Access Control and Privacy in Location-aware Services for Mobile Organizations

Maria Luisa Damiani
Dipartimento di Informatica e Comunicazione
Università di Milano, I
EPFL-IC-LBD, Lausanne, CH
damiani@dico.unimi.it

Elisa Bertino
Department of Computer Sciences,
Purdue University
bertino@cerias.purdue.edu

Abstract

In mobile organizations such as enterprises operating on field, healthcare organizations and military and civilian coalitions, individuals, because of the role they have, may need to access common information resources through location-aware applications. To enable a controlled and privacy preserving access to such applications, a comprehensive conceptual framework for an access control system enhanced with location privacy is presented.

1 Introduction

A challenging application area for mobile technology is represented by the development of location-based services (LBS) for mobile organizations. By *mobile organization* we mean a community of individuals that, because of the role they have, need to access common information resources through location-based services. Mobile organizations include enterprises operating on field such as fleet management and resources tracking, healthcare and leisure organizations, military and civilian coalitions created in response to a crisis, e.g., natural disaster, humanitarian relief and war. In these organizations the mobile members are characterized not only by an identity but also by a *role*. The concept of role is used in various contexts, e.g. CSCW, Security, Distributed Artificial Intelligence to organize and structure the division of responsibility. Broadly speaking a role is a set of rights and duties assigned to a subject who plays that role.

As running example consider an organization in charge of the management of a national park in which LBS are used to support the mobile personnel as well the visitors of the park. Individuals have different roles, say rangers, scientists, employees and tourists. Furthermore they are equipped with location-aware mobile terminals with which they invoke LBS, such as map services, needed for their activity. This scenario introduces a number of important requirements:

- Because of the different activities of the organization's members, it is reasonable to consider that LBS (services for short) are requested based on the roles of individuals. For example, the services which are available to a ranger may be different from those made available to tourists, not simply because of the individual preferences but mainly because of organizational and functional reasons. Such requirements calls for the development of an *access control mechanism* supporting the specification of which user can access which service in which context, based also on the role of the user.

- The accessibility of services may depend also on the position of the user. For example, the visitors of the park may be allowed to request services only when located in the area of the park. It seems thus important to extend the classical notion of access control mechanism to account for the mobility of the user under the hypothesis of a *bounded space*. Space may be bounded for several reasons: because of physical and technological constraints (e.g. the boundaries of the the park, the network extent), or because of marketing choices (e.g. the broader is the area assigned to the tourist, the more the tourist pays) or for security reasons (e.g. services in military zones). Furthermore, the extent of the reference space may depend on the user's role. We refer to the roles which are meaningful over a limited portion of space as *spatial roles*.

- Privacy concerns are also very relevant because of the capability of the technologies to collect, store and disclose the location of individuals. Privacy involving location is commonly referred to as *location privacy* [1]. Concern for location privacy raises primarily because each time a service is requested, the identity and the location of the user are transferred to the service provider and then possibly recorded. John, who is a manager of a business enterprise, however, fears that malicious parties may eavesdrop personal location data, and then, by relating these data with external sources, make inferences on his business activity. Therefore John wants to control what location data are retained. Several proposals have been developed like [4, 6, 7] to address such issue. A common approach, which has a lot of variations, is to perturb individual location data before they being transmitted or disclosed. In general, however, in these approaches the organizational dimension is not considered that is what we focus on here.

To address these requirements, access control systems must on one hand be based on functional roles and locations of users, and on the other hand be integrated with location privacy preserving techniques. Such an access control system would thus be able to support the specification and enforcement of location-based and function-based access control policies without however resulting in privacy breaches. To date no such comprehensive approaches have been developed.

The goal of this work is to propose such a comprehensive approach. In particular we propose an architecture for a privacy-preserving access control system for mobile organizations. The system filters the requests for service sent by the user, determines whether the request can be accepted based on the role and position of the user, and forwards an anonymous request together with a perturbed location to the application server which implements the requested service. The architecture is based on GEO-RBAC [2], an access control model which extends the RBAC model (Role Based Access Control Model) defined by NIST [5] with the concept of *spatial role*. In this model, a user is authorized to play a role in a session only when located in the space associated with the role, that is, the *role extent*. A spatial role can thus be in the *enabled* state or in the *disabled* state, depending on the user position. The *access control function* determines the set of enabled roles according to time and position and consequently the services which are available to the user at a given location and time instant.

In the paper we focus on the following aspects: i) the architecture of the access control mechanism; ii) the strategy for location privacy preserving. Both those aspects have not been investigated as part of our past work. In particular, issues related to privacy for GEO-RBAC and for other RBAC systems have not been yet investigated.

The major contributions of the paper are: i) definition of the semantics of the access control function. In particular we enhance the model of spatial role, introducing the distinction between *replaceable* and *non-replaceable* role. A replaceable role is a role which can be replaced by a "less powerful" role. An algorithm is then defined to determine the set of time varying enabled roles in the extended model. ii) Definition of an event-based architectural framework for the access control system. When the status of some user's role changes in time, the event is notified to the corresponding mobile terminal. The user can thus be aware at each time instant of the available services. iii) Definition of a privacy strategy which enables the user to dynamically control which location data are transmitted, according also to the organizational privacy rules.

The paper is organized as follows. In the next section we discuss the semantics of the access control function. In the subsequent section we introduce the architecture of the system and in particular the access control mechanism implementing the above function and the location privacy strategy. Next we discuss some open issues. Conclusions and future work are reported in the final section.

2 Preliminaries

Before discussing the semantics of the access control function, we briefly review the basic concepts underlying the reference model GEO-RBAC. We focus in particular on the following components of the model: a) the position model; b) the spatial role model; c) the spatial role hierarchy. For a complete description of the model, we refer the reader to [2].

2.1 The position model

The position model describes the position of the user. The model is based on the distinction between *real* and *logical* position. The *real position* corresponds to the position of the user on Earth acquired through some positioning technology. Real positions can be represented as geometries of different types since, depending on the chosen technology and accuracy requirements, they may correspond to points or polygons. By contrast, the *logical position* supports a representation of positions that is almost independent from the underlying positioning technology. Further, besides a geometry, it has a semantics. For example, logical positions can be a house, an address number, or a road. In compliance with current standards, the semantics is denoted by a spatial object type (*spatial feature type*). The logical position is computed from real positions by using a *location mapping function*. For example, a location mapping function can be defined to map a position acquired through GPS onto the closer road segment. Logical positions are crucial in supporting advanced LBS and in enabling knowledge-based spatial applications.

2.2 Spatial roles

A spatial role (role for short) describes a spatially bounded function for a user, or set of users. A role has a *role name* and a *role extent*. The role extent defines the boundaries of the space in which the role can be played by the user. For example $ParkRanger(Yellowstone)$ is a spatial role: *ParkRanger* is the role name and *Yellowstone* is the role extent, that is, the identifier of a spatial object describing a region in the reference space. Each role is assigned a set of *permissions*. A permission corresponds to a service. Thus saying that a permission prm is conferred on a role r, means that the users playing role r are enabled to access the service corresponding to prm However, for a role r to be effective the user must be located in the extent of r. When such an event occurs, the role is *enabled*; it is *disabled*, otherwise. Further, the model distinguishes between *role instances* and *role schemas*. Role instances are *ParkRanger (Yellowstone)* and *ParkRanger(AleshPark)*. The role schema specifies the properties which are common to the set of role instances having the same name. Specifically, a role schema defines: a common name for a set of roles, the type of role extent, the type of logical location, and the mapping function relating the real position with the logical position. For example: *ParkRanger(Park, ParkSector, m)* is the schema for park ranger roles defined for the areas of the park. The logical position of the users is identified by the sector of the park, supposed the park is subdivided in zones, which is computed by applying function *m* to the real position.

2.3 Spatial roles hierarchy

Spatial roles are organized in a hierarchy, defined by introducing a partial order $\preceq$ over the set of roles. We introduce the concept in a simplified manner through an example. Consider the two roles R1=$ParkEmployee(region1)$ and R2=$ParkRanger(region2)$. If we state that $R1 \preceq R2$ it means that: a) an individual which is a park ranger is also an employee. Therefore the permissions which are assigned to an employee are assigned also to a park ranger; the role extent of $R2$ is spatially contained in the role extent of $R1$, that is, *region2* is contained in *region1*; if role $R2$ is enabled, $R1$ is also enabled. Moreover the logical position of the park ranger in $region1$ is spatially contained in the logical position computed for the same user as employee in $region2$.

The use of role hierarchies is instrumental in simplifying both the specification and the management of roles as well as of their permissions. The hierarchy of roles is formally represented by extending the notion of *role graph* [11]. In particular we use the role graph to represent the set of roles through a lattice in which the nodes are the roles and the edges represent the precedence relationship. In addition to the user-defined roles, every role graph has a *MaxRole* ($\top$) and a *MinRole* ($\perp$). MaxRole has assigned the union of all permissions. Further, because the role is spatial, it has also an extent. Such extent results from the intersection of all extents, and can be empty. MaxRole is introduced to ensure that the common precedence relationship is always defined; however, it is most likely the case that, in order to improve security through separation of duty, no user is assigned the permission to use this role. MinRole represents the minimum set of privileges available to all roles, possibly empty. Like MaxRole, MinRole has an extent. In this case the extent is the whole reference space,i.e. *SPACE*. We draw the role graph without redundant edges through a Hasse diagram. In the paper we use the convention that the MinRole is drawn at the top. A role preceding r is an *ancestor* of r.

As an example, consider the set of roles R=$\{A(s_0), B(s_1), C(s_2), D(s_3), E(s_4), F(s_5)\}$ where $X(Ext)$ denotes the role X with a spatial extent identified by Ext. Without loosing in generality, we assume that roles are univocally identified by their names. Assume $A(s_0) \preceq B(s_1)$; $A(s_0) \preceq C(s_2)$; $A(s_0) \preceq F(s_5)$; $B(s_1) \preceq D(s_3)$; $B(s_1) \preceq E(s_4)$ and $C(s_2) \preceq E(s_4)$. The corresponding graph is reported in Fig.1(a) .

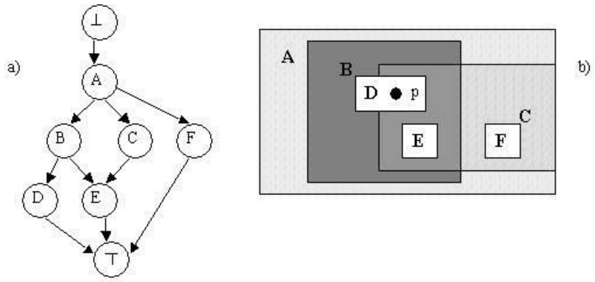

Figure 1. Role hierarchy (a) and role extents (b).

For sake of readability, in the graphical representation, the nodes of the graph are labeled only with roles names. An arrow from X to Y means $X \preceq Y$. We assume the role extents in Fig.1(b). Note that the extents of two non-comparable roles, thus roles which are not the one ancestor of the other, such as roles B and C, can overlap. Therefore, if a user is located in the intersection area of two roles, both of them will be enabled. Moreover, it should be noticed that the containment relationship between extents is not a sufficient condition for the roles to be comparable, since the role ordering is application-dependent. In the example, the extent of role F is contained in that of C , but the roles are not comparable.

3 Access control function

The first issue we address concerns the definition of the operational meaning of the access control function. Basically, the access control function determines whether a given permission, i.e. service, can be granted to a user in position p. When a user connects to the system a new *session* is activated and a number of roles are selected to be included in the *session role* set. However, for a session role to be enabled, the user should be located within the space of the role extent. Hence, as users are mobile, the set of enabled roles within a session changes in time. Let us consider now how such a set is computed. The problem can be formulated as follows: *given a set of session roles S and given a point $p \in SPACE$, determine the roles in S which are enabled when the user of the session is in position p.* With reference to the role graph in Fig.1, assume that the set S of session roles consists of the roles $S = \{D, E\}$. Which roles are enabled when the user is in position p?

We claim that there is no a unique answer. In the example, we can devise two interpretations: a) the first one is that the set of enabled roles is $ER = \{D, B, A\}$. The motivation is that *p* is contained in only one of the two session role extents, in particular in the extent of *D* , therefore ER contains *D* as well as its ancestors, because of the definition of hierarchy, that is B and A; b) the second interpretation is that $ER = \{D, C, B, A\}$, which differs from the previous one because of element C. The motivation is that *p* is contained not only in the extent of D but also in the extent of C. Because C is an ancestor of E, which is a session role, it seems reasonable to include it in the set of enabled roles. The intuition behind this second interpretation is that the user not only can play the roles that he has selected but also a weaker version of them, represented by their ancestors in the role hierarchy.

The above example shows that, in order to determine the set of enabled roles, we have to define what happens if a role is *not enabled*. As we have seen, there are two possible interpretations. Which of them is the most suitable depends on the semantics of roles and thus on the requirements of the applications. Therefore, for the sake of generality, we propose a model in which the behavior to adopt is assumed to be specific for each role and explicitly defined. When the user is allowed to play, in place of the session role r, a weaker role, in the sense discussed above, we say that r is a *replaceable role* (R-role); otherwise we say that r is a *non replaceable* role (NR-role). We call this property of the role *replacement property*.

3.1 A Model for R-roles and NR-roles

Given a role r, the replacement property of r is expressed by an attribute $dist$ (distance) of the role indicating the maximum distance which is allowed for an ancestor to replace the role. Specifically, $dist = 0$ means that the role cannot be replaced by any other role and thus it is a NR-role; if $dist > 0$ then the role can be replaced by a role at maximum distance $dist$ and therefore it is a R-role. By restricting the value of the distance attribute, we can introduce the constraint that a role cannot be replaced by a role that is *too far*, thus too generic. This results in a greater expressivity. The algorithm for computing the set of enabled roles for a user located in a given position is defined as follows.

Algorithm 1 *Given a session ss of user u, consider a set S of session roles in ss consisting of R-roles and NR-roles . Assign $ER = \oslash$. Then for each role $r \in S$ do:*

1. *Determine the real position rp of user u;*
2. *Compute the logical position lp, based on the role schema of r;*
3. *Determine whether lp is contained in the extent of r. If it is the case, then r is enabled and added to the set ER if not already present;*
4. *If the role r is not enabled but it is a* R-role*, determine whether one or more ancestor roles exist at the maximum distance specified by the* replacement *property which can be enabled. If it is the case, the roles are added to the set ER;*
5. *Add to the set ER the ancestors of its member roles.*

Consider the running example. Assume the session roles D and E to be R-roles which can be replaced by the roles at distance *dist=1*. Given the user in position p, let us build the set ER. Because p is contained in the extent of D, the session role is enabled; thus ER= $\{D\}$. Then, for each disabled session role, the ancestors which are not included in ER are considered. The only disabled role is E while the closest ancestor of E at distance 1 is C which can be enabled and thus added to the ER. The final set ER includes the ancestors of the previous roles, that is *ER*=$\{$*D,B,A,C*$\}$.

3.2 Extended roles in GEO-RBAC

In order to support the formal representation of the replacement property, we have extended GEO-RBAC. Specifically, the *replacement property* has been specified both in the role schema, to make it possible applying the property to all the instances of a role, and in the role instance definition, in order to characterize the single instance.

The role schema is thus extended with the $dist$ attribute which defines whether a role is a R-role or a NR-role. All the instances sharing the same schema inherit the same distance value. To enhance flexibility, the value of the property can however be specified also for single instances. The structure of the role instance is thus extended with an attribute labeled $dist$ which, if not NULL, overrides the value of the corresponding attribute defined at schema level. The value defined at schema level is thus the *default* value for all the instances. The definition of the extended schema and role instance is given next.

Definition 1 (Extended Role schema) *Let R, REXT_FT and LPOS_FT be the set of roles and the spatial feature types of, respectively, the role extent and the logical position. A Role Schema is a tuple:* $< r, ext, loc, m_{loc}, dist >$ *where:* $r \in R$*;* $ext \in REXT\_FT$*;* $loc \in LPOS\_FT$*;* m_{loc} *is a location mapping function for feature type* loc*;* $dist \in N$ *is the* distance *which indicates whether the role is replaceable or not.* $dist = 0$ *means that the role is a NR-role;* $dist > 0$ *means that the role is a R-role which can be replaced by any ancestor at a maximum distance* $dist$ *from the role.*

As an example, consider a possible schema for role E: $< E, Ext_E, Loc_E, m_E, 1 >$. In this case, role E is a R-role since the distance is 1. It means that, unless differently specified, all role instances of E have the same value for the replacement property. Specifically, the instances of E can be replaced by the roles instances at most at distance 1, that is those denoted as B and C. Note that a single role can be replaced by one or more roles. Further, a role cannot be replaced by the $\perp$ role.

Definition 2 (Extended Role Instance) *Given a role schema* r_s*, an instance* r_i *of* r_s *is a triple* $< r, e, dist >$ *where:* r *is the name of the role in schema* r_s*;* $e \in F$ *is a spatial feature of the type specified in the role schema, i.e.* $r_s.ext$*;* $dist \in N$ *the distance, defined as above, for the specific instance or undefined (*NULL*).*

It should be noticed that the distance property specified at the instance level is meant, if not NULL, to override the one defined in the corresponding schema.

Furthermore, for how the model is defined, given a set of enabled roles, we can map the real position of the user onto the semantic locations associated with the enabled roles, and thus obtain a set of *logical positions*. This set is partially ordered with respect to the spatial containment relationship, thus resulting in a lattice structure, that we refer to as *Location Graph* (LG). We will use the notion of LG later on in the paper.

4 A Reference Architecture for the Access Control System

After presenting the operational meaning of the access control function, we address now issues related to a reference architecture for an access control system based on the proposed model. We base our architecture on the general architectural framework reported in Fig.2. Such framework consists of three fundamental components:

- a set of *mobile users* equipped with mobile terminals and connected to a wireless network. Users are assumed to be identified by their terminal ID;
- the *Application Server* providing a set of location-aware information services;
- the *Access Control System* (ACS). It is a trusted component filtering the users' requests and protecting location privacy. To obtain the position of the user, the ACS accesses a *Location Server* which aggregates location data from different sources such as the network and mobile terminals equipped with GPS (connected by dotted lines in Fig.2) and responds to queries such as *retrieve the position of terminal ID*. Notice that in order to prevent uncontrolled disclosure of location data, the ACS is the only component enabled to query the Location Server.

A request for a service is processed as follows: the user requests the service by sending a request message to the ACS. The ACS then determines whether the request can be accepted and if it the case the request message is properly re-structured and sent to the Application Server. Finally, the requested information is then sent back to the ACS and then, through it, to the user.

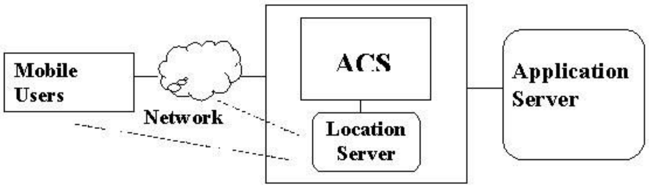

Figure 2. General architecture

In this section, we focus in particular on the key choices underlying the architecture, concerning: a) the definition of an event-driven approach to the specification of the access control mechanism; b) the definition of a privacy strategy which enables the user, at least to some extent, to control the location data which are transmitted to the Application Server.

4.1 The access control mechanism

The access control mechanism applies the access control function to determine the set of enabled roles and thus the services which are accessible to the user at a given location. For the computation of the enabled roles, in addition to the previously specified algorithm, we need to define at what stage these roles are determined. We devise two possible strategies:

- The status of roles is computed exclusively upon user request. When the user sends a request to the ACS, the system determines which roles are enabled and then based on this information, determines whether the request is accepted or rejected. The approach is thus *user-driven*;

- The status of roles is autonomously checked by an agent tracking the position of the user connected to the LBS system. As a state transition occurs for a role, that is, its status changes from enabled to disabled or vice versa, the new status is recorded and the event is notified to interested system components. The strategy is thus *event-driven*.

The simplest approach is the user-driven one because the position of the user is computed on demand and thus the status of roles is determined only when required. This approach has however a major drawback, in that the services which are available in a given session at a given instant are not known until the user makes an explicit access request. Therefore it may occur that the user requires a service which is not accessible in that position or that the user is not aware of available services at a given position.

Conversely when the event-driven strategy is adopted, the status of each role is determined asynchronously with respect to user requests. Therefore such information needs to be recorded by the ACS. Though more complex, this approach overcomes the drawbacks of the user-driven strategy: user requests can be more efficiently processed because the current status of roles is available at the time of the request and thus need not to be computed; the user can determine the effective roles she can play, before a request is made; finally, as we will see, this strategy supports services based on the push model.

Because it is arguably more flexible and, additionally, it is functional to the privacy-preserving strategy that will be introduced later on, the event-driven approach is the one we adopt. From an architectural point of view, the proposed organization for the ACS is based on the following major components:

- The *Policy DB* is a data base storing the security policies specifying, among other information, the spatial roles, the services available to each role, and the roles assigned to each user;

- The *Session DB* records the *status* of sessions. The status at time t of session s is represented by the tuple: $< s, t, SR, ER, LG >$ where: SR is the set of session roles, thus the roles selected by the user among those which have been assigned to her; ER is the set of roles enabled in s at time t; LG is the Location Graph corresponding to ER. The nodes of the Location Graph are the logical positions corresponding to roles in ER. The Location Graph thus represents the semantic location of the user at different levels of granularity. As we will see later on, this information is used ro protect location privacy .

- An agent called *Role Tracker*. It periodically retrieves from the Location Server the position of the users of current sessions and determines whether a state transition has occurred for the roles of each session. If this is the case, the event is communicated to the Event Manager;

- In response to a Role Tracker event, the *Event Manager* updates the status of sessions in *Session DB* and notifies the event to the corresponding terminal.

4.2 The privacy preserving access strategy

A user requiring a service transmits to the system, besides the service identifier, also her identity and position. To protect privacy, the user may wish to control the storage of personal information and in particular what information about location and user identity are transmitted.

To address this issue, we propose a strategy which aims at integrating privacy policies defined at organizational level with the preferences of the individual, by letting the user dynamically specify, at least to some extent, the granularity of the identity and of the location. More specifically, the key aspects of our approach are as follows:

i) Location data are perturbed before being transmitted to the Application Server. This strategy is adopted, in various forms, by most of the approaches supporting location privacy. Specifically, the strategy we adopt is to cloack the location by decreasing the spatial granularity of position, that is the detail of its geometric representation to obtain thus an uncertain position (*spatial generalization*);

ii) Spatial generalization is role-dependent, in that it can be differently applied depending on the role of users. For example the position of a field sales agent may have a granularity which is different from that of a marketing manager, not only because they are at different levels in the organizational hierarchy and thus may have different privacy rights, but also because the localization may have different purposes and relevance.

4.2.1 Location data cloaking

The techniques for perturbing location data based on spatial generalization, opposed to those based on the idea of confusing data [8], have the advantage that the resulting information, though more imprecise, preserves the correctness of data and thus can be used for analysis purposes, such as data mining. When a spatial generalization is performed over a position, the geometric shape of the object is replaced by a coarser geometry. For example the position along a road, at the maximum granularity is represented as a point and at a coarser detail is expressed by the whole road or even a road at different scale.

In the pioneering work on location privacy in [7], the generalization of a point location is the tile of a quadtree-based data structure containing such a point; in [6], which further develops this idea, the coarser geometry is not statically defined, but results from a dynamic computation. On the other hand, spatial generalization has been extensively investigated also in the GIS field (Geographical Information System) and approaches have been developed, which although not specifically conceived to protect privacy, can likely be useful for that purpose like [12]. In our model, for the sake of generality, we do not specify any spatial generalization criteria. Rather we define the mechanisms which enables its specification. The basic idea is to specify the generalized location of the user in terms of *logical position*. We recall that, in our model, the logical position is computed dynamically during the process of access control enforcement, by applying the *location mapping function*. The location mapping function implements the generalization criteria. As a result, given a user in a position p, the system determines not only whether a session role r is enabled in p, but also the corresponding logical position, i.e. the perturbed location. The ACS thus maps the actual position of the user onto a location according to the specified access policies which have been specified by the security and privacy administrator. The logical location is then forwarded to the application server.

4.2.2 Dynamic privacy preferences

The above strategy introduces the following issue: since a user can play simultaneously multiple roles and because roles are organized in a hierarchy, the logical position of a user is not unique. Specifically the logical location of the user is described by a *Location Graph* (see Section 2). For example if the user in a position is enabled to play at the same time the roles of *tourist* and *driver*, two logical positions (at least) can be devised, such as an address (because of the tourist role) and a road (because of the driver role). The question that next arises is which logical position, among the possible ones represented in the Location Graph, shall be transmitted.

A reasonable strategy is to introduce a priority over locations, so that in case of conflicting representations, the location at the highest priority is selected. Because there is no reason to privilege a prioritization criteria over another, we consider that such priority is defined by the user when an access request is made.

The user specifies the logical position and thus the granularity of location data by selecting a role among those currently enabled. The selected role is then enclosed in the request. For example, the previous user specifies in the message that the request is sent in her role as tourist. In such a way, not only the ACS can solve the ambiguity posed by the availability of multiple logical locations but also the user has some control over which personal data at which granularity are transmitted. As a result the user can dynamically express privacy preferences, which must however be compatible with the policies specified at organizational level. It is important to notice that the user is aware of the current enabled roles, since the ACS keeps tracks of the roles which are enabled in time and, through the *Event Manager*, transmits such information to the terminals. The information flow from the user to the Application Server is reported in Fig.3. The messages are enclosed in brackets. Consider a session s, in a given status *st*=$< s, t, SR, ER, LG >$ where in particular *ER* is the set of enabled roles in s and LG the corresponding Location Graph. To invoke a service p, the user of the session sends to ACS an access request in the form: $< s, r, p >$ where $r \in ER$ is the selected role name. Hence the ACS determines the logical position $lp \in LG$ corresponding to role r and then the message $< id, lp, p >$ is forwarded to the Application Server. Note that, to identify the multiple requests that can arrive from the same session and also to protect the identity of the requestor, we replace the session identifier with a *request identifier* (id), which is specific of each single request. The association between *id* and the user is then recorded by ACS.

Notice that since our focus is on the control of location granularity, we have not addressed specifically the issue of user anonymity. To ensure anonymity the removal of the user identity is not a sufficient condition, since the location can be linked with external data and thus reveal the identity of the user. To address this issue, a possible approach is to extend the concept of *k-anonymity* to location [6, 7, 13]. The implications of such an approach over our model will be investigated as part of future work.

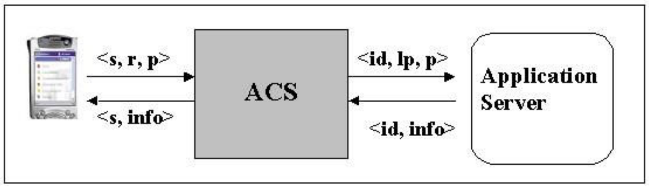

Figure 3. Access Control System: input and output.

5 Open issues

In designing the architecture of the ACS we have implicitly considered traditional LBS based on the *pull* model. Yellow pages and directory services fall in this category. We have not considered yet the class of services based on the *push* model. Under the push model services are still requested by the user (i.e. *subscribed*), but the information is provided on a continuous basis. An example is the service which allows one to be automatically notified about nearby traffic jams. Because in a location-aware context the availability of the service depends on the position of the user, if the user changes position in time, it may occur that the user is no longer in a position which authorizes him/her to access the service. To our knowledge this issue has not been addressed yet. Another important aspect to tackle is the integration of k-anonymity models for location and at the same time the extension of the privacy solutions from the case of single locations to the case of trajectories and paths. The definition of a metric for measuring the quality of service, when location data are perturbed, is also a major issue.

6 Related work

The most important contribution of this paper is the integration of spatially-aware access control policies and location privacy methods for users that require location-based services. To our knowledge this is the first approach which provides a comprehensive framework. In contrast there are several proposals which address only partially the problem by focusing either on the spatial dimension of access control or on privacy requirements. In particular, spatially-aware access control systems have been proposed to regulate the access to: raster and vector-based spatial data, spatial data defined at multiple granularities, and to deal with spatial contexts. On the other hand, approaches to location privacy, in addition to those focused on location cloaking and anonymity are concerned with privacy issues related to the disclosure of the information to third parties as in [9, 14]. In particular, most of these proposals adopt the idea that policies must be specified by a service provider explicitly stating how users location information can be used. In [4], a different research direction is proposed focused on the use of imprecise queries.

7 Conclusions

In the paper we have presented a location-aware access control model augmented with the capability of preserving location-privacy. The target users are individuals characterized by a functional role in a mobile community. The result of the work is a conceptual framework based on a well defined privacy and access control model the semantics of which has been specified both in set theoretic and operational terms. We have also emphasized the concept of *role*. We believe that the introduction of this concept in LBS opens the way to the development of a new category of services, we call *role-tracking*. Consider for example a tourist of the park wishing to locate a ranger in order to ask for some information. In this case a reasonable query is *Where is a ranger?*. The service is similar to the popular AT&T Find Friend, but in this case the object to be found is not a precise individual but rather an unspecified user playing the requested role. By combining our notion of role-tracking services with novel digital identity platforms [10], a variety of innovative LBS can be developed for use in a large number of domains.

As part of future work, we plan to investigate several directions. First we plan to develop a distributed architecture for the access control functions and to provide support for k-anonymity. We also plan to integrate this model with the X-GTRBAC system [3], an XML-based temporal access control model based on RBAC, in order to obtain an access control system supporting the specification and enforcement of a rich set of context-based access control policies. Finally, we plan to develop encryption-based access control techniques specifically tailored to space-based access control policies.

References

[1] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing. In *IEEE Pervasive Computing*, 3(1), 2003, pages 46–55.

[2] E. Bertino, B. Catania, M.L. Damiani,P. Perlasca. GEO-RBAC: A Spatially Aware RBAC. In *Proc. of the Tenth ACM Symposium on Access Control Models and Technologies (SACMAT 2005)*, Stockholm, Sweden, June 2005, pages 29-37.

[3] R. Bhatti, A. Ghafoor, E. Bertino, J. Joshi. X-GTRBAC: an XML-Based Policy Specification Framework and Architecture for Enterprise-wide Access Control. In *ACM Transactions on Information and System Security*, Vol.8, 2005, pages 187–227.

[4] R. Cheng, Y. Zhang, E. Bertino, S. Prabhakar. Querying Private Data in Moving-Object Environments. CERIAS Tech Report 2005-45 - Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette.

[5] D. Ferraiolo, R. Sandhu, S. Gavrila , R. Kuhn and R. Chandramouli. Proposed NIST Standard for Role-Based Access Control. In *ACM Transactions on Information and System Security*, Vol.4, 2001, pages 224–274.

[6] B. Gedik and L. Liu. Location Privacy in Mobile Systms:a Personalized Anonymization Model. In *Proc. of the 25th International Conference on Distributed Computing Systems (IEEE ICDCS)*, 2005.

[7] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. ACM/USENIX International Conference on Mobile Systems, Applications and Services (Mobisys 2003).*

[8] M. Gruteser, J. Bredin and D. Grunwald. Path Privacy in Location-aware Computing. In *Proc. of Workshop on Context Awareness, MobiSys 2004* , Boston, US, June 2004.

[9] C. Gunter and M. May. A Formal Privacy System and its Application to Location Based Services. D. Martin, A. Serjantov (Eds.): Privacy Enhancing Technologies, 4th International Workshop, PET 2004, Toronto, Canada, May 26-28, 2004, Revised Selected Papers. Lecture Notes in Computer Science 3424 Springer 2005.

[10] Identity-Management. Liberty alliance project. http://www.projectliberty.org.

[11] M. Nyanchama and S. Osborn. The role graph model and conflict of interest. In ACM Transactions on Information Systems Security,2(1):Pages 3 – 33

[12] S. Spaccapietra, C. Parent, and C. Vangenot. GIS Database: From Multiscale to MultiRepresentation. In *Abstraction, Reformulation, and Approximation*, Ed. B.Y.Choueiry and T.Walsh, LNAI 1864: Proceedings of the 4th International Symposium, SARA-2000, Horseshoe Bay, Texas, USA

[13] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. In *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10 (5), 2002; 571-588.

[14] M. Youssef, V. Atluri, and N. R. Adam. Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In *Proceedings of the 6th International Conference on Mobile data management*, Cyprus, 2005, pages 67–76.