User-Controlled Dynamic Access Credential Enrichment for Run-time Service Selection

Christoph Fritsch, Günther Pernul
Department of Information Systems
University of Regensburg
Regensburg, Germany
{christoph.fritsch|guenther.pernul}@wiwi.uni-regensburg.de

***Abstract*—Dynamic run-time selection and sourcing of service components provide considerable potential in todays changing business world. They provide means to counter agility, flexibility and the ability to integrate applications originating from systems of different security domains. While the advantages are obvious strong implications to security in general and authorizatin and access control in particular do exist. In this paper we present an infrastructure-based approach for en-route dynamic credential enrichment. It enables dynamic replacement of access-restricted service instances by implementing runtime supplementation of security tokens. If authorized, a security intermediary accesses user profiles and retrieves security tokens supplied by identity providers and needed for access control at dynamically selected access-restricted service instances.**

***Keywords*-Dynamic Service Selection, Service Access Control, Mediated Access Control;**

I. Introduction

While organizations have for decades been rewarded for consolidating around standard processes and for stockpiling assets, nowadays flexibility on operation and service allocation is becoming a key differentiator. Operations are no longer assumed to be confined within organizational boundaries but more and more often involve short-term collaboration with frequently changing external partners in a broader ecosystem. Smart business networks and other forms of short-term collaboration promise to allow for *"rapidly pick, plug, and play"* [1] business processes and services accross companies' borders.

Enterprises are demanding for effective and efficient ways to share and consume IT services and resources across companies' boundaries and research on dynamic service selection and sourcing divulged in industry and academia. For example, the European Union funded FP7 research project SPIKE[1] addresses exactly those short-term service-based federations and provides the test bed for the research presented in this paper.

This new openness to speedily establish virtual partnerships and the associated rapid and short-term integration of IT services with partnering organizations poses intense demands on organizations' access control measures [2] and still results in security being one of the most daunting challenges enterprise architects face in cross-organizational SOAs. Access control at companies' IT services has to be guaranteed even if they are supplied rapidly to external parties and the number and identities of people authorized to access single services vary frequently and swiftly. The more flexible and rapid an organization wants source or provide services, the more effective and powerful its access control schemes must be. Essential features such as heterogeneous platforms, ubiquitous accessibility and dynamic workflows that make services and SOA such an attractive paradigm frequently conflict with conventional security models and mechanisms [2].

Addressing these challenges, this work proposes a mediating access control infrastructure that helps organizations to prepare for rapidly sourcing IT services. Authorization and access control demands are dealt with by the mediated access control infrastructure that blends well with different approaches for dynamic service selection and sourcing.

The remainder of this paper is organized as follows: Section II discusses related work in both relevant fields, dynamic service selection and service access control. Subsequently, Section III introduces prevalent design considerations for our proposal. Section IV details our approach to dynamic credentials enrichment in detail before Section V provides information on the prototypical implementation and evaluation within the SPIKE project. Finally, we draw some conclusions in the concluding Section VI.

II. Related Work

VOs and enterprise networks as dynamic, inter-enterprise configurations for rapidly sharing IT resources have been identified as a promising alternative by [1]. Iyer et al. particularly stressed the issue of a flexible service selection and sourcing strategy [3]. *Dynamic Service Selection* addresses agility and flexibility challenges in those dynamic settings while *service access control* is an precondition for putting dynamic service selection approaches in action.

[1] http://www.spike-project.eu

A. Dynamic Service Selection

Schmidt et al. [4] noticed back in 2005 that *"SOA holds out the promise that services can be discovered [...] and bound together to form new and exciting, or simply more efficient applications"*. They developed the protocol switch and the service transformation patterns.

Content-based routing (CBR) in ESBs as mentioned by [5] and others is a rather mature approach. Based on message contents a component decides to which service instance a request is forwarded. DRESR [6] by Bai et al. introduces the idea of abstract routing paths (ARPs), where each service is identified by an URI and an abstract service name. To obtain concrete service instances, a central routing manager component selects a service instance from the pool of candidates for the given task. Another publication approaches service selection from the viewpoint of collaboration trust and reputation [7]. The iWeb framework [8] particulary considers QoS and other context data for selecting the best available service. Like our approach it is composed of a 3-layered architecture for service selection. VRESCo by [9] introduces an aspect-oriented extension for BPEL environments for monitoring and replacing partner links.

The concept of Dynamic Composition Handlers (DCHs) by Chang et al. [10] builds upon an ESB and clearly separates between service interfaces specified in the process and realized interfaces. The ESB-based ProBus approach by Mietzner et al. [11] is a concept for policy-driven dynamic service selection. In contrast to other work, this approach mainly focuses on the definition of selection criteria. ProBus matches the service requester's policy against resource properties of known services to obtain a service candidate.

Based on these works we presented an ESB-based approach that makes use of semantic service descriptions for discovery and mediation [12], [13].

B. Service Security

While dynamic service selection has frequently been addressed by researchers, dynamic service access control in these settings has not yet been considered sufficiently and only independently from dynamic service selection.

The OASIS published several standards such as WS-Security , WS-SecurityPolicy , WS-Federation and WS-Trust to provide general security standards for SOAs. However, all of them only define how to apply security mechanisms to individual SOAP messages, rendering their application to companies interested in opening their business processes a well-engineered but too low-level technical basis for unreflected deployment.

Bertino et al. [14] discuss three essential classes of security services – *identity management*, *authentication* and *access control* – and propose a service-oriented approach to security. The service-oriented security architecture presented by [15] considers the same services and bears a prototype based on an ESB similar to our approach. The FedWare middleware

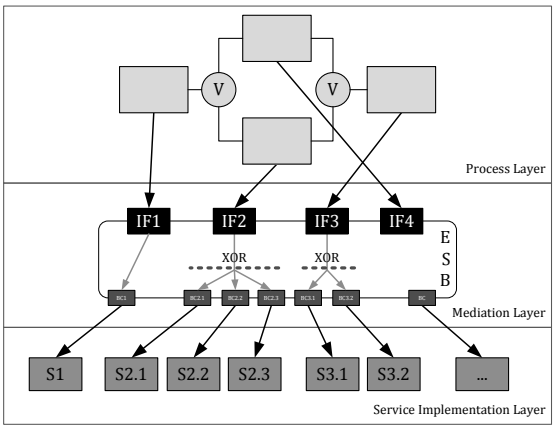

Figure 1. Service Selection and Credentials Enrichment Infrastructure

[16] employs an external Identity Provider (IdP) as we do but is based on the Sun Java System Identity Manager instead of open standards. The web service architecture for decentralized identity- and attribute-based access control by Hebig et al. [17] considers many of these issues but is particularly tailored to SOAP services while our approach is open for different kinds of services due to mediation capabilities of ESBs.

[18] tackles usage and access control in SOAs mainly from a conceptual perspective, focusing access control models and policy languages. Still inadequate understanding of the security issues and potential solutions together with the false belief that companies have to do costly investments into security infrastructures impede broad spreading.

III. Design Considerations and Decisions

Fig. 1 gives a conceptual overview of our ESB-based infrastructure for dynamic service selection that builds the basis for the dynamic credential enrichment approach laid out in this work. Details on the overal infrastructure and semantically supported dynamic service selection have already been illustrated in [12], [13], [19] so that we merely focus on dynamic security credentials enrichment in this work. The basic idea is to introduce a layer of indirection between service requesters and service providers (SPs). This layer aims at moving the point of time when a concrete service instance gets selected as close as possible to service invocation time. Due to this service binding indirection concrete service instances no longer have to be picked at modeling or development time of the service client or process model revealing issues of access control at these dynamically selected services.

The high-level design goals this concept follows can be subsumed on an organizational level as follows: (1) Overall, we aim at enabling *rapid formation of short-term collaborative processes* and *straightforward integration* with other companies' IT services in a SOA-based manner, i.e. by means of dynamically selected services and on demand built

service chains. (2) *Adaptive access control* is a key obstacle for that aim. Without proper access control procedures in place that are likewise applicable to intra- and inter-company scenarios, business relevant services and processes may never be exposed dynamically to others. Flexible but reliable cross-organizational access control and identity management approaches need to be available as an enabling technology. (3) Aiming for secure company-spanning business processes it may not be forgotten that IT services and their permanent and transient availability merely build the breeding ground on which companies process established businesses transactions and grab new business opportunities. Thus, any technological innovation has to *merge with companies' established IT systems and be applicable* to their IT and service landscapes.

On a more technical level further design goals can be identified: *Non-Intrusiveness* is one of them the proposed infrastructure has to meet. Just as the underlying dynamic service selection approach is transparent to clients and service providers, the credential enrichment approach has to be feasible in a minimum-invasive manner to integrate well with current service landscapes. Likewise it has to be *applicable to various kinds of services* and legacy systems in particular in open, federated and continuously developing environments. Along with non-intrusiveness comes the demand for *one-shot service selection and invocation*. Multi-step approaches that require several interactions and message exchanges between service requester and other components before an appropriate access-restricted service candidate gets selected are not applicable without major adjustments in current service landscapes and are thus not feasible. Finally, *standard compliance* marks an important desing goal. The proposed infrastructure aims at connecting various previously unacquainted organizations and in particular their IT systems and services. This goal can only be reached if the infrastructure complies with and supports widespread SOA and web service (security) standards.

Approaches for dynamic service selection and binding laid out in Section II have shown that dynamic service selection is feasible. The tighter but more flexible and loosely coupled this integration between previously unacquainted companies shapes up, the more essential is reliable access control. In the past it has been easy to distinguish between »good«, i.e. internal systems, and »bad«, i.e. the Internet, and security has been enforced at these borders. Nowadays these boundaries gradually become blurry and vanish, a trend that is notably pushed by the SOA paradigm.

As a result, we can identify a gap regarding application of cross-domain security in dynamic service selection settings. McGraw named this gap a "*trinity of trouble*" [20] and further subdivided it into security concerns resulting from *connectivity*, *extensibility* and *complexity*. Based on this we draw the following conclusions for this work: (1) *Connectivity* means that access-restricted services should still be in the run as candidate services for dynamic service selection. For that purpose an authorization and access control concept is essential that obtains data required for access control decisions on demand and in the right format. (2) *Extensibility* demands that the proposed approach shall be extensible to different kinds of services, be it modern web services or legacy services whose access control functions do frequently not consider modern SOA security standards. (3) *Complexity* results from connectivity and extensibility. It increases further when considering integration among independent companies for short time, only. Cross-organizational access management becomes an intrinsic element of the overall complexity.

To address these issues, we opted for implementing the *attribute-based access control (ABAC)* model that allows for expressing all data relevant for an access control decision in form of attributes and credentials that can easily be conveyed in form of security tokens. Along with the demand to span multiple organizations and various service candidates comes the necessity to dynamically ensure that *applicable credentials* are available for SPs to base access control decisions upon. Supplying required security tokens at the right time carrying required data in adequate formats is the main issue this works tackles. User attributes and credentials build the basis for access control decisions in ABAC systems which is why *freshness* of these data is of major importance. The proposed credential enrichment infrastructure thus has to provide means to ensure their actuality while data owners should still *be in control* for which purpose or service invocation which of their data are disclosed.

IV. Run-Time Credential Enrichment

En route dynamic enrichment of access control credentials builds the basis for flexible service selection and substitution. Both challenges are not independent but mutual dependent on one another. In particular minimal invasiveness and applicability to companies' current service landscapes demand for close integration.

A. Overview

The proposed infrastructure aims at dissolving access control issues coming along with dynamic service selection and binding. Infrastructure components are positioned in between service requester and SP to prepare and facilitate service access control even though particular service instances get selected on demand. These mediating components in combination with an extended IdP intercept and extend service invocations as required by service candidates' access control policies. The IdP provides interfaces inspired by the WS-Trust Security Token Service (STS) to enable the security broker to retrieve users' security tokens and credentials as requested by the dynamically selected service instance and as disclosed by the particular user.

Fig. 2 illustrates the main actors of the proposed dynamic service selection and mediated access control infrastructure and their mutual relations. Users interact with the IdP *(1)* and the mediation infrastructure which is split into

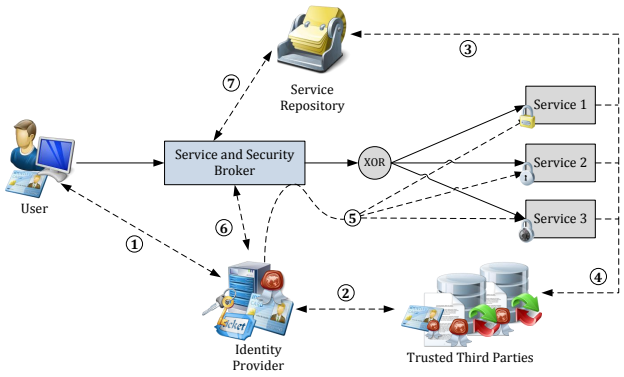

Figure 2. Involved Parties and Interactions

components for (a) dynamic service selection and mediation and (b) credentials enrichment. Interaction with the service and security broker occurs unnoticed by the user as the broker transparently intercepts and processes service invocation messages. Credentials enrichment components in turn form the central elements of the overall proposal. They resort to the service repository *(7)* for discovery and selection of candidate services and to internal semantic or static mediation and transformation facilities for message processing and adaption [12], [13]. Beyond that, they interact with the IdP *(6)* to dynamically fetch user credentials and security tokens in the format required by the dynamically selected service instance. Even though the IdP does not interact directly with particular service candidates and their SP, there still exists an implicit relation *(5)* between both: The IdP keeps security tokens ready based on which SPs decide whether access to their services is granted or denied. To ease establishing a chain of trust between users' IdPs and SPs, trusted third parties might be employed to verify and certify particular attributes of users and issue signed tokens, accordingly *(2)*. The relation between mediation infrastructure and service candidates is equivalent to the standard service invocation and message forwarding relation in any scenario that involves an ESB or other middleware. SPs finally register their services with a service registry *(3)* to make them available as service candidates for dynamic service selection. Beyond that, SPs optionally rely upon trusted third parties *(4)* to approve and certify user data in case they do not have confidence in IdP for ratifying these data.

B. Dynamic Credential Enrichment Procedure

Based on the overview of involved stakeholders and their relations in Fig. 2, Fig. 3 illustrates the overall process for dynamic service selection and credential enrichment as a whole. Steps marked with a »*« in Fig. 3 are optional.

1) Credential Enrichment: Steps *(1)* through *(5)* in Fig. 3 depict preparatory steps. SPs identify in an optional first step *(1)* parties whose assertions about users and confirmations about users' attributes they rely upon when making an access

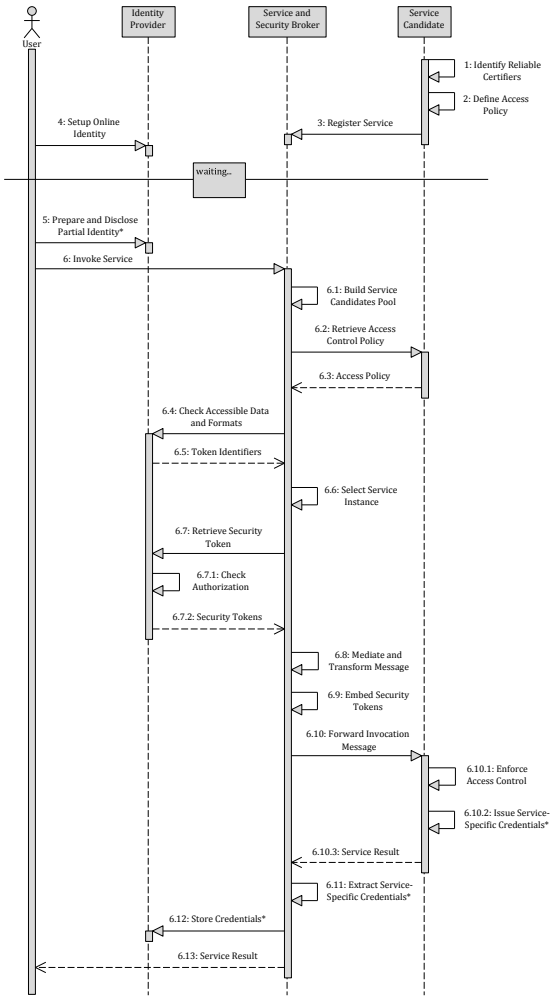

Figure 3. Dynamic Service Selection and Mediated Credential Enrichment

control decision (see relation (4) in Fig. 2). They define access control policies for their services (step *(2)* in Fig. 3, see Section IV-B2 for details) and register their services in a repository (step *(3)*). Users prepare for dynamic service discovery and credentials enrichment by registering with some IdP. Alternatively, they might make use of their companies' IdP if available to maintain credentials, security tokens and other attributes and prepare and disclose partial identities. Details on user actions are laid out in Section IV-B3. Except for partial identity definition and authorization *(5)* which is potentially carried out for any service invocation, all other steps are time- and execution-independent.

Step *(6)* and related sub-steps in Fig. 3 illustrate the dynamic service discovery and credentials enrichment flow. Our approach for dynamic service discovery and binding has already been explained in [12], [13]. First, users (optionally) prepare a dedicated partial identity and authorize the security broker to access contained data (step *(5)*) before they dispatch a service invocation message toward the virtualized service

interfaces operated by the infrastructure (step *(6)*). The invocation message contains a IdP discovery URL and an access token valid for a particular partial identity (see V-C for details). The security intermediary gathers all required data to process and forward the request accordingly. First, eligible service candidates are located (step *(6.1)*) and their respective access control policies are retrieved (step *(6.2)* and *(6.3)*). At the same time the security broker module discovers data accessible at the user's IdP (steps *(6.4)* and *(6.5)*) as further detailed in Section V-C. Based on this information and additional non-functional requirements for an applicable service instance is selected (step *6.6)*. For the particular service instance the security broker then retrieves required credentials from the user's IdP (steps *(6.7.x)*). Having all relevant data available the service request message is then transformed and access credentials embedded accordingly (steps *(6.8)* and *(6.9)*) before it is forwarded to the dynamically selected service instance (step *(6.10)*). In step *(6.10.1)* the SP matches received security tokens and their payloads against its access policies to make an access control decision which is then enforced. Finally, the service result is returned to the user in steps *(6.10.3)* and *(6.13)*.

Optionally, the SP might issue service-specific credentials in step *(6.10.2)*. For future access control decisions of the same user it does then no longer have to elaborately parse and evaluate various attributes encapsulated in different security tokens. At mediator side, these newly assigned tokens are extracted from the service response message (step *(6.11)*) and stored to the user's profile at the IdP (step *(6.12)*) to make them available for future invocations of the same service.

2) Service Access Policy Definition: We suppose that SPs do not resort to pre-established digital identities users assume to interact with a given service but rather base access control on various user attributes. We thus follow an ABAC model which is much more appropriate for dynamic service selection. Instead of authorizing particular users for their services, SPs define (alternative) sets of attributes prospective users have to hold to gain access to a given service. This approach relieves SPs from maintaining identities of all registered and authorized users in favor of a more flexible approach. On the other hand, any user, even previously unacquainted ones, that holds defined attributes can gain access to a service without a priori registration with the SP.

We propose to make use of XACML policies to define access control rules independently of particular service implementations. XACML policies inherently build upon attributes so that various types of security tokens can be expressed. In contrast to WS-SecurityPolicy, a XACML policy might define access restrictions based on attribute values conveyed within particular security tokens such as X.509 or SAML tokens. XACML policies enable the security broker to discover alternative policy sets that permit access to a given service (see step *(6.2)* in Fig. 3).

Please note that the basic principle for this approach to work is that SPs' access control policies are made available publicly. We argue that access control policies are no valuable assets and therefore not worth protecting. This is the exact same way WS-SecurityPolicy follows by supplying services' access policies as part of their WSDL descriptions. Policies identify credentials or tokens required to gain access to a service. Even if malicious requesters find out about feasible attribute sets, they still do not have access to required attributes issued by the `issuer` specified in the policy. In our view, reliability of access control should only depend on the ownership of predefined credentials and attributes and not on confidentiality of access policies.

3) Persona Preparation and Disclosure: Users define attribute sets – also denoted »*partial identities*« – they are willing to disclose for a particular server invocation to enable the security intermediary to obtain users' security tokens from their IdPs later on. Partial identities are composed of any number of attributes, credentials and security tokens a user holds. Users might define partial identities for different purposes such as job-related or private use or based on service types or inquired functionality, might maintain various partial identities at the same time or construct new ones for each invocation of novel service types. The IdP supports users via convenient GUI (see top left in Fig. 5) and returns an authorization tokens to forward to the intermediary in step *(6)* of Fig. 3.

To ensure controlled disclosure of those partial identities for a specific purpose, i.e. the interaction with a particular but dynamically selected service instance, our proposed infrastructure presupposes both users to disclose their data for a particular service invocation and define access restrictions for these disclosures of their online profiles. Hence, a user has to define his disclosure willingness concerning a particular attribute or sub-sets of his profile.

Privacy and data minimization or avoidance related research developed several approaches for that purpose. Several languages for privacy policies and privacy preferences such as P3P Xpref and others exist that allow users to express their willingness to disclose single attributes or a set of attributes to third parties. All these approaches primarily focus on privacy issues, i.e. prevention of data disclosure and processing. They seek to protect personal user data and strongly rely on the cooperation of SPs.

The initial point for the proposed dynamic credentials enrichment approach is different. In contrast to privacy policy approaches, users' primary intention here is not to disclose as few data as possible but instead to access a service type whose implementing service instance and its access control restrictions are not known until service invocation time. For that purpose, users authorize the security broker as intermediate party to retrieve a subset of the attributes they hold to gain access to the dynamically selected service. Even if users do not know in advance which of their attributes will be retrieved by the security broker to enable access control

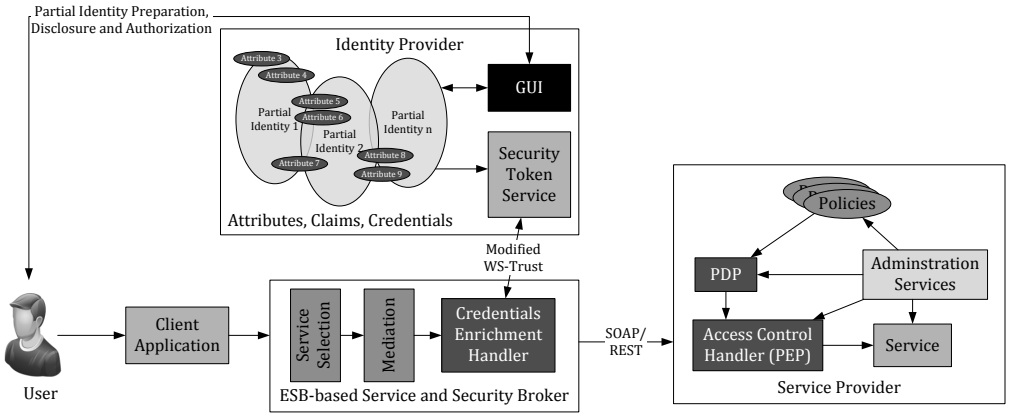

Figure 4. High-Level Architecture

decisions by SPs, they should still be enabled to limit the scope of available data and ensure that only authorized parties can gain access to their IdP-managed online identities. For these reasons, we propose a different approach for attribute release definition. Instead of users defining universal privacy policies ex ante, they merely define sets of attributes they are willing to disclose for a given service request for access control purposes. Applying this authorization, the security intermediary retrieves relevant security tokens at runtime. This approach enables users to easily retrace who retrieved which information as each attribute retrieval can be recorded and visualized by the IdP.

C. Summary

The overall dynamic credentials enrichment process demands for tight integration of both integral parts of the proposed infrastructure, service selection and mediation components on the one hand and mediated access control modules on the other hand. Fig. 4 illustrates the resulting high-level architecture. The ESB-based broker provides a transparent messaging infrastructure that interacts with users' IdPs to retrieve relevant security tokens. Users interact with their IdPs via a convenient GUI to define disclosed attribute sets and authorizations. SPs operate their services, define suitable access policies and implement access control measures to enforce these policies.

V. Prototype and Evaluation

Details on dynamic service selection (*(A)* and *(B)* in Fig. 5) have already been published in [12], [13] so that we only consider run-time credentials enrichment here. As can be seen in Figs. 4 and 5 we built our approach in conformance with the XACML and ISO10181-3 standards regarding separation and distribution of individual security components within the architecture. The client represents the access requester or initiator while the service instance represents the target or resource. The IdP conforms to the functions of the policy information point (PIP). The policy enforcement point (PEP) or access control enforcement function (AEF) is provided by the service instance inbound security module just as the policy decision point (PDP)/access control decision function (ADF). Finally, the policy administration point (PAP) does not exist as a single component but, following the idea of the WS-Security standards, rather each service instance is capable of providing its security policy in a machine-readable form as part of its interface description. Exemplary request-/response messages as well as interface descriptions of our STS service and discovery documents are available[2]

A. ESB-based Transparent Intermediary

Fig. 5 gives an overview of the ESB-based infrastructure prototype for dynamic service selection an credentials enrichment. Apache ServiceMix ESB provides the runtime environment into which several components are deployed to implement required functionality for dynamic service selection and mediation [13]. The underlying JBI standard[3] ensures reasonable and standardized extension points in form of BCs and SEs to implement new functionality for message processing, transformation and enrichment. Relevant functionality for dynamic credential enrichment has been implemented as custom functionality in form of ServiceMix-Bean SEs *((3)* and *(8)* in Fig. 5). These components interact with users' IdPs to retrieve relevant security tokens at runtime. *(3)* ensures that only those services are considered for dynamic selection that do not demand for more or different security tokens and attributes than the user disclosed for the current invocation (see *(6.2)* to *(6.5)* in Fig. 3). *(8)* then retrieves relevant security tokens from the IdP as soon as a concrete service instance has been selected (consistent with *(6.7.x)* and *(6.9)* in Fig. 3). Data on the user's IdP and an access token to its STS interfaces are extracted from the incoming message. This information is then used to obtain information on security tokens and attributes accessible at the IdP before

[2]http://www-ifsresearch.wiwi.uni-regensburg.de/ICEBE/icebe.zip
[3]http://www.jcp.org/en/jsr/detail?id=208

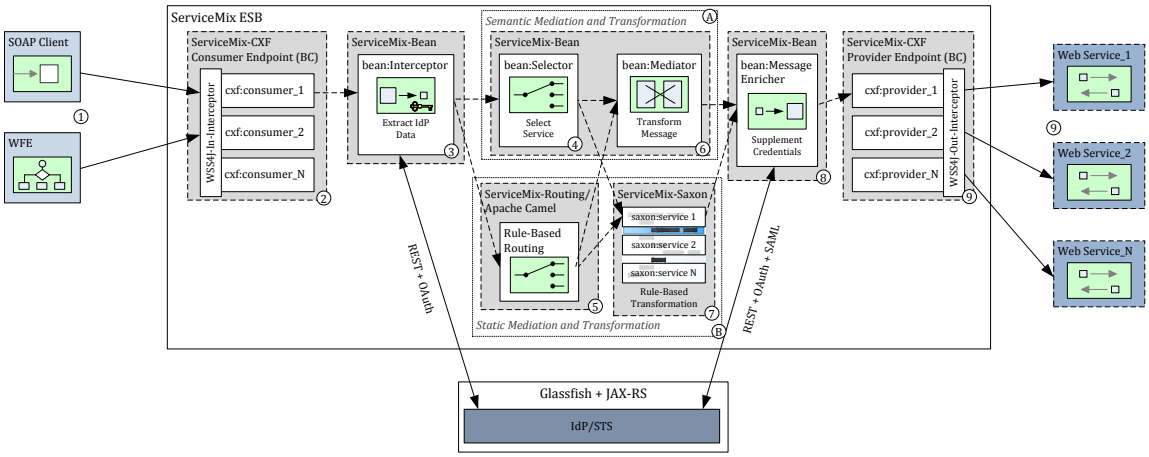

Figure 5. Prototype Overview

authorization data required by a particular service candidate are requested via the IdP's STS interface. Finally, these data are embedded into the service request message.

Thus the ESB-based infrastructure is transparent to client and service. Clients issue service requests to the infrastructure's virtualized interfaces *(2)* and services receive proper invocation messages that contain relevant security tokens from the infrastructure's outgoing BC *(9)*.

B. IdP\STS Interfaces

The IdP forms the user-targeted counterpart of the security intermediary. The prototype comprises two complementary interfaces: (1) A user-facing GUI to let users define sets of attributes they are willing to disclose, configure access to these personas and manage authorized requesters. (2) RESTful STS service endpoints that enable the credential enrichment services to interact with the IdP to retrieve users' security tokens and attributes in various formats and representations.

To retrieve security tokens we make use of WS-Trust `<wst:RequestSecurityToken>` messages. In contrast to standard messages we embed a SAML `Attribute-Query` in its `SecondaryParameters` to let the security broker define in detail what attributes should be enclosed in the requested security token. An exemplary query that returns a SAML security token containing the user's eMail address, name and identifier (lines 12–14) as SAML token (line 2) is shown in Listing 1. To specify requested attributes uniformly, our prototype makes use of identifiers specified in the OASIS Identity Metasystem Interoperability, X.500(96) User Schema for use with LDAPv3, the inetOrgPerson LDAP Object Class and axschema.org.

```
1  <wst:RequestSecurityToken xmlns:wst="...">
2    ..
3    <wst:SecondaryParameters>
4      <saml2p:AttributeQuery>
5        <saml2:Attribute Name="http://axschema.org/company/name">
6        <saml2:Attribute Name="http://axschema.org/contact/email">
7        <saml2:Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/
               identity/claims/privatepersonalidentifier">
8      </saml2p:AttributeQuery>
9    </wst:SecondaryParameters>
10   ..
```

Listing 1. Exemplary SAML Attribute Query

The IdP REST endpoints are implemented with the Jersey framework based on the JAX-RS specification. OAuth-based access control at these REST endpoints is realized by means of the Jersey-OAuth extension[4] and another open-source OAuth library for Java[5]. Query processing and issuing of SAML tokens is implemented with the OpenSAML[6] library while X.509 attribute certificates are constructed with the help of the Bouncy Castle library.

C. IdP and Credentials Discovery

Our proposal allows each user to maintain its security tokens and attributes at a different IdP which is why various IdPs may exist. Thus, the security intermediary has to be able to *discover* users' IdPs and their REST endpoints. IdP discovery is implemented in our prototype via the Yadis protocol which in turn makes use of XRDS capabilities documents. The provided sample XRDS document defines endpoints required by the OAuth-based access control scheme and describes the IdP's STS and auxiliary interfaces. The URI given for the `»oauth.net/core/1.0/endpoint/resource«` service type denotes the actual STS endpoint URL where the security intermediary may retrieve security tokens and attributes in various formats. Other URLs specify the endpoints for retrieving attribute identifiers and security token formats supported by this IdP and a list of attributes accessible to the security intermediary under the current access credentials.

[4]http://java.net/projects/jersey/sources/svn/show/tags/jersey-1.5/jersey/contribs/jersey-oauth

[5]http://oauth.googlecode.com/svn/code/java

[6]http://www.shibboleth.net/downloads/java-opensaml/

Users convey a discovery URL pointing to a XRDS discovery document (see provided example) to the security intermediary within service request messages (*(6)* in Fig. 3) from which the intermediary can learn all relevant data.

VI. Conclusions

In this paper we presented a mediated access control infrastructure that makes dynamic replacement of access restricted services possible. Transparent infrastructure components retrieve and attach security tokens required for access control at dynamically selected services instances. An identity provider enables users to define bundles of attributes and credentials and disclose them for particular purposes and service invocations. Via its access restricted REST interfaces it enables the security intermediary to retrieve users' data as requested and in the format required by the particular dynamically selected service instance.

The overall infrastructure has been prototyped and tested in the European FP7 research project SPIKE. Results show that the presented approach brings forward dynamic service selection even in those settings where service access control is of importance. As the overall concept is built around and extends well-known standards it can rather easily been implemented in current service landscapes.

Acknowledgment

The research leading to these results received funding from the European Community's Seventh Framework Programme under grant agreement no. 217098 and by "Regionale Wettbewerbsfähigkeit und Beschäftigung", Bayern, 2007-2013 (EFRE) as part of the SECBIT project[7].

References

[1] E. van Heck and P. Vervest, "Smart Business Networks: How the Network Wins," *Communications of the ACM*, vol. 50, no. 6, 2007.

[2] E. Bertino, L. D. Martino, F. Paci, and A. Squicciarini, *Security for Web Services and Service-Oriented Architectures*. Springer Berlin, 2010.

[3] B. Iyer, J. Freedman, M. Gaynor, and G. Wyner, "Web Services: Enabling Dynamic Business Networks," *Comm. of the AIS*, vol. 11, 2003.

[4] M.-T. Schmidt, B. Hutchison, P. Lambros, and R. W. Phippen, "The Enterprise Service Bus: Making Service-oriented Architecture Real," *IBM Systems Journal*, vol. 44, no. 4, 2005.

[5] M. P. Papazoglou and W.-J. van den Heuvel, "Service oriented Architectures: Approaches, Technologies and Research Issues," *The VLDB Journal*, vol. 16, no. 3, July 2007.

[6] X. Bai, J. Xie, B. Chen, and S. Xiao, "DRESR: Dynamic Routing in Enterprise Service Bus," in *Proc. of the IEEE Int. Conf. on e-Business Engineering (ICEBE)*, 2007.

[7] http://www.secbit.de

[7] S. Wang, X. Zhu, and H. Zhang, "Web Service Selection in Trustworthy Collaboration Network," in *Proc. of the IEEE 8th Int. Conf. on e-Business Engineering (ICEBE)*, 2011.

[8] S. Luo, B. Xu, K. Sun, Y. Bai, P. Zhang, J. Hu, and Z. Li, "iWeb: A Service-Oriented Web Application Framework with Service Selection over QoS and Context," in *Proc. of the IEEE 8th Int. Conf. on e-Business Engineering (ICEBE)*, 2011.

[9] A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar, "End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo," *IEEE Transactions on Services Computing*, vol. 3, 2010.

[10] S. H. Chang, J. S. Bae, W. Y. Jeon, H. La Jung, and S. D. Kim, "A Practical Framework for Dynamic Composition on Enterprise Service Bus," in *Proc. of the IEEE Int. Conf. on Services Computing (SCC)*, 2007.

[11] R. Mietzner, T. van Lessen, A. Wiese, M. Wieland, and F. Leymann, "Virtualizing Services and Resources with ProBus: The WS-Policy-Aware Service and Resource Bus," in *Proc. of the 7th Int. Conf. on Web Services (ICWS)*, 2009.

[12] C. Fritsch, P. Bednár, and G. Pernul, "DS^{3}I – A Dynamic Semantically Enhanced Service Selection Infrastructure," in *Proc. of the 12th Int. Conf. on E-Commerce and Web Technologies (EC-Web)*, 2011.

[13] K. Furdík, P. Bednár, G. Lukác, and C. Fritsch, "Support of Semantic Interoperability in a Service-Based Business Collaboration Platform," *Scientific Int. J. for Parallel and Distributed Computing – Scalable Computing: Practice and Experience*, vol. 12, no. 3, 2011.

[14] E. Bertino and L. D. Martino, "A Service-oriented Approach to Security - Concepts and Issues," in *Proc. of the 8th Int. Symp. on Autonomous Decentralized Systems (ISADS)*, 2007.

[15] C. Opincaru and G. Gheorghe, "Service Oriented Security Architecture," *Enterprise Modelling and Information Systems Architectures Journal*, vol. 4, no. 1, 2009.

[16] T. Hoellrigl, J. Dinger, and H. Hartenstein, "FedWare: Middleware Services to Cope with Information Consistency in Federated Identity Management," in *Proc. of the 5th Int. Conf. on Availability, Reliability and Security (ARES)*, 2010.

[17] R. N. Hebig, C. Meinel, M. Menzel, I. Thomas, and R. Warschofsky, "A Web Service Architecture for Decentralised Identity- and Attribute-based Access Control," in *Proc. of the 7th IEEE Int. Conf. on Web Services (ICWS)*, 2009.

[18] A. Pretschner and F. Massacci, "Usage Control in Service-Oriented Architectures," in *Proc. of the 4th Int. Conf. on Trust, Privacy and Security in Digital Business (TrustBus)*, 2007.

[19] C. Fritsch and G. Pernul, "Security for Dynamic Service-Oriented eCollaboration - Architectural Alternatives and Proposed Solution," in *Proc. of the 7th Int. Conf. on Trust, Privacy & Security in Digital Business (TrustBus)*, 2010.

[20] G. McGraw, *Software Security: Building Security*. Addison-Wesley Professional, Feb. 2006.