Multilateral Security in Mobile Applications and Location Based Services

for Europe's Independent IT Security Conference - ISSE 2002, Oct 2nd to 4th

Mario Hoffmann[1], Jan Peters[2], and Ulrich Pinsdorf[2]

[1] Fraunhofer Institute for Secure Telecooperation
Rheinstraße 75, D-64295 Darmstadt, Germany
Tel: +49-(0)6151/869-267, Fax: +49-(0)6151/869-224
mario.hoffmann@sit.fraunhofer.de

[2] Fraunhofer Institute for Computer Graphics
Fraunhoferstraße 5, D-64283 Darmstadt, Germany
{jan.peters|ulrich.pinsdorf}@igd.fraunhofer.de

Abstract. Due to the many current weaknesses of security mechanisms in mobile technology, location based services essentially depend on security aware middleware and reliable multilateral security concepts. Neither the latest operating systems of mobile devices nor current concepts in wireless communication GPRS, WLAN, or Bluetooth meet the security requirements needed to establish sustainable trust between consumers and producers.
This paper presents an approach for multilateral secure mobile applications which is based on mobile agent technology. We use mobile agents as highly dynamic deployment mechanisms for service components. Our approach allows us to enforce fine-granular multilateral security policies for all involved participants: service distributors, service providers, and users accessing the service.
Our work is applied to the MOBILE project funded by the German Federal Ministry for Education and Research.

Keywords: Mobile applications, location-based services, mobile agent systems, multilateral security, privacy, profiles

Refers to following topic: Security integration in applications

Refers to interest level: For application developers (interested in IT security)

1 Introduction - Security follows technology

During the last twelve months a large number of discussions about *Internet Security* have left their mark on the development of secure hardware and software. Improvements have been made in automatically analysing electronic mail, enforcing software patents and tracking the user's location. In other fields, however, there is still much to do. Biometry such as face and fingerprint recognition for example has been promoted to secure access to both restricted areas and the electronic infrastructure, but it's still very easy to deceive the new technology. In a similar way, many aspects of *mobile security* —comprising all efforts to secure a fast growing new area— are still topic of different research projects. Operating systems of personal digital assistants (PDAs) for example offer poor integrated access control and only basic user authentication. Even encrypted radio traffic via GSM[1] or WLAN[2] can be scanned for compromising information (*eavesdropping*), and, finally, it's quite simple to make use of current mobile services without paying for anything.

Nevertheless, we develop *secure* mobile applications and location based services to establish multilateral security between the users of mobile technology, network providers, and service providers, but this can only be done step by step. In fact, we believe that there is still a long way to go before we are able to integrate fingerprint or smart card technology to PDAs, to rely on cryptographically secure WLAN and Bluetooth, to predict the precision of a global positioning system, and for profiles to remain under the user's full control. Unfortunately, these are all essential security requirements for the sustainable impact of mobile web services on the market. And before all these technologies can be integrated —most probably never— we present an alternative way of realising end-to-end-security.

This paper introduces an approach to providing *end-to-end-security* based on a secure middleware platform taking advantage of mobile agent technology. That means that multilateral secure mobile applications and location based services will be offered using mobile agents as highly dynamic deployment mechanisms for service components. All involved participants in our role based concept like users, service brokers, service providers and information providers (see section 2.3) are able to maintain their personal respectively service agents and configure fine-granular multilateral security policies and their profiles. The whole scenario is divided into two security domains. First, the user's security domain which contains one or more mobile devices and a homebase —an always-on agent server providing access to security policies and profiles. Second, the outer world (most likely the Internet) where service brokers, service providers, and information providers offer mobile applications and services (see section 3).

In the following, we give a short definition of mobile agents adapted to the specific context of the MOBILE[3] project and an overview of related work in similar projects other research groups are working on. Section 2 introduces our role-based approach in detail, which is integrated into the project's general architecture (see section 3). Subsequently, section 4 focuses on the security concept before we conclude in section 6.

[1] GSM = Global System for Mobile Communication
[2] WLAN = Wireless Local Area Network
[3] MOBILE: www.mobile-projekt.de

1.1 Mobile Agent Technology

The most promising way, in our opinion, to integrate mobile services is by means of so-called *mobile software agents.* Franklin et al. [2] have collected and examined 11 definitions of agents in the context of information technology and computer science. Based on those specific definitions we have combined several characteristic features to a basic agent type including: *reactivity*, agents react on environmental changes; *autonomy*, agents have control over their actions; *proactivity*, agents do not only react on changes of the environment; *continuousness*, the agent's process runs without interruption from its creation until the process finally dies.

When we talk of mobile agents in the following, we refer to an agent as a combination of these features adding mobility and the ability of communication. Such an agent can be seen as a tuple of program code, data and execution state, which *migrates* from one execution environment (an *agent server*) to another. Making use of the local resources, assigned to an agent by the execution environment, an agent is able to collect, process and publish data. A detailed view on our agent model will be given in section 2.2 and section 4.2.

1.2 Related Work

Since the hype surrounding UMTS[4] —which is supposed to go online at the end of 2003— a lot of research has been done to take advantage of the benefits of UMTS such as bandwidth, quality of service, and packet-oriented transmission. New and innovative mobile services are heralded almost daily promising a richer mobile experience, but before this can become commonplace, a number of fundamental problems need to be addressed. Many project teams around the world have dedicated their work to mobile computing, mobile security, and especially location based services. Just a few are also based on a multi-agent system and three of them will be shortly introduced in the following.

In 1999, Ernö Kovacs and his team at Sony International (Europe) GmbH in cooperation with Siemens AG presented the results of the ACTS AMASE project which had adapted a mobile agent system for use in wireless networks to support mobile users [6]. They concentrated on issues related to making mobile agent-based services aware of the context of the user. They examined a mechanism that allowed agents to adapt their behaviour to the current situation in the wireless network, and a system mechanism that automatically adapts agent execution to the given context. Again, security played only a small part. The issues of the privacy of data and agent-based computing in general were mentioned but neither introduced in detail nor sufficiently solved.

Currently, at Carnegie Mellon University, for example, Norman Sadeh and his research group are developing *MyCampus*, an agent-based environment for context-aware mobile services [14, 15]. Their work revolves around the development of a growing collection of task-specific agents that users can pull into their personal environment. Using ontologies for representing user preferences and contextual attributes enables agents to automatically access and exploit relevant user preferences and contextual attributes. Security issues such as who exactly has access to user preferences and when have not been addressed, so far.

[4] UMTS = Universal Mobile Telecommunication System

Last but not least, the DAI-Lab[5] in Berlin is working on various interesting projects in the areas of telematics services supporting mobility, e-commerce, and service and network management. All projects are based on an agent architecture implemented in Java. Similar to our context, the *BerlinTainment* project develops an agent based serviceware framework in order to allow a faster realisation of intelligent, open, scalable, flexible, adaptable, device independent and location based information services [18, 19]. However, even in those projects distinctive security concepts have not yet been implemented.

2 Approach

In contrast to the projects mentioned above the MOBILE project aims to develop a platform for the day-to-day deployment of multilateral secure dynamic services which are dependent on time, place and person. By taking personal preferences and settings into account, an individualised selection of available services can be offered to users according to their current location and the current time. Security requirements and data protection issues for both service provider (e. g. authentication, secure data transmission) and customer (e. g. control over personal data) need to be considered to the same degree as personal preferences in the choice of services.

2.1 Mobile Scenario

In general, location based services as a category of mobile web services depend on the time, the place and the user accessing the service. The idea is that by taking personal preferences and settings (the user's profile) into account, service providers and brokers are able to offer an individualised selection of web based services to users according to their current location and the current time. The core services themselves have to be provided by information providers.

The security aspects in such a scenario are manifold and sometimes conflicting. Users mainly do not want to divulge too much information and to spend too much time with the configuration before using a service, especially, when they have to reenter information for every different service and/or provider. It is more effective to collect those entries and maintain the latest settings in the user's profile under the user's control and to provide the appropriate entries in case of a service request. Service providers, however, are naturally interested in gathering as much as possible information about the user's surfing behaviour in order to maximise their return on investment. Finding an appropriate balance between the different interests and individual security requirements of all participants is called multilateral security (see section 4).

2.2 Mobile Agent Benefits

In our opinion, the most promising way to represent the different roles, their interests and goals in mobile scenarios is the use of a multi-agent system as defined in section 1.1. Once

[5] The DAI-Lab is part of Faculty IV Electrotechnics and Information Technologies at the Technical University Berlin, Germany

let loose, mobile agents roam the network, seek information, carry out tasks on behalf of their senders autonomously, and return to present the results of their queries. In the meanwhile the user is freed of the obligation to permanently monitor the application's progress. This makes mobile agents particularly useful in mobile environments, because no permanent network connection must be maintained in order to run the agent-based application [6]. Mobile agents also offer great benefits to applications in "wired" networks by adding client-side intelligence and functionality to server-side services unified under a homogenous access paradigm. Furthermore, mobile agents offer considerable network bandwidth savings because they can migrate to, and process data, at the source of that data, which therefore need not be shipped back and forth across the network. Applications based on mobile agents are inherently distributed. Agents are often independent of particular hardware or operating system, and can be deployed in heterogenous environments [7, 8].

The *mobility* property brings with it some important benefits (see section 2.2) but also raises new security issues, which can be categorized as follows (compare Karjoth et al. [4, 5] or Sanders and Tschudin [16]):

Malicious agent problem It is important to protect agent servers and other agents from malicous agents by means of for example unauthorized access to resources, denial of service attacks, impersonation of a foreign identity.

Malicious host problem Agents should also be protected against malicious hosts by means of for example agent manipulation, denial of service, monitoring.

As middleware for access to information resources we need an infrastructure of mobile agent servers which provide the runtime environment for the agents and interfaces to locally installed databases. Keeping the above mentioned security issues in mind, it is quite clear that this platform should follow a security aware design concept.

Implemented completely in Java, the multi-agent system SeMoA (Secure Mobile Agents) [17] used as middleware in our architecture is independent of a particular hardware platform or operating system, and can be deployed in heterogenous environments — even on PDAs running Linux and in the near future on Java-enabled cellphones. The SeMoA project is developing an open server for mobile agents with a special focus on all aspects of mobile agent security, including protection of mobile agents against malicious hosts, and is a "best effort" to provide adequate security for mobile agent systems — servers as well as agents [12].

In connection with security domains identified in a mulitlatural security concept (see section 4) another advantage of mobile agent systems can be named: Let's assume that an agent collects information from different information resources and has the task of automatically evaluating them using some private comparison criteria from its owner which should not be revealed. Since the problem of running private processes in a foreign execution environment has not yet been satisfactorily solved, it is advantageous to be able to migrate data together with corresponding algorithms into a trusted security domain, where these processes can be run. Especially within an infrastructure containing mobile devices with little processing power the migration of an agent to a host in the same security domain which can deliver the required data and computational resources can be helpful.

[6] This might not be a critical aspect in terms of "always-on" and DSL flatrates, but effectively hides the current location of the user and prevents the monitoring of the user's physical movements when using a positioning system based on cell phones.

2.3 Role-based approach

In order to give a better understanding of the different parties and their interests in the mobile scenario, we have first identified the different roles to be modelled by mobile agents appearing in our approach (see figure 1). Starting from the bottom (the information provider) and going up to the top (the user requesting information) the following subsections introduce the different parties and the relationships between them.

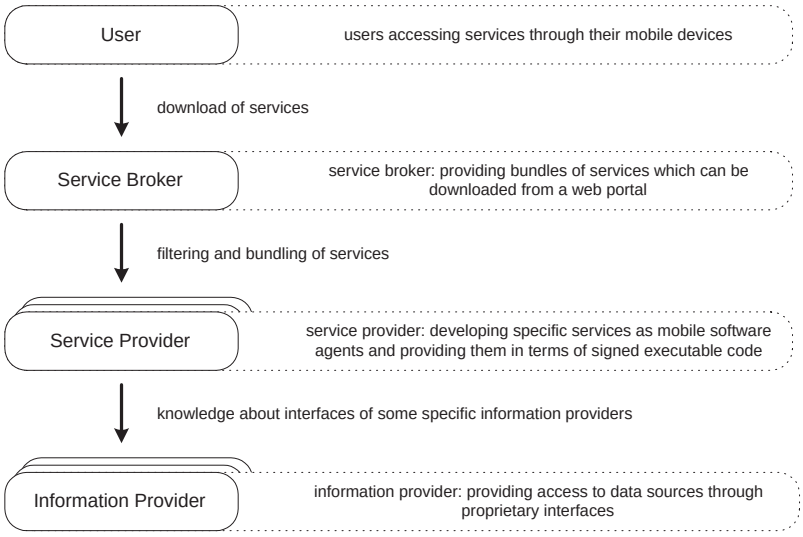

Fig. 1. Roles appearing in our approach.

Information Providers: *Information providers* in our scenario can be compared with well known internet services, which can be queried, for example, to get the weather forecast for a certain region, to receive the latest news according to a chosen topic, to book rooms in a hotel having a fixed price limit, to locate a restaurant for vegetarian tourists, to provide the nearest 24 hour dry-cleaning services or simply to look up an address in an online telephone book.

These providers host *basic internet services* with either direct access to databases containing the desired information or via connection to other cooperating information providers which can in turn deliver the desired information upon request.

In order to provide the information on offer in our scenario an information provider simply has to install an agent server, which furthermore grants incoming agents access through a certain service interface, whereas the interfaces and protocols to access the provided information are predominantly *proprietary*. In contrast to common internet services, which often provide information through a web portal, in our scenario information can be requested directly using the agent server as middleware which provides a known excecution environment for service developers (see service providers).

Service Providers: *Service providers* have knowledge of a certain number of information providers dealing with basic internet services, which in each case for example allow the

booking of rooms in a given hotel, in a specific city. The service provider can then offer a new service component, which allows users to find the cheapest room in this city by comparing the offers of the different information providers. In general, with predefined comparison and evaluation criteria this service component gathers the particular query results and provides a high-value service.

Service providers have detailed knowledge about the service interfaces of certain information providers and know about the data structures of the information provided. In the context of location based services and for a localised area this is a realistic assumption.

With this knowledge a service provider is able to develop *complex service components* by means of mobile software agents (cf. section 2.2), which are able to seek, process and compare data and subsequently return results with added value to the user. A service component can therefore consist of several cooperating agents, which can perform their task in a concurrent way.

The service provider can either be employed by several information providers or more probably sell his service components to interested service brokers.

Service Broker: The task of the *service broker* is to collect, review and publish service components developed by service providers. Besides structuring and filtering the variety of provided services, quality inspection is another major issue.

Bundles of service components, provided as executable code, can be offered through a web portal in terms of mobile software agents. The user downloads these agent as executable code archives to his mobile device and instantiates individual agents by execution within an installed agent server.

Users: The *user* first has to find an appropriate service depending on his needs, the specific situation and his context. When he has found such a service or a bundle of services on the web portal of a service broker, he downloads the corresponding agents and instantiates them as mentioned above.

A running agent can then be instructed to gather the requested information (so-called *pull services*) or looks for up-to-date information autonomously and presents it in case of particular events (so-called *push services*). In either case the agent possesses the knowledge of where and how to gather the information needed to perform its task; it automatically migrates to those information providers which can deliver this information. Once the user has downloaded and instantiated an agent he can access it as a service whenever he needs to.

A user has many individual requirements of mobile services and (mobile) services have many requests to a user. In order to bother the user as little as possible the user's individual requirements are derived from personal settings configured in a *personal profile*. Note: the profile is not stored at one or more of the various mobile devices the user may own (e.g. notebooks, organizers, cellphones), but at a trusted place on the web (for details refer to section 3).

3 Underlying Architecture

The following section describes the underlying architecture of MOBILE in order to give a better understanding of the relationships between users, providers, and information sources

before you proceed with the multilateral security concept in section 4.1. Basically, the architecture is divided into two security domains, the user's security domain —comprising the mobile device and the user's homebase— and the outside world —consisting of service portals, service providers, and information sources. Based on figure 2 the following subsections are giving a detailed explanation of each part of the architecture.

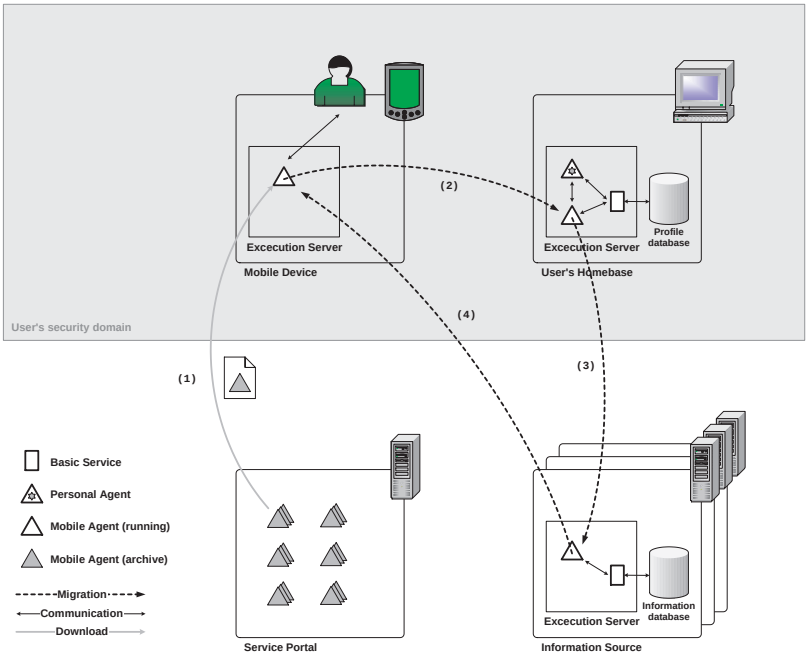

Fig. 2. Infrastructural architecture of our approach.

Mobile device: In our szenario we are concentrating on mobile users with mobile devices such as notebook/laptop computers, organizers, and cellphones. Additionally, we recommend to use a GPS receiver for globally available and consistent localisation. Although it is also generally possible to get a fairly good position from mobile network providers via cell-id-triangulation, for typical mobile szenarios (asking for specific information depending on the precise location of the user) an accuracy of measurement of 50 to 100 meters is not close enough.

Furthermore, mobile devices such as organizers have really poor authentication mechanisms and almost none access control integrated. Instead of installing additional mostly incompatible software on each mobile device we prefer to maintain private data only at a trusted place on the web, the *homebase*. The item *User's homebase* describes this place in more detail. For identification and authentication purposes a smartcard reader will be provided.

As opposed to the current use of mobile devices —particularly for storing addresses, managing appointments, and taking notes— for security reasons our approach advises to shift important data from your mobile device to the homebase. Not only that you don't loose all your private data if your mobile device gets lost or your batteries are empty, but

also you don't have to deal with inconsistencies between your desktop computer at work, your personal computer at home, your laptop, your organizer and mobile, not to talk about different operating systems and browsers where you usually manage inconsistent sets of bookmarks.

Running a Java Virtual Machine —available for organizers, smartphones, and more and more cellphones— gives us the opportunity to integrate the mobile device as a node in the multi-agent infrastructure. In order to maintain the personal agent properly, to add new services, or just to use the underlying security mechanisms to send data signed and encrypted we can take advantage of the multi-agent system and migrate the personal agent and service agents when needed from the homebase to the mobile device and back again (see step 1 and 2 in figure 2). On mobile devices which are not Java-enabled we support at least a web-based graphical user interface via a SSL-connection to provide secure access to the homebase and mobile services.

User	→ Mobile Device
	→ User's Homebase
Service Broker	→ Service Portal
Service Provider	→ Service Provider
Information Provider	→ Information Source

Table 1. Mapping between roles and corresponding entities in architecture

User's homebase: As mentioned above the user's homebase is part of the user's security domain. Therefore, the homebase must be installed on a trusted place, where the user has full control over his private data nobody else can spy out or misapply. The location of this trusted place diversifies in relation to the user's security requirements. A trusted place can be installed in your company, for example. Assuming that a secure webserver already supports secure mobile access via a web browser, additionally, your local administrator only has to set up a server for the multi-agent system. Another possibility for advanced users is the installation and configuration of the multi-agent system at your personal computer at home securely connected to the Internet via an ISDN- or DSL-flatrate. Finally, there is the opportunity to ask a trustworthy web space provider to manage your private profile.

As a common node of the agent's infrastructure service agents can migrate to one's homebase to offer new services or already subscribed service agents return after processing certain tasks, for example, getting the best-price hotel in Paris (see step 3 in figure 2). Security policies, profiles and private data can only be queried and edited at the user's homebase and only trusted and authorised agents have access to personal and private data. The only agent with full control is the user's personal agent. Unknown or unauthorised service agents, first, have to negotiate with the user's personal agent who grants specific rights, for example, to obtain a subset of the user's profile or certain information about the user. Finally, authorising service agents and delegating access rights is always in the responsibility of the user.

Other information the personal agent (PA) is responsible for are location awareness — the PA knows most of the time where you are— and reachability management —the PA knows how to contact you or just leaves a message. A user will therefore not be

inconvenienced by information requests of authorised service agents until a requested piece of information cannot be found in the profile. Only in that case the user is asked for the required information which is subsequently stored in the profile.

Service Portal: Service portals are part of the outside world. On these portals service providers (see table 1 for the complete mapping between roles and entities of the architecture) offer high-value services such as travel agencies or city portals. These portals contain well configured sets of basic information agents, for example, travel agencies might be composed by a hotel reservation agent, a booking agent for trains and airplanes, and a routing agent if you go by car. Depending on the various target groups many different service portals are necessary in the near future. And the underlying agent infrastructure supports an easy way to interoperate between service agents, information agents, and personal agents.

Basically, there are two opportunities for service providers who run the service portals to get into contact with and to serve interested users. First, service agents are able to proactively migrate to homebases in order to find out if any of the users represented by their personal agent is interested in a travel agent. Second, the personal agent itself might look for an appropriate travel agent on behalf of its user to subsequently subscribe, download and install it. Once a travel agent is installed on a homebase or a mobile device (see step 1 and 4 in figure 2) every time the user needs to plan and organise a journey or a business trip he can use this specialised agent instantly.

Information Provider: Basic information agents are also part of the outside world. Providing simple but specialised agents and services, for example route planning agents or restaurant guide agents, service providers and end users have to trust the agent developers to a certain level, but trust is good, control is better. Therefore, we recommend to sign and certify basic information agents, as described in section 5.

After signing and certifying the information agents can be easily integrated into high-value services by service providers. Moreover, they can be downloaded and installed by the user directly if he is looking for a certain service (see step 1 and 4 in figure 2).

4 A multilateral security concept

The architectural approach has many implications for the security concept. For example the filtering and bundling of services is the main task for service providers and brokers, but how can they guarantee the availability, integrity, and even the harmlessness of the underlying core services? Service components and their agents want to access the user's personal profile, but how can they certify their identity and how can the personal agent satisfactorily ensure the user's intentions? These issues are addressed and discussed in this section.

4.1 A definition of multilateral security

In contrast to a simple communication protocol between two processes, we present a complex system of interacting entities in a distributed infrastructure. Several parties cooperate to provide users with different profiles with the information they need. Since the different entities in such an open communication system do not trust each other in the first place, each party can be seen as potential attacker.

To meet the security requirements of all participating entities in our approach we thus have to develop a so-called *multilateral security* concept for the whole system. That means realizing specific security mechanisms, which establish a certain level of trust between the interacting entities.

In this context we first have to assure that local data processing is secure by means of confidentiality, integrity, availability and accountability. Subsequently, we have to extend these security goals to the whole infrastructure as far as needed to meet every entity's requirements.

4.2 Security mechanisms provided by the mobile agent system

For the moment, we can presume that a secure mobile agent system such as SeMoA already provides us with a set of basic security mechanisms such as encryption and digital signatures. These mechanisms guarantee the integrity of mobile agent code and assure the corresponding identity of the user and the mobile services accessed. A detailed analysis of security risks and corresponding requirements regarding agent transport, agent structure and agent execution can be found in [11, chapter 4.4].

In order to underline the security context using such a mobile agent system, we want to emphasize the following statement:

> *A protection of data in free-roaming mobile software agents against malicious hosts is based on cryptographic protocols. In general, the objectives are twofold: first, an agent carries confidential information that is revealed only while the agent is on a trusted host, and second, the agent transports partial results back to its origin in a way that assures the integrity (and optionally the confidentiality) of the partial results. Furthermore, the owner of the agent shall be able to derive the identity of the host on which given partial result was acquired.*

Furthermore we are able to monitor our service components during their lifetime using an agent tracking service [13] or verify their itineraries upon return to their owner [10].

Hence, secure agent systems offer the possibility of maintaining and configuring fine-granular security policies on each server for every single service and agent. Since we can not rely on the different more or less insecure transport protocols in wireless networks (as mentioned in section 1), we actually employ these mechanisms to realize end-to-end security between the different components described in our approach. Our task is to adopt and extend the given security mechanisms to enable multilateral security concepts which cover the specific requirements of each of the involved entities.

4.3 Interaction between the different entities in our architecture

In the light of the role-based approach introduced above and its underlying architecture (see sections 2 and 3) we can identify several threats. According to the method proposed in

the Common Criteria [1], we develop corresponding security requirements, describe mechanisms [7] to enforce them and finaly evaluate remaining risks. This is done in the following, understanding a multilateral security concept as subsumption of security mechanisms for certain actions. These actions take place between specific entities within our architecture. If a security mechanism is already provided in the context of the underlying mobile agent system it is marked with ^{SeMoA} in the following.

Special tasks of the personal agent: The user's personal agent as a special kind of service component runs as a stationary agent within the agent server on the user's hombase, therefore it does not perform requests on information sources provided by external information providers. Besides granting foreign agents access to the user's personal profile (see below), it may itself inspect the personal profile and act as push-service. By checking the user's diary situated on his homebase the personal agent is able to remind the user of appointments, for example.

- → THREATS: In this case there are no direct threats from other entities, but the risk of a server breakdown or unavailability due to network problems, which prevent communication between the user's homebase and the mobile device.
- → SECURITY REQUIREMENTS: Availability of network connection and execution environments.
- → SECURITY MECHANISMS: A solution to reduce the risks is the migration of such a push-service together with the diary as corresponding information source onto the mobile device assuming sufficient resources.
- → REMAINING RISKS: A server breakdown on the mobile device is still possible, but less probable than the occurrence of network problems.

Interaction between user and personal agent/profile: The user is able to access his personal profile via his personal agent to keep track of the currently stored preferences and to modify, add or delete entries within the profile. Furthermore, the user's personal agent independently inquires and automatically stores information from the user, if a request for a specific profile entry could not be resolved in the first place.

- → THREATS: During message exchange between mobile device and user's homebase the user's preferences might be spyed on. Furhermore an unauthorised user might try to access the personal profile and read or manipulate its entries.
- → SECURITY REQUIREMENTS: Identification and authorization of the user towards the personal agent. Confidentiality of exchanged messages.
- → SECURITY MECHANISMS: A challenge-response protocol using the user's private key as shared secret for encryption of protocol messages.
- → REMAINING RISKS: Little to no remaining risks in the context of strong cryptography.

Filtering and bundling of services: The relation between service provider and service broker consists of two aspects. In the first place, the service provider is interested in financial return on the knowledge about information sources of specific information providers,

[7] Encryption and signing of data as a security mechanism is meant in a cryptographical way using public/private key algorithms.

which has flowed into the development of services. In contrast, the service broker filters, bundles and subsequently provides service components through a web portal assuring a certain quality of service. These aspects are beyond the scope of mobile agent security, since we are talking about the simple transfer of executable code. No instantiation of a service component has occurred, yet.

- → THREATS: 1. Depending on the protocol chosen, usage of service components without paying for the intellectual property they contain might be possible; 2. The Service component might be malicous in an arbitrary way.
- → SECURITY REQUIREMENTS: 1. Payment model for the exchange of intellectual property in terms of functionality implemented within the agent's code; 2. Harmlessness of the service component.
- → SECURITY MECHANISMS: 1. A complex payment protocol used between service broker and service provider; 2. Since it is very difficult and mostly impossible to completely eliminate the harmfulness, the service broker can only do some statistical analyses of the executable code and furthermore trust the service provider (cf. section 5).
- → REMAINING RISKS: 1. Dependent on the protocol chosen; 2. Dependent on the level of trust between service broker and service provider.

Service download: The user downloads a service component from the service broker's web portal expecting the functionality promised there. With this action we are still beyond the scope of mobile agent security mechanisms mentioned in section 4.2.

- → THREATS: Code manipulations during download.
- → SECURITY REQUIREMENTS: Assurance of the agent's code integrity.
- → SECURITY MECHANISMS: The service broker signs the agent's code (see section 5).
- → REMAINING RISKS: Little to no remaining risks in context of strong cryptography.

Query execution: The personal agent instantiated by the user on his mobile device performs a specific query by migrating through the network, gathering and processing information from information providers.

- → THREATS: Malicious information providers might try to read or manipulate gathered information or monitor the agent user's habits.
- → SECURITY REQUIREMENTS: 1. Confidentiality, integrity and authenticity of gathered information; 2. Anonymity of the agent's user.
- → SECURITY MECHANISMS: 1. The information provider encrypts provided information for the user and signs them ^{SeMoA}. Secure computations are only computed within the user's security domain, i.e. on the user's homebase or directly on the mobile device; 2. Usage of a pseudonym instead of the user's identity as agent's owner. Dependent of the contract between user, service broker and information providers, the usage of pseudonyms is only possible in some specific scenarios ^{SeMoA} (compare section 5).
- → REMAINING RISKS: As far as all security mechanisms are applicable and the user trusts the service broker as independent entity, only little remaining risks.

Service components accessing the personal profile: In some cases a service component has to know about certain user preferences stored in the user's personal profile to perform a query. Therefore, the service component has to send requests to the personal agent.

→ THREATS: A malicious agent might try to access the personal profile and read or manipulate entries.
→ SECURITY REQUIREMENTS: 1. Identification and authentication of foreign agent's owner and authorisation to allow access to the profile, 2. Integrity and confidentiality of inquired profile entries.
→ SECURITY MECHANISMS: 1. The agent's owner signs the agent's code ^{SeMoA}, 2. Requested profile entries are encrypted only for those hosts on which the agent is allowed to operate on ^{SeMoA}.
→ REMAINING RISKS: Little to no remaining risks in the context of strong cryptography.

Service component execution on agent server of corresponding broker: Beside specific offerings comprising several service components, the service broker provides agent servers which can be used by service components downloaded from its web portal, for example to perform calculation tasks which assume a powerful host.

→ THREATS: 1. Foreign agents might use these agent servers for execution without having properly subscribed to the service broker to e.g. enable payment schemes; 2. Monitoring and manipulation of agent data procession through the service broker.
→ SECURITY REQUIREMENTS: 1.1. Identification of the service broker the agent has been downloaded from; 1.2. Identification of the agent's owner. This allows adjustment of the agent's access rights on a foreign host depending on its origin; 2. Secure computations.
→ SECURITY MECHANISMS: 1.1. The service broker signs the agent, that includes specific information about the customer necessary to run the corresponding core service (compare section 5); 1.2. Signature of the agent's owner on the agent's code ^{SeMoA}; 2. Dependent on the trade-off between trust and communication overhead a protocol to enable secure computations might be chosen [9].
→ REMAINING RISKS: 1. Little to no remaining risks in context of strong cryptography; 2. Dependent on effort to be made.

Service component execution on agent server of corresponding information provider: When the service component wants to access information sources of an information provider it has to be executed on a locally installed agent server.

→ THREATS: 1. Foreign agents might access information sources without having properly subscribed to the information provider, to enable payment schemes for example; 2. Monitoring and manipulation of agent data processing by the information provider.
→ SECURITY REQUIREMENTS: 1. Authorization of the agent; 2. Secure computations.
→ SECURITY MECHANISMS: 1. Presentation of a valid ticket (see 5); 2. Dependent on the trade-off between trust and communication overhead a protocol to enable secure computations might be chosen [9].
→ REMAINING RISKS: 1. Little to no remaining risks in the context of strong cryptography; 2. Dependent on the effort to be made.

Interaction between service components: Services consisting of more than one service component have two possibilities of collaboration. In the first case, they migrate to one agent server and exchange messages through a locally installed interface. In the second case, they make use of inter-agent communication mechanisms which enable message exchange transparent from their current location.

- → THREATS: 1. A malicous agent might take over a foreign identity as communication partner; 2. Exchanged messages might be read or manipulated.
- → SECURITY REQUIREMENTS: 1. Mutual authentication of the agent user's identity and subsequently authorization of communication, 2. Confidentiality and integrity of exchanged messages.
- → SECURITY MECHANISMS: 1. Cryptographical signatures on the agent's code ^{SeMoA}, 2. Encryption and signatures on the exchange messages ^{SeMoA}.
- → REMAINING RISKS: Assuming trust in the underlying execution environment, which provides secure communication as feature, there are only little to no remaining risks in the context of strong cryptography.

Interaction between user and service components: To start a service component the user generally has to initiate it with a set of start parameters. Subsequently service components present a result after they have performed their task. In some cases interaction is even necessary while the service component is processing its tasks.

- → THREATS: A malicious entity might take over the control of a foreign service component.
- → SECURITY REQUIREMENTS: Authorisation of the user facing the service component.
- → SECURITY MECHANISMS: Shared secret.
- → REMAINING RISKS: Dependent on the attacking entity (Since the agent has to use its shared secret within a foreign execution environment, this execution environment can take over control subsequently).

5 Code-Signing

Code-signing as a mechanism to ensure the code's integrity and to allow the unequivocal identification of the corresponding signer has already been implemented as a basic security mechanism in SeMoA (see section 1.1) and in the context of Signed Java Archives [3]. In this section we introduce signing of agent code in the context of *secure software distribution* within our mobile scenario.

From the time when an agent is instantiated as a mobile service component on an agent server by the user and subsequently is running and migrating within the mobile agent system, we can rely on the security mechanisms provided by SeMoA. But if we consider the process of agent creation and download, we are out of scope of the security concerns of any execution environment for agents. In this case we have to provide additional security mechanisms to ensure the stated security requirements in section 4.3. Especially when thinking about the entire lifecycle of an agent in our scenario: an agent developed by a service provider, provided by a service broker, downloaded and instantiated by the user and finally accessing information sources of some information provider, two questions arise.

– Who guarantees that the agent a user has just downloaded from a service broker's web portal, is fulfilling exactly the tasks announced on the web portal?
– Why should an information provider let an agent be executed on his agent server and grant this agent access to his information sources, if he doesn't even know who has programmed that agent?

SeMoA already provides the agent with a signature of its owner to identify the user, and with a signature of the last sender to identify the agent server the agent comes from. But in this case it might furthermore be interesting being able to identify the service provider and service broker of an arbitrary agent.

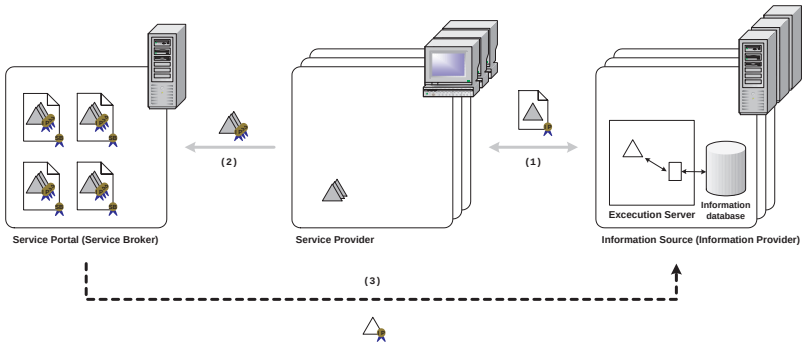

Fig. 3. Relationship between Service Portal, Service Provider and Information Provider.

The solution we propose is slightly different (see figure 3). Since the information providers, service providers and service brokers often work together very closely, we do not enable the identification of the service broker and service provider of an agent, but the identification of the service broker of an agent coupled with the identification of each information provider the agent is going to cooperate with.

With the service broker's signature of an agent, he assures the integrity of the agent's code and thereby his responsibility for the agent's behaviour. The user can appeal to this responsibility if something unexpected happens during the agent's execution. Another signature of the service broker and the information provider's signature can be seen as a *ticket* to allow usage of the service broker's resources in terms of his agent server and to allow access to provider's information sources. To prevent monitoring the first signature of the service broker should be ripped of the agent's code when it is initiated on the user's server. Furthermore the signatures used as tickets should be encrypted, so that only the corresponding issuer is able to decrypt and verify his signature in terms of access control.

The signing process can be realized in serveral ways, depending on the relationships between service broker, service provider and information provider:

1. In case the service broker and the information providers work together very closely, it should be the job of the service broker to ensure the harmlessness of the agent to avoid the malicious agent problem, as far as possible. In a first step he acquires the encrypted

signatures of all involved information providers as part of a contract. Next he signs the agent's code himself and thereby certifies his responsibility for the agent's behaviour.
2. In case the service provider and the information provider work together very close, the information provider should directly inspect the agent and add his encrypted signature as ticket to permit the agent to access his information sources later on. Next the service broker again signes the agent.
3. The service provider and information provider can even be seen as one entity in our infrastructure, if the information provider directly develops agents operating on his own information sources. This situation is similar to the second one.

6 Conclusion

Secure mobile and location based services rely on trustworthy mobile devices, secure communication technologies and infrastructures, end-to-end security, and a reasonable multilateral security concept. Due to current weaknesses of available communication technology for mobile services on the one hand and poor security and authentication mechanisms within mobile devices on the other hand one way to provide end-to-end security is on the basis of a multi-agent platform.

This paper has introduced a multi-lateral secure architecture for mobile applications and location based services representatively based on the multi-agent system SeMoA. Although a multi-agent system at a first glance is just another proprietary middleware platform, the underlying concept makes mobile agents particularly useful. In our approach mobile agents represent both the users and the services using a homogenous access paradigm and making sure that each party involved easily interoperate with each other.

The concept of two security domains separates the user's security domain (mobile device and homebase) from the outside world, i. e. the service and information providers. The relationships and interactions between these roles and their related agents are described in detail.

Multilateral security, i. e. balancing the interests of users and service and information providers, has been realised by identifying and analysing every single transaction taking place between specific instances in our architecture. Especially, these transactions are taking into accout the user's profile stored at the user's homebase. Once the user's profile is configured the personal agent manages the access autonomously on the user's behalf and grants and delegates certain access to certified service agents.

In the meanwhile the user is freed of the obligation to monitor the application's progress permanently. In order to protect the user's privacy the user's profile is securely stored on a trusted server instead of on the mobile device and is only accessible through the personal agent.

References

1. Common Criteria for IT Security Evaluation (CC), June 1999. ISO IS 15408, Version 2.1.
2. Stan Franklin and Art Graesser. Is it an agent, or just a program? In *Intelligent Agents III*, volume 1193 of *Lecture Notes in Artificial Intelligence*, pages 21–36, Berlin, 1997. Springer Verlag.

3. Li Gong. Java^{TM} Security Architecture (JDK1.2). Technical report, Sun Microsystems Inc., October 1998.
4. Günter Karjoth, Danny B. Lange, and Mitsuru Oshima. A security model for aglets. *IEEE Internet Computing*, pages 68–77, July·August 1997.
5. Günter Karjoth, Danny B. Lange, and Mitsuru Oshima. A security model for aglets. In *Mobile Agents and Security*, volume 1419 of *Lecture Notes in Computer Science*, pages 1–14. Springer Verlag, 1998.
6. Ernö Kovacs, Klaus Röhrle, and Björn Schiemann. Adaptive mobile access to context-aware services. *ASAMA - First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents*, October 1999.
7. Danny B. Lange and Mitsuru Oshima. Seven good reasons for mobile agents. *Communications of the ACM*, 42(3):88–89, March 1999.
8. Tod Papaioannou. *On the Structuring of Distributed Systems: The Argument of Mobility*. Ph.d. thesis, Loughborough University, February 2000.
9. Frank Piessens, Bart De Decker, Erik Van Hoeymissen, and Gregory Neven. On the trade-off beteween communication and trust in secure computations.
10. Volker Roth. Secure Recording of Itineraries through Cooperating Agents. In *Proc. 4th ECOOP Workshop on Mobile Object Systems: Secure Internet Mobile Computations*, pages 147–154, Brussels, Belgium, July 1998. INRIA.
11. Volker Roth. *Sichere verteilte Indexierung und Suche von digitalen Bildern*. Ph.D. thesis, Technische Universität Darmstadt, Wilhelminenstraße 7, 64283 Darmstadt, Germany, June 2001.
12. Volker Roth and Mehrdad Jalali. Concepts and Architecture of a Security-centric Mobile Agent Server. In *Proc. Fifth International Symposium on Autonomous Decentralized Systems (ISADS 2001)*, Dallas, Texas, U.S.A., March 2001. IEEE Computer Society Press.
13. Volker Roth and Jan Peters. A Scalable and Secure Global Tracking Service for Mobile Agents. In *Proc. Mobile Agents 2001*, Lecture Notes in Computer Science. Springer Verlag, December 2001.
14. Norman Sadeh. A semantic web environment for context-aware mobile services. *Wireless World Research Forum Conference*, September 2001.
15. Norman Sadeh, Enoch Chan, and Linh Van. MyCampus: An Agent-Based Environment for Context-Aware Mobile Services. *AAMAS - First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2002.
16. Tomas Sander and Christian F. Tschudin. Protecting mobile agents against malicious hosts. In *Mobile Agents and Security*, volume 1419 of *Lecture Notes in Computer Science*, pages 44–60. Springer Verlag, 1998.
17. SeMoA Project. Website. Available at URL `http://www.semoa.org`.
18. Ralf Sesseler and Sahin Albayrak. Service-ware framework for developing 3g mobile services. *Sixteenth International Symposium on Computer and Information Sciences*, November 2001. Available at URL `http://www.dai-lab.de/`.
19. Ralf Sesseler, Alexander Keiblinger, and Nicolas Varone. Software agent technology in mobile service environments. *13. International Symposium on Methodologies for Intelligent Systems*, June 2002. Available at URL `http://www.dai-lab.de/`.