Negotiation Protocols for LBAC Systems

Abstract

Location-based Access Control (LBAC) systems are based on applications whose access control policies include location predicates. The enforcement of location predicates is performed by an Access Control Engine (ACE) and requires complex location services integrating sensing technologies able to gather users' physical location and components that process this information according to LBAC specifications. A specialized Location Middleware (LM) provides such location services. In this paper, we consider that the quality of such particular location services could be adjusted according to different Service Level Agreements (SLAs) expressed through the exchange of specific metadata. To this end, we address the issue of negotiating location service attributes between an ACE and a LM and introduce some protocols to carry out this coordination process. We start from a basic negotiation protocol that shows the core aspects of our proposal, to introduce an enhanced protocol that takes 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.

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 identities, roles, and affiliations. Writing and evaluating an access control policy based on requesters' location information requires to take into account 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 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. *Location-based Access Control* (LBAC) systems are designed to evaluate authorization policies based on requesters' location information in addition to conventional credentials [4, 9, 11, 20, 24]. Other works took a different approach with respect to location information by considering them resources to be protected against unauthorized access [6, 7].

In a LBAC system, specialized location components 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). A LBAC system cannot interact directly with this complex location infrastructure without the help of an appropriate interface and mediation functionality. Today, most commercial location platforms include a gateway that mediates between location providers and

location-based applications [16]. 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. 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]. As we will see, the main difference with our work is that mobile phones only are considered and no negotiation process is included. 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 [15, 22]. However, coordination among multiple wireless networks and location negotiation protocols for mobile commerce are not considered. Some recent papers present architectures designed for pervasive environments and architectures incorporating mobile data for security management [14, 17, 18].

In this paper, we focus on the particular architectural issue arising when an access control component asks a location verification to a location middleware. The LM is in turn faced to more location providers, serving the needed location information at different service quality due to technological differences or contractual agreements. Most advanced location-based platforms (e.g., OpenWave Location Manager [16]) 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. The availability of different location services makes possible to the location middleware to offer several options to the access control component, based on quality and cost. Negotiation protocols could be established between an access control component and a location middleware 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 middleware adopt to agree upon and set quality of services attributes and the corresponding service cost.

2 LBAC requirements

LBAC systems are subject to a set of specific functional and architectural requirements along with model requirements that we describe in the following.

2.1 Functional and architectural requirements

LBAC has larger architectural requirements than traditional access control systems, because user locations are gathered by means of specialized sensing technologies operated, in general, by companies of the telecommunication sector. Therefore, a LBAC architecture must integrate components logically tied with the applications that need access control enforcement and components providing location-services.

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.

To this end, one common assumption [5, 12, 13, 19], is to include a specialized middleware in charge of managing interactions between applications and location providers. Such a location middleware receives requests from LBAC components asking for location services. At this stage, we use the expression "location service" in a broad sense, meaning every possible request that demands the discovery of location information of an entity. It could be asked explicitly the location of a particular user or the answer to a location-based

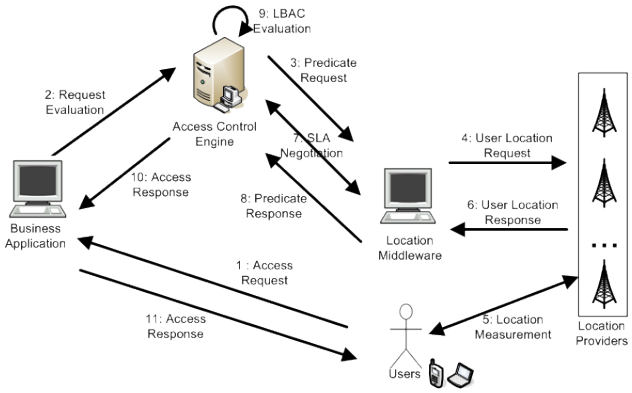

Figure 1: Reference LBAC Architecture

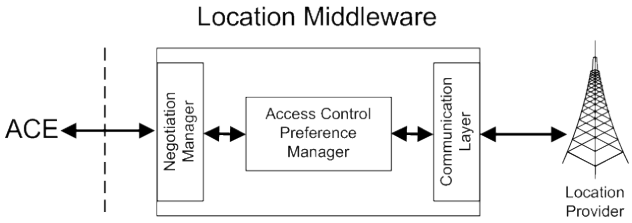

Figure 2: Location Middleware

condition (e.g., the number of users in a certain area, whether or not a particular user is in a given area or close by a given position). The location middleware collects users locations from a pool of location providers, elaborate location information, if needed, according to the request from LBAC components, and produces an answer. The logical components of an LBAC architecture are therefore the following.

- *User*: individuals to be located through their mobile terminals.
- *Business applications*: customer-oriented application that access resources protected by LBAC policies.
- *Access Control Engine (ACE)*: component that stores and enforces LBAC policies by requesting location services to the Location Middleware.
- *Location Providers (LPs)*: components using location sensing technologies to provide location data (we assume location providers are operated by mobile phone operators).
- *Location Middleware (LM)*: entity that interacts with different LPs, and provides location services and information to the ACE.

Figure 1 shows the layout of our reference LBAC architecture. The LM illustrated in Figure 2 is then logically realized by integrating the following components.

- *Communication Layer (CL)*: component responsible for managing the communication process with LPs. It hides low-level communication details to other components.
- *Negotiation Manager (NM)*: component that acts as an interface with ACE. NM provides negotiation functionalities and implements the negotiation protocols.

– *Access Control Preference Manager (ACPM)*: component responsible for SLA management and metadata specification.

Communication among the different components of our architecture is performed via request/response message exchanges. Here, we focus on the distinctive aspects of location-based services and location-based conditions evaluation and develop a set of possible communication protocols for the exchange of location-based attributes and data (see Section 3). The underlying assumption is that the ACE will *negotiate* the service conditions before requesting the actual location parameter evaluation. Note also that 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 LM with defined SLA parameters; or providing more elaborate patterns to achieve a complete *cost/benefit analysis* of the different service conditions.

2.2 Model requirements

Location-based conditions, which are conditions whose evaluation depends on location measurements performed by a Location Provider, are a core aspect of any LBAC system. To support location-based conditions in an authorization language we has to identify how these conditions can be expressed, how location information is queried, and what kind of response the location service returns.

Typically, location-based conditions are represented via *location-based predicates* that have been investigated since long by the wireless network research community [2]. In particular, we identify three main classes of predicates [4]: *movement-based* evaluates conditions on the mobility of the users (e.g., their velocity, acceleration, or the direction where they are headed); *position-based* evaluates 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* evaluates conditions relating multiple users or entities (e.g., the number of users within a given area). Two key issues related to these location-based predicates and that are specific to LBAC systems are:

– *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 LP 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 [8]. 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 [8].

Traditional location-based services [10] usually assume queries to the location service to be of the form of *range queries* 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 approach we consider queries to

[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.

be of a simpler (although largely equivalent) form; namely, we shall deal with *boolean queries* asking the LM 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*. The *confidence value* expresses the level of reliability that the LM is willing to guarantee to the assessment (`True` or `False`), according to accuracy, environmental conditions, granularity of the requested location, and measurement technique.

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 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 one location provider. The LM handles the ACE's request, interacts with LPs, and evaluate the predicate replying with a true or false value, together with the confidence it has in such a response. The LM 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 ACE to evaluate location-based conditions without taking into account technological details of the location measurement process. 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 LM, both the confidence and the timeout associated with location measures 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 LM 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 Negotiation protocols

Before discussing the different negotiation protocols that can be established between the ACE and the LM, we should consider how SLA conditions of location-based services can be represented. In our approach, the LM defines SLA conditions by means of a collection of *metadata*, one for each location technology offered (e.g., cellular phones, GPS, and WiFi cells). Those metadata are used by the ACE during the negotiation phase with the LM. The main benefit to ACE is that service conditions are presented by the single LM through metadata, hiding the technical peculiarities and the complexity of the LP infrastructure. 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 are to be intended as the "best offer" the LM proposes to the ACE. However, more evolved negotiation protocols, as described in the remainder of this section, could even re-negotiate the SLA conditions when the "best offer" is not satisfactory. For instance, it is likely that the price and quality of a service are directly related, that is, a service with high level of accuracy will

```
<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 >
```

Figure 3: An example of a fragment of XML metadata document defined by a LM

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 [21]).

For metadata browsing as well as for the evaluation of conditions that may determine whether or not a given LM'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 [23]. Figure 3 shows an example of XML-based metadata describing the technologies used by the LM and some SLA attributes. We are now ready to describe our negotiation protocols.

3.1 Basic negotiation protocol

The *basic negotiation protocol* is a lightweight negotiation protocol between the ACE and the LM that defines a simple interaction between the two parties to enhance the timely selection of a location service and the following location-based predicate evaluation. The basic negotiation process is composed by four different phases: a *meta-evaluation* phase, followed by a *selection* phase, the actual *evaluation request* phase , and the *access control decision* phase.

Phase 1: Meta-Evaluation Request of Location-Based Predicate. This phase results in a negotiation between the ACE and the LM of the SLA location-based attributes associated with a requested predicate. The goal is therefore to negotiate service conditions in form of XML metadata. To this purpose, the ACE gathers the list of available location services offered by an LM for the evaluation of location predicates. Based on this list, the ACE requests a specific predicate meta-evaluation. The ACE waits the answer from the LM expressed as XML metadata and representing SLA-based offered conditions for the service provision. To avoid indeterminate waiting periods, timeouts are set. If the timeout associated with a location service expires and the ACE had not received the answer, the corresponding meta-evaluation is discarded.

Phase 2: SLAs Evaluation. This phase is carried out by the ACE that selects the best location service among the ones offered by LM, according to the exchanged metadata and based on a given selection algorithm

that minimizes a cost function Z. Upon reception of the metadata from LM, the ACE therefore computes a cost function on all SLA conditions expressed by LM. 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 the LM into the metadata and therefore the nominal cost is:

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

Then, the confidence value declared by each location service 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 an 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 location service whose SLA conditions minimize the cost $C_{virtual}$ is selected by the ACE for predicate evaluation.

Phase 3: Evaluation Request. This 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 LM.

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 LM. 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 the ACE and the LM, 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 favour of simplicity and performances. The following enhanced negotiation protocol takes into account these issues.

3.2 Enhanced negotiation protocol

The *enhanced negotiation protocol* differs from the basic one because it takes into account that the ACE is likely to perform a cost/benefit analysis before accepting the SLA conditions of the LM. To this purpose, we make use of metadata defined by the ACE that include 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 LM. Over the maximum confidence level `maxConfidence`, the SLA is acceptable, below the minimum confidence level `minConfidence`, the SLA 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 after `MaxTries` re-negotiation steps, an acceptable confidence level is not achieved, the 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 location services that provide for a confidence level greater than or equal to `maxConfidence`. If an acceptable solution (`maxUnitCost`$< Z = min\ \{C_{virtual_i}\}$) is found among these, the corresponding service is selected. Otherwise

```
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
```

Figure 4: Enhanced Negotiation Protocol

the SLAs that provide 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 4 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 4, the following two major steps have been introduced in the enhanced evaluation algorithm.

1. SLAs proposed by LM 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 with a confidence level between the two thresholds (function* intermediateSelect*),* which are included in the second array `IntermediateSLA` 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 array `UpConfSLA`, if not empty. Each SLA is evaluated against ACE metadata. As for the basic protocol, we calculate the cost Z, representing the minimum virtual cost asked for the corresponding location service (LS in Figure 4). 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 service is selected for actual evaluation. If array `UpConfSLA` is empty or if none of the SLAs produces a cost Z below or equal to `maxUnitCost`, the negotiation process restarts with the set of location services 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 and the requirements in term of temporal validity that the ACE may need.

3.3 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 the LM proposes a SLA containing a price of 100.000 euros and a timeout of 100.000 seconds and another 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 location service 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 5.

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.

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 into account the 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 the LM 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 SLA 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.

```
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
```

Figure 5: Timeout-based Negotiation Protocol

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.

4 Conclusions

We presented a general architecture for evaluating LBAC conditions under the assumption that a specialized location middleware hides the complexity of dealing with multiple functionally equivalent providers. Our architecture relies on novel negotiation techniques between an ACE and a LM to support the provision of location services. We also provided a discussion about metadata (SLA) and different negotiation processes. We then analyzed the different evaluation algorithms, adopted by different negotiation protocols, which are in charge of comparing several SLAs agreed between an ACE and a LM. The evaluation algorithm outcome represents the more suitable location service for ACE purposes. A description of the different algorithms and approaches are discussed.

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] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04)*, Orlando, FL, USA, March 2004.

[6] 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.

[7] 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)*, Yorktown Heights, New York, USA, June 2004.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] H. Naguib, G. Coulouris, and S. Mitchell. Middleware support for context-aware multimedia applications. In *Proc. of the IFIP TC6 / WG6.1 Third International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, pages 9–22, Deventer, The Netherlands, 2001.

[13] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li. QoS-aware middleware for ubiquitous and heterogeneous environments. *IEEE Communications Magazine*, pages 140–148, November 2001.

[14] 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.

[15] 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.

[16] Openwave. *Openwave Location Manager*, 2006. http://www.openwave.com/.

[17] 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.

[18] 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.

[19] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. H. Campbell, and M. D. Mickunas. Middlewhere: A middleware for location awareness in ubiquitous computing applications. In *ACM/IFIP/USENIX 5th International Middleware Conference (Middleware 2004)*, Toronto, Ontario, Canada, October 2004.

[20] 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.

[21] *The data documentation initiative codebook DTD - version 1.0*, March 2000. http://www.icpsr.umich.edu/DDI/CODEBOOK.TXT.

[22] U. Varshney. Location management for mobile commerce applications in wireless internet environment. *ACM Transactions on Internet Technology*, 3(3):236–255, August 2003.

[23] W3C. *World Wide Web Consortium (W3C). XML Path Language (XPath) Version 1.0*, November 1999. http://www.w3.org/TR/xpath.

[24] 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.