Security for Dynamic Service-Oriented eCollaboration*

Architectural Alternatives and Proposed Solution

Christoph Fritsch and Günther Pernul

Department of Information Systems,
University of Regensburg, 93053 Regensburg, Germany
{christoph.fritsch,guenther.pernul}@wiwi.uni-regensburg.de
http://www-ifs.uni-regensburg.de

Abstract. Current challenges on the markets cause companies to interact with one another and strive after becoming members of virtual organizations assuming that in doing so they can achieve sustainable competitiveness and remain successful despite increased competition. This new openness has strong implications and poses intense demands on organizations' security systems. In this paper we present architectural considerations and our concept of a security infrastructure to cope with these challenges. The presented approach aims at minimizing the lead-time before usage of external services can start by employing a security intermediary for mediation purposes.

Keywords: SOA Security, ESB Access Control, Virtual Organizations.

1 Introduction and Motivation

Today's companies are facing strong challenges and pressure in the markets: the economic crisis demanded maximum flexibility to survive and increased competition due to the globalization requires shorter innovation cycles and continuous improvement of products and value creation processes. More and more companies are realizing that their ways of doing business have to be advanced as they can no longer solely trade as fully self-contained actors. Many of them are beginning to reconsider their entrenched business structures and aim for collaborative value chains and flexible cross company business network structures to perform future business with anybody, anywhere, anytime regardless of underlying information technology infrastructures [13]. Surveys by [3] or [10] back this trend and predict a significant increase of virtual organizations (VOs) and eCollaboration in the forthcoming years.

* The research leading to these results is receiving funding from the European Community's Seventh Framework Programme under grant agreement no. 217098. The content of this publication is the sole responsibility of the authors and in no way represents the view of the European Commission or its services.

S. Katsikas, J. Lopez, and M. Soriano (Eds.): TrustBus 2010, LNCS 6264, pp. 214–226, 2010.

Especially information and communication technology (ICT) has repeatedly been identified as the most critical success factor for efficiently running collaborative projects [7], [16]. The need for tight integration of cross-organizational value chains is still rapidly increasing and organizations' boundaries are becoming more fluid and permeable.

This new openness to speedily establish VOs and the associated rapid but tight integration of IT systems with partnering organizations has strong implications and poses intense demands on organizations' security systems. Flexible security measures and infrastructures to enforce them have to be in place. Security of the company's IT properties has to be guaranteed at all times even if the number and the identities of people authorized to access single services vary frequently and swiftly. The more flexible and rapid an organization wants to join a virtual business network, the more effective and powerful its security infrastructure in general and its access control schemes in particular must be. In this paper we therefore present our fundamental considerations regarding several architectural alternatives for developing a flexible access control infrastructure for networked enterprises, particularly tailored to rapid but still reliable and trustworthy linkage of single services from a pool of service candidates. We assume the following cooperation model: A VO is composed of a set of partnering organizations each offering a set of services that contribute to a single goal common to all collaboration partners. The goal is defined through a business process model in which each task either refers to a service or a human task performed by a collaboration partner. Services are either statically defined (i.e. pre-assigned at collaboration design time) which requires non-negligible start-up efforts such as searching and opting for appropriate service providers out of a pool of several candidates and establish relationships with him by some means or other. Or services are dynamically selected at the latest possible time, i.e. at service invocation time which is roughly sketched in Fig. 1.

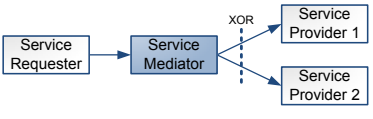


Fig. 1. Ideal service-oriented eCollaboration

For that reason we consider a service broker reasonable and essential. In contrast to previous pure service registry approaches, our service broker is not limited to service registration functionality but introduces an additional layer of indirection and mediation. The service broker can provide standardized interfaces for different kinds of services (e.g. a travel booking services) and service providers (e.g. travel companies) register their service instance for a given type of service. Several authors such as [8], [9] and our initial prototype proved the feasibility of this approach.

The remainder of this paper is organized as follows: In Chapter 2 we present the fundamental security functions for flexible service-oriented VOs from different perspectives. Subsequently, Chapter 3 introduces our approach to tackle the particular security requirements in short-term business networks in more detail and lay out some preliminary implementation considerations in Chapter 4. Chapter 5 provides information on related work and similar approaches before we draw some conclusions and identify future work in the concluding Chapter 6.

2 Security Functions and Implications

We first want to exemplify the basic security requirements our approach is governed by. As a basic non-security principle for our proposed security infrastructure we try to get along with as little need for adaption as possible at both client and service provider side. We aim at relieving both from performing extensive efforts before they can benefit from the newly gained flexibility to rapidly offer existing services to new customers and embed these into their applications.

2.1 Fundamental Security Functions

The flexibility gained from our understanding of VOs strengthens the need for a security infrastructure that is highly reliable and adaptive at the same time. Appropriate security and access control mechanisms in particular are required to ensure that only authorized actors can invoke supporting service while the business process flows from one activity to the next.

In more detail, the following security functions are essential for secure VOs: Usually users first name their claimed identity, termed *identification*. Closely connected is *authentication* during which the system validates the user's claimed identity. Typically both steps precede *access control* which aims at preventing unauthorized use of a resource as well as use of resources in an unauthorized manner. To perform reasonable access control, resource owners first have to specify and allocate access rights to potential users, termed *authorization*. In many cases access rights are directly assigned to user identities but further more elaborate approaches for specifying access rights based on different criteria such as role membership or various attributes of users are well-engineered[18]. These more powerful approaches partially permit *anonymization* or *pseudonymization*, i.e. service usage without disclosing the user's authentic identity. During the interaction of users with resources and services, different *communication security functions* such as encryption and digital signature arise to ensure security aims such as confidentiality and integrity. Last but not least *auditing* allows for recording and reviewing all security-related events.

2.2 Different Perspectives: End User, Service Provider, Broker

Different stakeholders in VOs have different demands concerning security functions. We therefore briefly sketch the perspectives of the most important actors.

End users are usually mainly interested in uncomplicated utilization of required services. If confidential data is transfered communication security functions are inquired, whereas identification is only considered important if it bears advantages such as enhanced or eased functionality due to personalization. Anonymization or pseudonymization might be of interest if users do not want to disclose their authentic identity to access a particular resource.

Service providers mainly focus on authorization, access control and auditing to govern access to their resources and track potential violations. They might be interested in validated identities of their users for accounting purposes and in communication security functions in case they offer confidential information. User anonymization/pseudonymization is usually rather irrelevant.

Service Brokers are trying to please both, users and SPs. Depending on the degree of trust both parties put in one another, service brokers may mediate for example between user's demand for anonymization and service providers' request for authentic user identities for billing purposes.

2.3 Interim Implications

From our definition of dynamic service-oriented eCollaboration can easily be deduced that identity-based authorization and access control does not fit our needs as they imply service providers to know all potential service requesters in advance. If service requesters are permitted access to single services if they can prove their identity – as it is common nowadays – the results are users holding separate accounts at each service provider. Indeed, single-Sign-On (SSO) solutions such as for example Shibboleth[1], OpenID[2] or Cardspace[3] alleviate the problem, still they do not represent fully applicable solutions mainly due to their limitation to services requiring a web browsers as user interface. If services are considered in a broader sense, including modern (SOAP-based) web services as well as legacy applications made available either way, other access control approaches such as role- (RBAC) and attribute-based access control (ABAC)[18] in particular are far more eligible. These approaches introduce an additional layer of indirection between individual users and their access rights and thereby allow for more flexible and dynamic definition of access control policies. More generally spoken, identification becomes less important for the benefit of identity-independent authorization and access control.

3 Approaching Flexible Access Control for VOs

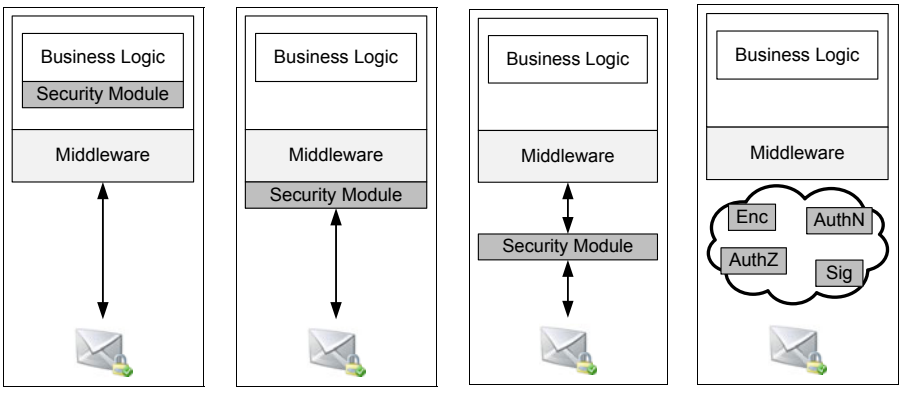
This chapter introduces the system architecture of the SPIKE[4] access control infrastructure and the conceptual model behind in more detail. Our approach allows for dynamic service selection at collaboration run-time and particularly

[1] http://shibboleth.internet2.edu/
[2] http://openid.net/
[3] http://www.microsoft.com/windows/products/winfamily/cardspace
[4] http://www.spike-project.eu/

considers access control as the most relevant security function in dynamic service-oriented VOs.

3.1 Architectural Alternatives

The most popular approaches to embed security modules into distributed systems are depicted in Fig. 2 and briefly outlined below. Typically, security modules have to be implemented on both client- and server-side, yet both parties do not necessarily have to decide on the same architectural alternative.



(a) Embedded into the Application (b) Security in the Middleware (c) Security as Infrastructure (d) Security as a Service

Fig. 2. Architectural Alternatives for Implementing Security

Security embedded into the Application. Fig. 2(a) depicts the concept of security components integrated into each and every application. Developers of services and clients have to extend the business logic of service and client respectively by additional security functionality which has to be invoked explicitly. As a result, security functionality is strongly interlinked with the business logic making both potentially more complex, hard to maintain and hard to test for correctness. In addition, interlinkage of business and security functionality might complicate service usage for potential customers. Developers at client- and service-side have to be expert in both, business logic of the service and security issues. If qualified developers are available this approach can provide superior performance due to the direct API communication between business and security logic.

This architectural alternative imposes further disadvantages such as poor scalability and reusability of both, business and security functionality which applies to service providers as well as to service requesters. Most notably, the security modules of service clients have to be particularly tailored to a single service instance rendering this approach cumbersome for the kind of dynamic eCollaboration we aim for.

Security in the Middleware. In contrast to the embedded security approach, security components as part of the middleware (Fig. 2(b)) allow for clear separation of business and security logic. Considering that in most cases services are not operated stand-alone but are deployed into some runtime environment or middleware, this approach seems plausible without causing additional complexity. The approach enables business experts to take care of new business logic without considering security while security can be added in a second step by security experts, which in turn do not have to crasp every detail of the business logic. Thus the security in the middleware approach enables implicit and configurable integration of security.

While this approach apparently meets the situation at service provider side, it might be different for potential service requesters. Security components in the middleware are only feasible for them if the service client is running within some middleware which might be the case for example if the service interface at client side is integrated into some web application deployed to some application server. Still in many cases external services are integrated into stand-alone applications rendering the security in the middleware approach mainly feasible for service providers, not necessarily for service requesters.

Security as Infrastructure. The Security as infrastructure approach (Fig. 2(c)) is quite similar to security in the middleware. While both allow for separation of business and security logic, this approach goes one step further regarding positioning of the security modules. Instead of providing the security modules as part of the middleware, security nodes are freely deployed between service implementation and service client. Usage of the external security nodes is typically configured at the network routing layer. In comparison to other alternatives, this approach does not only decouple security from the business logic but both purely communicate by means of message exchanges, i.e. the security components intercepts messages from and to particular services and clients. Usually neither service nor client notice the existence of the security modules in between implying that security functions are utilized implicitly making it suitable for both service providers and service requesters.

It must be mentioned that this approach imposes severe security implications if applied faulty. While it might be reasonable to provide services without any security functionality within the own company's frontiers for example for simplicity and performance reasons, it has to be guaranteed that all service communication with external organizations implicitly passes the security node. However, if applied correctly, this approach provides maximum flexibility and great reusability and extensibility. In contrast to security in the middleware it is even applicable to stand-alone services and allows defining appropriate security configurations and security credentials depending on sources and targets of messages.

Security as a Service (SaaS). Security as a Service (Fig. 2(d)) is regularly proposed as the most promising approach for realizing security functions especially in SOAs. The common promise is that neither service client nor service provider have to pay attention to security components but can solely focus on

business logic. Security functionality is provided by distributed security components operated by various providers in the 'cloud' and is implicitly integrated into communication between service requester and service provider. Only rarely a clear usage scenario that includes integration and allocation of distributed security components is explained. In our opinion issues such as sequence of and orchestration control over different security services are not yet considered satisfactory and certain security functions such as encryption or signature can not be provided by external parties in a reasonable way.

As a result we do not consider security as a service – at least in its currently prevalent denotation – a well-engineered and mature architectural alternative. Rather its reasonable and practicable sub-concepts are already known from and implemented in security in the middleware and security as infrastructure approaches.

3.2 Conceptual Model

The SPIKE approach clearly separates between communication security on the one hand and identification, authentication, authorization and access control on the other hand. This work clearly focuses on the latter. To meet the general requirements of minimum initial adjustment efforts shortly addressed in Chapter 2, we aim at establishing as much functionality as possible neither at the client nor at the service but as part of the infrastructure in between. The conceptual model of our proposed flexible access control infrastructure for service-oriented VOs can be derived from Fig. 3 and is based on the security as infrastructure approach. The main actors are users, service providers (SPs), identity providers (IdPs) and the service broker acting as a security intermediary at the same time.

A *user* does not invoke services directly but needs some client application to do so. For the sake of simplicity and because the client application only provides technical means to employ given services, we do no distinguish between both. In our scenario, a user demands a given business functionality and does neither pay attention to the technical realization nor does she pay attention to the chosen service provider as long as it performs reliably.

Service providers aim at attracting as many service users as possible. Therefore, their rationale is not on shielding their services from unknown users but rather on ensuring that the unknown users confirm to their conditions for service usage. As a result, service providers are not primarily interested in the identity of service requesters but in further attributes e.g. for billing purposes. For that reason they may define the access policy for their services based on security tokens or attributes potential users have to hold and exhibit to gain access.

We assume that service users manage their profile at some *identity provider*. The profile or 'digital identity' consists of all attributes, security and access tokens the user holds. This is an established concept proven by several implementations such as Shibboleth, OpenID or Cardspace.

Furthermore, we assume that service requesters try to access new services of previously unacquainted service providers frequently and rapidly which is why an intermediate *service broker* seems reasonable. In addition to its service

selection and mediation functionality, we employ it as a security intermediary. Otherwise dynamic selection of appropriate service instances by a service broker was only possible if all service candidates had the same security policy and therefore required the same security credentials for successful access control. On the contrary, our security intermediary may complement service requests by additional security tokens and mediate between several formats.

The overall service invocation procedure is as follows: For any kind of available service the security broker provides a generic interface for which service providers register their particular implementation. Client applications are built against the service broker interfaces. For invoking a particular service capability, a user dispatches a service request message to the broker. The service broker selects an appropriate service instance from the pool of registered services before the security and access policy of the chosen service is analyzed to extract required 'access tokens' for that service. These tokens are requested from the user's IdP, are attached to the service request which is finally dispatched to the selected service instance. The service instance checks the obtained security token and, based on the result, access is approved or denied.

3.3 Proposed System Architecture

Fig. 3 sketches the proposed SPIKE security architecture which focuses on enabling flexible access control in particular. We positioned security components at five locations distributed across the different parties. Required preparatory work that has to be completed before the access control infrastructure can be employed is narrowed down to basically three preconditions: (1) service users hold their attributes and access tokens at some freely chosen IdP (2) a service broker publishes an interface description for the inquired service (3) one or several service providers registere their service implementation at the service broker.

A service request, constructed in a way to match the service interface offered by the service broker, originates from the user's client application. The outbound security component (No. 1 in Fig. 3) complements the message by information on the user's IdP, i.e its address and an access token to gain access to the user's profile. Potentially further communication security mechanisms as defined in

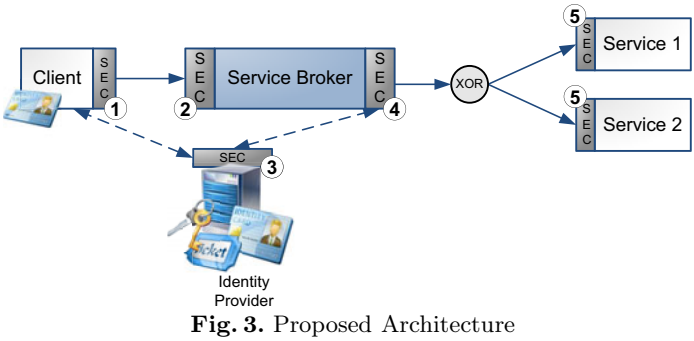


Fig. 3. Proposed Architecture

the service's interface description can likewise be considered by the outbound security module at client side.

The inbound security module at the service broker (No. 2 in Fig. 3), i.e. SPIKE platform side, extracts the information on the user's IdP and the access token from the service request. As a next step, an appropriate service instance for the inquired service functionality is selected from the pool of registered services.

The outbound security module (No. 4 in Fig. 3) retrieves and analyzes the security and access control policy of that service to find out about alternative access tokens or attribute sets that are required to gain access to the selected service instance. It tries to fetch required credentials from the user's IdP leveraging the access token transferred by the user. In case of the user not holding one of the access tokens defined in the security policy of the service, the process is aborted. No. 3 in Fig. 3 depicts the security module at IdP side that protects the user's profile from illegitimate access. As a final step the outbound security module at service broker side enriches the service request by the security tokens received from the user's IdP and forwards the request to the selected service provider. Further security mechanisms such as encryption can likewise be applied here.

Finally, the inbound security module at service provider side (No. 5 in Fig. 3) checks the incoming request for existence and validity of requested security tokens. If the check is successfully, access to the service is granted.

As can be seen from the descriptions above, we built the SPIKE approach regarding separation and distribution of individual security components within the architecture in conformance with the XACML [19] and ISO10181-3 [1] standards. 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. Our current design assumes that the access control decision purely depends on the availability of some security token in the service request. However, depending on the implementation of the SPIKE security infrastructure, the policy decision point (PDP)/access control decision function (ADF) can be performed by the outbound security module of the service broker or the inbound security module of the service instance, respectively. 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.

4 Implementation Considerations

For the evaluation of our security and access control infrastructure we are currently in the process of detailing all individual components and building a prototype which is going to be tested within the SPIKE project. A lively open source community provides numerous individual software components we can reuse and base our implementation upon.

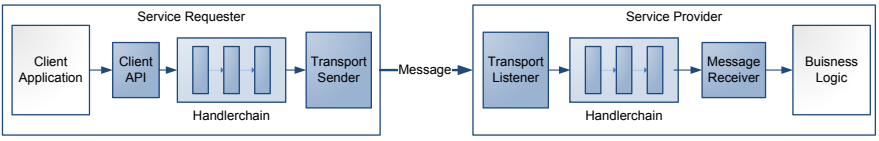


Fig. 4. Client and Service Security Handlers

An Enterprise Service Bus (ESB) offers many message transformation and mediation functionalities required by our service broker component. The JBI specification [22] marked an important step towards a common understanding of the term ESB and currently several rather mature open-source implementations of that standard such as Apache Service Mix[5] or Sun's OpenESB[6] are available. To put the outbound security module at client side and the inbound security module at service side into practice, implementations of the WS-Security standards provide appropriate ground work. Fig. 4 depicts the concept of a chain of security handlers as it is implemented by WS-Security implementations such as Axis2[7] or WSIT[8]. These handlers are configured to intercept the information flow at client and service side and configurably take care of security functionality transparent to the business logic. Last but not least, the concept of in- and out-interceptors in ESBs as depicted in Fig. 5 provides an applicable starting point for implementing required functionality for security components No. 2 and No. 4 in Fig. 3. The Apache CXF binding componentfor JBI-based ESBs employs this concept currently only for SOAP-based web services. From our current point of view, propagation of this concept to other kinds of services should be possible.

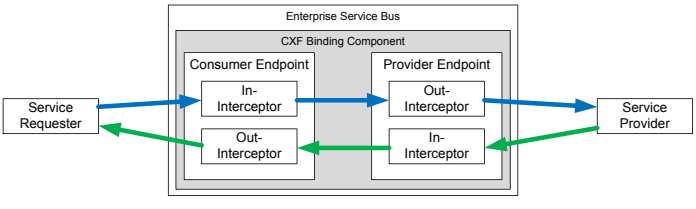


Fig. 5. ESB In- and Out-Interceptors

5 Related Work

VOs and enterprise networks as a dynamic, inter-enterprise configuration for sharing resources and competencies have been identified as a promising alter-

[5] http://servicemix.apache.org/
[6] https://open-esb.dev.java.net/
[7] http://ws.apache.org/axis2/
[8] https://wsit.dev.java.net/

native by several authors such as [13], [10] and [3]. [15] stressed the issue of a flexible software and service selection and sourcing strategy while [17] particularly emphasized the short-term nature of virtual enterprises which conflicts with time being the most important factor in the development of trust between collaboration partners. As a result, novel ideas such as the trust negotiation and authorization approach for VOs by [23] have been developed.

OASIS published several standards such as WS-Security[9], WS-SecurityPolicy[10], WS-Federation[11] and WS-Trust[12]. 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 – holding hundreds of individual services, resulting in thousands of different SOAP messages – a well-engineered but too low-level technical basis for unreflected deployment.

Bertino et al. [6] discuss three essential classes of security services – identity management, authentication and access control – in more detail and propose a service-oriented approach to security. The service-oriented security architecture presented by [20] considers the same services and bears a prototype based on an ESB like our approach.

The FedWare federated identity management middleware service by [14] employs an external IdP as we do but instead of open standards they base their approach on the Sun Java System Identity Manager. In contrast, the distributed access control infrastructure by [5] does not employ an IdP and does not permit user participation regarding transfer and usage of the access tokens. The web service architecture for decentralized identity- and attribute-based access control by [12] considers many of these issues but is particularly tailored to web services while our approach is open for all kinds of services due to mediation capabilities of ESBs. The security credential mapping approach by [2] introduces a concept to mediate between different credential formats such as X.509 certificates, SAML and username tokens and Kerberos tickets, rendering this work an oportune starting point for extending our IdP. Still this work is currently determined to GRID services, only.

Several further authors such as [21] or [4] tackle usage and access control in SOAs and VOs in particular 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 [11] impede broad spreading.

6 Conclusion and Future Work

In this paper we have presented the architectural concept of a security infrastructure for dynamic service-oriented VOs. The presented approach aims at min-

[9] http://www.oasis-open.org/specs/#wssv1.1
[10] http://docs.oasis-open.org/ws-sx/ws-securitypolicy
[11] http://docs.oasis-open.org/wsfed/federation
[12] http://docs.oasis-open.org/ws-sx/ws-trust

imizing the leadtime before usage of external services can start by employing a security intermediary for mediation purposes. We primarily focus on access control but the overall architecture permits implementing further security functions as well. We presented several architectural alternatives and the conceptual model and architecture of our approach in detail and completed this work by preliminary technical considerations towards the implementation of a first prototype. This prototype is then going to be evaluated within the SPIKE project.

Beyond that tentative prototype, current and future work covers detailing and refining several aspects of our approach. Access control to users' security tokens and attributes hosted at the IdP is not yet fully sorted out but an adapted OAuth[13] protocol seems to provide a promising approach. Furthermore, availability and absence of required access tokens might even be considered during the selection phase of an adequate service instance just as other user defined service selection criteria such as service availability, price range, load or other quality of service criteria. Last but not least even users' 'privacy attitude', i.e. which access tokens or which set of attributes are they willing to disclose for a particular type of service, might be considered during the service selection phase and necessitates further research.

References

1. International Organization for Standardization (ISO): Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 3: Access Control. ISO/IEC 10181-3 (1996)
2. Ahsant, M., Gonzalez, E.T., Basney, J.: Security Credential Mapping in Grids. In: Proc. of the 4th International Conference on Availability, Reliability and Security (ARES'09), pp. 481–486 (2009)
3. AT&T: Collaboration across borders: An AT&T survey and white paper in cooperation with the Economist Intelligence Unit (2008)
4. Aziz, B., Arenas, A., Martinelli, F., Matteucci, I., Mori, P.: Controlling Usage in Business Process Workflows through Fine-Grained Security Policies. In: Furnell, S.M., Katsikas, S.K., Lioy, A. (eds.) TrustBus 2008. LNCS, vol. 5185, pp. 100–117. Springer, Heidelberg (2008)
5. Belsis, P., Gritzalis, S., Skourlas, C., Tsoukalas, V.: Design and Implementation of Distributed Access Control Infrastructures for Federations of Autonomous Domains. In: Lambrinoudakis, C., Pernul, G., Tjoa, A.M. (eds.) TrustBus 2007. LNCS, vol. 4657, pp. 125–134. Springer, Heidelberg (2007)
6. Bertino, E., Martino, L.D.: A Service-oriented Approach to Security - Concepts and Issues. In: Proc. of the 8th International Symposium on Autonomous Decentralized Systems (ISADS'07), pp. 7–16 (2007)
7. Broser, C., Fritsch, C., Gmelch, O., Pernul, G., Schillinger, R., Wiesbeck, S.: Analyzing Requirements for Virtual Business Alliances - the Case of SPIKE. In: Proc. of the International ICST Conference on Digital Business, DigiBiz 2009 (2009)
8. Chang, S.H., La, H.J., Bae, J.S., Jeon, W.Y., Kim, S.D.: Design of a Dynamic Composition Handler for ESB-based Services. In: Proc. of the IEEE International Conference on e-Business Engineering (ICEBE '07), pp. 287–294 (2007)

[13] `http://oauth.net/core/1.0a/`

9. D'Mello, D.A., Ananthanarayana, V.S.: Quality Driven Web Service Selection and Ranking. In: Proc. of the 5th International Conference on Information Technology: New Generations (ITNG '08), pp. 1175–1176 (2008)
10. Eid, T.: Gartner Research: Gartner Says Worldwide Web Conference and Team Collaboration Software Markets Will Reach $2.8 Billon in 2010 (2007)
11. Gutiérrez, C., Fernández-Medina, E., Piattini, M.: Web Services Security: Is the Problem Solved? In: Proc. of the 2nd International Workshop on Security In Information Systems (WOSIS 2004), pp. 293–304 (2004)
12. Hebig, R.N., Meinel, C., Menzel, M., Thomas, I., Warschofsky, R.: A Web Service Architecture for Decentralised Identity- and Attribute-based Access Control. In: Proc. of the 7th IEEE International Conference on Web Services (ICWS'09), pp. 551–558 (2009)
13. van Heck, E., Vervest, P.: Smart Business Networks: How the Network Wins. Communications of the ACM 50(6), 28–37 (2007)
14. Hoellrigl, T., Dinger, J., Hartenstein, H.: FedWare: Middleware Services to Cope with Information Consistency in Federated Identity Management. In: Proc. of the 5th International Conference on Availability, Reliability and Security (ARES '10), pp. 228–235 (2010)
15. Iyer, B., Freedman, J., Gaynor, M., Wyner, G.: Web Services: Enabling Dynamic Business Networks. Communications of the AIS 11, 525–554 (2003)
16. Kasper-Fuehrer, E., Ashkanasy, N.: The Interorganisational Virtual Organisation: Defining a Weberian Ideal. International Studies of Management & Organisation 33, 34–64 (2003)
17. Lawson, R., Hol, A., Hall, T.: Challenges of eCollaboration among SMEs. In: Proc. of the 20th Bled eConference: eMergence (2007)
18. Lopez, J., Oppliger, R., Pernul, G.: Authentication and Authorization Infrastructures (AAIs): A Comparative Survey. Computers & Security 23, 578–590 (2004)
19. Moses, T.: eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard (2005)
20. Opincaru, C., Gheorghe, G.: Service Oriented Security Architecture. Enterprise Modelling and Information Systems Architectures Journal 4(1), 39–48 (2009)
21. Pretschner, A., Massacci, F., Hilty, M.: Usage Control in Service-Oriented Architectures. In: Lambrinoudakis, C., Pernul, G., Tjoa, A.M. (eds.) TrustBus 2007. LNCS, vol. 4657, pp. 83–93. Springer, Heidelberg (2007)
22. Ten-Hove, R., Walker, P.: Java Business Integration (JBI) 1.0. Java Specification Request 208 (2005)
23. Winslett, M., Lee, A.J., Perano, K.J.: Trust Negotiation: Authorization for Virtual Organizations. In: Proc. of the 5th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '09). pp. 1–4 (2009)