Enhancing Customer Privacy for Commercial Continuous Location-Based Services

Jens Bertram, Carsten Kleiner, and David Zhang*

Fachhochschule Hannover, Fakultät IV
Ricklinger Stadtweg 120
30459 Hannover
Germany
jens.bertram1@stud.fh-hannover.de, ckleiner@acm.org,
david.zhang@fh-hannover.de

Abstract. The likelihood of consumers to use commercial location-based services significantly depends on their perception of privacy protection by the service provider. In this paper we discuss existing privacy-enhancing architectures for LBS and argue that they are either not applicable or insufficient for services requiring continuous location queries. In order to offer such services providers often prefer to refrain from storing fine-grained location information of their customers. Instead some form of data aggregation on the mobile device is used and only aggregated information is released to the service provider upon approval of the customer. This leads to a rather loose integration of the mobile device into the backend process. We explain our concept for such an enhanced architecture and discuss some implementation aspects. The work has been motivated by a specific application scenario in an insurance context for which we are currently developing a prototype.

Keywords: Enterprise Location-based Service, Privacy, Mobile Device, Continuous Query, Navigation, Data Aggregation.

1 Introduction

Location-Based Services (LBS) have been considered a very important application setting in an increasingly mobile society for a couple of years. In recent years most of those services are rather simple consumer-oriented services being offered for free or at a small charge (e.g. find the nearest restaurant of a specific type from a given location). Such services typically do not require a previous registration of the consumer with the provider. This will most probably change in the near future because there are also very interesting and commercially promising business applications based on LBS. These can be pure business applications such as automated fleet management which are already in place in some companies. But it

* The participants in the research project *DaSimoD* leading to the results presented in this paper have been funded by a grant of the *German Federal Ministry of Education and Research* under grant no. FKZ17N5309.

Y. Cai et al. (Eds.): Mobilware 2010, LNICST 48, pp. 326–337, 2010.

might also be business to consumer applications where both parties gain advantages by the service. An example is the Pay-As-You-Drive scenario for a car insurance company which has been used as motivation for this paper. It is described in more detail in section 2.

LBS are of particular interest in the context of devices such as mobile phones or PDAs since modern devices typically provide some kind of position technology on the device. As many LBS are used based on the current position of the client and the mobile device is carried by the consumer most of the time, it is very simple to use LBS from a mobile device.

Unfortunately in this setting large privacy concerns arise; most people would not want a service provider (maybe even unknown to them) to be able to record their position at any time when they consume the service. By combining or aggregating such information potentially from different services used by the same device detailed motion profiles could be assembled. Note that even without a registration the identity of the client might be revealed to the service when submitting a request (e.g. by IMEI, phone number, MAC address). In addition even when LBS are used anonymously or with pseudonym it is possible to determine the identity of the user through the indirect location privacy problem. This is even more true when using passive position technologies.

There has already been some research on how to offer LBS privacy-friendly. We will review the existing suggestions in section 3. Each has its individual strengths and weaknesses. In this paper we introduce an example business application that relies on continuous location information. For this application scenario which will be described in section 2 we will argue that none of the existing approaches is both applicable and sufficient. Consequently we present an extended privacy-friendly architecture for our and similar application scenarios in section 4. Neither our nor any of the existing approaches offers perfect privacy at an acceptable computational cost; we explain why in sections 3 and 4. In section 5 we briefly discuss some implementation aspects for our architecture, before we conclude in section 6 with a summary and some ideas for future work.

2 Application Scenario “Pay-As-You-Drive”

Our research has been motivated by a specific application scenario for mobile LBS. This scenario leverages mobile devices, location information and LBS to allow for a “Pay-As-You-Drive” car insurance model. It refers to a specific tariff option for car insurance. The insurance company offers a reduced tariff for drivers which comply with certain rules. An example might be young drivers that do not drive at night where the risk of accidents for them is particularly high. Or drivers that claim to always stick to the posted speed limits might be reimbursed.

More generally speaking “Pay-As-You-Drive” is a car insurance model, where the insurance premium depends on the driving behavior of the policyholder (e.g. type of road, time, speed, break of speed limits, driven distance, etc.). Based on this information, the insurance company may calculate the risk more accurately, which could result in flexible and potentially lower costs for the policyholder.

Obviously this application has potential to put the clients' privacy at risk. Our goal is to find a solution that protects the user's privacy, so that he will not reveal more about himself than he has agreed to by contract. This means that only data needed for the risk calculation is collected and aggregated in a way that is sufficient to fulfill the requirements as provided in the contract. Additionally the system architecture should prevent the ability to create profiles of movement of the different users. For example it is not necessary to know where and when exactly a user/car was, it might be sufficient to know the type of road, distance driven and average speed. Note that we consider online access to road information mandatory due to space constraints on the mobile device combined with the requirement for a high degree of actuality of data and metadata.

On the other hand the information assembled in such services is so extremely sensitive, that not even insurance companies want to have detailed information about the client on their servers in order to prevent being forced to reveal it (e.g. to governmental parties).

We have designed a privacy-friendly system architecture for offering Pay-As-You-Drive insurance contracts. Note that this task is not restricted to use the classical point-based LBS services internally. But it is also an option to use trajectories or similar extended and/or aggregated geographic data as parameters for the services in the architecture. Thus the architecture may be used for services that go far beyond classical LBS as well. Some different possible options will be discussed in section 4.

3 Related Work

The increasing use of LBS has created new privacy risks for users. Using LBS involves confiding personal data like current location to the location-based service provider (LBSP). This data may convey personal details about the client. Even when using anonymous LBSs (a service that does not require users to convey their identity) the identity of the client may be revealed from the location data by the indirect location privacy problem. This has been nicely explained in [9]. A lot of research has been done in this area to handle privacy problems. Among the methods to protect the privacy are location k-anonymity [5], false dummies [1], false locations [2] and private information retrieval (PIR) [3, 4].

The k-anonymity technique which is described in [5] protects privacy by providing anonymity for clients based on trusted third party architecture. This approach implies that a client uses a service in an anonymous way i.e. he does not send his real identity and may also hide his network address by technologies like onion routing. Additionally the trusted third party provides k-anonymity (i.e. an individual request may not be distinguished from at least k-1 other requests) and so ensures that the client cannot be practically identified based on position data. To achieve this, precision of location information is reduced both in spatial and temporal dimensions. The degree of reduction is based on statistics and is chosen in a way that the location information sent to the service may have come from k different entities, where k may be chosen so that it is not practical to identify the actual client. This approach works well for snapshot queries. But if a client continuously uses a LBS, anonymity may be reduced by maximum movement bounds. Moreover in our use case we require precise

location information in order to be able to use the service in a meaningful manner. Approximate information is not sufficient for navigation services.

Temporal cloaking in contrast is possible to a certain degree. For example the time of the day does not have to be precise to minutes or maybe even hours. Because only a weekly or monthly report is required, the time of the day may be important but not the exact date. Details depend on the particular service and contract.

The idea how to protect location privacy with false dummies is to hide the true location among a number of false locations. Whenever a client uses a LBS he will not only request this service for his true location but also for some dummy locations. Because the service cannot distinguish the true from the false locations, privacy is protected. This works well for snapshot queries. If on the other hand a LBS is used frequently, true locations will form a route and dummies could be easily identified. Therefore a smart algorithm has to be used to generate dummies in this case. In [1] it is suggested to remember the last dummies sent to a service and to generate new dummies in the neighborhood. This will make it more difficult to identify dummy requests, because they will also form a route. In our use case this will not be sufficient because we track cars driving on roads. The true route will be easily identified because it is the only one following a road. The dummy generating algorithm would need to be improved to generate dummies in the neighborhood that are on a road as well. This will require the client to have a complete map available. To further improve the quality of dummy requests in this use case the algorithm may also take constraints like one way streets and speed limits into account. In our use case clients do not have road maps available and so dummy requests would not be generated in a smart way. That is why dummy requests cannot be applied in this case.

Another privacy protecting technique is SpaceTwist [6]. It uses false locations for nearest neighbor queries sent to the LBS. The client specifies the false location called an anchor. Then he queries the service for the nearest neighbors of this anchor. The service will return points of interest in ascending distance of the anchor. The client terminates the request when the answer covers the area around its position sufficiently. In our use case the service would have to return fragments of roads with consistent metadata until the client identifies a fragment which includes its position. SpaceTwist has only been studied for snapshot queries, not for continuous queries. For continuous queries the anchors may form an approximate route for the movement of the client. This again would reduce location privacy.

In [3] a framework based on private information retrieval (PIR) is presented. PIR is based on the Quadratic Residuosity Assumption which states that it is computationally very expensive to find the quadric residues of a large number $N = q1 * q2$ where q1 and q2 are prime. This framework requires that the database is indexed appropriately. A request to a LBS using PIR does not contain spatial information. The points of interest are retrieved based on an object index. Therefore location privacy can be guaranteed here. Unfortunately a PIR request is a quite costly operation both in terms of computational complexity on the server side as well as regarding message sizes. Our use case implies a large number of clients accessing the LBS in high frequency. In [4] a scalable approach leveraging PIR called SPIRAL is described. This framework is based on trusted third party architecture and blinds the LBS so it will not know which objects have been sent to the client. While this might remedy the complexity issues on server

side it does not reduce messages sizes significantly. The latter is not acceptable for services on mobile devices.

In summary, for LBS used continually, it becomes even more difficult to protect privacy. All of the single locations form a route. False dummies may then be easily identified. Also false locations will form an approximate route. Because of a high frequency of requests PIR may not be applicable. And k-anonymity achieved by cloaking locations to regions will reveal maximum movement bounds. Apparently there is no perfect solution to protect privacy for a continuous LBS application like Pay-As-You-Drive. In the following chapter we will propose a system architecture for this application which respects the users' privacy to a high degree and is a good tradeoff between privacy and performance.

4 Privacy-Friendly System Architecture

In this chapter we present and discuss the architecture we designed for a Pay-As-You-Drive application. Because until now there is no ultimate solution to protect location privacy for continuous queries we propose a somewhat pragmatic solution. In the first section we present and discuss the general architecture of commercial applications for which we propose privacy enhancements. The Pay-As-You-Drive application fits well in this architecture. The second section focuses on measures to provide practical location privacy by means of the Pay-As-You-Drive example. These measures can be transferred to other applications that fit in the general architecture.

4.1 General Architecture

The architecture described in this section mainly consists of three different and independent systems (cf. Fig.1). A mobile device (e.g. a smart phone) provides raw GPS position data like latitude, longitude, altitude and speed. It integrates position data with additional meta information and aggregates it to create a report for a business partner. This might be for example a report about driving behavior which is sent to an insurance company. The required meta information related to the position like road-type and speed-limit are provided by an external service provider (LBS in Fig. 1). They are periodically requested by the mobile device. The meta information is typically stored together with geometries in a GIS/Spatial Database, which allows fast detection of a specific road from a given position based on road geometries. The third part of this architecture is the business partner system, which receives the reports. For the Pay-As-You-Drive example this is the insurance system which stores the user-reports and calculates the risk and insurance premium accordingly. Note that location privacy against the business partner is considered sufficient by providing only aggregated information in weekly or monthly reports. This assumption is reasonable in all practical cases. Thus the remaining privacy issues are against the LBS provider. Since the most relevant part concerning the users' privacy in the Pay-As-You-Drive concept is related to the interaction between mobile device and LBS, the following part focuses on this issue.

To be able to create an accurate report, the mobile device will continuously request the LBS in short time periods (about seconds). This behavior is similar to a tracking

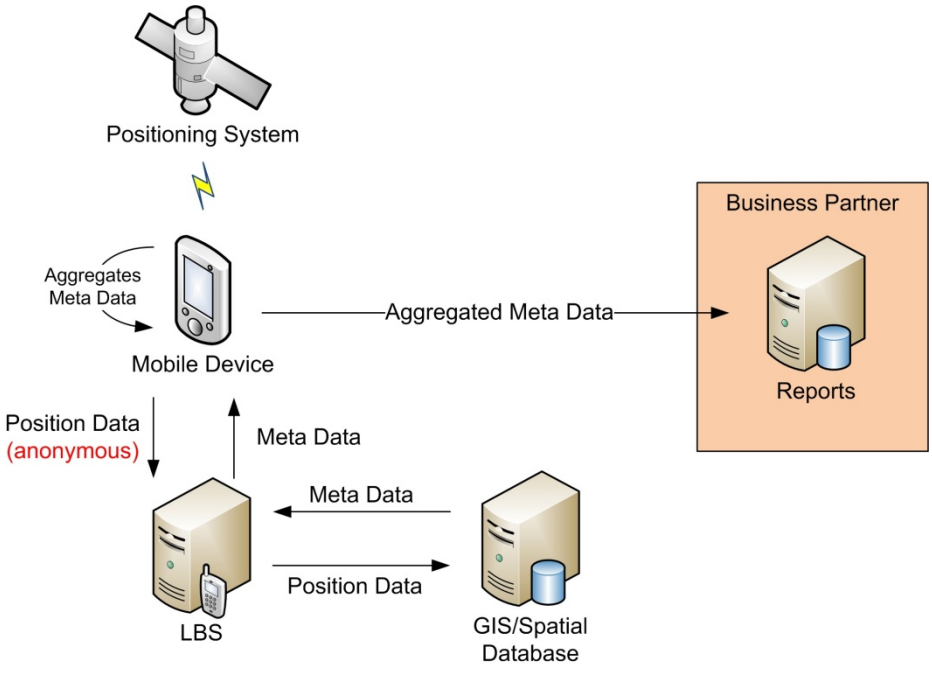

Fig. 1. System Architecture

system, where a car's position is periodically reported to a monitoring system to be able to locate a specific vehicle. It is this side effect we want to prevent; the LBS shall not be able to locate or track a specific vehicle/user. Please note, that the results of the requests are not needed immediately; they are aggregated to a report which is only evaluated in regular time intervals.

By using an anonymous connection to the LBS, which implies that no authentication is required and the service provided by the LBS is free of charge, an incoming request gives no information about the users' identity. Otherwise, if the service is not free of charge, authentication is necessary but it would be sufficient if an authentication mechanism is used that authenticates a user as member of a group and not providing any further information about his identity. But using an anonymous connection to the LBS is not enough. The user may still be identified by the indirect privacy problem. Because of the high frequency of client requests they will form a route which might help to identify the user, for example the regular route from home to work place. Once a client is identified, the route can be used to find out where else that person went.

After receiving the meta information according to the previously sent position the client (mobile device) aggregates the data. As for the Pay-As-You-Drive application the report describes the users driving behavior by providing information about several parameters such as distance driven on a specific road type, violation of speed limits and average accident rate. Details depend on the specific contract. The reports are created for a certain time period and are periodically submitted to the insurance system.

While the connection to the LBS is made anonymously, there still exists an indirect location privacy problem. As the LBS receives all position data of the clients it could store the data, analyze it and search for certain patterns in the data set to identify individuals. In the next section we present suggestions for improving on this situation.

4.2 Considerations to Improve User Privacy

The architecture presented in chapter 4.1 has one major drawback: it continually sends its position to the LBS and thus it is vulnerable to the indirect location privacy problem. In this section we present an improved concept protecting location privacy based on the same simple client-server architecture: a client communicates with an anonymous location-based service. The improvements proposed in this section are made by means of the Pay-As-You-Drive example, but can be transferred to other applications which fit in the general architecture described in the previous section.

The Pay-As-You-Drive application depends on accurate location information to measure the driving profile of the policy holder. But the report will only be assembled periodically i.e. at the end of the month. This fact can be used to increase location privacy. The basic idea is to remove temporal correlation of the requests. The location data send to the LBS is required not to contain any temporal attributes of the specified position. To achieve this, the request will not specify the time at which the client was at the location nor the current speed or direction and it will also not be sent immediately. Because the report only needs to be assembled e.g. monthly sending can be deferred for hours, days or even weeks. Also the meta data for locations will not be retrieved in the sequence they were recorded but in random sequence.

The result of this method is that it becomes difficult to impossible to identify routes from the locations sent to the server and thus it is difficult to identify the person that sent the request. If there was only one client using this anonymous LBS in a city, this method would not protect his anonymity against the indirect privacy problem. If the LBSP would display all the requests he received on a map, he would see where the person went and even how often. From this knowledge he could derive the identity of that client. But it would not be possible to determine at what time the client was at a specific place or in which direction he moved from there. If there are two or even more clients in that city, it will become more difficult to identify them. Also it becomes difficult to determine which client went to a specific place that is not already known to be related with one of the clients. The degree of anonymity increases with the number of clients moving around in a common area.

Until the LBS is queried for the meta data for a specific location the client device will have to store the location data. The local storage must retain the sequence of locations and may also contain a timestamp for each entry. The client will choose randomly from stored location entries to issue a request. After it received the response from the LBS for some location entries that have been recorded in sequence, it can aggregate metrics into the corresponding report. Now the inner entries of the location sequence are not needed anymore and are removed from local storage. The first and last entries of a sequence will still be needed to compute metrics for the neighboring location sequences.

Because locations may be stored for a longer period of time this allows for some optimizations to reduce network traffic and to increase privacy even further. When the

client asks the LBS for metadata for a specific location the service could include the geometry of the area associated with this metadata. Then the client does not have to query for all the neighboring locations which would yield the same result. They can now all be answered by a single request. These may include locations that have been recorded in the area over a couple of days. Thus every request will resolve coherent sequences of location entries that can be aggregated and removed from local storage.

The next step is to improve the algorithm choosing the next location entry from the local storage to be evaluated by the LBS. To minimize the overall number of requests the algorithm could take the location data of the entries into account. It could favor to choose locations that are in a less frequently used area. Thus location entries for frequently visited areas would accumulate in local storage and be answered with a single request. The effect is that frequently visited areas are not frequently sent to the LBS. Obviously these measures reduce the number of requests sent to the service and therefore also reduce the data that LBS could collect about the client.

Our concept uses temporal cloaking and is applicable to Pay-As-You-Drive, because the client application does not need short-term answers. In contrast to the k-anonymity concept (e.g. in [5]) it is not based on a trusted third party architecture. On one hand this simplifies the overall architecture. On the other hand it is not possible to ensure a given degree of anonymity. The degree of anonymity depends on the number of clients using the service in a common area. However, it is possible to derive this number from the number of policyholders within a common area.

Note that in order to ensure a certain degree of anonymity a trusted third party may also be employed in our architecture. In this sense the approaches are orthogonal and may be combined if desired. But the practical improvement of privacy might be rather small, especially in urban areas, when adding k-anonymity to our architecture. Thus the tradeoff between simplicity of the architecture and privacy increase would favor using our approach purely. The optimizations described above have the potential to significantly reduce the number of requests and so cloaking the movement of the client. Altogether this is a practical approach to implement Pay-As-You-Drive and other similarly structured commercial services with a reasonable degree of privacy.

5 Implementation Considerations

The previously described architecture relies on a continuous communication of the mobile device and the LBS. This is necessary to receive additional information based on the current location and to be able to record the policyholders driving behavior.

Timing of Messages

The accuracy of the aggregated data depends on different aspects. As the mobile device itself does not store location related information like the geometry of a road, the driven distance between two different positions has to be interpolated, which might not represent the actual driven distance, if the two positions are far apart. This means the higher the resolution of the route (positions/time) is, the more accurate is the aggregated data. As positioning itself and data aggregation run in constant time, the resolution of the route mainly depends on the time between position data is sent

and the service response is received. This in turn depends on the structure of data, the network quality and bandwidth and the protocol used for message exchange.

The data sent by the mobile device to the LBS mainly consists of the values of latitude and longitude. Optionally the altitude and velocity could be included to distinguish nearby roads like parallel running roads or bridges. Based on this position data the service responds with related meta data. The report sent from the mobile device to the insurance system contains characteristics about the driving behavior.

Today's smart phones and mobile devices provide fast network connections and are designed for an "always-online" usage. But there are still areas and situations where only low bandwidth or even no network service is available. As a fallback procedure in this case all position data combined with a timestamp could be stored in local storage on the device and send later (without timestamp), when the network is available again or the bandwidth increases. In case of the improved concept described in section 4.2 this procedure is an implicit part of the concept already.

Message Protocol

The processing time and so the resolution of the route also depends on the protocol used for the message exchange between the different systems. There are several requirements affecting the choice of a protocol like data structures, implementation effort, reusability and security. In case of Pay-As-You-Drive the previously described data is structured in a static and simple way without complex and varying elements and attributes, which is well suited for the use with different protocols. We will now describe different protocols for the data exchange and their advantages and disadvantages in the Pay-As-You-Drive or similar scenario.

Based on the statically structured data a binary format, e.g. an ordered set of key-value pairs, or a specifically structured XML format could be used to describe data. The main advantage of the binary format is the reduced data volume and the faster processing speed for parsing the message, but it requires a specific and matching implementation on both client and server side. Both, the binary and an XML based format could be sent through a socket based connection, without the need of establishing a new connection for each piece of position data, which decreases the transmission time. As a drawback this approach requires a specific implementation and further network configuration, e.g. firewalls, which reduces the reusability of the service for future applications. Also security aspects like encryption and authorization are not provided automatically and would need to be implemented within both the client and service application.

Another approach for messaging and data exchange is the use of Web Service technologies like REpresentational State Transfer (REST) [11] and SOAP. A conceptual comparison of REST and SOAP is e.g. given in [7].

REST uses a HTTP connection for data transmission and the service and service method is identified by an URI in the HTTP header. As an improvement the use of a HTTP connection does neither require a specific interface implementation nor a specific network configuration since HTTP traffic is typically allowed in most networks. Most mobile device platforms will provide an API for HTTP connections and thus support RESTful Web Services directly. The message payload format can be chosen as desired in this case, for example the previously described binary format or

an XML based format could be used. An advantage of the use of XML is that message elements are reusable and customizable. Thus services can be reused for different purposes, for example by providing a list of points of interest (POI) for a given location or other location based services. As with sockets the advantage of a binary format is its smaller data volume which might help for mobile devices with rather thin bandwidth. It may be used for REST with the same disadvantage of reduced reusability as before. Regardless of the format the data volume is increased by the HTTP header representing a message overhead when compared with sockets. From a security point of view end-to-end encryption is available for REST by using HTTPS. Any further security mechanisms such as authentication have to be implemented within the application.

Similarly SOAP can be used based on HTTP as transport protocol, but several others like FTP and SMTP are also possible. As the main difference SOAP itself is based on XML and defines a message format containing a header and a body part with the payload described in XML. The binary format could still be used with SOAP as a value of an XML element, but this would lead to the same disadvantage as before. The advantages of the use of XML are the same as described for REST, mainly the reusability of the service for different areas of application. Differing to REST not only the HTTP header but also the whole SOAP message implies large overhead and thus increases the transmission time. The message overhead of SOAP and REST is compared in [8]. As an advantage the WS-Security Specification [12] for SOAP specifies security aspects like encryption and authentication. Unfortunately the support for SOAP is not wide-spread on mobile device platforms nowadays, especially the support of WS-Security is far from perfect.

As an additional improvement an OGC OpenLS Standard [10] compliant service would increase the ability to reuse the service in a large area of applications and also increases the ability to choose a service from a set of equivalent services (Provider Change [9]). This may be combined with either REST or SOAP but would incur an additional significant message overhead.

Fig. 2. Increased flexibility and message overhead for service implementation protocols

A remedy to the message overhead issue would be to send a set of positions within a single request and to receive a set of related meta-information in a single response. This results in fewer but bigger messages within a given time. The overhead of HTTP and SOAP header is significantly reduced as for n positions sent at once only a single header is included instead of n headers in n messages. For the privacy aspect n should be chosen small enough to only represent a short part of the route and to decrease the risk of the indirect location privacy problem.

As shown in Fig. 2 the message protocol to be chosen depends on the particular application. There is a classical tradeoff between the desired degree of flexibility of the service on one hand and the message overhead on the other side. In the current situation we would suggest to choose an XML-based REST protocol. This observes

current bandwidth restrictions on one hand. But on the other hand it is rather easy to extend it to more flexible protocols in the near future when the bandwidth restrictions are no longer relevant.

6 Conclusion and Future Work

In this paper we have presented a promising commercial application for mobile LBS in the Pay-As-You-Drive application for car insurance. We have explained why privacy issues are of particular interest in this as well as other mobile LBSs. Thereafter it has been shown that none of the existing solutions for observing privacy in LBS is applicable and sufficient in this context (see [13] also). Consequently we have proposed a new architecture which is a good solution from both a user privacy point of view as well as from a pragmatic perspective. Location privacy is enhanced in two stages: on one hand we introduce a party independent from the original service provider; this LBS provider may offer its services in the context of several different commercial applications. In order to improve on this classical trusted-third-party architecture we introduced the second stage: data sent to the LBS provider is anonymous and temporally blurred (cf. section 4.2). Thus the LBS provider does not have to be trusted as it only receives less sensitive data than in the original scenario.

We have suggested several improvements to the basic architecture which may be added to increase privacy as well as efficiency. Finally we have discussed some important implementation issues regarding the timing of messages as well as protocols to be used. These issues are specific to mobile devices and are thus subject to continuous change as technology evolves. Therefore we have presented a general discussion with recommendations in the current situation.

Currently we are working on a prototypical implementation of the basic architecture as presented in sections 4 and 5. The client components are developed for Android based smartphones. We decided to use multi-purpose client devices as opposed to proprietary hardware as in many currently available tracking applications. This is due to the fact that they contain all necessary technology and will soon have a sufficient market share. Thus we don't expect proprietary devices to survive for a long time anymore. In the near future we plan to extend the prototype to include the improvements explained in section 4.2 as well.

Apart from Pay-As-You-Drive there are many other similarly structured commercial LBS (namely ones involving continuous queries) for which the findings in this paper are relevant. Therefore it would be interesting to extend the prototype to different services in this area and evaluate the advantages of flexible protocols further.

From a commercial point of view there has to be a business model for the LBS provider in our architecture. Apart from a classical ad-based business model we also see an open source offering e.g. based on Open Street Map as well as a provider (e.g. the insurance company) sponsored model for LBS providers. In the latter case a LBS provider could offer basic services for many different applications thus remedying the potential influence of a single application provider.

Our approach only works for client based positioning which is not really a restriction nowadays. Nevertheless an important improvement would be to extend the architecture to include the use of a tracking platform which is still widely used and important for certain kinds of enterprise applications. On those platforms positioning of devices is initiated by the tracking platform.

References

1. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Proceedings of International Conference on Pervasive Services, ICPS 2005, pp. 88–97. IEEE, Los Alamitos (2005)
2. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, MobiSYS 2004, p. 177. ACM Press, New York (2004)
3. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.: Private queries in location based services. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, p. 121. ACM Press, New York (2008)
4. Khoshgozaran, A., Shirani-Mehr, H., Shahabi, C.: SPIRAL: A Scalable Private Information Retrieval Approach to Location Privacy. In: 2008 Ninth International Conference on Mobile Data Management Workshops, MDMW, pp. 55–62. IEEE, Beijing (2008)
5. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proceedings of the 1st International Conference on Mobile systems, Applications and Services, MobiSys 2003. ACM, New York (2003)
6. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In: IEEE 24th International Conference on Data Engineering 2008, pp. 366–375. IEEE, Los Alamitos (2008)
7. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. ”big” web services. In: Proceeding of the 17th International Conference on World Wide Web, WWW 2008, p. 805. ACM Press, New York (2008)
8. Aijaz, F., Ali, S.Z., Chaudhary, M.A., Walke, B.: Enabling High Performance Mobile Web Services Provisioning, pp. 1–6. IEEE, Los Alamitos (2009)
9. Decker, M.: Location Privacy-An Overview. In: 2008 7th International Conference on Mobile Business, pp. 221–230. IEEE, Los Alamitos (2008)
10. Mabrouk, M.: OpenGIS Location Services (OpenLS): Core Services. Open Geospatial Consortium Inc. (2008)
11. Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures. Ph.D:180 Building (2000)
12. WS-Security specification (March 2004), `http://www.oasis-open.org/specs/index.php#wssv1.0`
13. Kulik, L.: Privacy for Real-time Location-based Services. The SIGSPATIAL Special 1(2), 9–14 (2009)