Secure Content Distribution for Digital Libraries *

Mariemma I. Yagüe, Antonio Maña, Javier López, Ernesto Pimentel, José M. Troya

Computer Science Department
University of Málaga. Spain.
{yague, amg, jlm, ernesto, troya}@lcc.uma.es

Abstract. Security is a very relevant aspect in the implementation of most digital libraries. Two important security issues in these systems are distributed access control and secure content distribution. This paper presents a system that provides these two features for digital libraries. The system is based on the combination of the use of an external authorization infrastructure, a software protection mechanism and a modular language to specify how access to the contents is controlled. The extensive use of semantic information makes possible the integration of such a complex set of subsystems.

1 Introduction

Digital Libraries (DLs) integrate a variety of information technologies that provide opportunities to assemble, organize and access large volumes of information from multiple repositories. Regarding access control, DLs usually contain objects with heterogeneous security requirements. The problems of controlling access to such objects present many similarities with those found in other distributed systems. However, some of the problems are specific, or more relevant, for DLs. Some of those problems are: (i) libraries are offered to previously unknown users; (ii) payment or other activities (like the execution of copyright agreements) must be bound to the access to the objects; (iii) the originator or owner of the object must retain control over it even after it is accessed by users; (iv) a high degree of flexibility is required because of the heterogeneous nature of the objects; (v) the ability to change the access control parameters dynamically and transparently is also essential in most DLs; and finally (vi) due to the large amount of objects, it is important to establish access conditions automatically, based on information about objects.

This paper presents XSCD-DL (XML-based Secure Content Distribution for Digital Libraries), a particular application of the XSCD infrastructure [1]. XSCD-DL provides distributed access control management and enforcement, as well as secure content distribution in digital libraries. To achieve our goals we combine an external authorization infrastructure, a modular language called *Semantic Policy Language* (SPL) to specify how access to the contents is controlled, and a software protection mechanism (*SmartProt*). The extensive use of semantic information makes possible the integration of such a complex set of subsystems into XSCD-DL.

* Work partially supported by Spanish Ministerio de Ciencia y Tecnología. Project TIC2002-04500-C02-02

The rest of the paper is organized as follows. Section 2 summarizes some related work. Section 3 describes the fundamentals and global structure of XSCD-DL. Finally, section 4 summarizes the conclusions and presents ongoing and future work.

2 Related work

Different projects, such as the ADEPT Digital Library [2] or the Alexandria Digital Library [3], have focused on handling various structural and semantic issues, while providing users with a coherent view of a massive amount of information. The use of metadata is essential in these systems, but the application to the security issues is not considered. The Stanford Digital Library Project [4] covers most of the different issues involved in this field. One important outcome is the FIRM architecture [5] that proposes the separation of objects that implement control from objects that are controlled, enhancing the flexibility of the system.

Regarding access control, several proposals have been introduced for distributed heterogeneous resources from multiple sources [6][7]. Unfortunately, these proposals do not address the specific problems of distributed access control in DLs. Traditional access control schemes such as *mandatory access control* (MAC), *discretionary access control* (DAC) or even *role based access control* (RBAC) are not appropriate for complex distributed systems such as digital libraries. It has been shown that an approach based on attribute certificates represents a more general solution that fits more naturally in these scenarios [8]. In fact, MAC, DAC and RBAC schemes can be specified using the attribute-based approach.

Because of the specific requirements imposed to the access control systems of digital libraries, the most widespread architecture is that of a federated set of sources, each one with a centralized access control enforcement point. This architecture has important drawbacks such as the reduced system performance produced because the centralized access control enforcement point becomes a bottleneck for request handling. Other drawbacks are that (a) the control point represents a weak spot for security attacks and fault tolerance, (b) it does not facilitate the deployment of owner retained control mechanisms, and (c) it usually enforces homogeneous access control schemes that do not fit naturally in heterogeneous user groups and organizations.

On the other hand, distributed access control solutions proposed so far do not provide the flexibility and manageability required. An interesting approach based on the concept of mobile policies [9] has been proposed to solve some of the limitations of RBAC schemes [10]. This is a limited improvement because of the requirement to execute the access control policies in trusted computers (object servers in this case). Furthermore, when access to an object is granted, this object has to be sent to the client computer where control over it is lost. Finally, because object and policy are compiled in a package, any single change in the policy that controls an object requires that the object-policy package is recompiled and distributed to all trusted servers.

Several XML based languages have been developed for access control, digital rights management, authentication and authorization. These languages do not support powerful features such as policy modularisation, parameterisation and composition. Furthermore, some of their features are not necessary in DLs [11]. Two relevant

proposals are the Author-X system [12] and the FASTER project [13][14], which propose two similar systems for access control to XML documents. Both systems define hierarchic access control schemes based on the structure of the document. The FASTER system does not support any content protection mechanism. FASTER access control is based on user groups and physical locations following the well-known technique of defining a subject hierarchy. In scenarios such as digital libraries, this approach is not adequate because a fixed hierarchy can not represent the security requirements of all the different contents, users and access criteria. On the other hand, content protection in Author-X is founded on the concept of (passive) secure container, which introduces disadvantages from the point of view of security and access control system management. Author-X is based on credentials that are issued by the access control administrator. Therefore, in practice, each credential will be useful only for a single source, limiting interoperability. A direct consequence of this approach is that users must subscribe to sources before they can access their contents.

3 XSCD-DL Fundamentals and Global Structure

New solutions are required to address the need of some of the new distributed applications such as DLs, web services or grid computing. Some of the problems found on existing access control systems are:

- The security administration in current DLs is very complex and error prone. A flexible and powerful policy language that incorporates features to manage the complexity inherent to DL environments represents a step towards the solution of this problem. Automated management tools also serve this objective.
- The explicit static allocation of policies to objects is not adequate in highly dynamic environments with heterogeneous contents, where new resources are often incorporated to the system and security requirements change frequently. In order to solve this problem, dynamic allocation of policies to resources and the definition of an access policy language designed to ease the management of the system must be considered.
- The access control criteria are usually defined either explicitly or on the basis of the structure of the contents. These approaches present severe drawbacks as we will show later.
- In new environments we deal with a large number of (possibly anonymous) users. Existing schemes, based on user identity, need to collect some profile information in advance. Therefore, a registration phase is used to collect information about the user and issue the corresponding local credentials. The semantic integration of an external PMI represents a step towards the solution of the interoperability of different DLs with heterogeneous access control systems.
- The access policy depends on the administrator of the server where the object is stored. In DL environments, it would be desirable that originators of the contents are able to define the applicable access policy to apply in a dynamic and transparent way, regardless of the object storage location.
- Finally, no secure content distribution mechanisms are used.

Because our system includes different elements that are combined to solve some of these problems, in the following subsections we will focus on each of the problems.

3.1 Modular language for flexible security administration

XML is widely considered a suitable candidate for a policy specification language [7]. Existing XML-based languages for access control, authorization and digital rights management are not based on a modular approach and do not provide some important features such as policy composition and parameterisation. These ones play an essential role in the flexibility of management of the access control system [11].

The definition of access control policies is a complex and error prone activity that presents many similarities with computer programming. Therefore, we have included some of the mechanisms used to reduce the complexity in programming languages such as modularity, parameterisation and abstraction. In order to provide the simplicity and flexibility required in complex systems such as digital libraries, our solution is based on the modular definition of policies. Modularity in our solution implies: (a) the separation of specification in three parts; that is, access control criteria, allocation of policies to resources and semantic information (properties about resources and context); (b) the abstraction of access control components; (c) the ability to reuse these access control components; and (d) the reduction of the complexity of management due to previous properties. Moreover, the use of semantic information about the context allows the administrator to include contextual considerations in a transparent manner, also helping the (semantic) validation task.

Usual components of access policies include the target resource, the conditions under which access is granted/denied and, sometimes, access restrictions. Opposed to other languages, SPL policy specifications do not include references to the target object. Instead, a separate specification called *Policy Applicability Specification* (PAS) is used to relate policies to objects dynamically when a request is received. Both SPL policies and PAS use semantic information about resources included in *Secured Resource Representation* (SRRs) and other contextual information documents, which is an original contribution. SPL policies and PAS can be parameterised allowing the definition of flexible and general policies and reducing the number of different policies to manage. Parameters, which can refer to complex XML elements, are instantiated dynamically from semantic and contextual information. Finally, policies can be composed importing components of other policies without ambiguity. This compositional approach allows us to define the abstract meaning of the elements of the policies, providing a mechanism to achieve abstraction, which also helps in reducing the complexity of management. Tools to graphically manage the relations among policies and with other components are also essential for a simple and flexible management.

The schema for SPL specifications is represented as a set of XML-Schema [15] templates that facilitate the creation of these specifications, allowing their automatic syntactic validation. Figure 1 shows the structure of the SPL language.

SPL policies can include locally defined components as well as imported elements. The ability to import elements enables the modular composition of policies based on the XPath standard [16]. An SPL Policy is composed of a set of *access_Rule*

elements, each one defining a particular combination of attribute certificates required to gain access, associated with an optional set of actions (such as *Notify_To*, *Payment* and *Online_Permission*) to be performed before access is granted. In this way provisional authorization is enabled in SPL.

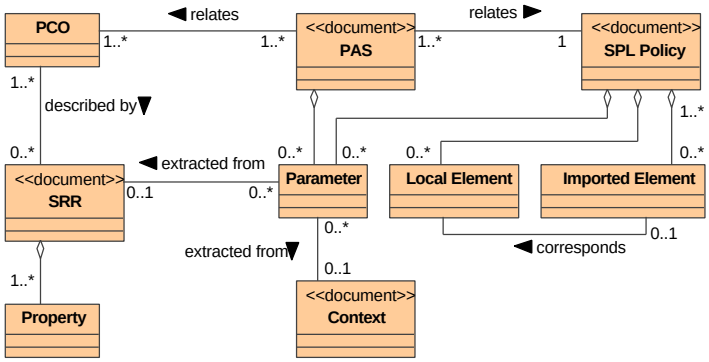

Fig. 1. Conceptual model of the SPL Language

The *Policy Applicability Specification* provides an expressive way to relate policies to resources, either explicitly or based on the metadata about the objects (e.g. type of content, owner, price, etc.). PAS documents include three main elements: policy, objects and instantiation. The policy element indicates which policy is applicable to the specified objects. Objects are defined by their location and conditions to be fulfilled by the semantics of these objects (SRRs). Optionally, operation elements can be used to define which operations of the target object are controlled by the declared policy, allowing a finer grained access control. In case no operation element is included, the policy is applicable to all of the object operations. The instantiation element describes the mechanism to instantiate parameters in the policies. Figure 3 shows an example of applicability rules for SPL policies to objects.
The *Secured Resource Representation* is a simple and powerful mechanism to describe properties about resources. Properties described in SRRs are used for the instantiation of policies and PAS, and to locate the applicable policies. An example of an SRR is also included in figure 3. The SRR is designed specifically for the process of dynamic allocation of policies to resources. Dynamic allocation is a very flexible and useful mechanism that solves the problem of associating policies to newly created objects. The use of dynamic policy allocation needs a rich set of metadata about the resources. This semantic meta-model is used to locate the right policy for each resource, based on its relevant properties.

3.2 Local credentials versus external PMI

Most of current access control schemes base their authorization approaches on locally issued credentials that are linked to user identities. This type of credentials present many drawbacks. Among them we highlight: (a) they are not interoperable; (b) the

same credentials are issued many times for each user, what introduces management and inconsistency problems; (c) credentials are issued by the site administrator; however, in most cases, the administrator does not have enough information or resources to establish trustworthy credentials; and (d) they are tight to user identity. In practice, it is frequent that the identity of the user is not relevant for the access decision. Sometimes it is even desirable that the identity is not considered or revealed. Furthermore, in systems based on identity, the lack of a global authentication infrastructure (a global PKI) forces the use of local authentication schemes. In these cases, subscription is required and users have to authenticate themselves to every accessed source. To solve the aforementioned problems, single-sign-on mechanisms have been used in last years. These mechanisms are based on federation of sources that represent a limited improvement because credentials remain local (not to a site, but to a set of them). Moreover, all federated sources must agree on a homogeneous access control scheme.

On the other hand, digital certificates can securely convey authorizations or credentials. Attribute certificates bind attributes to keys providing means for the deployment of scalable access control systems in the scenarios that we have depicted. These authorizations are interoperable and represent a general and trustworthy solution that can be shared by different systems. Taking into account security, scalability and interoperability, the separation of the certification of attributes and access control management responsibilities is widely accepted as a scalable and flexible solution. In this case, the access control system needs to be complemented by an external component: the *Privilege Management Infrastructure* (PMI)[17]. The main entities of a PMI, known as *Source of Authorizations* (SOAs), issue attribute certificates. Usually, each SOA certifies a small number of semantically related attributes. This scheme scales well in the number of users and also in the number of different factors (attributes) used by the access control system.

With this approach, each access control system will select which SOAs to trust and which combination of attributes to use. Because they are separate systems, a mechanism to establish the trust between the access control and the PMI is required. Metadata described about the PMI, represented as *Source Of Authorization Description* (SOAD) documents, is the key to achieve the necessary interoperability. SOADs are RDF [18] documents protected by digital signatures [19] that express the different attributes certified by each SOA, including their names, descriptions and relations. These descriptions state a series of facts about the environment of the system using metadata to represent the semantics of the different attributes that are certified by the SOA, including names, descriptions and relations among attributes.

In our scheme SOADs represent the semantic description mechanism that establishes trust between the PMI and the access control system. The semantic information about the certificates issued by each SOA is also used to assist the security administrators in the creation of access control policies. Additionally this semantic information allows the detection of possible inconsistencies in our SPL policies, during the semantic validation process.

3.3 Structure versus semantics as the basis of the access control scheme

Usually, conditions and restrictions of access depend on the semantic properties of the target object that are neglected in structure-based approaches. Because the security requirements for each object in the DL depend on different properties about the object, an approach based on semantic descriptions of the contents is much more flexible and natural. Moreover, it is easy to incorporate structure-based requirements in the semantic model. Finally, the structure is much more volatile than the semantics. In particular, when the contents are XML documents, the structure is dynamic and flexible, introducing serious problems in the security administration. Some works have appeared to automate the process of translating the authorizations for the transformed XML document, although the proposed solutions are very limited [20].

In order to illustrate the advantages of content-semantics over content-structure as the basis for the policy specification, lets consider the case of a digital library of proceedings of scientific conferences. Usually, titles, authors and abstracts of all papers are public while full papers have different access control requirements depending on the publisher and the type of paper. Figure 2 shows how the ideal structuring of the contents for cataloguing and searching purposes does not match the one for security requirements. Moreover, two different possibilities appear in the latter case, illustrating that the structure-based approach is not adequate for the case of digital libraries, where contents have heterogeneous security requirements. The incompatibility between the structure required for the application domain and the ones that match the security requirements confirms that structure-based approaches are not able to represent these situations in a natural way. Furthermore, if we were to use the structuring in figure 2a then multiple authorizations of opposite sign (grant/deny) would be required to match the security requirements of each piece of content.

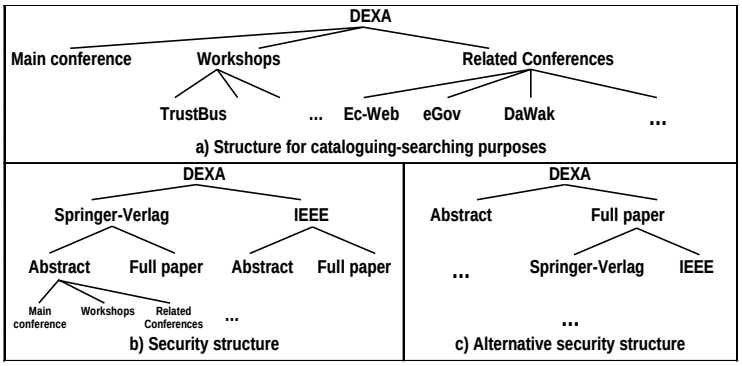

Fig. 2. Different structuring of the contents of a digital library

Another drawback of structure-based approaches is that the number of policies becomes very large. In fact, these approaches usually imply the definition of several policies for each resource. Positive and negative authorizations are used in these cases to facilitate the definition of simple policies and to reduce the number of policies. The price to pay is the introduction of ambiguities, which in turn requires the definition of conflict resolution rules. Consequently, the administration of the system becomes

complex and difficult to understand increasing the chance of producing incorrect policies. The semantic-based and modular approach adopted in XSCD-DL facilitates the definition and management of policies avoiding the use of positive and negative authorizations. Tools provided to support the policy specification, composition and validation also serve this objective.

The semantic-based approach adopted in XSCD-DL is illustrated in figure 3. It shows how the use of semantic information, the modularisation, parameterisation and dynamic instantiation of policies results in simple and flexible policies reducing the complexity of management of the system. The specifications also demonstrate how the system can manage dynamic changes in a transparent way. No modifications are necessary in the policies when requirements or properties of resources change.

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:PAS xmlns:spl="http://www.lcc.uma.es/ICADL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/ICADL pas.xsd">
    <spl:parameter>proceeding_Publisher</spl:parameter>
    <!--  PAS parameters are instantiated directly from the target SRR,
therefore no instantiation tag is required -->
    <spl:policy>FullPaper.xml</spl:policy>
    <spl:object>
      <spl:object_Location>
          http://www.dexa.org/2002/ecweb/
      </spl:object_Location>
      <spl:conditions>
        <spl:condition>
          <spl:property_Name>type</spl:property_Name>
          <spl:property_Value>full_Paper</spl:property_Value>
        </spl:condition>
        <spl:condition>
          <spl:property_Name>Publisher</spl:property_Name>
          <spl:property_Value>*paper_Publisher</spl:property_Value>
        </spl:condition>
      </spl:conditions>
    </spl:object>
    <spl:instantation>
      <spl:formal_Parameter>Publisher</spl:formal_Parameter>
      <spl:actual_Parameter
    path="//Publisher_Info[@name='*proceeding_Publisher']/parent::">
      Publishers_Context_Info.xml</spl:actual_Parameter>
    </spl:instantation>
<!--  Publisher_Context_Info contains details about each Publisher -->
</spl:PAS>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:policy xmlns:spl="http://www.lcc.uma.es/ICADL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/ICADL Policy.xsd">
<spl:parameter>Publisher</spl:parameter>
<spl:access_Rules>
    <spl:access_Rule>
        <spl:attribute_Set>
            <spl:attribute>
        <spl:attribute_Name>Registered</spl:attribute_Name>
        spl:attribute_Value>*Publisher[@name]</spl:attribute_Value>
        <spl:SOA_ID>*Publisher[@SOA]</spl:SOA_ID>
            </spl:attribute>
        </spl:attribute_Set>
    </spl:access_Rule>
  </spl:access_Rules>
</spl:policy>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:SRR xmlns:spl="http://www.lcc.uma.es/ICADL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/ICADL SRR.xsd"
resource="http://www.dexa.org/2002/ecweb/Lopez_acdacs.pdf">
<!-- This SRR states access control about the paper Lopez_acdacs.pdf  -->
  <spl:property>
      <spl:property_Name>type</spl:property_Name>
      <spl:property_Value>full_Paper</spl:property_Value>
  </spl:property>
  <spl:property>
      <spl:property_Name>proceeding_Publisher</spl:property_Name>
      <spl:property_Value>Springer_Verlag</spl:property_Value>
  </spl:property>
  <!-- e-mail of the responsible -->
  <spl:property>
      <spl:property_Name>responsible</spl:property_Name>
      <spl:property_Value>manager@ecweb.com</spl:property_Value>
  </spl:property>
</spl:SRR>
```

Fig. 3. Example Policy, its corresponding PAS and the SRR for a ‘full paper’

3.4 Content protection

Two important issues arise when considering content protection in digital libraries: the content distribution mechanism itself and the owner-retained-control issue. The first one must ensure that contents are protected so that only the intended recipients can access them. In the case of digital libraries it also entails other requirements such as the need to bind the execution of digital rights agreements, payment or other actions to the access to the contents. This is known as provisional authorization or

provision-based access control (PBAC) [21]. The second one deals with enabling that owners of the contents retain control over them even when contents are stored in external untrusted servers.

Our solution to the previous problems is based on the use of secure active containers. A *secure active container* [22] is a piece of protected mobile software that conveys the contents and forces the user to fulfil the applicable policy before access is granted. By "protected software" we mean that it is neither possible to discover nor to alter the function that the software performs and it is also impossible to impersonate the software. In our scheme, this is achieved using a variant of the *SmartProt* system [1]. *SmartProt* partitions the software into functions that are executed by two collaborating processors. One of those processors is a trusted computing device that enforces the correct execution of the functions and avoids that these functions are identified or reverse engineered. We are currently using smart cards for this purpose although other alternatives are possible. Our secure active containers are implemented as Java™ applets that we call *Protected Content Objects* (PCOs). They include the contents to be accessed (which are encrypted), the access control enforcement mechanism, and a cryptographic link to the *Mobile Policy* (MP) required to gain access to the contents. We extend the concept of mobile policy described in [9] by allowing their execution in untrusted systems. Moreover, in our solution policies are bound to the object but not integrated with. This modification makes possible that policies are dynamically changed in a transparent manner. The definition of the MP structure allows a high degree of flexibility.

The PCO generation process is independent of the customer card and will be performed just once for each piece of content. PCOs can be distributed and copied freely. One important constraint to the free distribution of protected contents in our system is that originators of those contents must be able to dynamically change the applicable access control policy regardless of the storage location of the PCO. In order to fulfil this requirement, MP and PCO must be separated. In this way, the MP is retrieved from the originator DL during the execution of the PCO. Requesting the MP at access time from the originator slightly reduces the performance of the system but, in return, it allows a high degree of flexibility and gives the originator more control over the application of the policies. To improve the efficiency and flexibility we have included validity constraints in MPs that can be used to control the need for an online access to the originator server. As a result, originators can define certain validity constraints for each MP (based on number of accesses, time, etc. depending on the smart card features). Hence, MPs can be cached by clients and used directly while they are still valid. The generation of MPs is a fast process while the generation of PCOs is slower. Furthermore, PCOs are much more stable than policies. Finally, opposed to PCOs, each MP is specific for a smart card. As each PCO has its own key, we can manage them individually, which is not possible in other software protection proposals where all applications are protected using the same key.

When the client requests some data object from a DL server, it receives the PCO containing it. Before the PCO can execute the protected sections of its code it has to retrieve the corresponding MP sending a request containing the certificate of the public key of the client smart card. In case the server from where the PCO was retrieved is the originator of the PCO, it produces the MP for that PCO. Otherwise the server just forwards this request to the PCO originator. When the MP is received by

the client smart card, it is decrypted, verified and stored inside the card until it expires or the user explicitly decides to extract it. Once the MP is correctly installed in the card the protected sections of the PCO can be executed, which requires the cooperation of the card containing the MP. The protected sections of the software do not reside in the cards. Instead, during the execution of the PCO, these sections are transmitted dynamically as necessary to the card, where they are decrypted using the installed MP and executed. When finished, the card may send back some results. Some other partial results will be kept in the card in order to obtain a better protection against function analysis and other attacks.

3.5 Global Structure

A general overview of the main components of the system and their relation is depicted in figure 4. The first component is the *SmartProt* protection system. This component transforms unprotected content objects in the originator DL server into PCOs as described in section 3.4.

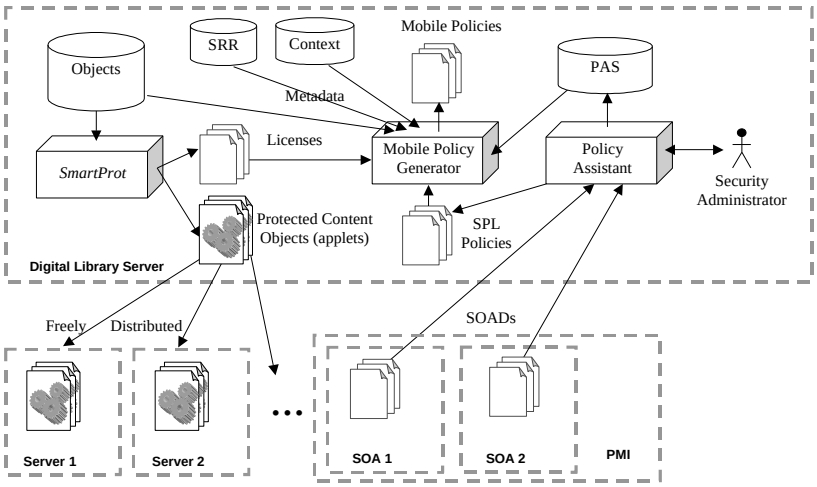

Fig. 4. XSCD-DL Infrastructure for Digital Libraries

The second component, called *Policy Assistant*, is designed to help security administrators in the specification, management and validation of access control policies. This component uses the SOADs as a basis for the specification of SPL policies and PAS. It is also responsible for the automated validation of policies at different levels. SPL policies are validated syntactically using XML-Schema. Semantic validation is made possible by the use of a specific *Semantic Policy Validator* (included in the Policy Assistant) that uses the DOM API to parse the SPL specification validating it. Finally, as an extension of the semantic validation, policies can also be validated in the context where they will be applied. Policy context validation uses the semantic information contained in the SOADs for the detection of

possible inconsistencies in the SPL policies. Therefore, the Policy Assistant integrates all the tools to facilitate the administration of the access control system.

The third component, called *Mobile Policy Generator*, attends requests from end users producing MPs dynamically. To determine the set of applicable policies for a given PCO, the generator uses different sources of metadata. After receiving a request the Mobile Policy Generator analyses the semantic metadata available for the target PCO, which is contained in SRRs, finds the appropriate PAS and retrieves the necessary SOADs. Using this information, the Mobile Policy Generator is able to find the applicable SPL policies. These policies are then analysed and instantiated using the metadata about the resource (SRRs) and the context. Finally, these policies are combined. The combination of policies helps reducing the complexity of administration while enabling more expressive and flexible policies to be considered in the access decision.

4 Conclusions and future work

The system presented in this paper, XSCD-DL, combines an external PMI, a modular language (*Semantic Policy Language*, SPL) and a software protection mechanism (*SmartProt*) for the specification of the access control policies in order to provide distributed access control management and enforcement and secure content distribution in digital libraries. XSCD-DL allows policies to be dynamically changed by the owner or originator of the resource in a transparent manner.

An important feature is the extensive use of XML metadata technologies to facilitate the security administration in digital libraries and other complex environments. It also enables interesting functionalities of the system such as the contextual validation of policies. In our system, XML-metadata technologies are applied at different levels to express the semantic information. On one hand, metadata are used for the creation and semantic and contextual validation of access control policies. Likewise, metadata about the objects included on the DL enables dynamic policy allocation and parameter instantiation. On the other hand, metadata is an essential tool for the integration of the external PMI in the access control system.

To summarize, XSCD-DL represents a solution applicable in different distributed scenarios, is flexible, solves the originator-retained-control problem, can be applied regardless of the attribute certification scheme, implements distributed access control management and enforcement mechanisms, incorporates secure content distribution and allows the dynamic modification of policies transparently and efficiently.

A prototype of this system has been implemented for a Digital Library scenario. In such environment, PCOs are implemented using Java applets. The *e-gate Cyberflex*™ USB Java smart cards are used as secure coprocessors. The high capacity and the transfer speed of these cards makes possible that the performance of the PCO is very good. A set of techniques, such as temporary authorizations, is used to improve the performance. We are currently working on the formalization of SPL specifications. A fair payment mechanism has been designed to complement the system and is currently being implemented.

References

1. López, J., Maña, A., Pimentel, E., Troya, J.M., Yagüe, M.I. *An Infrastructure for Secure Content Distribution*. To appear in Proceedings of ICICS'02. Springer-Verlag. 2002.
2. Janée, G., Frew, J. *The ADEPT Digital Library Architecture*. Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '02). 2002.
3. Coleman, A. *Metadata Rules the World: What Else Must We Know About Metadata?*. http://www.alexandria.ucsb.edu/~acoleman/mrworld.html
4. Baldonado, M., Chang, K., Gravano, L.P, Paepcke, A. *The Standford Digital Library Metadata Architecture*. International Journal of Digital Libraries, 1(2), February 1997.
5. Ketchpel, S., Garcia-Molina, H., Paepcke, A. *Shopping Models: A Flexible Architecture for Information Commerce*. Proceedings of the Second ACM International Conference on Digital Libraries. 1996.
6. Thompson, M., et al., *Certificate-based Access Control for Widely Distributed Resources*. Proceedings of the 8th USENIX Security Symposium. pp. 215-227. 1999.
7. Chadwick, D. W. *An X.509 Role-based Privilege Management Infrastructure*. Business Briefing. Global Infosecurity 2002. http://www.permis.org/
8. López, J., Maña, A., Yagüe, M.I. *XML-based Distributed Access Control System*. In Proceedings of EC-Web'02. Springer-Verlag, LNCS 2455. 2002.
9. Fayad, A., Jajodia, S. *Going Beyond MAC and DAC Using Mobile Policies*. In Proceedings of IFIP SEC'01. Kluwer Academic Publishers. 2001.
10. McCollum, C.J.; Messing, J.R.; Notargiacomo, L. *Beyond the pale of MAC and DAC - Defining new forms of access control*. Proceedings of the IEEE Symposium on Security and Privacy, pp. 190-200. 1990.
11. Yagüe, M. I. *On the suitability of existing access control and DRM languages for mobile policies.* University of Málaga. Department of Computer Science Technical Report nb. LCC-ITI-2002/10. 2002.
12. Bertino, E., Castano, S., Ferrari, E. *Securing XML documents with Author-X*. IEEE Internet Computing, 5(3):21-31, May/June 2001.
13. Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P. *Controlling Access to XML Documents.* In IEEE Internet Computing, vol. 5, n. 6, November/December 2001, pp. 18-28.
14. Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P. *A Fine-Grained Access Control System for XML Documents.* In ACM Transactions on Information and System Security (TISSEC), vol. 5, n. 2, May 2002, pp. 169-202.
15. W3C. *XML-Schema*. http://www.w3.org/XML/Schema
16. W3C. *XML Path Language*. http://www.w3.org/TR/xpath
17. ITU-T Recommendation X.509. *Information Technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks*. 2000.
18. W3C. *Resource Description Framework* (RDF). http://www.w3c.org/RDF/
19. W3C. *XML-Signature Syntax and Processing.* http://www.w3.org/TR/xmldsig-core/. 2002.
20. Chatvichienchai, S., Iwaihara, M., Kambayashi, Y. *Translating Access Authorizations for Transformed XML Documents.* In Proceedings of Ec-Web'02. Springer-Verlag, LNCS 2455. 2002.
21. Kudo, M., Hada, S. *XML Document Security based on Provisional Authorization*. In Proceedings of the 7th ACM Conference on Computer and Communications Security. 2000.
22. Maña, A., Pimentel, E. *An Efficient Software Protection Scheme*. In Proceedings of IFIP SEC'01. Kluwer Academic Publishers. 2001.