A Privacy-Enhanced Attribute-Based Access Control System

Jan Kolter, Rolf Schillinger, and Günther Pernul

Department of Information Systems, University of Regensburg,
D-93040 Regensburg, Germany
{jan.kolter,rolf.schillinger,guenther.pernul}@wiwi.uni-regensburg.de

Abstract. Service-oriented architectures (SOAs) are increasingly gaining popularity due to their considerable flexibility and scalability in open IT-environments. Along with their rising acceptance comes the need for well suited security components. In this respect, access control and privacy emerged to crucial factors.

Targeting the demands of a SOA, many promising authorization models have been developed, most notably the attribute-based access control (ABAC) model. In this paper we take up concepts from the OASIS XACML and WS-XACML specifications and introduce a dynamic ABAC system that incorporates privacy preferences of the service requestor in the access control process. Separating the Policy Decision Point from the service provider's premises, our infrastructure enables the deployment of alternative PDPs the service requestor can choose from. We employ a PKI to reflect the sufficient trust relation between the service provider and a potential PDP. Our work is carried out within the European research project Access-eGov that aims at a European-wide e-Government service platform.

1 Introduction

The trend towards large distributed IT-systems is ongoing and promises many economical and technical advantages. Especially the service-oriented architecture (SOA) paradigm [1] emerged to a popular and widely adopted technology, as SOAs allow the easy wrapping of existing distributed applications into platform independent web services. Distributed architectures and SOAs in particular bear large potential for both service providers and service consumers. For service providers the main incentives of a SOA lie in the low maintenance of large service infrastructures. Furthermore, SOAs facilitate the bundling of single, physically divided services. With a flexible SOA a service provider can easily outsource selected links of its value chain to specialized providers. Service consumers, on the other hand, profit from easy access to services as well as standardized discovery and execution of web services.

The distributed and flexible character of SOAs calls for well fitted security mechanisms in place. Especially access control and privacy are security concerns that represent decisive factors for the building of a secure and trustworthy service

S. Barker, G.-J. Ahn (Eds.): Data and Applications Security 2007, LNCS 4602, pp. 129–143, 2007.

infrastructure. Access control has been addressed frequently in the context of SOAs. Targeting the flexible and dynamic requirements of SOAs the attribute-based access control (ABAC) model is considered an appropriate approach for an access control system [2]. Recent standardization efforts of the OASIS have created the XACML specification [3], a standard suitable for the implementation of an ABAC system. The WS-XACML working draft [4] specifies this standard for the use in a SOA environment.

Apart from flexible access control, privacy increasingly moves to the center of attention in distributed IT infrastructures. Respective studies show that users become more and more concerned about the disclosure of private attributes [5]. In the ABAC authorization model, however, the disclosure of personal attributes (subject attributes) is essential for the access decision. Consequently, privacy plays a special role in distributed architectures that employ an ABAC system.

In this paper we introduce an ABAC system that incorporates privacy-preserving components. We enable the service requestor to define individual attribute disclosure rules. These privacy preferences are utilized to select one out of many alternative Policy Decision Points (PDPs). As we decouple the PDPs from the service provider's premises, it becomes more likely that the characteristics of one PDP match the service requestor's attribute disclosure rules. Introducing a PKI in our infrastructure, we address the fact that the service provider must fully trust the entitled PDPs in their unbiased decision making capabilities.

The content presented in this paper has been developed within the European research project Access-eGov[1]. The project's goal is to develop a European-wide e-Government platform that facilitates the semantic discovery of individual and bundled services. A cornerstone of this platform is a powerful security infrastructure that provides dynamic authorization functionality for public authorities and protects the citizen's privacy at the same time.

The remainder of this paper is organized as follows: Section 2 discusses relevant ABAC approaches and selected privacy technologies. In Section 3 we present the concept and implementation details of our privacy-enhanced ABAC infrastructure. Section 4 lays out details about the integration into the research project Access-eGov. After describing related work in Section 5, Section 6 finally concludes the paper and gives an outlook on planned future work.

2 Fundamentals

In this section we describe basic access control and privacy concepts and technologies. Based on these inputs we present our privacy-enhanced security architecture in Section 3.

2.1 Access Control

Access control is generally defined as the prevention of unauthorized use of a resource, which includes the prevention of use of a resource in an unauthorized

[1] European Union, IST Program, Contract No. FP6-2004-27020.

manner [6]. As confidentiality is a basic requirement for every interaction, access control is considered an important security functionality. Over time various authorization models have been developed, most notably the Discretionary Access Control (DAC) model, the Mandatory Access Control (MAC) model and the Role-based Access control (RBAC) model [6]. However, all of these approaches do not meet the requirements of large-scale distributed environments like SOAs (see Section 1). Here, an access control system is needed, which is flexible enough to handle the inherent heterogeneity of users.

Addressing the needs of distributed systems, the attribute-based access control (ABAC) model evolved. Due to its flexibility and its capability to express complex access control semantics ABAC is deemed a suitable authorization model for SOAs [2]. The ABAC model moves away from the static definition of access permissions and is based on the comparison of subject and object descriptors, allowing for dynamically grouping objects and subjects [7]. Unlike an RBAC approach, this grouping is not done manually by a system administrator but implicitly by subject and attribute values. A big advantage of ABAC becomes evident, if one considers an access policy that grants access to a certain resource depending on the service requestor's age. As an attribute like the age changes continually, the access policy needs to be evaluated dynamically.

The XACML specification [3] is an OASIS standard that supports the integration of subject and object attributes in access policies, a feature that is essential for ABAC policies. The standard defines a powerful policy language that supports complex, fine-grained rules. Rules are aggregated to policies that control the access to a resource. For a detailed review of the XACML policy elements the reader is referred to [3].

Along with the policy language the XACML standard defines an authorization infrastructure that is generic enough to implement the ABAC authorization model [8]. Fulfilling the needs of distributed architectures, the XACML architecture logically separates the access control components responsible for policy definition, policy enforcement and policy evaluation. Specifically, the XACML architecture specifies the implementation of a Policy Enforcement Point (PEP), a Policy Administration Point (PAP), a Policy Decision Point (PDP), a Policy Information Point (PIP), and a Context Handler. Each of these actors is devoted to one specific task of the access control process: The PEP receives access requests and forwards them to the PDP which is responsible for the evaluation of attributes and the access decision. The PIP supplies the PDP with subject and object attributes that are relevant for the access decision. The access policy is provided by the PAP that stores and maintains the access rules. The XACML architecture also employs a Context Handler to collect and broker data flows.

The Web Service Profile of XACML (WS-XACML) specification [4] defines means for the application of XACML in a SOA. While WS-Security [9] introduce security tokens in the context of web services, WS-XACML specifies the Authorization Token, a format that allows the transfer of the access decision to a trusted third party. This enables the trusted third party to make an access decision on behalf of the service provider. Furthermore, WS-XACML defines a

format for the expression of authorization, access control, and privacy policies of web services, areas that are not covered by the common W3C standard WS-Policy [10]. By means of policy assertions, WS-XACML facilitates the definition of requirements and capabilities with respect to authorization and privacy on client and service side. Matching semantics regulate that the capabilities of the service provider must match the requirements of the client, and vice versa. Additionally, WS-XACML defines the relation of P3P policy preferences and XACML policy assertions.

2.2 Privacy

User behavior and the way users disclose personal data have changed significantly over time. A main reason for this development is the growth of the Internet to a multi-million user platform that is used for various needs and wants. Especially eCommerce and trends like interactivity of web sites and personalized offers strongly rely on personal user data.

An increasing number of users, however, perceive this trend as a privacy threat, as they need to disclose more and more personal information to a growing number of providers [5]. Addressing these concerns the Platform for Privacy Preferences (P3P) [11] provides a privacy policy language which enables service providers to advertise their individual privacy policy. A P3P privacy policy describes how personal data of users are dealt with, including information about the purpose and the recipients of the collected data. On client side, privacy preferences of the user are collected and translated into "A P3P Preference Exchange Language" (APPEL) [12]. A privacy agent then uses this information to signal if a web site's privacy policy is in line with the user's pre-defined privacy preferences.

3 A Privacy-Enhanced ABAC System

In this section we lay out the conceptual and technical aspects of an ABAC-based access control infrastructure that incorporates privacy disclosure rules of the client. The section starts with a brief outline and a description of our goal.

3.1 Outline and Goal

As described in Section 2, the basic idea of an ABAC system is to grant access to a resource based on static or dynamic attributes of the service requestor (subject) and the resource itself. The main advantage of ABAC lies in its dynamic character which makes the access control process flexible and satisfies the needs of SOAs. However, ABAC strongly relies on the disclosure of privacy-sensitive subject attributes service requestors are not willing to share in any circumstance. Especially users of large-scale IT-infrastructures have a rising interest in the controlled disclosure of their attributes. Addressing these concerns of users, we introduce a security infrastructure that enhances the ABAC system specified by the XACML specification [3] with privacy preserving components.

In order to enable the client to control the transfer of privacy-sensitive attributes, we propose to take up common ideas from privacy enhancing technology (PET) approaches. As a manual approval by the client for each service access seems obviously unrealistic, clients should define their individual privacy preferences (attribute disclosure rules) [11,12]. These rules should specify what attributes a client is willing to disclose under which circumstances. When accessing a resource protected by an ABAC system, these preferences should be considered dynamically when the system determines the set of attributes a client needs to provide (see example in Section 3.2).

With clients defining their individual privacy preferences, it is not unlikely that attribute disclosure rules forbid the transfer of certain attributes to a particular service provider. For this reason, we utilize the flexible character of the XACML architecture (see Section 2) and separate the Policy Decision Point (PDP) from the service provider's premises [13]. In this scenario a direct transfer of client attributes to the service provider itself is not necessary, as the outsourced PDP is the only actor that collects and evaluates attributes.

The separation of the PDP from the service provider even facilitates the deployment of many different PDPs a client can choose from. Each individual PDP offers a variety of capabilities. In a dynamic selection process a PDP is chosen whose capabilities match a client's privacy preferences. Consequently, the access decision is made by a PDP the client trusts and is comfortable to share certain attributes with. This scenario makes it more likely for a client to find a suitable PDP that does not conflict with personal attribute disclosure rules.

At this point it is noteworthy that a breakup of the PDP from the service provider's access control infrastructure is only possible, if the service provider fully trusts the evaluation and decision making processes of the PDP. The proposed infrastructure acknowledges this aspect and incorporates a Public Key Infrastructure (PKI) that is used to reflect the service provider's level of trust in a PDP. In the following the described infrastructure is presented in detail starting with a conceptual view of the architecture.

3.2 Architecture

The ABAC system sketched in the last section is built on the XACML architecture (see Section 2). In the XACML architecture the access control process is split between several actors, namely a Policy Enforcement Point (PEP), a Policy Decision Point (PDP), a Policy Administration Point (PAP) and a Policy Information Point (PIP).

In order to dynamically choose a PDP that matches a client's individual privacy preferences, we extend the XACML architecture with elements and actors that contribute to privacy and trust in the dynamic access control process. Figure 1 depicts our proposed architecture.

Apart from the common access control actors, the figure shows a set of alternative PDPs that are capable of performing the access decision. The PDP Selector on client side is responsible for the browsing, matching, and the selection of a proper PDP.

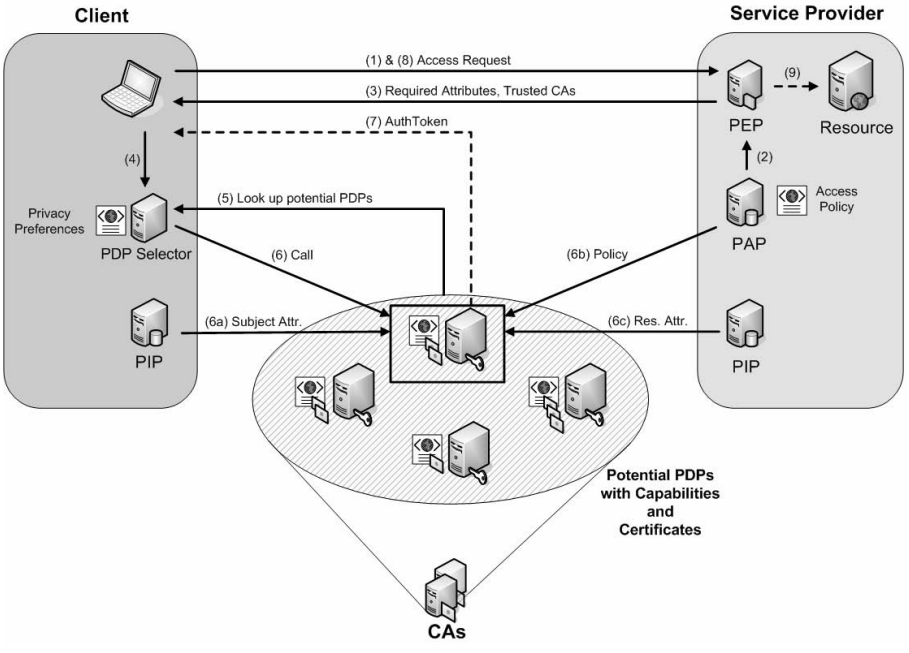

Fig. 1. An ABAC system enabling the dynamic selection of a privacy-conform PDP

The fact that the separated PDPs are entitled to make the access decision outside the influence of the service provider requires a maximum level of trust of the service provider, as an outsourced PDP is solely responsible for the evaluation of the required attributes and the processing of the access policy. Employing a PKI, we introduce the idea of trust certificates being held by each PDP. These certificates reflect the trust of service providers in the evaluation capabilities of a PDP. Specifically, they state that a certain PDP is authorized to evaluate a certain attribute. As not every service provider will trust every Certificate Authority (CA), each PDP can hold attribute evaluation certificates from several CAs for a particular attribute.

Focusing on the sequential steps of the access control process, our scenario starts with a client accessing a protected resource of the service provider (1). The responsible PEP receives the access request and calls the PAP that holds the access policy of the resource (2). The PAP then reads the access policy and determines the subject attribute set required for an access decision, which is subsequently forwarded to the client (3), e.g. a credit card number and the date of birth. At this point, the client is also supplied with the CAs the service provider trusts.

We point out that the required attributes can also include alternative attribute sets. Increasing the chance to get access to the requested resource, the client can opt to transfer several alternative attribute sets. However, in the process

of finding a proper PDP it is likely that only the transfer of a few alternative attribute sets will be in accordance with the client's privacy preferences.

After the required attributes have been transmitted, the client calls the PDP Selector (4) to browse for potential PDPs (5). In a first step, the PDP Selector filters all PDPs that do not hold a certificate for the required attributes which need to be evaluated. Certificates must be issued by a CA the service provider trusts. The PDP Selector is provided with that information in step (2) and (3).

As mentioned before, a suitable PDP must also meet the privacy preferences of the client. Such requirements for example limit the disclosure of a credit card number to financial institutions. They could also define that the date of birth is only transferred, if a secure connection has been established. For this reason, each potential PDP advertises a set of capabilities that define service attributes and technical aspects. After the PDP Selector has looked up and filtered available PDPs, the capabilities of the remaining PDPs are matched against the client's privacy preferences. If for example the credit card number and the date of birth are required subject attributes, only PDPs under the control of a financial institution that can establish a secure connection are actors the client is comfortable to transfer the required attributes to.

Once a suitable PDP has been selected, the PDP is called with the service request (6) and the required subject attributes from a client side PIP (6a). It also receives the access policy (6b) and required resource attributes (6c) provided by the PAP and the PIP of the service provider. Subsequently, the PDP evaluates all required attributes, processes the access policy and makes an access decision. In case of a positive access decision ("Permit"), the PDP issues an authorization token to the client (7). The client in turn uses the authorization token to access the resource (8). The token is acknowledged by the PEP which grants access to the resource (9).

Figure 2 shows a sequence diagram of the presented access control process, which points out the role of each participating actor in the access control architecture. The following section focuses on technical aspects and standards we employed to develop the proposed infrastructure.

3.3 Technical Details

Employing the proper technical means for the dynamic selection of a PDP is vital to the success of our envisioned infrastructure. Section 2.1 and Section 2.2 already presented specific policy languages capable of describing privacy preferences. P3P is a notable representative, as its advantage lies in the most important aspect of privacy descriptions, the user-friendliness. P3P, however, is not suitable to describe the actual application of privacy preferences. This task rather requires policy languages like XACML. For this reason, P3P and XACML are considered as complimentary technologies [14], with P3P describing the preferences on a rather high level and XACML allowing an "instantiation" of a P3P policy, describing it at a lower level which involves specific attributes and rules.

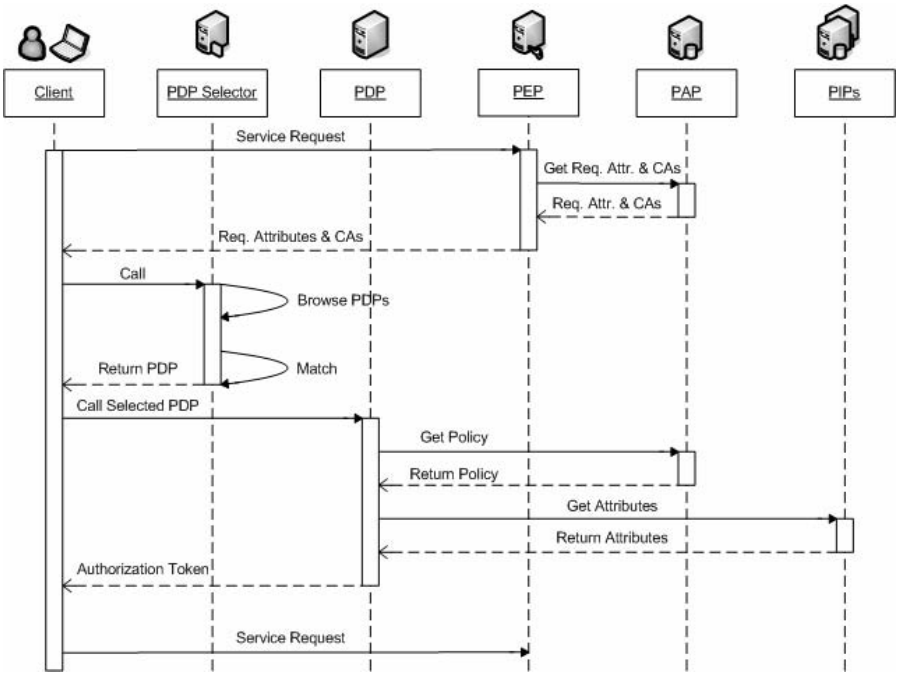

Fig. 2. Sequence diagram of the privacy-enhanced ABAC process

As the presented access control system targets the controlled disclosure of individual client attributes, the following will assume available privacy information at a low representation level.

Apart from privacy-related information on client side, the proposed infrastructure also relies on similar information pieces of the service provider, which needs to publish obligations regarding the client's data it commits itself to. This information represents the counterpart of privacy preferences of the client. As in our scenario a physically separated PDP is the target of client attributes and as the presented infrastructure facilitates multiple alternative PDPs, this information must be advertised by each available PDP.

A suitable candidate for the uniform encoding of privacy preferences of the client and obligations of the PDPs is XACML, which however is not built for the dynamic negotiation of policies, as the intersection of policies is not defined in the XACML standard [15]. The SOA-focused Web Service Profile of XACML (WS-XACML) directly addresses this shortcoming and offers a solution for the storing of the above mentioned information pieces. For a proper representation we utilize the two WS-XACML policy elements Requirements and Capabilities, which are wrapped into assertions, enabling a suitable matching in a web service environment.

Considering authorization information, the same WS-XACML elements can be used to map client attributes and access rules of the service provider. Table 1

Table 1. Requirements and capabilities of clients and PDPs

	Client	**PDPs (Service Provider)**
`ws-xacml:Requirements`	Privacy preferences, obligations to the service provider	Access Policy, rules built with client attributes
`ws-xacml:Capabilities`	Client attributes	Committed obligations of the service provider

categorizes privacy and authorization-related information into WS-XACML Requirements and Capabilities of the client and the PDPs, which in our case represent the service provider.

Requirements and Capabilities consist of XACML rules and policies expressed in a certain vocabulary. For our scenario they are wrapped in the following WS-XACML-specified privacy and authorization assertions, which are derived from the `XACMLAbstractAssertionType`:

- `XACMLPrivacyAssertion`
  The Requirements element of this assertion type issued by the client contains privacy preferences. As the Requirements element has to contain a Vocabulary element and as WS-XACML defines a P3P vocabulary, encoding the privacy preferences with the P3P vocabulary is a logical choice. The specification furthermore offers the client the possibility to define a Capabilities element, containing obligations the client is willing to accept. The privacy assertion of each PDP, on the other hand, expresses privacy-related obligations in the Capabilities field and might contain Requirements a client must meet.

- `XACMLAuthzAssertion`
  The authorization assertion's Requirements element filled by the PDP defines access rules and uses standard XACML attributes. According to the specification, every vocabulary is valid; XACML, however, already provides all needed elements. Like in the privacy assertion, the WS-XACML specification allows a Capabilities element to express the PDP's authorization-related obligations. Authorization assertions issued by the client contain personal attributes using the Capabilities field.

WS-XACML also describes an algorithm for matching assertion types. In our case, on the domain-specific level, matching is done between assertions of the same element name. The matching evaluates to "true", if all Requirements of one assertion can be fulfilled by Capabilities present in the other assertion. Certain special cases are also addressed; the interested reader is referred to [4] for detailed matching rules.

Once a suitable PDP has been selected using the privacy assertions of the client and those of all potential PDPs, the entitled PDP has to arrive at an access decision using the authorization assertions. If access to the requested service is granted, the PDP creates a `xacml-saml:XACMLAuthzDecisionStatement` token.

The token is passed to the service requestor which subsequently uses this token to gain access to the requested resource.

As mentioned earlier, an access control system that provides PDPs outside the service provider's premises must also incorporate trust-related security mechanisms. Choosing a malfunctioning PDP could result in the fraudulent or at least illegitimate access to a service. A malfunction in this context could either occur accidently in form of software bugs or deliberately through attacking the PDP. If, on the other hand, the user hands out his personal data to a malfunctioning PDP, his privacy might be violated or substantial financial losses have to be sustained, if an attacker manages to acquire some of the user's credentials.

One way to tackle this problem is to hardwire the system, effectively taking away every possibility of dynamic extension or reconfiguration. Therefore, we introduce a PKI scheme which, in practice, resembles the handling of the Transport Layer Security Protocol (TLS) [16]. In our scheme, the CAs attest that a particular PDP on a certain IP is allowed to evaluate certain attributes, by filling these information bits in the Distinguished Name field of the certificate and cryptographically signing it. Given that the service provider trusts that CA, implied by trusting the CAs root certificate, it can easily check if the digital certificate is valid and consequently the PDP is trustworthy. Such a PKI based scheme greatly increases the overall security of the dynamic PDP selections. Even the client can benefit from this PKI by using the same procedure to verify the validity of a certain PDP.

We point out that - following the concepts and technical aspects presented in this section - the whole access control process from request to response can be handled dynamically at runtime. No hard coding of the relationship between service provider and PDP is required, enabling the client to choose a privacy-conform PDP based on individual preferences.

4 Integration into Access-eGov

The European project Access-eGov[2] targets the interoperability of e-Government services by facilitating the composition of semantically annotated services to complex process definitions. Particular attention is paid to the fact that many e-Government services are not available online yet. Even those online services are rarely semantically annotated web services. For this reason, the whole platform is geared towards supporting offline and traditional e-services through the use of the specialized concept of a "complex goal" and an own process model.

A careful analysis of user requirements and a standard software development process led to a functional specification of the Access-eGov architecture's components [17,18]. Figure 3 gives an overview of the resulting service-oriented architecture. Users, service providers, and the Access-eGov platform are main actors, supported by management tools and specially crafted service ontologies. Users are represented by their digital personal assistant, which is responsible for invoking services and storing the state of execution. Apart from service-related

[2] http://www.access-egov.org/

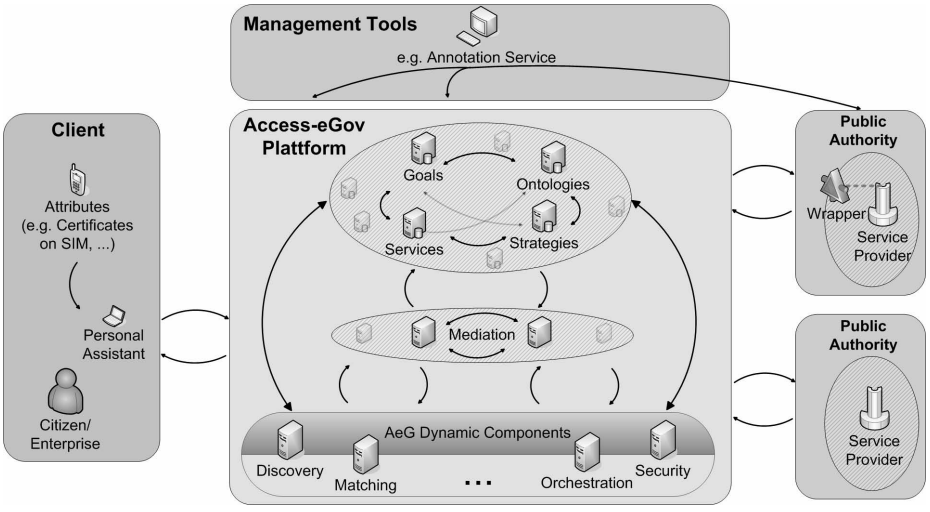

Fig. 3. Access-eGov service architecture

functions, the user's personal assistant maintains a user model containing profile information in the form of attributes. This profile also contains the user's privacy preferences and can be stored in a number of locations inside the Access-eGov platform or on user side, e.g. on a smartcard.

The dynamic components of Access-eGov operate on a peer-to-peer networked set of repositories, storing ontologies, annotated services, goals and strategies. A strategy represents the overall task a user wants to accomplish. The discovery and matching components split that task into well-defined sub-tasks, so called goals, which can be orchestrated and finally executed. The figure also shows the above mentioned option to integrate traditional e-services through the use of a wrapper. Once the discovery, matching and orchestration have been completed and a single service has been requested, the access control process introduced in Section 3 is started.

The personal assistant incorporates general as well as attribute-specific privacy preferences of the user in the corresponding XACML assertions. Utilizing the characteristics of a SOA-based peer-to-peer network, the security component in the platform does the same for the service provider's privacy obligations and requested attributes, which are part of a service repository. This enables a PDP Selector subcomponent to look up potential PDPs, either through brute force or through recursive use of the Access-eGov platform.

To single out a suitable PDP, the personal assistant tries to call the retrieved PDPs one by one. Once a PDP is found, which fulfills all the client's and provider's requirements, the provider checks its corresponding trust certificate. If the certificate is valid and issued by a CA the service provider trusts, the

access request can be completed. In case of a successful completion, the personal assistant may get an Authorization Token, which then can be used by the client to access the originally requested service.

From an integration point of view, the best approach is to take the Authorization Token concept one step further. Making the government services and the Access-eGov platform a single-sign-on domain would greatly minimize the administrative effort and the need for redundant data storage. In the course of this project, it already showed, however, that single-sign-on solutions are not feasible with current local laws. The public authorities cannot abandon control of the authorization process for users of their services.

Targeting a flexible security infrastructure, the components of the Access-eGov security subsystem are modeled as services themselves. After semantically annotating them, using a specially crafted security ontology, those services are stored in the same service repositories as services offered by the public authorities. As the concept of a Goal in Access-eGov is not limited to serve users with specific tasks, goals can also be used to describe tasks and workflows unrelated to a specific user, but of a more technical nature. Above mentioned dynamic components can resolve a "security strategy" in exactly the same way, allowing the infrastructure to create new security systems on the fly.

5 Related Work

Focusing on ABAC and privacy aspects, our work builds on several research initiatives in the respective areas. In [19] the authors introduce a attribute certificate-based ABAC system using the AKENTI engine within the context of grid computing. Their approach, however, does not integrate any privacy-related mechanisms. On a theoretical level [20] describe a uniform framework that formulates and evaluates logical rules controlling service access and information release. Defining a powerful policy language, the uniform framework especially focuses on theoretical definitions of access control and information release policies and their logical matching. In [8] the authors enrich an ABAC system with an inference engine, targeting the semantic interoperability of security policies.

The definition of privacy preferences (disclosure rules of personal attributes) has been addressed by several PET initiatives [11,21]. Within the scope of the PRIME project the definition of data handling policies is proposed [22]. These policies define how personal data of users are dealt with at the receiving party. In that context, in [23] a privacy obligation management model is introduced, providing means to monitor and enforce privacy rules of end-users. In [24] XACML-based attribute release policies are utilized in the context of identity providers. Like our approach the author underscores the suitability of XACML to model attribute release policies. The author primarily addresses the controlled attribute release of an identity provider, not an access control infrastructure as a whole.

6 Conclusions

Distributed IT-infrastructures like the service-oriented architecture (SOA) increasingly rely on well fitted security mechanisms that protect both the privacy of clients and the resources of service providers. An attribute-based access control (ABAC) authorization system is flexible enough to satisfy the needs of a SOA. However, the ABAC model heavily relies on the disclosure of personal attributes, a characteristic that could conflict with privacy preferences of the client.

In this paper we introduced an ABAC system that provides a set of alternative, physically separated Policy Decision Points (PDPs). After the definition of individual privacy preferences (attribute disclosure rules), our approach facilitates a client to dynamically select a PDP that is in line with his privacy preferences. Addressing the necessary trust a physically separated PDP must provide, we embed a PKI in the access control process. For the process of defining, advertising and matching privacy preferences and PDP capabilities we take up concepts from the OASIS WS-XACML specification. Our approach enables the whole access control process to be handled dynamically at runtime. No static relations between a PDP and the service provider are necessary. The presented solution is part of a service-oriented security infrastructure within the research project Access-eGov.

Future work will involve performance and usability tests in the Access-eGov system. Furthermore, we are pursuing the notion of dynamically built PDP cascades, which will mitigate the effect of strict privacy and trust preferences of users and service providers, respectively. Finally, we are checking the potential of Trusted Computing technologies. With a trusted environment on client side the PDP could be moved directly to the client, an option that would obviously suit any privacy disclosure rule.

Acknowledgment

The work reported in this paper was in part funded by the European Union within the project Access-eGov (Sixth Framework Program, Contract No. FP6-2004-27020). We thank our project partners for helpful comments and stimulating discussions.

References

1. MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R.: Reference Model for Service Oriented Architecture 1.0. OASIS Standard (October 2006)
2. Yuan, E., Tong, J.: Attributed Based Access Control (ABAC) for Web Services. In: Proc. of the IEEE International Conference on Web Services (ICWS'05), Washington, DC, United States, pp. 561–569. IEEE Computer Society Press, Los Alamitos (2005)

3. Moses, T.: eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard (February 2005)
4. Anderson, A.: Web Services Profile of XACML (WS-XACML) Version 1.0. OASIS Working Draft, vol. 8 (December 2006)
5. Earp, J., Baumer, D.: Innovative Web Use to Learn About Consumer Behavior and Online Privacy. Communications of the ACM 46(4), 81–83 (2003)
6. Lopez, J., Oppliger, R., Pernul, G.: Authentication and Authorization Infrastructures (AAIs): A Comparative Survey. Computers & Security 23(7), 578–590 (2004)
7. Priebe, T., Dobmeier, W., Muschall, B., Pernul, G.: ABAC - Ein Referenzmodell für attributbasierte Zugriffskontrolle. In: Proc. of the 2nd Jahrestagung Fachbereich Sicherheit der Gesellschaft für Informatik (Sicherheit '05), Regensburg, Germany, pp. 285–296 (2005)
8. Priebe, T., Dobmeier, W., Kamprath, N.: Supporting Attribute-based Access Control with Ontologies. In: Proc. of the 1st International Conference on Availability, Reliability and Security (ARES '06), Washington, DC, United States, pp. 465–472. IEEE Computer Society Press, Los Alamitos (2006)
9. Nadalin, A., et al.: Web Services Security: SOAP Message Security 1.1. OASIS Standard Specification (2006)
10. World Wide Web Consortium: Web Services Policy 1.2 - Framework (WS-Policy). W3C Member Submission (April 2006)
11. Cranor, L., et al.: The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. W3C Working Group Note (November 2006)
12. Cranor, L., Langheinrich, M., Marchiori, M.: A P3P Preference Exchange Language 1.0 (APPEL 1.0). World Wide Web Consortium Working Draft (April 2002)
13. Kolter, J., Schillinger, R., Pernul, G.: Building a Distributed Semantic-aware Security Architecture. In: Proc. of the 22nd International Information Security Conference (SEC2007), Sandton, South Africa, May 2007 (to Appear)
14. Anderson, A.: The Relationship Between XACML and P3P Privacy Policies (November 2004), `http://research.sun.com/projects/xacml/XACML_P3P_Relationship.html`
15. Andersson, A.: Sun Position Paper. W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement (October 2006)
16. Dierks, T., Rescorla, E.: RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1. Internet RFCs (April 2006)
17. Klischewski, R., Ukena, S., Wozniak, D.: User Requirements Analysis & Development/Test Recommendation. Access-eGov deliverable D2.2 (July 2006)
18. Tomasek, M., Paralic, M., et al.: Access-eGov Components Functional Descriptions. Access-eGov deliverable D3.2 (November 2006)
19. Thompson, M., Johnston, W., Mudumbai, S., Hoo, G., Jackson, K., Essiari, A.: Certificate-based Access Control for Widely Distributed Resources. In: Proc. Proc. of the 8th USENIX Security Symposium, Washington, DC, United States (1999)
20. Bonatti, P., Samarati, P.: A Uniform Framework for Regulating Service Access and Information Release on the Web. Journal of Computer Security 10(3), 241–271 (2002)
21. Hansen, M., Krasemann, H.: Privacy and Identity Management for Europe PRIME White Paper. PRIME deliverable D15.1.d (July 2005)

22. Ardagna, C., De Capitani di Vimercati, S., Samarati, P.: Enhancing User Privacy Through Data Handling Policies. In: Proc. of the 20th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2006), Sophia Antipolis, France (July 2006)
23. Casassa Mont, M.: Towards Scalable Management of Privacy Obligations in Enterprises. In: Proc. of the Third International Conference on Trust, Privacy, and Security in Digital Business (TrustBus '06), Krakow, Poland, pp. 1–10(Septmeber 2006)
24. Hommel, W.: Using XACML for Privacy Control in SAML-Based Identity Federations. In: Communications and Multimedia Security, pp. 160–169 (2005)