Charged Location Aware Services

Krzysztof Piotrowski[1], Peter Langendörfer[1], Michael Maaser[1], Gregor Spichal[2] and Peter Schwander[2]

[1] IHP, Im Technologiepark 25,
15236 Frankfurt (Oder), Germany

[2] Lesswire, Im Technologiepark 25,
15236 Frankfurt (Oder), Germany

Abstract. Location aware services have been envisioned as the killer application for the wireless Internet. But they did not gain sufficient attention. We are convinced that one of the major pitfalls is that there is up to now no way to charge for this kind of services. In this paper we present an architecture which provides the basic mechanisms needed to realize charged location based services. The three major components are: a location aware middleware platform, an information server and a micro payment system. We provide performance data that clearly indicates that such a system can be deployed without exhausting the resources of mobile devices or infrastructure servers.

1 Introduction

The convergence of mobile communication and Internet allows to realize new kind of services, business models, as well as the possibility to integrate company employees which are working remote. The amount of mobile devices in everyday use increases rapidly creating a big market for potential applications. One of the most interesting services classes are location aware services. Location aware services are the next step in the evolution of information access methods. These services are very interesting and their importance in commercial applications grows.

Imagine a service that informs a customer about current price cuts just when she reaches the shopping center. Or a service in a gallery that shows the information about the painting the visitor is just looking at. The number of such applications is limited only by the imagination of service providers. Of course location awareness is strongly connected with mobility, thus usually mobile device such as PDAs and cellphones will be used. But they have very limited resources with respect to computational and battery power.

We are convinced that location aware services will be widely deployed only if there is a suitable means to charge for their usage. With our approach presented here we show that it is feasible to charge a client for a location based service without exhausting its resources.

Our architecture consists of an information server, a location aware platform and a micro payment scheme. This combination provides the expected functionality of the

Piotrowski K., Langendörfer P., Maaser M., Spichal G. and Schwander P. (2005).
Charged Location Aware Services.
In *Proceedings of the 4th International Workshop on Wireless Information Systems*, pages 33-41
DOI: 10.5220/0002579700330041

whole system. Thus, our solution allows to provide the client with location dependent information that she wishes to get. But what is important and interesting, the service provider gets money for this information. This makes our solution more practical from the commercial point of view. In addition our approach shows very good performance parameters, e.g. HP IPAQ 5550 (400 MHz) needs 0.1 milliseconds per e-cash token to execute the most computational expensive part of the payment procedure.

The work on e-cash schemes started in the late eighties [3]. Since then many electronic cash schemes were proposed, e.g. [1], [2], [9], but they were not optimized for mobile devices. The design of location aware middleware platforms has recently attracted a lot of attention, e.g. NEXUS [4], [5], CATIS [8], Alipes [11], PLASMA [7], FAME2 [13]. But those platforms do not provide means to charge a user for services running on top of the platform.

In the following section we shortly describe the main features of each building block. We provide the details that are necessary to explain the interactions between components and to explain the functionality of the proposed system in whole. Section 3 describes our system, the idea behind and an exemplary scenario. After that we provide an estimate of the performance of the system. The paper concludes with a short summary and an outlook on next research steps.

2 Underlying components

This section describes the components we used to build up our system. The specification provided refers only to the main features of each component and specific details necessary to understand the idea of our approach.

2.1 Location aware platform — PLASMA

This section presents our platform called PLASMA (PLAtform Supporting Mobile Applications) [7]. The description focuses on the components for location handling that are needed to realize the sample scenario. But PLASMA consists of event and profile handling components as well. The application of these components would open the opportunity to set up more sophisticated services.

Infrastructure. PLASMA may be deployed in a hierarchical structure and it allows replicating the infrastructure servers. The tree structure allows to split geographical regions with high density of mobile devices into several new geographical regions, so that the load per infrastructure node is reduced. The servers of the lowest hierarchy level that directly communicate with mobile clients are called leaf servers. Figure 1 depicts the structure of all platform components (engines) inside a platform server.

In the following paragraphs we focus on the engines on the infrastructure side that are involved in the sighting process, i.e., receiving, collecting and delivering the location data.

The *communication engine* handles target references, which are used for the communication between clients and platform servers and for platform-internal communication. In fact, a target reference is made up of the machine’s IP address and the number of

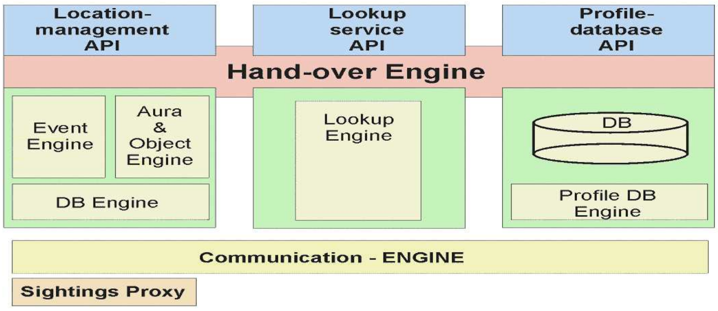

Fig. 1. PLASMA components and their structure inside a platform server

the port used by PLASMA. The communication engine is also responsible for an initial communication establishment for newly appearing device, e.g. when it is powered on.

The *sighting proxy* resides only on leaf servers, since only these servers communicate with mobile clients. The sighting proxy receives the position information from registered clients. It converts the position information provided by the positioning system into geographical coordinates that are required by the platform. Then the location information is forwarded to the *database engine* (DB).

The database engine stores the location information and provides it to other platform components and to applications upon request.

Figure 2 shows the engines involved in the sighting process and the flow of the location data.

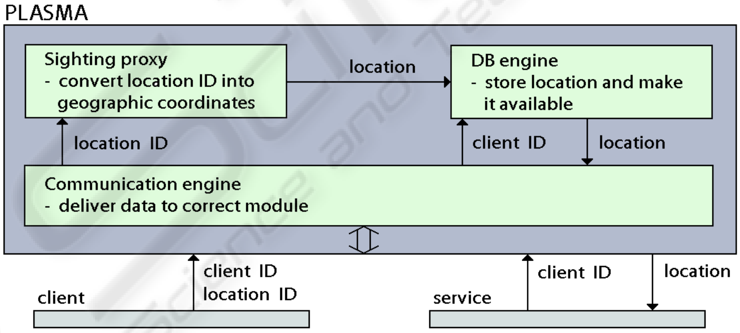

Fig. 2. PLASMA components that are involved in the sighting process and the flow of the location data. Service represents the party that asks about the location of the client

PLASMA is based on Java. Thus, in order to enlarge the number of clients that are able to use the platform it was necessary to introduce a gateway module between the Java and non-Java world — the Mediator (see fig. 3). It uses the SOAP protocol to communicate with non-Java clients. SOAP can be used to do remote procedure calls between any two programming languages. Therefore, we need only one kind of mediator for all non-Java clients.

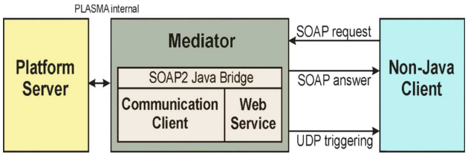

Fig. 3. For the support of non-Java clients, the mediator translates SOAP calls into the protocol for the platform communication and vice versa

Client. Both client types (Java and non-Java) are active with respect to the sighting system by reporting their current position to the positioning proxy. The position information may be delivered by any positioning system e.g. by GPS in outdoor scenarios or by IR beacons in indoor scenarios.

2.2 Micro payment — MONETA

This section presents MONETA — our off-line micro payment system [10]. It was designed to fit today's need for a multi-purpose cash scheme that allows a user to pay small amount of money without high costs.

MONETA provides all features that are required for an e-cash scheme to be considered as secure, i.e., it provides non repudiation, is framing proof and unforgeable.

Our cash scheme is optimized for mobile devices. It uses an asymmetric security architecture that reduces power and memory consumption [6]. Elliptic Curve Cryptography private key is used on the client side, and the RSA private key on the infrastructure side.

MONETA provides anonymity to its clients while also being secure against theft and double spending of money. To secure it against theft, we introduced a trusted third party — the MONETA Certificate Authority, that issues certificates for pseudonyms. These pseudonyms are linked to the e-cash tokens. To reveal the identity of the user in case of double spending we applied a mechanism based on the one proposed in [1].

There is only one value of the e-cash token (coin), which should be the smallest, indivisible amount of money (e.g. 1 Euro cent). This assumption makes the money handling easier, because there is no need to give the change and it is the easiest way to form different amounts of money. This also leads to a straight flow of money (see fig. 4).

Client, bank and service provider (service) are involved in the money flow. As shown on figure 4 there are three steps on the money flow path.

withdrawal - the client gets the coins from the bank. The coins are stored in a database (wallet) on the client device. The corresponding amount of real money is removed from the client's bank account,

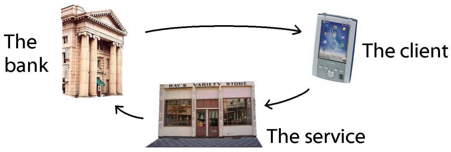

Fig. 4. The usual money flow in the MONETA micro payment scheme

payment - the service receives the coins from the client in return of goods or services it provides. The client removes the coins she spent,

refund - the service sends the coins back to the bank and its account is refunded.

2.3 Information server — WebTag^{TM} system

This section presents our information server. The information packet it delivers is called WebTag^{TM}. This packet binds (tags) the resources URL with a location of a client and a group of recipients. The clients are allowed to define their own WebTags^{TM}.

The WebTag^{TM} system consists of the server and the client part. The client part has similar functionality to an e-mail client. In order to receive WebTags^{TM} the client has to be registered and logged in. The WebTag^{TM} server manages the databases of WebTags^{TM}, places (positions) and users. It is triggered by external information about the location of a certain client. According to this information the server pushes a WebTag^{TM} to clients that are currently logged in and are on the list of recipients (see fig. 5).

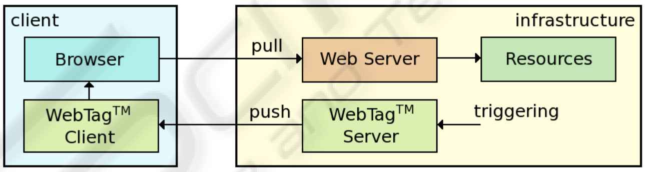

Fig. 5. According to the location information the WebTag^{TM} server pushes the WebTag^{TM} to the client, who can then access the resources from the web server via pull operation

3 Location aware service with micro payment charging

In order to make this section more illustrative, we start defining a scenario, which we then use to explain the overall architecture as well as the message flow. Any other scenario will follow the same lines.

First we have to specify the parties and the environment. There are two clients in this scenario: the visitor and the host. The visitor changes her position and the host pays to receive notifications according to the position of the visitor. This scenario requires a positioning system that provides the visitor with her current location to be available.

Imagine that our clients are going to have a business meeting. The visitor travels to reach the meeting place. The host is interested in the location of the visitor, but not in a sense of continuously updated coordinates. The host wants to know that the visitor reached a particular place on the way, e.g. that she just arrived to a city or to an airport. To get this information the host subscribes to WebTags^{TM} that bind these places with the visitor, and sets herself as the recipient of these WebTags^{TM}. Thus, the host receives the information according to the location of the visitor. The host is charged for this information just before it arrives to her device.

In our example scenario we have divided the functionality of the client into two parts, i.e., the host and the visitor. In other scenarios the client can act as the host and the visitor at the same time, i.e., the client gets information depending on her own location.

Figure 6 shows the structure of our system and sketches the data flow. It provides information about the parties of the system and about components used by each party as well.

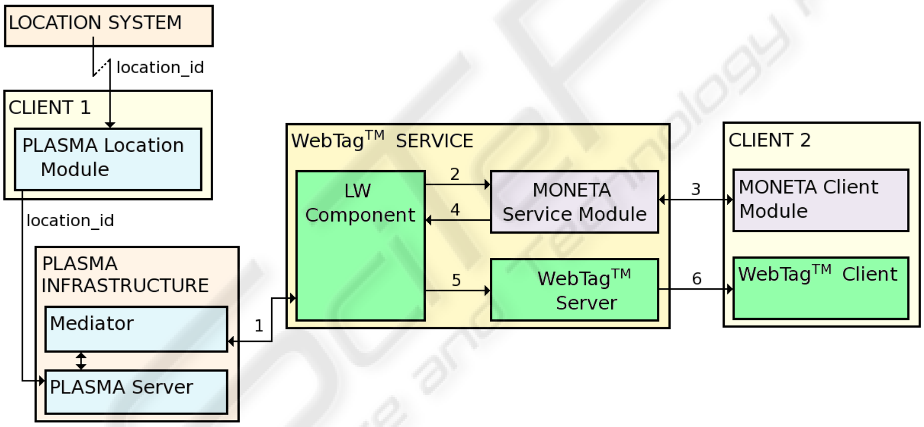

Fig. 6. The client 1 represents the traveling party (visitor). Respectively, the client 2 refers to the client (host) that gets the information depending on the location of the traveling party. The infrastructure parties are the WebTag^{TM} service and the PLASMA infrastructure. We omitted the MONETA bank and the Web server to make the figure more clear

The components mentioned in the previous section use different communication protocols, but each is built on top of TCP/IP. To make all parts of the system work together we had to provide an additional module — the LessWire (LW) component. On one hand this component is a kind of extension to the WebTag^{TM} server, since it manages the user identification and accesses the WebTag^{TM} server's databases. But it also provides a kind of business logic, i.e., it gets the location information from PLASMA and triggers the pushing of the WebTag^{TM} only if the visitor is in the specified location and the procedure of charging the host completed successfully.

The data flow is divided into two parts. The first one is collecting location data done by the PLASMA server. The second is the main procedure of charging and delivering the location dependent information.

Depending on the kind of positioning system the location information (*location_id*) is delivered to the visitor once a specified time period or every time she changes the location. The *location_id* identifies a location with the accuracy depending on the granularity of the positioning system. The visitor forwards the location information to the PLASMA server. After converting the *location_id* to a common location format — geographic coordinates, the PLASMA server stores the location together with the visitor's identity in its database. This procedure is independent (asynchronous) from the remaining data flow.

The charging and delivering procedure forms a sequence of six steps. The order of these steps is shown in figure 6 as a number next to the arrow representing each data exchange. As mentioned before the communication is based on TCP/IP.

step 1. Once a specified period of time the LW component asks the PLASMA server about the current position of the visitor. Since the LW component is not a Java PLASMA client it uses the Mediator to get the information. the LW component accesses the database of WebTags^{TM} to check if there is a WebTag^{TM} defined for the visitor and the obtained location. If this is the case then the LW component gets the list of IP addresses of recipients, i.e., clients that were set to receive information depending on the position of the visitor. In our scenario only the host is on this list.

step 2. The LW component sends an request to the MONETA service module asking to charge the host.

step 3. The third step is the payment procedure. MONETA client and service modules communicate in order to transfer coins from the host's device to the service's machine.

step 4. The LW component receives the result of the payment procedure.

step 5. If the payment procedure was successful then the LW component triggers the WebTag^{TM} server to push the WebTag^{TM} that contains the URL to the specified resource, e.g. an information ,,The visitor has just arrived to Berlin".

step 6. The delivery of the WebTag^{TM} is the sixth and last step of the sequence. Now the host accesses the resources described by the received WebTag^{TM}.

4 Performance evaluation

To estimate the performance of the whole system we estimate the computation load of each component.

The client[3] uses a mobile device with limited computation power. That is why the operations done on the client side have to be as lightweight as possible. During the forwarding of *location_id* and the reception of WebTag^{TM} the client does almost nothing in sense of computing power and data transmission. The payment procedure requires

[3] From now on we refer to the client as to the merged functionality of the host and the visitor.

the client to calculate two arithmetic multiplications per coin using big integer numbers (up to 233-bit long), but these operations are also not extremely expensive. They take 0.1 ms per coin on HP IPAQ 5550 with 400 MHz XScale processor.

The size of one coin is 280 bytes. Thus, the price of the WebTag^{TM}, i.e., the amount of coins used in the payment procedure influences the amount of transferred data as well.

The PLASMA server's functionality is limited to database operations and conversion of the location information. One PLASMA leaf server can handle up to 16000 position updates per second. The WebTag^{TM} server together with the LW component are doing simple operations and transfers of small amount of data. The most expensive operations are done by the MONETA service component. During the payment procedure it has to perform four elliptic curve scalar point multiplications per coin in order to verify it. The time needed by this operation depends on the ECC library used. Our first version of pure Java ECC implementation allows to calculate one scalar point multiplication in about 50 ms using a standard PC (2 GHz). This result was achieved for the P-224 elliptic curve recommended by NIST [12].

Table 1 presents our measurement results more precisely. The location query is a bidirectional operation, thus the result specifies the time period between sending the query and receiving the answer. It includes the delay caused by the network connection. Since the location forwarding is one-way operation we provide the time needed by the PLASMA server to process and to store the location data without the network delay. For the payment procedure we measured the amount of time needed by the client and the service to process each coin. We do not have exact measurement results for the WebTag^{TM} creation operation, but it should be comparable with the result for location forwarding.

Table 1. Amount of time needed by each party to perform the operations

Operation	Client	PLASMA	Service
Location forwarding	0,0625 ms		—
Location query	—	45 ms	
Payment	0,1 ms / coin	—	200 ms / coin

5 Summary and Outlook

In this paper we have investigated an architecture which provides all means to implement location aware services and to charge for their use. The system consists of three major components: a location handling middleware platform, a micro payment system and an information server. All these components have a very good performance, e.g. the location handling platform can manage up to 16000 position updates per second, and the most expensive operation on the mobile device, i.e., the payment procedure requires only 0.1 milliseconds per coin.

Despite the system shows already a good performance figures we will optimize the overall performance in our next research steps. Currently the information server poll the location database in order to get the current position of all users. In a next step we will replace this by an event based mechanisms which is provided by the middleware platform used. In addition we will tune the implementation of the cryptographic operations. This includes the integration of C and assembler programming parts as well as the integration of hardware accelerators for elliptic curve cryptography. We are also going to investigate the privacy issues.

References

1. Baumgart, M., Neumann, H., Schweitzer, N.: Optimistische Faire Transaktionen mittels zeitlich beschränkter Münzen. Verlässliche IT-Systeme - VIS 2001. Vieweg-Verlag (2001).
2. Brands, S.: Untraceable Off-line Cash in Wallets with Observers. In: Proc. Of Crypto ’93, Lecture Notes in Computer Science 773. Springer-Verlag.
3. Chaum, D., Fiat, A., Naor, M.: Untraceable Electronic Cash. In: Shafi Goldwasser (Ed.): Advances in Cryptology CRYPTO ’88. Springer-Verlag (1988).
4. Hohl, F., Kubach, U., Leonhardi, A., Schwehm, M., Rothermel, K.: Nexus: An open global infrastructure for spatial-aware applications. In: Proc. of the fifth International conference on Mobile Computing and Networking (MobiCom 99). ACM Press (1999).
5. Leonhardi, A., Kubach, A.: An architecture for a distributed universal location service. In: Proc. of the European Wireless 99 Conference. VDE Verlag (1999).
6. Langendörfer, P., Dyka, Z., Maye, O., Kraemer, R.: Power Security Architecture for Mobile Commerce. In: 5th IEEE CAS Workshop on Wireless Communications and Networking. IEEE Society Press (2002).
7. Langendörfer, P., Maye, O., Dyka, Z., Sorge, R., Winkler, R., Kraemer, R.: Middleware for Location-based Services: Design and Implementation Issues. In: Q. Mahmoud (ed): Middleware for Communication. Wiley (2004).
8. Pashtan, A., Blattler, A., Heusser, A., Scheuermann, P.: CATIS: A Context-Aware Tourist information System. In: Poceedings of the 4th International Workshop of Mobile Computing. Rostock (Germany), June 17-18, 2003.
9. Petersen, H., Poupard, G.: Efficient scalable fair cash with off-line extortion prevention. Technical Report LIENS–97–07. May 1997. 33 pages.
10. Piotrowski, K., Langendörfer, P., Kulikowski, D.: Moneta: An Anonymity Providing Lightweight Payment System for Mobile Devices. Reviewed Paper. IFIP/GI Workshop on Virtual Goods. Ilmenau (Germany), May 28-29, 2004.
http://virtualgoods.tu-ilmenau.de/2004/MONETA.pdf (last viewed: January 31, 2005).
11. Synnes, K., Nord, J., Parnes, P.: Location Privacy in the Alipes platform. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-36). Big Island, Hawaii, USA, January 2003.
12. U.S. Department of Commerce / National Institute of Standards and Technology (NIST): Digital Signature Standard. FIPS PUB 186-2. Federal Information Processing Standards Publication (2000). http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf (last viewed: January 31, 2005).
13. Wuest, B., Drögehorn, O., David, K.: FAME2: Software Architecture for B3G and 4G Networks. In: IST Summit on Mobile and Wireless Communication. Aveiro, Portugal, 2003, p.837-841