LBS System Architecture with Privacy

D. B. Aredo, H. J. Rivertz and J. I. Vestgården
Norwegian Computing Center
P. O. Box 114 Blindern, N-0314 Oslo, Norway
{aredo,rivertz,jornv}@nr.no

ABSTRACT

In this paper, we present results of a research work on architectural and implementation issues related to location-based services (LBS). The architecture provides mechanisms for protecting privacy of actors involved, especially the privacy of the consumers of services provided by an LBS system. This work focuses on an LBS system architecture and investigates its scalability, performance, and security mechanisms by analyzing, designing, and simulating the architecture. Based on the results obtained, we propose adaptation to the architecture and specify communication protocols that satisfy the security requirements.

Keywords: LBS architecture, Security, Privacy

1 INTRODUCTION

A basic difference between mobile systems and ordinary stationary computer systems is that the actors of the former do not have a fixed physical location - they bring the communication units along with them to office, on bus, to home, and wherever they go. This makes physical location a basic parameter of mobile system, a fact most users of such systems are aware of. This location information is important for the users of the systems, which can be seen from the simple observation that most cellular phone conversations start with the question "where are you?".

The term location-based services (LBS) is used to denote services where location is an important parameter. The service can be an extended or improved version of an already existing services, or a completely new kind of service. So far the location based technologies and possibilities are not very well developed and exploited, although existing cellular networks such as GSM already natively contain such capabilities. In near future, the GPS technology chips might also be included in cellular phones, as well as other kinds of communication equipments that can make positioning with tremendous accuracy feasible.

The general setting of our work is that of some entity, e.g. a cellular phone network operator, that can obtain fairly accurate location data. This data is then 'sanitized', in order to reduce its accuracy before it is released to the entity that delivers the location-based services.

Lately, network providers have begun to offer a limited number of location based services. In the USA, the FC-911 mandatory states that network providers shall be able to track all cell phones so that the 911 emergency service can get the caller location in order to provide quick emergency responses. In Norway, the two major cellular network providers, Telenor and NetCom, have begun to provide services where users can locate each other. However, today's location based services are rather limited and it is possible to think of a vast number of services based on location, not only *pull* kind of services as those mentioned above, but also *push* services, where a service provider pushes a service based on location information. A key question is then how to protect privacy, i.e. to control who shall be able to get your location information and under what circumstances? The solution proposed by Snekkenes in [16] is that the distribution of location information shall be controlled by a *policy* formed by the locatable object owner. The policy is kept at an entity called *policy custodian*, which may not have commercial interests in the location information. In order to ensure the freedom of users to choose policy custodians where their policy will be stored, a register, called the *policy custodian directory*, must be established. The register tells interested actors such as the location provider where a policy of a given locatable object is stored.

1.1 The problem statement

Location-based services are becoming more common in our daily activities. Surveys show that a great deal of users of LBS services, and service providers are concerned about their privacy [1],[17],[18]. The availability of the services is also a crucial factor for the success of a business involving LBS. Hence, there is a need for an LBS system architecture providing security mechanisms that protect privacy by enforcing appropriate policies.

The objective of the work presented in the sequel is to investigate an architecture of LBS systems and to establish a technological platform suitable for running user-related experiments. This will be achieved by validating, and possibly adapting different variants of the architecture proposed by Snekkenes in [16]. The following research questions are investigated:

- Is the architecture sufficiently scalable – if not, how can it be adapted?
- In operational settings, what are the key requirements - scalability, robustness, realizability, or security?
- Are there requirements that may act as 'show stoppers'? If so, can the architecture be adapted to address this?
- What protocols should be used? Is it necessary to develop new protocols, or can the existing protocols handle the situation?
- How much load will the architecture put on the network infrastructure and components?
- Where and when can a network congestion may occur?

Based on the results of investigating the above issues, variants of the architecture and scenarios are proposed and evaluated. Some of them will be discussed in later sections.

1.2 The outline of the paper

The rest of the paper is outlined as follows. In Section **2**, a brief review of related work is presented. Section **3** is devoted to discussion of major architectural issues. In Section **4**, we discuss simulation of different scenarios in the architecture. Finally, in Section **5**, we conclude and present issues that need further investigation.

2 RELATED WORK

The context and the main focus of this work is the architecture of LBS systems proposed by Snekkenes in [16]. The architecture and the location privacy policy language presented by Snekkenes will be investigated and used as a source of concepts as well as user scenarios and architectural designs throughout this paper.

Authentication mechanisms are essential for the final implementation of any privacy protecting system, and Hirose *et al.* [6] discuss an anonymous user identification mechanism. A language for formal specification of location policy is essential for a solid privacy policy enforcement. *Ponder* is an example of already existing policy languages which might be extended to handle location policies [10],[5],[15].

A radio-frequency tracking system to be used inside buildings is described in [2], which claims a spatial accuracy of less than two meters. For the paranoid, the Observer describes a UK defense tracking system [3], which will be able to track cell-phones directly (not through network provider).

The interest of business companies in location based services is growing, all aiming to get shares somewhere in the location based services market. Telecommunications companies are interested in extending existing cellular phone systems with spatial tracking capabilities. In addition, some companies sell dedicated tracking devices such as the TruePosition [19] and the ABS Digital Angle [4]. The Environmental Systems Research Institute (ERSI) [9] is interested in the geographical information system (GIS) necessary to provide location based services. Oracle [12],[11] focuses on database used in the infrastructure and for the GIS.

3 ARCHITECTURAL ISSUES

The system is intended to provide a 'Universal' location service, where location data typically is produced by the GSM/UMTS network operator. It is assumed that the location provider has some interest, e.g. for compliance with privacy laws, in allowing subscribers to have some control on what location information is released, when and to whom. There is at least one country, namely Norway, where a governmental organiza-

tion, called the Brønnøysund Register Center [14] that runs a free public register service. Individuals can register that they do not want to receive direct marketing or sales phone calls or mail. In some sense, this offers individuals of Norway a privacy policy in the sense of [13]. The system architecture proposed in [16] includes the following main entities:

- *Personal Location Privacy Policy*: Statement of what can be released to whom and when. Each located object will have an associated policy.
- *Policy custodian*: Where the policies are stored and possibly also enforced. The set of permitted operations may include read, write, modify, and query etc. depending on the identity of the requesting entity.
- *Policy custodian directory*: A directory that shows policy custodian where the policy is stored.
- *Location provider*: Entity providing the location data. Any release of data should be subject to the policy.
- *Service provider*: Entity that is combining location data with other data to produce some service.
- *Service consumer*: Entity to which the service is presented for consumption.
- *Service initiator*: Entity that would like and/or accept that the service is produced.
- *Service requestor*: Entity that makes a request to the service provider for the service to be produced.
- *Request*: Generic request.
- *Located object*: The entity whose location data will be required to deliver a service. The present location of the located object may or may not be known.
- *Owner of located object*: The entity that owns the located object.

The intention is that there should be established some (free central public) register (the policy custodian directory) at some well known address, that for each located object contains a pointer to the location where a personal location privacy policy for that located object is stored. A location provider would then be obliged to receive 'release approval' from the policy custodian before any location data could be released. In many cases, it may be convenient to make the network operator a policy custodian. As the policy itself might be sensitive, some access control measures to the policy might be required.

There are two main challenges with the feasibility of the proposed architecture: the first is the administrative challenge of creating and founding a Policy Custodian Directory. Most likely, it can be founded by a governmental organization or a consortium of major Location Providers and Location Service Providers. The second challenge is the possible performance bottleneck caused by policy transfer/caching. When the policy is located at a Policy Custodian and enforced at the location provider the policy must be transfered, either at request or in advance. If it is transfered in advance, the time before policy updates take effect will be limited by the allowed cache-time. Thus, a long chance-time will make frequent policy updates impossible. On the other hand, a short cache-time can give severe performance implications if the total load is high, i.e. it might imply scalability concerns.

These two challenges can be addressed in various ways. For example, it is possible to design architectures with a general policy but without a global policy custodian directory. However, the cost of such solutions are tighter integration between Location Provider and Policy Custodian, which will limit the users' choice of Policy Custodian.

Policy transfer/caching can be avoided by letting the Policy Custodian enforce the policy. The drawback with this approach is that everything the policy might depend on must be sent to the Policy Custodian. If an architecture with policy caching is implemented, the performance challanges can be solved with a reasonably designed policy transfer protocol and a suitable policy language. Specially important is that there there will be no frequent transfere of the largest parts of the policy, e.g. any geographical information or maps that the policy might depend on.

Clearly, there are several ways to ensure that release of location data is in accordance with personal policy. The following represents one possible sequence of events:

- The location data requestor (e.g. the service provider) sends a location query to the location provider. The request may include several located objects.
- The location provider must identify the currently applicable personal location policies by

contacting policy custodian directory and then the policy custodian. For performance and scalability reasons, one may want to do this periodically rather than per request by caching policies. In many cases, it may be acceptable to let the cached policies be valid for a short period of time, e.g. for one-week.

• The location provider forwards the request to the custodian, which responds appropriately.

• Depending on the response, the location provider responds to the location data requestor with the location data according to the response received, which will be in accordance with the privacy policy.

4 SIMULATION

A program that simulates possible scenarios in the LBS system architechture and network load is implemented in Java. The program simulates the load on the system as requests on their way through the communication channels and in storages in the system. A place may be a cluster of servers, one server, even procedures in a program on a server or any other well defined entity. We will call lines and places for states. In the present version all requests are virtual. They will only carry some time information, the size of the requests are modeled by the capacities of the diferent lines. Each state has in the model a list of pairs, each consists of a request and the next recipient of that request. The recipients will be called receivers, and they are chosen at random on arrival of the requests in a state. The senders are called communicators. The time when the request is planned to arrive at the receiver is calculated and stored together with the request.

The flow is controlled by a central controller and a time variable attached to each request, which says when it should be moved to another state in the model. The time variable is updated on the arrival of the request and depends on the receiver.

The controller controls that events are handled chronologically and that the simulation is run in visual real time if possible. It asks all communicators when their next events will occur and tells the

5 CONCLUSION

We have analyzed the impact of privacy protecting mechanisms in connection with location based services on capacity, performance, and scalability. The analysis includes both studies of each actor of the model separately and the communication between the actors. In addition to the analytical analysis, a tool programmed in Java has been developed to make numerical simulation of different architectures of the networks.

In the analysis we have addressed critical implementation requirements related to privacy-protecting mechanisms in connection with location based services. Possible scenarios and architectures have been identified and analyzed. The focus of the discussion has been on dependencies between parameters rather than absolute quantities, in order to better identify the scalability issues of the suggested alternatives for system architecture. The analytical study concludes that the privacy architecture of Snekkenes [16] seems to address well the security, performance, capacity, and scalability requirements. Alternative architectures, without the Policy Custodian Directory, has been discussed as well, and found viable.

The network protocols will be of a vital importance for the capacity, performance, scalability, and security of the architectures. The attention of the protocol analysis has been to identify the high level network protocols for the architecture. One main issue is policy transfer/ caching. The conclusion is that high policy cache-rate is of critical importance for the scalability of the architecture. The cache-rate will depend on policy update-time (the duration for a policy change to take effect) and the direct choice of policy format.

We have made a prototype platform for the simulations but there have not been enough resource to do the simulations. Some test simulations have been done but they are ment for testing of the platform and we did not get any concrete results from them. What remains to be done before one can perform effective tests is an interface for setting up experiments and automatic tests. In the current version all parameters must be entered manually, and many of them in the program code itself. In the simulation platform, there are three types of behaviors, i.e. reactions, when the load is high. The first is the trivial type which

do not react to overload at all. The second is the queue type which queues all new requests when there is no capacity to handle them. The third is the delay type which delays requests when the load is high. Our simulation tool combines all these types of behaviors.

The main conclusion is that the privacy protecting architectures seem feasible with regard to capacity and scalability. There is, however, one concern regarding the political/administrative feasibility of a policy custodian directory. At the present stage, it is not clear who will be interested in financing such entity. The location providers will probably want to make their own proprietary solutions, whereas, public administration can be interested without providing to the necessary funding to run such an organ.

5.1 Future work

The work on the LBS architecture has raised a lot of interesting prospects for future work. Some of the prospects that can be investigated further are as follows:

- Choice of policy language: The policy language will have critical influence on performance. On one hand, the policy must be designed in such way that it can be easily cached/transfered and rather be small. On the other hand, it must also reflect the true meaning of the policy owners capacity to control their policy.
- Detailed analysis of the security mechanisms: In this report it has been assumed that the actors comply with the "rules of the game". One future task can be to analyze and implement how the actors can be forced to comply with the rules of the game by using, for example, cryptographic techniques and legal bindings.
- Further development of the prototype platform and using other techniques such as timed colored Petri-nets [8],[7] in the simulation. Petri-nets is a graphical mathematical tool. The benefit of using Petri-nets is not only that they are very general tools, but also they are well known. Use of petri-nets would increase the target audience for the project and would provide a tool for analysis and visualization.

Acknowledgement

The authors are grateful to Ragni Ryvold Arnesen and Ingvar Tjøstheim for reviewing the draft of the paper and providing invaluable comments. This work was supported by the Research Council of Norway (NFR) grant number 152211/431.

REFERENCES

[1] M. S. Ackerman, F. C. Lorrie, and R. Joseph. Privacy in e-commerce: Examining user scenarios and privacy preferences. In *Proc. of ACM Conference on Electronic Commerce*, 1999.

[2] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the IEEE Infocom 2000, Tel-Aviv, Israel*, volume 2, pages 775–784, March 2000.

[3] Jason Burke and Peter Warren. How mobile phones let spies see our every move. The Observer, http://www.observer.co.uk /uk_news /story /0,6903,811027,00.html, October 13 2002.

[4] Digital Angel Corporation. http://www. digitalangel. net.

[5] Lupu Demianou, Dulay and Sloman. The ponder policy specification language. *Springer-verlag LNCS*, pages 18–39, 1995.

[6] Shouichi Hirose and Susumu Yoshida. A user authentication scheme with identity and location information. *Springer-Verlag, ACISP 2001, LNCS 2119*, pages 235–264, 2001.

[7] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Analysis Methods*, volume 2 of *Monographs in Theoretical Computer Science*. Springer-Verlag, 1997.

[8] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Basic Concepts*, volume 1 of *Monographs in Theoretical Computer Science*. Springer-Verlag, 1997.

[9] Ian Koeppel. What are location services? the gis perspective. Environmental Systems Research Institute (ERSI) White Paper, 2000.

[10] Dulay Lupu, Sloman and Damianou. Ponder: Realising enterprise viewpoint consepts. *EDOC*, pages 66–75, 2000.

[11] Oracle. Enhansing mobile application with location-based services. Oracle Business White Paper, June 2001.

[12] Oracle. Leveraging location-base services for

mobile applications. Oracle Techical White Paper, 2001.

[13] Kai Rannenberg. How much negotiation and detail can users handle? experience with security negotiations and the granularity of access control in communications. In *Computer Security - ESORICS 2000, 6th European Symposium on Research in Computer Security, Toulouse, France, October 4-6, 2000, Proceedings*, 2000.

[14] Reservasjonsregisteret. The brønnøysund register centre. http://www. brreg. no /oppslag /reservasjon /index.html.

[15] Sloman and Lupu. Policy specification for programmable networks. *Springer-Verlag LNCS*, 1999.

[16] Einar Snekkenes. Concepts for personalk location privacy policies. In *ACM Conference on Electronic Commerce (EC'01), 14-17 October 2001 Tampa, Florida, USA.*, pages 48–57. acm press, 2001.

[17] I. Tjøstheim, K. Boge, R. Arnesen, and K. Fuglerud. Online-consumers and privacy – a national study of what the e-consumers are willing to share of personal information. report 979, Norsk Regnesentral, December 2001.

[18] I. Tjøstheim, B. Nordlund, J. Lous, and K. Fuglerud. Travelers and location-information in the mobile environment - consumer attitudes and a prototype of a service for early adapters of mobile intenet service. In *Information and Communication Technologies in Tourism 2003, Proc. of the International Conference in Helsinki, Finland*, January 29-31 2003.

[19] TruePosition. http://www. trueposition. com/.