones and has been implemented as an application for SIM-cards (Java-card applet). It includes the following steps:

1. The location info could be requested right from the SIM-card (smart card) as Cell ID info. This information exists always and Java-card applet can read it (local info).

2. Cell ID information could be translated into "human"-readable form of (latitude, longitude) pair. There are several public services over the Net that let us do that. The typical example is OpenCellId [4]. Actually, it is just a public HTTP based API.

3. Using the data obtained in the step 2, we can create a link to the map. Our original development used Google Static Map. The Google Static Maps API lets applications embed a Google Maps image on your webpage without requiring JavaScript or any dynamic page loading. In our case Google Static Maps API let us build a map (actually – an image for map) based on the latitude – longitude pair obtained through the step 2. For the smartphones we can create the similar link with Google Maps API (there are no more JavaScripts limitations).

4. URL shortening service could be deployed. In order to make sure our geo-related URL's complies with SMS restrictions (simply – they are no more than 140 symbols) we can deploy URL shortening service and make our signature smaller. In our application we use bit.ly shortening service [5].

5. In order to add our location aware URL to the message (to SMS or to email) we will deploy URI Scheme for GSM Short Message Service and The *mailto* URL scheme [1]. So, our final step included dynamical generation of the mobile web page with links for messaging:

sms:?body=our_geo_aware_URL
and
mailto:?body=our_geo_aware_URL

As soon as the mobile user will hit one of the links, the native (it is very important!) messaging client (e.g. the native SMS client) will be launched with the text (body) field pre-populated with the given URL. So it is enough just to select the target phone (address) from the address book, add some text (optionally, of course) and send the message. After all, this service presented a mobile mashup (mobile web mashup) that passes user through the series of screens where the last one offers for the user customized messages sending links. And the whole process is

a) completely automated
b) does not require any authorization in external services
c) very portable and will work on any mobile phone

For HTML5 applications we can use its geo-capabilities [6]. The modified web client is illustrated on Figure 1.

Figure 1. Geo-Mail client

And Figure 2 illustrates the delivered map.

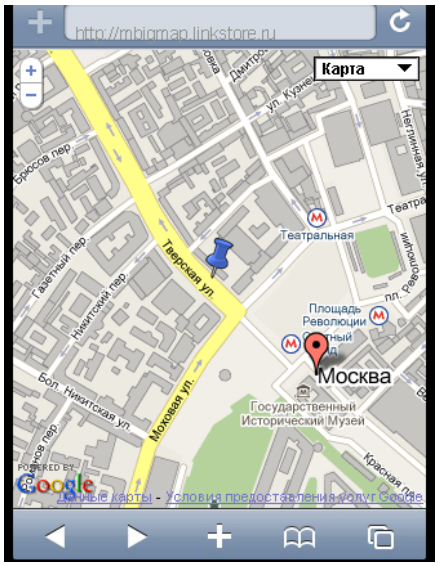

Figure 2. A delivered map

It displays two markers upon opening. One of them shows the shared location and the second one displays own location for the reader (receiver). Note, that in all cases maps are useless without the message. There is no way to get identity info without the message.

Originally, the above described approach was implemented as a proof of concept for SIM-cards (Java cards) manufactures. It used so called SCWS Servlet. It is Java servlet located on the Smart Card Web Server. This servlet requested location info from SIM-card. We mean here proactive command Provide Local Information, with Command qualifier '00' (as per GSM 11.14 standard) [7]. This command lets us obtain Location Information (MCC, MNC, LAC and Cell Identity).

Further, such an approach has been consistently used in a variety of applications. At first hand, it is Android application which lets create pre-filled (geo-signed message) message. Due to the Android's architecture (intents, in particular) it can use any available messaging system: email, SMS, instant messages, etc.

As the next step, we can mention here a series of HTML5 web applications, similar to the above mentioned Geo Mail. For example, Geo SMS creates a pre-populated link for SMS delivery, Geo Tweet provides the same service for Twitter, Geo Signature has been implemented as an extension for Google Chrome, etc.

By the similar manner, we can create geo-signed posts (messages) for social networks. An appropriate mashup can use either public API for creating pre-populated forms (forms with geo-links), either use email-based posting. For example, a new message to Facebook's timeline could be added by email. This email could be geo-signed too.

Another modification could be concentrated on the geo-link itself. Firstly, instead of referring to the map, we can use some landing page. Landing page could be created automatically too. Also, we can use one time readable (accessible) landing pages. They could be automatically

destroyed (deleted) right after the first access. This approach lets us completely remove the history of movements.

We do not need to share location in the form of geo-coordinates only. Shared party, for example, can pick up some location (place) and use this information in the geo-signature ('I am here' message). For example, a location sharing client (e.g. a new mail client) requests location via public Foursquare API [8], client selects location and selected name could be used as a signature. A typical example is 'Never Eat Alone' service. Mobile client can quickly share a name for café or restaurant he is waiting his friends (Figure 3,4). Note, that it is some analogue for check-in service, just without any particular social network. An appropriate link for sharing a location name as well as any other location-related information (address, special offers) via email (SMS) could be presented as QR-code right on premise. Such a service could be vendor agnostic of course. So, instead of Foursquare (Swarm) we can use any appropriate location API, which let us select nearby places (Google, Facebook, etc.).

As it is stated in several papers, geo-coordinates could be replaced in some services with network proximity information. We can use information about wireless signal strength for the distance estimation [9]. It means also, that we can use information for the wireless network environment as a "pseudo" geo-information. At least, we can show to mobile users other players nearby, regardless of the real geo location. It has been used in several services [10,11]. In this connection we can talk also about sharing location information, but now in the form of so-called network fingerprint. Fingerprint describes the existing network environment (e.g., names for Wi-Fi access points and signal strengths). Shared link contains information which could be compared with fingerprint of receiving party. We can mention here our paper [12]. It presents a customized QR-code reader. It automatically adds to the scanned URL information about network fingerprint.

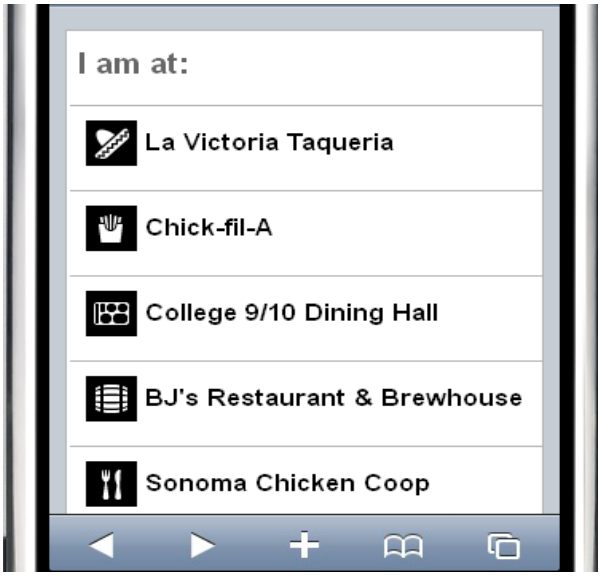

Figure 3. List of places

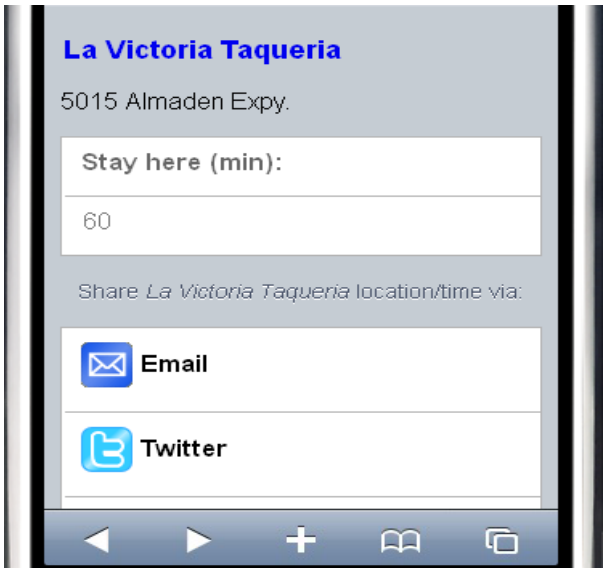

Figure 4. Share location info

Note, the fact that we do not need a specially installed application for sharing location info (it is just a link), lets us use the simple trick for distributing this approach. If 'share location info' application is just a link, we can use this link on our geo-signature. In other words, the shared party will send this link (with shared location info, of course) to the target party. In this case the receiver can use obtained link (link being presented in his incoming email/SMS) for the reply back his own location. Returning back to the original question 'Where are you now?', it is a way to ask a question and share the link ("application") for answer this question.

Returning back to the telecom operators position (it was heavy presented in [1]), we should mention Secure User Plane (SUPL) [13]. With SUPL positioning data is sent over the user's traffic channel using a secure IP connection between the smartphone (in standard it is called SET - SUPL enabled Terminal) and SUPL Location [14]. Assisted GPS (A-GPS) with SUPL uses assistance data received from the network to obtain a faster location calculation compared with GPS alone [15]. It lets telecom operators enrich communication services with location data with most effective way comparing with GPS/Cell ID approach presented in the original paper [1]. We can name Geo Signatures as a "telecom way" for sharing location info.

3. Related Works

As it is mentioned in study [16], mobile users want to disclose what they think would be useful to the requester or deny the request. In other words, it is difficult to dictate to users what do they need to share. Sharing location information is some like voluntary action. The same is confirmed in [17]. Privacy is one of the biggest concerns and obstacles for all location-based services. It is the biggest problem for the wide adoption of location-based services (LBS). In general, location information is preferably shared on a need to know basis, not broadcast.

Participants were biased against sharing their location constantly, without explicit consent each time their location is requested. This suggests that people are cautious about sharing their

location and need to be reassured that their private information is only being disclosed when necessary and is not readily available to everybody [18].

Authors in [19] suggest some requirements for LBS development. They could be combined into four high-level groups:

- a decentralized architecture, where as much personal information about an end-user is captured, stored, and processed on local devices owned by that end-user;
- a range of mechanisms for control and feedback by end users over the access, flow, and retention of personal information, to support the development of pessimistic, optimistic, and mixed-initiative applications;
- a level of plausible deniability built in;
- special exceptions for emergencies.

By our opinion, the key point for privacy concern for any existing LBS service is some third party server that keeps identities and locations together. We can vary the access levels, placement for the servers, but we could not remove the main part in privacy related concerns – the third part server itself. This third party server (servers) presents the weakest part of LBS. What can we do if the server itself is compromised?

There are plenty of papers devoted to privacy preserving in LBS. We can mention, for example, privacy preserving solutions based on cryptographic techniques that totally hide the location information in requests [20]. Another popular technique is spatial cloaking [21]. Spatial cloaking (in LBS) describes how to generalize the spatial information transmitted to the service provider. By receiving generalized locations only, the service provider can only return approximate results on the presence of close-by group members and their positions [22]. Of course, there is always a trade between generalization (increased privacy) and quality of the service.

The typical spatial cloaking service consists of two main components, the location anonymizer and the privacy-aware query processor [23]. The location anonymizer used to blur the exact location information. As the next step, anonymizer cloaks spatial regions based on user-specified privacy requirements (e.g., city level, street level, etc.) The privacy-aware query processor deals with the cloaked spatial areas rather than the exact location information.

The K-anonymity model supports user privacy profiles, indicate that the mobile user wants to be k-anonymous, i.e., not distinguishable among other k users. So, user can hide his location information within an area (e.g. not distinguishable on the street level) and (or) within the group of users.

Classically, location privacy is the ability to prevent other parties from learning one's current or past location [24]. By this reason pseudonymity is also deployed in LBS. In non-anonymous applications, for example, the user's identity is also transmitted to the service provider. According to the classification in [25], LBS can be non-anonymous, pseudonymous and anonymous. Of course, a non-anonymous LBS needs a user's location information and his real identity. A

pseudonymous LBS needs a user's pseudonym but not his real identity. Pseudonyms should allow LBS distinguish users. An anonymous LBS does not require a user's identity at all.

Our own development in this area (as a part of Geo Messages approach) includes distributed pseudonymous schema [26].

4. Safe Location Sharing

The above mentioned Geo Messages approach is simple to implement and it really eliminates identity revealing problems. But it is a pure peer-to-peer by the nature. By this reason, it now not very convenient to monitor several location data feeds simultaneously. It is not so easy to jump from one message to another.

Here we bring another peer-to-peer service that solves the privacy issues and lets you deal with several location feeds simultaneously. The main idea behind Geo Messages (Geo Signatures) was about separating location info and identity. Location info has been presented as a link in the message. Identity info has been presented as a header for the message. The second key moment is the fact that identity for the sender is known for the receiver.

Our approach WATN (Where Are They Now) [26] extends Geo Messages ideas for location sharing deals across multiple participants. It uses a distributed architecture which combines anonymous server-side data with local pseudonymity records.

In WATN we separate location info and identity data just in three steps:

a) assign to any participant some unique ID (just an ID, without any links to the personality). This ID should be assigned during the first request, without any registration;
b) save location data on the server with links to the above-mentioned IDs;
c) keep the legend (descriptions for IDs, who is behind that ID) locally. Each participant should save own legend. It is the key moment: the legend is personalized. One ID could have different legends simultaneously (just because each participant can set own legend for that ID).

In this case any participant may request location data for other participants from a third party server (as per sharing rules, of course), get data with IDs and replace IDs (locally) with legend's data. With such replacement we can show location data in the readable form. For example: name (nick) plus location. And in the same time the server (third party server for our users) is not aware about nicknames, because they will be saved locally for the each participant.

Technically is means that in this schema location server keeps two things.

a) Location information with meaningless IDs:
ID_1 -> ($latitude_1$, $longitude_1$)
ID_2 -> ($latitude_2$, $longitude_2$)
ID_3 -> ($latitude_3$, $longitude_3$)
etc.

Each such record presents a pair of current coordinates for users. Each user (participant) has been presented via own ID.

b) Social graph information. It describes who is sharing location information to whom. E.g.,
ID_1-> (ID_2, ID_3)
ID_3 -> (ID_1)
etc.

The above mentioned record states, for example, that user ID_1 shares location data with users ID_2 and ID_3

At the same time, any client (participant) keeps locally the own legend:
ID_1 -> (name or nick)
ID_2 -> (name or nick)

Note again, that in WANT model each client keeps own legend info. Clients are not aware about each other. So, there are no lists for clients and legends. And there are simply no ways to obtain list of all clients (participants) or list of all legends. Accordingly, there are no third-party server servers that know all about registered clients. It means, obviously, that in this model the same ID may have different legends. Each client technically can assign own name (nick) for the same ID. WATN's social graph saves information (links between participants) using our meaningless IDs only. And the human readable interpretation for that graph can vary of course from client to client. Actually, the last statement is probably very close to the real life, where the same person could be known under different names (nicks) depends on the context (e.g., one nick for work space and another nick for family space).

So, WATN keeps social graph, location and identity info in a distributed database. But it is distributed on the server-client level rather than on the server-server level.

WATN has been implemented as a mobile web application. HTML5 usage is significant there. The application uses W3C geo location [27] and local storage specification [28]. As per W3C documents HTML5 web storage is local data storage. It lets web pages store data within the user's browser.

The application itself is just an URL. As soon as any user reaches it (opens the web page), the application can restore user's ID from a local store. If ID is missing (it is a new client), the application can generate new ID and save it in the local storage. So, finally, each participant opened web page has got some ID. The application (JavaScript code on web page) can use this ID for passing requests to the server and obtain the social graph associated with this ID. This social graph (see above) contains IDs for participant which shared location with current user. So, we can request locations for collected IDs from the server. And as a final step – replace IDs with nicknames, using a locally stored legend. This chain is illustrated below:

ID (locally saved) -- Social Graph (server-side) – location info (server side) – IDs replaced with locally saved legend

Now let us describe how the legend is filled. Each participant can voluntarily decide to share own location information. For doing this he sends (via the same web application, of course)

email or SMS message with notification. Of course, the user can stop this sharing any time. This model uses peer-to-peer sharing: any user shares own location directly to another person. Actually, the location could be shared with any person with a known email address. So, this notification link plays a role of application's invitation too.

This notification is just a link to the same application with the parameter presents user's ID. As soon as notification is received and link (an application) is opened (Figure 5), the target party can accept (decline) this 'I will share the location with you' request.

As soon as this link is fired, WATN application (client) becomes aware about two IDs: own ID_1 for this client (it is restored from the local storage – see description above) and ID_2 from the "share with you" link (originated request ID). So, if notification is accepted, we can add social graph record (on the server) like

ID_2 -> ID_1
(participant ID_2 shares location information with participant ID_1)

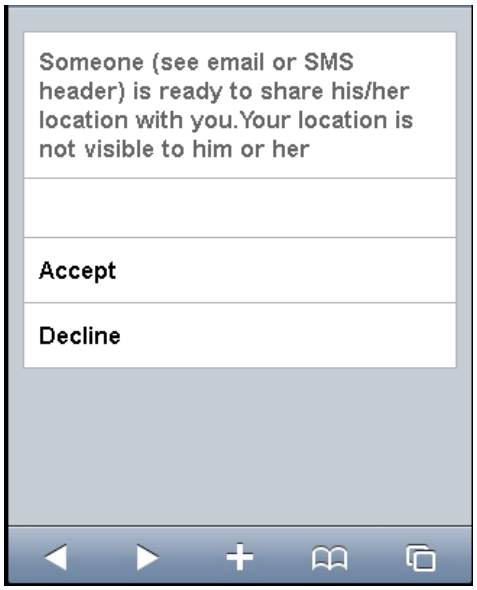

Figure 5. Share location request

But the notification link has been opened (see above) from any message (e.g., email). So, the receiver is aware about the author (right from the email's header). It means, that it could be asked also about setting some nickname for sender's ID (ID2 in the above presented example). It is illustrated in Figure 6.

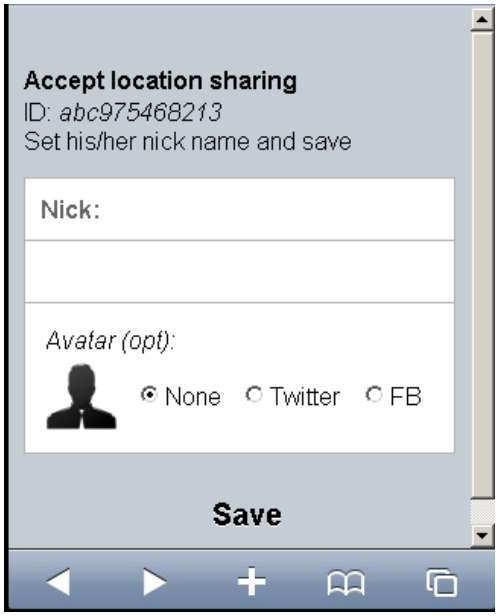

Figure 6. Set nickname

The whole schema work like a typical two phase commit in distributed database [29]. The application saves social graph info on the server and legend information locally.

The process of sharing location related data is automated. If we return back to the above described initialization process, the user's ID is always obtained as a first step. So, we can use this ID and save (on the server) its current location. So, as soon as the participants to whom this ID is shares own location will run the system (open the web page), they will get information about last know location for ID. As it follows from this explanation, location is saved only when the user reaches the system (opens the web page). So, it is like a check-in.

And as it follows from this explanation, notification link is, actually, the same Geo Message being described above. It uses absolutely the same idea. There is a geo-link in the message, and message's header describes the sender.

5. Conclusion

Geo Signatures approach presents a new way for sharing location information for mobile users. This approach uses the idea of peer to peer sharing and does not require from participants any presence in the social networks. The Geo Signatures approach uses existing communications channels for LBS and combines standard messaging with geo-location. It could be a very natural way for telecom operators to add internet services to their standard packages. It could be also a base for many additional mobile services, due to its easy to embed background.

References

[1] Namiot, D., “Geo messages”, *In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 14-19, 2010.
[2] Falk, H., “Applications, architectures, and protocol design issues for mobile social networks: a survey”, *Proceedings of the IEEE*, pp.2125-2129, 2011.
[3] Namiot, D., & Sneps-Sneppe, M., “Customized check-in procedures. In Smart Spaces and Next Generation Wired/Wireless Networking”, Springer Berlin Heidelberg, pp. 160-164, 2011.
[4] Yang, G., “Discovering Significant Places from Mobile Phones–A Mass Market Solution. In Mobile Entity Localization and Tracking in GPS-less Environnments”, Springer Berlin Heidelberg, pp.34-39, 2009.
[5] Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E. P., & Karagiannis, T., “The web of short URLs”, *In Proceedings of the 20th international conference on World Wide Web* , pp. 715-724, 2011.
[6] Hu, W. C., Kaabouch, N., Yang, H. J., & Wang, W., “Location-Based Services Using HTML5 Geolocation and Google Maps APIs”, *In Proceedings of the Midwest Instruction and Computing Symposium*, 2013.
[7] Guthery, Scott, Roger Kehr, and Joachim Posegga. “How to turn a GSM SIM into a web server”, *Smart Card Research and Advanced Applications*, Springer US, pp. 209-222, 2000.
[8] Lubke R., Schuster D., Schill A., “Mobilisgroups: Location-based group formation in mobile social networks”, *Pervasive Computing and Communications Workshops*, pp. 502-507, 2011.
[9] Namiot, D., & Sneps-Sneppe, M., “ Geofence and Network Proximity In Internet of Things, Smart Spaces, and Next Generation Networking”, Springer Berlin Heidelberg, pp. 117-127, 2013.
[10] Namiot, D., & Sneps-Sneppe, M., “Proximity as a service”, *In Future Internet Communications (BCFIC)*, pp. 199-205, 2012.
[11] Sneps-Sneppe, M., & Namiot, D., “Spotique: A New Approach to Local Messaging”, *In Wired/Wireless Internet Communication*, Springer Berlin Heidelberg, pp. 192-203, 2013.
[12] Namiot, D., Sneps-Sneppe, M., & Skokov, O., “Context-Aware QR-Codes”, *World Applied Sciences Journal*, vol,25, no.4, pp.554-560, 2013.
[13] Goze, T., Bayrak, O., Barut, M., & Sunay, M. O., “Secure user-plane location (SUPL) architecture for assisted GPS (A-GPS)”, *In Advanced Satellite Mobile Systems*, pp. 229-234, 2008.
[14] Namiot, D. “GeoFence services”, *International Journal of Open Information Technologies*, vol.1, no.9, pp. 30-33, 2013.
[15] Li, B., Mumford, P., Dempster, A. G., & Rizos, C., “Secure User Plane Location: concept and performance”, *GPS solutions*, vol.14, no.2, pp.153-163, 2010.
[16] Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J., & Powledge, P., “Location disclosure to social relations: why, when, & what people want to share”, *In Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, pp. 81-90, 2005.
[17] D. Wagner, M. Lopez, A. Doria, V. Kostakos, I. Oakley, and T. Spilitiopoulos, “Hide and seek: location sharing practices with social media”, *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, Lisbon, Portugal, pp. 55-58, 2010.
[18] Namiot, D., & Sneps-Sneppe, M., “Peer to Peer Location Sharing”, *The Eighth International Conference on Digital Telecommunications*, pp. 20-25, 2013.
[19] J. Hong and J. Landay, “An Architecture for Privacy-Sensitive Ubiquitous Computing”, *MobiSys'04*, Boston, Massachusetts, pp. 177-189, 2004,

[20] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., & Tan, K. L., "Private queries in location based services: anonymizers are not necessary", *In Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ACM, pp. 121-132, 2008.
[21] Chow, C. Y., Mokbel, M. F., & Liu, X., "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments", *GeoInformatica*, vol.15, no.2, pp.351-380, 2011.
[22] Mascetti, S., Freni, D., Bettini, C., Wang, X. S., & Jajodia, S., "On the impact of user movement simulations in the evaluation of lbs privacy-preserving techniques", *In International Workshop on Privacy in Location-Based Applications*, CEUR-WS, pp. 61-81,2008.
[23] Mokbel, M. F., Chow, C. Y., & Aref, W. G., "The new Casper: query processing for location services without compromising privacy", I*n Proceedings of the 32nd international conference on Very large data bases*, VLDB Endowment, pp. 763-774, 2006.
[24] Cheng, R., Zhang, Y., Bertino, E., & Prabhakar, S., "Preserving user location privacy in mobile data management infrastructures", *In Privacy Enhancing Technologies Springer Berlin Heidelberg*, pp. 393-412.
[25] Beresford, A.R., Stajano, F., "Location Privacy in Pervasive Computing", *IEEE Pervasive Computing*, vol. 2, no.1, pp.46-55, 2003.
[26] Namiot, Dmitry, and Manfred Sneps-Sneppe, "Where Are They Now–Safe Location Sharing", *Internet of Things, Smart Spaces, and Next Generation Networking*, Springer Berlin Heidelberg, pp.63-74, 2012.
[27] Rost, M., Cramer, H., Belloni, N., & Holmquist, L. E., "Geolocation in the mobile web browser", *In Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, ACM, pp. 423-424, 2010.
[28] Casario, M., Elst, P., Brown, C., Wormser, N., & Hanquez, C., "Html5 local storage", *In HTML5 Solutions: Essential Techniques for HTML5 Developers* , pp. 281-303, 2011.
[29] Thomson, A., Diamond, T., Weng, S. C., Ren, K., Shao, P., & Abadi, D. J., "Calvin: fast distributed transactions for partitioned database systems", *In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 1-12, 2012.