Location-based Metadata and Negotiation Protocols for LBAC in a One-to-Many Scenario

Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani,
Sabrina De Capitani di Vimercati and Pierangela Samarati

Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
Via Bramante 65 - Crema - Italy
{ardagna,cremonini,damiani,decapita,samarati}@dti.unimi.it

Abstract. Location-based Access Control (LBAC) techniques allow the definition of users' access rights based on location predicates that exploit the users' physical location. However, evaluating the physical location of a user is a specialized activity that is unlikely to be performed by the same entity (e.g., organization or system) in charge of the access control decision. For this reason, location evaluation is usually assumed to be provided by specific Location Services (LSs) possibly coexisting in a same area and competing one with the others.
In this paper, we address the issues related to the communication and negotiation between an Access Control Engine (ACE) enforcing access rules that include location-based predicates and multiple, functionally equivalent, LSs. We introduce metadata for the exchange of service level agreement attributes between the ACE and the LSs. Based on such metadata we develop different negotiation protocols, from a basic negotiation protocol that shows the core aspects of our proposal to an enhanced protocol that enriches the interaction by taking into account a cost/benefit analysis and some service requirements. Finally, we present an extension to the enhanced protocol to consider possible time validity constraints on access control decisions.
Keywords: Access Control, Mobile System, Location-based Services, SLA, Negotiation Protocol

1 Introduction

Access control mechanisms have been traditionally designed based on the assumption that requesters must be identified through adequate information, often called *credentials*, to decide what actions they are authorized to perform on protected resources. Such a fundamental assumption still holds when among credentials we consider *location-based information*. However, the peculiar nature of this information requires to deal with them differently from what we are used to do with more conventional credentials such as, for instance, identities, roles, and affiliations.

Writing and evaluating an access control policy based on requesters' location information requires to take into account the many distinct aspects of monitoring users' locations, including the intrinsic *dynamics* that makes the information strictly time-dependent, the unavoidable *measuring error* that makes it dependent upon the location technology adopted and environmental conditions, and the many *privacy concerns* that a service for monitoring people locations inevitably arises. The first two aspects have direct consequences on the evaluation of access control policies that must be designed to deal with both the *time-variance* of location credentials, due to the on-going motion of requesters, and the *approximation of the measure* due to technological limitations. Privacy aspects, instead, are more related to the overall architecture that permits an access control component to receive location-based information. Location measures are performed by specialized location providers that own the technology and the infrastructure for collecting such an information. How they disclose location information to access control components must be ruled according to privacy conditions. To deal with these issues, *Location-based Access Control* (LBAC) systems have been developed. They are designed to evaluate authorization policies based on requesters' location information in addition to conventional credentials, and whose development have been fostered by the ongoing rapid development of a new generation of wireless and mobile networking devices suitable for being used as sensors by location technologies.

LBAC systems exploit a *Location verification* feature, provided by specialized components, which must be able to tolerate rapid context changes, because users, instead of being forced to work in a fixed, pre-set position like a computer, can now wander freely while initiating transactions by means of terminal devices like cell phones, smart phones and palmtops able to join the telephone network and/or a wi-fi network. For each mobile communication technologies, location verification may rely on different techniques, like measuring signal power losses and/or of transmission delays between terminals and wireless base stations or on specialized location sensing techniques like the well known *Global Positioning System* (GPS).

Mobile communication technologies and location sensing techniques can provide a rich set of location-based information to access control modules, not limited to the position of a requester when a certain access request is submitted. The direction where she is headed, her velocity and acceleration are other available information. Moreover, when location measures are coupled with a contextual description, for example the

topology of the environment where the requester is moving (e.g., a city map) and the type of motion (e.g., walking, by car, by train etc.), then advanced reasoning methods can be applied to foresee the requester position in a time frame. Also, in the near future, location-based services will provide a wealth of additional environment-related knowledge (for instance, whether or not the user is alone in a given area).

In this paper, we focus on the architectural issues arising when an access control component is faced to more location service providers, all able to serve the needed location information about requesters. Most advanced location-based platforms, like OpenWave Location Manager [19], can rely on multiple sources of location information (e.g., provided by multiple wi-fi or mobile phone operators) and on multiple techniques including Cell ID, assisted global positioning system (A-GPS), Angle Of Arrival (AOA), Enhanced Observed Time Difference (E-OTD) and others. Service providers have the technical means to provide a location service for LBAC systems, which, then, needs strategies for deciding which location service is the most convenient to join, based on its quality and cost. In particular, we discuss the communication and negotiation protocols that could be established between an access control component and many location service providers according to different scenarios of increasing complexity. To support these protocols, we define a set of metadata distinguished on the type of location technology and on the negotiated location predicates. The concept of Service Level Agreement (SLA) is used as the contractual means that an access control component and a location service provider adopt to agree upon and set quality of services attributes and the corresponding service cost.

2 Basic Concepts and Reference Scenario

2.1 Reference Architecture with Multiple Location Services

Evaluation of LBAC policies involves context data about location and timing that are made available by third parties through service interfaces called *location services*. In other words, a LBAC system evaluating a policy is not likely to have direct access to location information, since the location sensing technology is operated by specialized organizations (e.g., mobile phone companies) which cannot be freely share the information due to privacy constraints. Therefore, a LBAC system must interact with different location services by sending them location requests and waiting for the corresponding answers. Of course, the characteristics of these location services depend on the communication environment where

the user transaction takes place. One of the major challenges faced by a location-based scenario is the development of an infrastructure that permits the communication and negotiation between one service provider and several location services allowing the agreement on a particular SLA that represents the preference of the parties involved in the communication. This negotiation phase allows the LBAC system to select the most suitable location service for its purposes. Here, we focus on a mobile network, where location services are provided by mobile phone operators. Our LBAC architecture includes the following three entities.

Requestor: the entity whose access request to a service must be authorized by a LBAC system. We make no assumption about requestors, besides the fact that they carry terminals enabling authentication and location tracking.

Access Control Engine (ACE): the entity that implements the LBAC system (location-based service provider) used to authorize accesses to the available services. For evaluating access requests according to some LBAC policies, the ACE must communicate with a Location Service for gathering location information.

Location Services (LSs): entities that provide the location information at different levels of granularity and with different Quality of Service (QoS). The types of location requests that a location service can satisfy depend on the specific mobile technology, the methods applied for measuring users position, and environmental conditions.

Figure 1 shows the reference architecture. Interactions among a Requestor, the Access Control Engine and multiple Location Services are carried out via request/response message exchanges. In particular, in this paper we focus on the interaction between the ACE and the LSs. When the Access Control Engine receives requests that need the evaluation of some location-based attributes of the requestor, it chooses a LS from which to collect such an information. This task is carried out through a negotiation phase potentially involving all LSs able to provide this information. At the end of the negotiation, the ACE selects the most suitable LS, evaluates the location-based conditions, and finally returns a response to the Requestor. This functional decomposition emphasizes the fact that location functionalities are fully encapsulated within remote services set up and managed by specialized operators and that it is likely to have at the same time more operators able to provide functionally equivalent services.

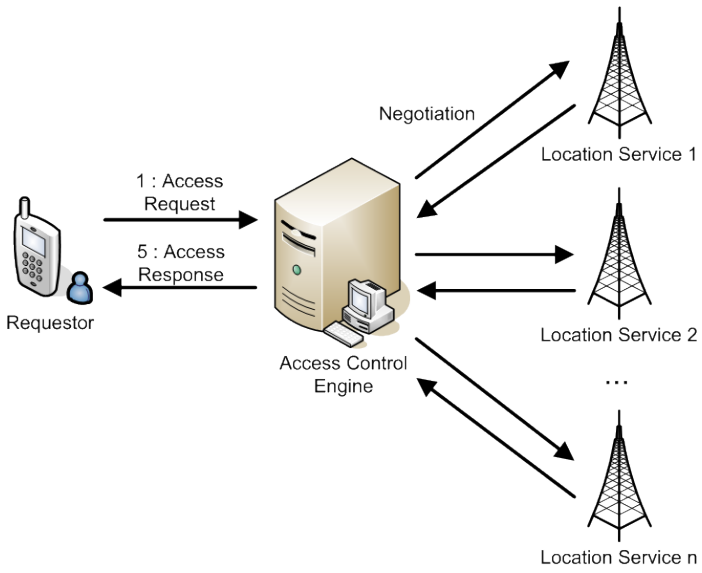

Fig. 1. Location-based Architecture

2.2 Metadata Associated with LSs

Semi-structured metadata formats are increasingly important for distributed services and are at the basis of important initiatives like the Semantic Web of the World Wide Web Consortium (W3C) for Web-based services. While traditionally exploited for information discovery and retrieval in networked environment, metadata have been also used in access control systems for selectively releasing data based on conditions on their metadata [5].

In our proposal, metadata are used to describe SLA conditions of location-based services. Each LS declares its service conditions by means of a collection of metadata, one for each location technology offered by the LS (e.g., cellular phones, GPS and WiFi cells). Those metadata are used by an ACE during the negotiation phase when one LS among the many available must be selected. For instance, metadata may report QoS levels such as the confidence level of a given location evaluation and the price for such a service. These service conditions, expressed by means of a collection of metadata, are to be intended as the "best offer" a LS proposes to an ACE. However, more evolved negotiation protocols, as described in the following sections, could even re-negotiate the SLA condition when the "best offer" is not satisfactory. For instance, it is likely that the price and confidence of a service are directly related, that is, a service with high

```
<SLA>
  <technology>GPS</technology>
  <price currency="Euro">5,00</price>
  <confidence>0,80</confidence>
  <timeout unit="second">120,00</timeout>
</SLA>
<SLA>
  <technology>Cellular Phone</technology>
  <price currency="Euro">13,00</price>
  <confidence>0,70</confidence>
  <timeout unit="second">240,00</timeout>
</SLA>
<SLA>
  <technology>802.11g</technology>
  <price currency="Euro">7,00</price>
  <confidence>0,60</confidence>
  <timeout unit="second">240,00</timeout>
</SLA >
```

Fig. 2. An example of a fragment of XML metadata document defined by a LS

level of confidence will be more expensive than one less reliable. The ACE may want to choose according to its own cost/benefit function. To make our approach generally applicable, we do not make any assumption on the format of metadata, which can be in the form of textual or semistructured documents (e.g., XML [1] or DDI [23]).

For metadata browsing as well as for the evaluation of conditions that may determine whether or not a given LS's SLA is acceptable, it is useful to evaluate metadata. While for textual metadata, we limit the granularity to the whole document, for semistructured metadata, we allow reference to finer grained content at the level of properties. In this way, we can specify, for example, a filtering expression stating that only metadata related to location technologies should be considered by the ACE. Such properties (elements and attributes, in the XML terminology) are referenced by means of path expressions written, for example, with the XPath language [25].

Figure 2 shows an example of XML-based metadata describing the technologies used by a LS and some SLA attributes.

3 Location-based conditions and predicates

A *location-based condition* is a condition involving a predicate whose value depends on location measurements performed by a Location Service. Location-based predicates have been investigated since long by the wireless network research community [2], trying to address critical issues like their time and space dependency. However, two key issues are specific to LBAC:

- *interoperability*: location tracking could rely on different sources of location information, depending on availability and cost;
- *uncertainty*: an approximation is intrinsic to each location measure that a Location Service performs.

While interoperability largely depends on roaming agreements between mobile phone operators and is more business-oriented in nature, uncertainty needs to be tackled effectively by a LBAC. Today, in the mobile network scenario, no technology is available ensuring a user location with 100% of accuracy [11]. In addition, location measurement is often unstable because of changing environmental conditions, such as reflection or interferences that may corrupt the signal [3]. However, different techniques are available, each one providing a specific level of granularity and precision and requiring different costs to be operated. This makes it possible to define different selection strategies based on different contexts and on the requirements in which the particular evaluation takes place [11].

3.1 Expressing location-based conditions

A first step for the support of location-based conditions in the authorization language is to identify how location information is queried and what kind of response the location service returns.

Traditional location-based services [13] usually assume queries to the location service to be of the form of *range queries* asking the Location Service to return an estimated range of values (possibly collapsing to a single one) for a predicate. Range queries can be modeled as functions of the form *predicate(parameters)* $\rightarrow$ *[range, accuracy, timeout]* stating that the evaluation of a location *predicate* over *parameters* returns a result of *range* (e.g. the center and radius of the circular area where a terminal is located). The range has a given *accuracy*, that is, an upper bound to the

deviation w.r.t. the true value, guaranteed by the location service[1] and is to be considered valid within the timeframe specified by *timeout*. For the sake of simplicity, in our model we consider queries to be of a simpler (although largely equivalent) form; namely, we shall deal with *boolean queries* asking the Location Service to assess whether a given value (or range thereof) of a location predicate is true or false. Boolean queries can be modeled as functions of the form:
predicate(parameters, value) $\longrightarrow$ *bool_value, confidence, timeout*; stating whether or not (depending on whether *bool_value* is `True` or `False`), *predicate* over *parameters* has the stated *value*. For instance, a query may ask whether a terminal is located inside a given region. Here, the assessment (`True` or `False`) has again a time validity specified by a *timeout* parameter; but instead of providing a measure accuracy, we assume that the location service attaches to answers a *confidence value* defined as follows:

- the *confidence value* expresses the level of reliability that the Location Service is willing to guarantee to the assessment (`True` or `False`), according to accuracy, environmental conditions, granularity of the requested location and measurement technique. While confidence is associated with measurement *accuracy*, which in turn depends on the technology used for localizing the requester, the quality of the location service and so forth, the form of this dependency is encapsulated within the location service.

Note that the *timeout* takes into account the fact that location values may change rapidly, even during policy evaluation. If the evaluation of a condition involving a predicate happens to start after the predicate timeout is expired, a predicate re-evaluation is triggered. Intuitively a range query with a condition on the returned range can be expressed as a boolean query where the condition is moved within the predicate itself. For instance, a condition requesting the area where the user is located (via the predicate `inarea`) and then evaluating whether the area is `Milan`, can be represented as a condition requesting whether it is true or false that the user is in `Milan` area. However, we remark that in range queries the location service can respond with different ranges and accuracy levels, thus varying the granularity of the response. For instance, the service can choose between an high-accuracy answer specifying a wide range (e.g., a

[1] The accuracy is a qualitative concept and should not be confused with *precision*, that is, the closeness of agreement between independent test results obtained under stipulated conditions.

city) or a low-accuracy answer specifying a smaller region (e.g., a building). In boolean queries, the accuracy is essentially established a-priori through a SLA negotiation step that considers both the requirements stated by the ACE and the functionalities offered by a LS. The LS, then, responds stating whether the predicate is true or false, together with the confidence it has in such a response.

The rationale behind our choice (i.e., boolean queries with a level of confidence) is that we want to decouple the physical measurement error (specific to the technology adopted, the environmental conditions and the service quality of the LS) from the access control conditions that the access control engine has to evaluate. The Location Service is in the best position for providing a confidence estimate, because associating confidence with a range requires educated guesses about the measured variable probability distribution, as well as the knowledge of the number of physical measurements actually taken by the sensors. Furthermore, our solution enables the Access Control Engine to evaluate location-based conditions without taking into account technological details of the location measurement process. An additional benefit of our approach is to foster interoperability between the ACE and multiple LSs, possibly relying on different location technologies. Given a certain confidence in the evaluation of a location-based predicate (e.g., a user has been positively localized in a given area with a confidence level of 90%), the ACE could compute the final outcome of boolean location-based predicates by means of its local *confidence thresholds*. Considering the agreements between a ACE and a LS, both the confidence and the timeout associated with location measures provided by a LS can be considered as QoS attributes and then formally negotiated and set as reference values through *Service Level Agreements* (SLAs). This way, the ACE has reference values contractually defined with a LS to estimate the reliability of a location measurement and can then decide whether or not trust it in the evaluation of an access control policy.

3.2 Location-based predicates

The definition of location-based predicates requires the identification of the kind of conditions that it might be useful to include in access control policies and whose evaluation is possible with today's technology. We identified three main classes of conditions:

- *movement-based* predicates evaluate conditions on the mobility of the users, such as their velocity, acceleration, or the direction where they are headed;
- *position-based* predicates evaluate conditions on the location of the user at a given instant (e.g., to evaluate whether a user is in a certain building or city or in the proximity of other entities);
- *interaction-based* predicates evaluate conditions relating multiple users or entities. An example is the number of users within a given area.

For the definition of specific location-based predicates corresponding to the above-mentioned classes, we refer to [4].

4 Communication and Negotiation between ACE and Multiple LSs

The communication between one ACE and several LSs could be carried out according to different well-established protocols developed in the Distributed Computing area. Considering communication, we focused on the distinctive aspects of location-based services and predicates evaluation and developed a set of possible communication protocols for the exchange of location-based attributes and data. The underlying assumption is that an ACE, when faced to more functionally equivalent LSs, will *negotiate* the service conditions before requesting the actual location parameter evaluation.

The negotiation phase could be designed according to many different requirements, stressing *performance* and then providing lightweight communication patterns, requiring the compliance to previously established *SLA agreements* that set minimum QoS standards and then matching service conditions offered by LSs with defined SLA parameters, or providing more elaborate patterns to achieve a complete *cost/benefit analysis* of the different service conditions offered by LSs.

4.1 Communication Strategies

Before discussing specific negotiation strategies, we describe the two *one-to-many* communication approaches that can be applied in our reference scenario. The first approach is a *parallel* communication between the ACE and all LSs. The rationale behind this choice is that in the parallel case, the ACE prefers to collect all service offerings from LSs and, among the received offers, to apply a selection criteria and enter in a negotiation

phase. The benefit of the parallel approach is that the location evaluation service selected will be the best among all those available at the time of the request. The drawback is the performance penalty due to the required computational efforts (i.e., the computational complexity is $O(N \cdot M)$ where N is the number of parallel negotiations and M is the average number of negotiation steps required to conclude a single process) and possible network latencies in gathering all offerings. The scalability of this approach is also limited. The second approach is a *serial* communication between the ACE and one LS at time. In this serial case, the ACE applies a threshold strategy, which consists in defining the values of all service parameters required to consider an offering as satisfactory. Next, according to a selection criteria the ACE selects a LS, negotiates the service conditions and decides whether or not the offering is satisfactory w.r.t. the service parameter thresholds. If the service is not satisfactory, another LS is selected and, the negotiation process is restarted. In the most general case, when all LSs are functionally equivalent, a *random choice* can be adopted. Otherwise, *prioritize techniques* could be applied, based, for example, on reputation of LSs or statistical data generated from previous transactions such as the rate given by $\frac{\#positive\_negotiations(LS_i)}{\#total\_negotiations(LS)_i}$, which is calculated by the number of successful negotiations time the number of negotiations. In the worst case, the serial strategy bounds the computational complexity to $O(N \cdot M)$ where N is the number of parallel negotiations and M is the average number of negotiation steps required to conclude a single process. In the best case, instead, it requires $O(M)$.

A *mixed strategy* can also be applied when LSs can be logically or physically grouped into *clusters*. Logical clusters might be formed, for example, according to ACE preferences (e.g., all LSs that provide location by using a common technology), contractual aspects (e.g., all LSs owned by a company that has contractual agreements with the ACE), or reputation (e.g., grouping LSs based on the number and feedback of previous experiences). Physical clusters of LSs can be defined based on the geographical location of LSs. The mixed strategy, then, consists in selecting one cluster at time (serial strategy), and negotiating with all LSs belonging to that cluster (parallel strategy).

4.2 Negotiation Strategies

Given a strategy to carry out the one-to-many communication, different negotiation strategies can be adopted by the ACE. In the following,

we describe different negotiation strategies to the parallel one-to-many communication.[2]

Basic Negotiation Protocol. The *basic negotiation protocol* is a lightweight negotiation protocol between an ACE and a LS that defines a simple interaction between the two parties to enhance the timely selection of a LS and the following location-based predicate evaluation. The basic negotiation process is composed by a *meta-evaluation* phase, followed by a *selection* phase and the actual *evaluation request* and *access control decision* phases. In short, with meta-evaluation we intend a communication process that results in a negotiation between the ACE and a LS of the SLA location-based attributes associated with a requested predicates. The goal of this phase is to negotiate service conditions in form of XML metadata, so no real predicate evaluation is performed. The selection phase, instead, is carried out by the ACE that selects the best LS according to the exchanged metadata and based on a given selection algorithm that minimizes a cost function Z. The following evaluation request phase consists in the actual location-based service provision. In particular, it is a request of a location-based predicate evaluation that the ACE submits to the selected LS. When the LS returns the value calculated for the requested predicate, the ACE is able to complete the access control decision phase. Figure 3 presents the detailed steps of the basic negotiation protocol, while Figure 4 shows a graphical example of message exchange during a negotiation.

The operations performed by an ACE during the basic negotiation protocol are grouped into four different phases.

Phase 1: Meta-Evaluation Request of Location-Based Predicate. The ACE gathers the list of available *LS*s offering services for the evaluation of location predicates. Such a list could be statically maintained by the ACE or dynamically created after an on-line lookup. It could be also further refined, for example, based on location evaluation technologies employed by LSs. We omit these details for brevity. Based on the list of available LSs, the ACE communicates with each of them requesting a specific predicate meta-evaluation. The ACE waits the answers from all LSs expressed as XML metadata and representing SLA-based offered conditions for the service provision. To avoid indeterminate waiting periods, timeouts are

[2] The application of negotiation protocols presented to the serial or mixed communication strategy is qualitatively identical to those discussed here. Therefore, we are presenting only the parallel strategy case, for brevity.

Protocol 1 *Basic Negotiation protocol*

Initiator: An ACE
Communication Counterparts: A set {LS_1,...,LS_n} of functionally equivalent LSs.

INITIATOR (ACE)
Phase 1: Meta-Evaluation Request of Location-Based Predicate
1.1 Gather available *LS*s for a specific location predicate evaluation.
search*(LS)*
1.2 Start the predicate meta-evaluation process for each *LS*.
meta_Eval(*predicate(*$attr_1$,...,$attr_n$*)*,LS_i), i=1,...,n
1.3 Wait for meta-evaluation from each LS_i
Phase 2: SLAs Evaluation
2.1 Receive the set of SLAs described in step M.3 and execute the LS selection algorithm.
2.2 Calculate the nominal cost as $C_{nominal_i} = \frac{price_i}{timeout_i}$, i=1,...,n
2.3 Calculate the virtual cost as $C_{virtual_i} = \frac{C_{nominal_i}}{confidence_i}$, i=1,...,n
2.4 Calculate the goal function Z as the minimum $C_{virtual_i}$, i=1,...,n
2.5 Select the LS_i that minimizes the goal function Z.
Phase 3: Evaluation Request
3.1 Request the location-based predicate evaluation to the *LS* selected at step 2.5.
predicate_Eval(*predicate(*$attr_1$,...,$attr_n$*)*)
3.2 Wait for the predicate evaluation.
predicate_Eval_Reply(*predicate(*$attr_1$,...,$attr_n$*), Eval*)
Phase 4: Access Control Decision
4.1 Receive the predicate evaluation (Step P.4) and perform the access decision operation.
4.2 Grant or deny User's access request.

LOCATION SERVICE (*LS_i*)
Meta Evaluation Request
M.1 Receive a meta-evaluation request (meta_eval message) and perform the meta-evaluation algorithm
M.2 Generate metadata containing, for each available location technology, three attributes: `price`, `timeout`, and `confidence`
M.3 Return the metadata to the ACE representing the SLA offered for service provision. (SLA_i)

Predicate Evaluation Request
P.1 Receive a request for predicate evaluation and start the evaluation process.
P.2 Select the location technology
P.3 Evaluate the predicate
P.4 Return the predicate evaluation

Fig. 3. Sequence of messages and operations in the basic negotiation protocol

set. If the timeout associated to a LS expires and the ACE had not received the answer, the corresponding meta-evaluation is discarded.

Phase 2: SLAs Evaluation. Upon reception of the metadata from LSs, the ACE computes a cost function on all SLA conditions expressed by

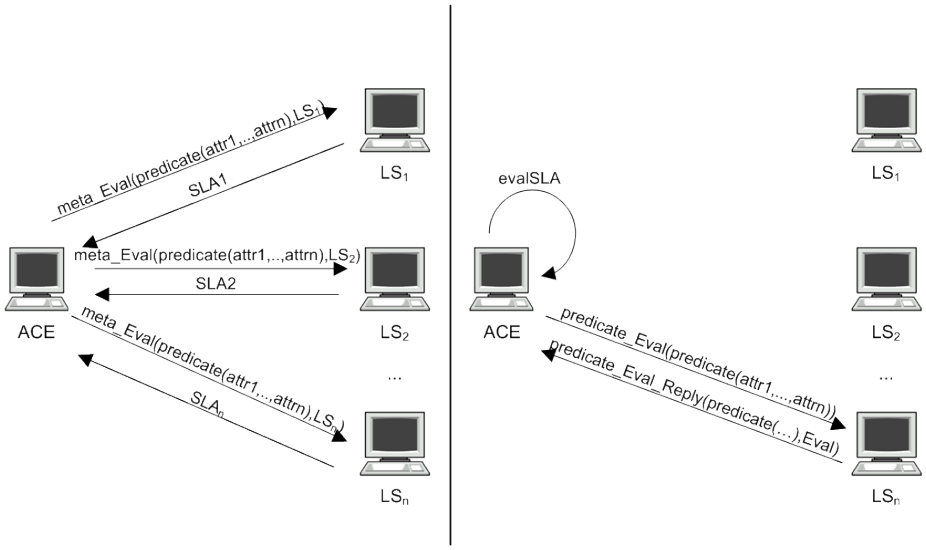

Fig. 4. Message Exchange Example in the Basic Negotiation Protocol

LSs. The cost of the location service is calculated firstly as a *nominal cost* $C_{nominal}$ that represents the price of the service for time unit. *Price* and *Timeout* are declared by a LS into the metadata and therefore the nominal cost is:

$$C_{nominal} = \frac{Price}{Timeout}$$

Then, the confidence value declared by each LS is considered. We recall that the lower is the confidence, the highest is the risk of having an unreliable location evaluation. Concerning the cost function, the idea is that such unreliability could be seen as a loss in terms of service quality and then evaluated as a cost, called C_{risk}.

Hence, the full cost function represents the cost perceived by the ACE, called $C_{virtual}$, logically composed by two terms: the actual price per unit time and the degradation of the service quality due to a confidence ratio lower than 1.

$$C_{virtual} = C_{nominal} + C_{risk} = \frac{C_{nominal}}{confidence}$$

In the basic negotiation protocol, the LS whose SLA conditions minimize the cost $C_{virtual}$ is selected by the ACE for predicate evaluation.

Phase 3: Evaluation Request. Given a selected LS, the ACE requests the actual location predicate evaluation.

Phase 4: Access Control Decision. Upon reception of the location-based predicate evaluation, the ACE can evaluate all applicable access control policies together with the location-based predicate value provided by the LS. Finally, an access decision is taken and the end-user gains or not the access to the requested resource protected by the ACE component.

In summary, the basic negotiation protocol assures that a suitable solution is always found. However, although the simple negotiation process implemented by the basic protocol can speed up the interaction between one ACE and a set of LSs, neither a cost/benefit analysis of the selected service nor possible minimum SLA requirements asked by the ACE have been taken into account. The basic negotiation protocol assumes that such issues have been disregarded in favor of simplicity and performances. The following enhanced negotiation protocol takes into account these issues.

4.3 Enhanced Negotiation Protocol

The *enhanced negotiation protocol* differs from the basic one because it takes into account that an ACE is likely to perform a cost/benefit analysis before accepting the SLA conditions of a LSs. To this purpose, we make use of metadata defined by the ACE that includes all the information to set the lowest acceptable SLA conditions and maximum costs. In particular, `maxUnitCost` is the cost in time unit that the ACE is willing to pay for the predicate evaluation; [`minConfidence`,`maxConfidence`] are the threshold values for the confidence to be negotiated with LSs. Over the maximum confidence level `maxConfidence`, the LS's SLA is acceptable, below the minimum confidence level `minConfidence`, the LS offer is just discarded and between the two thresholds the meta-evaluation phase is restarted a maximum `MaxTries` number of times to re-negotiate the SLA for possibly better conditions. If a better confidence is obtained from a LS so that the new level is greater than or equal to `maxConfidence`, the SLA is deemed acceptable. If, after `MaxTries` re-negotiation steps, an acceptable confidence level is not achieved, the LS's SLA is discarded. In this way, the ACE is sure that the negotiated SLA is suitable for its purposes and respects additional restrictions on the cost and benefit of the evaluation. In particular, only those SLAs that provide adequate quality level in term of confidence are compared with the `maxUnitCost` defined by the ACE. The idea is to first consider for a cost/benefit analysis only those LSs that provide for a confidence level greater than or equal to `maxConfidence`. If an acceptable solution (`maxUnitCost`$< Z{=}min$

```
Procedure evalSLA(maxUnitCost,minConfidence,maxConfidence,SLA[],MaxTries)
    /* UpConfSLA is an array containing the set of SLAs with a
    proposed confidence greater than or equal to maxConfidence */
    UpConfSLA[]=upSelect(SLA[],maxConfidence);
    /* IntermediateSLA is an array containing the set of SLAs
    with a confidence between minConfidence and maxConfidence */
    IntermediateSLA[]=intermediateSelect(SLA[],minConfidence,maxConfidence);
    Z = +∞;
    LS = "";
    if (count(UpConfSLA[])>0)
      For Each S in UpConfSLA[] do
        C_nominal = S.cost / S.timeout;
        C_virtual = C_nominal / S.confidence;
        if (C_virtual< Z)
          Z = C_virtual;
          LS = S.LS;
        endif
      endfor
      if (maxUnitCost< Z)
        restartNegotiationProcess(IntermediateSLA[],MaxTries-1);
      else
        startPredicateEvaluationProcess(LS);
      endif
    else
      restartNegotiationProcess(IntermediateSLA[],MaxTries-1);
    endif
```

Fig. 5. Enhanced Negotiation Protocol: Evaluation Algorithm

$\{C_{virtual_i}\}$ is found among these, the corresponding LS is selected. Otherwise the LSs that provides a confidence level between `minConfidence` and `maxConfidence` are considered for re-negotiation. At the end of this new meta-data evaluation phase, either an acceptable solution is found or the negotiation is aborted. Figure 5 presents the evaluation algorithm used in the enhanced negotiation protocol.

Enhanced Negotiation Protocol: Evaluation Algorithm. A major goal of the enhanced protocol is to include business-related parameters as the ones that more likely drives the selection of a location-based service provider in a one-to-many scenario.

As showed in Figure 5, the following two major steps have been introduced in the enhanced evaluation algorithm.

1. SLAs proposed by LSs are divided in three categories: *i) SLAs with a confidence level below the* `minConfidence` *value,* which are immediately discarded; *ii) SLAs with a confidence level greater than the* `maxConfidence` *value (function* upSelect*),* which are included in the array `UpConfSLA` that will be processed in the next steps of the algorithm; *iii) SLAs those with a confidence level between the two thresholds (function* intermediateSelect*),* which are included in the second array `IntermediateSLA` that is used only if a positive selection is not found by using the first array of SLAs.
2. The second phase represents the selection phase and takes as input the set of SLAs from the first array, if not empty. Every SLA is evaluated against ACE metadata. Firstly, as for the basic protocol, we calculate the cost Z, representing the minimum virtual cost offered by the LSs. Then, we compare Z with the maximum cost set by the ACE and only when the offered cost is below or equal to the `maxUnitCost` value, the related LS is selected for actual evaluation. On the contrary, if the first array is empty or if none of the SLAs from the first array produces a cost Z below or equal to `maxUnitCost` the negotiation process restarts with the set of LSs that provide a confidence between the minimum and the maximum confidences.

In summary, the enhanced protocol does not assure that a satisfiable solution will be found. However, if a solution is reached, it assures that ACE requirements are fulfilled by respecting constraints on cost and confidence. Although this protocol seems the best option for location-based service negotiation, an additional issue must be taken into account to avoid a potential problem that may arise. Such a problem is due to possible differences in the validity timeouts for a location-based predicate evaluation offered by LSs and the requirements in term of temporal validity that the ACE may need.

4.4 Timeout-based Negotiation Protocol

The *Timeout-based Negotiation Protocol* is motivated by what we have called the *unit cost problem* of the enhanced protocol, which can be easily described with an example. Suppose, for example, that one LS proposes a SLA containing a price of 100.000 euros and a timeout of 100.000 seconds and another LS has a SLA with a price of 100 euros with a timeout of 50 seconds. Following our algorithm, since the ACE always considers costs per time unit rather than absolute values, the first SLA will be considered the best. This, however, although correct by considering unit

costs, is likely not to be what the ACE wants to select, either because the total cost of the service is excessive or because such a long timeout validity is useless. Hence, considering only the unit cost $C_{nominal}$ is not sufficient, because additional metadata are needed to express the ACE preferences in terms of total cost and duration of the evaluation.

To this end two additional parameters, called `STOmin` and `STOmax`, are introduced. Together they represent the temporal interval of validity of an access control decision computed by the ACE. In particular, the two values have the following meaning: *i) before* `STOmin`*,* the ACE does not want to renew the access control decision; *ii) after* `STOmax`*,* the access control decision must be renewed; *iii)between* `STOmin` *and* `STOmax`*,* the access control decision must be considered safe and valid. The reason for this is that if the validity of the evaluation of location-based predicates is too short, the ACE is forced to request new evaluations too frequently, each time paying costs for the service and for the negotiation efforts. Otherwise, the ACE is concerned with costs for the location-based service. If the evaluation of location-based predicates has an excessively long timeout, much greater than the temporal validity required by the ACE for the access control decision, the cost of the location-based service is more than necessary (e.g., in the previously example, with proportions voluntarily exaggerated, should show this effect).

Concerns about the temporal validity of an access control decision based on location-based predicates could be found in many practical cases. Suppose, for example, that a service tracks the path of a felon. Location-based predicates will be requested and evaluated, for example, every ten seconds. In this case the ACE will be more willing to accept a proposal with low timeout and low price. On the other side, if a service requests an evaluation every ten minutes, ACE will be more interested to select a LS that provides an evaluation with a validity in the range of minutes, rather than seconds.

This approach, named *Timeout-based Negotiation Protocol*, relies on the evaluation algorithm depicted in Figure 6.

Time-based Negotiation Protocol: Evaluation Algorithm The timeout-based protocol differs from the other protocols because it takes into account requirements from the temporal validity of an access control decision based on location predicates. To this purpose, the additional metadata (`STOmin` and `STOmax`) are defined to set such a temporal validity, as required by an ACE.

```
Procedure evalSLA(maxUnitCost,minConfidence,maxConfidence,SLA[],MaxTries,
                  STOmin,STOmax)
  /* UpConfSLA is an array containing the set of SLAs with a
  proposed confidence higher than maxConfidence */
  UpConfSLA[]=upSelect(SLA[],maxConfidence);
  /* IntermediateSLA is an array containing the set of SLAs
  with a proposed confidence between minConfidence and maxConfidence */
  IntermediateSLA[]=intermediateSelect(SLA[],minConfidence,maxConfidence);
  Z = +∞;
  LS = "";
  if (count(UpConfSLA[])>0)
    For Each S in UpConfSLA[] do
      if (S.timeout > STOmax)
        C_nominal = S.cost / STOmax;
      else
        if (S.timeout < STOmin)
          discard(S);
          /*LS is willing to release a data with a validity lesser than the minimum required*/
          next(S);
        else
          C_nominal = S.cost / S.timeout;
        endif
      endif
      C_virtual = C_nominal / S.confidence;
      if (C_virtual< Z)
        Z = C_virtual;
        LS = S.LS;
      endif
    endfor
    if (maxUnitCost< Z)
      restartNegotiationProcess(IntermediateSLA[],MaxTries-1);
    else
      startPredicateEvaluationProcess(LS);
    endif
  else
    restartNegotiationProcess(IntermediateSLA[],MaxTries-1);
  endif
```

Fig. 6. Timeout-based Negotiation Protocol: Evaluation Algorithm

For what concerns the timeout-based evaluation algorithm, the difference with respect to the enhanced protocol algorithm is that the $C_{nominal}$ calculation has been modified to take in account the LS's SLA *timeout* and the ACE's `STOmin` and `STOmax`.

For instance, if the ACE needs to re-evaluate the location-based predicate every 10 seconds ($STOmax$ value) and a LS replies with a SLA's *timeout* of 60 seconds, the nominal cost calculated now is $\frac{SLA.cost}{STOmax}$, instead of $\frac{SLA.cost}{SLA.timeout}$ as in the enhanced protocol. This way the ACE considers that the validity of the evaluation will be 10 seconds instead of 60 seconds and calculate accordingly the real unit cost. On the other side, if a LS provides an evaluation with a timeout lower than `STOmin`, the solution is discarded because it is useless for the ACE evaluation that requires a stable evaluation for at least `STOmin` time units.

In summary, the timeout-based protocol does not assure that a satisfiable solution will be, eventually, found. However, it represents a solution aware of access control time constraints in addition to cost and confidence levels.

5 Related work

Technologies for integrating multiple sources of location information have been investigated for several years [16]. Today, most commercial location platforms include a gateway that mediates between location providers and location-based applications [19]. In those architectures, the location gateway obtains subscriber's location information from multiple sources and delivers them, possibly modified according to privacy requirements or to location-based applications. Regarding our work, this increased diffusion, accuracy, and reliability of location technologies have suggested novel ways to use location information within access control systems. To this end, the definition of LBAC models that includes the negotiation of QoS parameters based on SLA agreements is an emerging research issue that has not been yet fully addressed by access control researches [4]. Some early mobile networking protocols linked the notion of physical position of a terminal device with its capability of accessing network resources [2]. The main difference with our work is that mobile phones only are considered and no negotiation process is included. Widespread adoption of wireless local networks has been the subject of some recent studies focused on location-based information for the monitoring of user movements, based on Wireless-Lan [6] and 802.11 Networks [7]. A methodology for aggregating location data from multiple sources is described in [15]. This approach improves on the location tracking features by providing a solution for the composition of different location tracking techniques that increases the precision. In our work, instead, we focused on proto-

cols for the selection of the most convenient location technique according to different scenarios, SLA agreements and requirements.

Other researches are related to ours with regard to the underlying description of the architecture and operations of an access control server in a LBAC context. For instance, the need for a protocol-independent location technique has been explored by Nord et al. [18], which assume heterogeneous positioning sources like GPS, Bluetooth, and WaveLAN for designing location-aware application. In [18], a generic positioning protocol for interchanging position information between position sources and client applications is introduced and different techniques for merging position information are presented. Our techniques enriches this work by providing different negotiation protocols that involves several heterogeneous location techniques for the discovery of the most suitable location service. Another work, by Varshney [24], whose approach is close to ours, studies location-based information and their management in the area of mobile commerce applications and presents an integrated location management architecture to support composite location requirements. However, coordination among multiple wireless networks and location negotiation protocols for mobile commerce are not considered.

Few papers, instead, consider location information as a means for improving security. Sastry et al. [22] exploit location-based access control in sensor networks. Zhang and Parashar [27] propose a location-aware extension to Role-Based Access Control (RBAC) suitable for grid-based distributed applications. Other papers take into account time variant information for querying database containing location information [12, 14].

Other works took a different approach with respect to location information by considering them resources to be protected against unauthorized access. For instance, in [10], a mechanism to protect a user's location information by means of electronic certificates, delegation and trusted location-based services is described. The same problem is addressed in [9] by proposing a privacy-aware architecture for a global Location Service, which should permit users to define rules for the access to location information. With respect to our work, privacy requirements and management is a complementary issue that we have not addressed but can be seamlessly integrated with. In addition to privacy issues related to location information, there could be privacy issues related to metadata information too. For instance, location providers could require the location gateway to not publicly disclose SLA agreements, from which the competitors may benefit. We plan to address such privacy issues in future works.

Some recent papers present architectures designed for pervasive environments and architectures incorporating mobile data for security management. In [21], the development of a location-based service for the web is described. In [20] an architecture and a proof of concept implementation of a security infrastructure for mobile devices is presented. The work is focused on enforcing policies in pervasive environments. In [17], the evolution of applications using information that are bound to locations is discussed. A platform for such applications, named NEXUS and similar to the World Wide Web, is introduced for new applications and new information providers. This platform relies on distributed servers, whose federation provide an integrated view of location-based information to the applications.

Finally, the work by Zeimpekis et al. [26] attempts to identify the different indoor and outdoor positioning techniques that can be used for the provision of mobile and wireless applications and services. The authors also propose a novel taxonomy of these techniques based on the accuracy needed for different mobile location-based services. In a similar vein, Giaglis et al. [8] explore how a large number of indoor environments can benefit from location-based applications and services, describing the many related technological and application challenges.

6 Conclusion

In this paper, we presented a general architecture for evaluating LBAC conditions under the assumption that multiple functionally equivalent providers (LSs) are available. Our architecture relies on integrating multiple sources of location information via novel negotiation techniques involving one ACE and multiple LSs. We also provided a discussion about communication protocols, involved metadata (SLA) and different negotiation processes. We then analyzed the different evaluation algorithms, adopted by different negotiation protocols, that are in charge of comparing several SLAs agreed between an ACE and all the available LSs. The evaluation algorithm outcome represents the more suitable LS for ACE purposes. A description of the different algorithms and approaches are discussed.

7 Acknowledgments

This work was supported in part by the European Union within the PRIME Project in the FP6/IST Programme under contract IST-2002-507591 and by the Italian MIUR within the KIWI and MAPS projects.

References

1. S. Abiteboul, P. Buneman, and D. Suciu, editors. *Data on the Web: From Relations to Semistructured Data and XML*. Academic Press/Morgan Kaufmann, 1999.
2. I.F. Akyildiz and J.S.M. Ho, editors. *Dynamic mobile user location update for wireless PCS networks*. Wireless Networks, 1995.
3. M. Anisetti, V. Bellandi, E. Damiani, and S. Reale. Localize and tracking of mobile antenna in urban environment. In *Proc. of the International Symposium on Telecommunications*, Shiraz, Iran, September 2005.
4. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Supporting location-based conditions in access control policies. In *Proc. of the ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan, March 2006.
5. P. Bonatti, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. A component-based architecture for secure data publication. In *Proc. of the 17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, USA, December 2001.
6. D. Faria and D. Cheriton. No long-term secrets: Location-based security in overprovisioned wireless lans. In *Proc. of the Third ACM Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, USA, November 2004.
7. S. Garg, M. Kappes, and M. Mani. Wireless access server for quality of service and location based access control in 802.11 networks. In *Proc. of the Seventh IEEE Symposium on Computers and Communications (ISCC 2002)*, Taormina/Giardini Naxos, Italy, July 2002.
8. G.M. Giaglis, A. Pateli, K. Fouskas, P. Kourouthanassis, and A. Tsamakos. On the potential use of mobile positioning technologies in indoor environments. In *Proc. of the Fifteenth Bled Electronic Commerce Conference - e-Reality: Constructing the eEconomy*, Bled, Slovenia, June 2002.
9. C. Hauser and M. Kabatnik. Towards Privacy Support in a Global Location Service. In *Proc. of the IFIP Workshop on IP and ATM Traffic Management (WATM/EUNICE 2001)*, Paris, France, September 2001.
10. U. Hengartner and P. Steenkiste. Implementing access control to people location information. In *Proc. of the ACM Symposium on Access Control Models and Technologies 2004 (SACMAT 2004)*, IBM, Yorktown Heights, USA, 2004.
11. S. Horsmanheimo, H. Jormakka, and J. Lahteenmaki. Location-aided planning in mobile network trial results. *Wireless Personal Communications: An International Journal*, 30(2-4), September 2004.
12. H. Hu and D.L. Lee. Energy-efficient monitoring of spatial predicates over moving objects. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 28(3):19–26, 2005.
13. U. Leonhardt and J. Magee. Towards a general location service for mobile environments. In *Proc. of the 3rd Workshop on Services in Distributed and Networked Environments (SDNE '96)*, Macau, China, June 1996.
14. M.F. Mokbel and W.G. Aref. Gpac: generic and progressive processing of mobile queries over mobile data. In *Proc. of the 6th international conference on Mobile data management*, Ayia Napa, Cyprus, May 2005.
15. J. Myllymaki and S. Edlund. Location aggregation from multiple sources. In *Proc. of the 3rd IEEE Int.l Conf. on Mobile Data Management (MDM 02)*, Singapore, January 2002.

16. Jussi Myllymaki and Stefan Edlund. Location aggregation from multiple sources. In *Mobile Data Management*, pages 131–138, 2002.
17. D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang. A model-based, open architecture for mobile, spatially aware applications. In *Proc. of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD '01)*, Redondo Beach, CA, USA, July 2001.
18. J. Nord, K. Synnes, and P. Parnes. An architecture for location aware applications. In *Proc. of the 35th Hawaii Int.l Conference on System Sciences*, Hawaii, USA, January 2002.
19. Openwave. *Openwave Location Manager*, 2006. http://www.openwave.com/.
20. A. Patwardhan, V. Korolev, L. Kagal, and A. Joshi. Enforcing policies in pervasive environments. In *Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004): Networking and Services*, Cambridge, MA, USA, August 2004.
21. J. Ranchordas and A. Lenaghan. A flexible framework for using positioning technologies in location-based services. *IEEE Region 8, EUROCON2003*, II(6):95–98, September 2003.
22. N. Sastry, U. Shankar, and S. Wagner. Secure verification of location claims. In *Proc. of the ACM Workshop on Wireless Security (WiSe 2003)*, San Diego, CA, USA, September 2003.
23. *The data documentation initiative codebook DTD - version 1.0*, March 2000. http://www.icpsr.umich.edu/DDI/CODEBOOK.TXT.
24. U. Varshney. Location management for mobile commerce applications in wireless internet environment. *ACM Transactions on Internet Technology*, 3(3):236–255, August 2003.
25. W3C. *World Wide Web Consortium (W3C). XML Path Language (XPath) Version 1.0*, November 1999. http://www.w3.org/TR/xpath.
26. V. Zeimpekis, G.M. Giaglis, and G. Lekakos. A taxonomy of indoor and outdoor positioning techniques for mobile location services. *ACM SIGecom Exchanges*, 3(4):19–27, 2003.
27. G. Zhang and M. Parashar. Dynamic context-aware access control for grid applications. In *Proc. of the 4th International Workshop on Grid Computing (Grid 2003)*, Phoenix, Arizona, November 2003.