published as: Michael Menzel, Christian Wolter, Christoph Meinel: Access Control for Cross-Organisational Web Service Composition; In Proceedings of the 2nd International Workshop on Secure Information Systems (SIS 2007); Wisla, Poland, October 2007. pp. 701 - 711, ISSN: 1896-7094;

Journal of Information Assurance and Security 2 (2007) 155–160

Access Control for Cross-Organisational Web Service Composition

Michael Menzel[1] , Christian Wolter[2] and Christoph Meinel[1]
[1]Hasso-Plattner-Institute for IT Systems Engineering
University of Potsdam, Germany
{*michael.menzel, meinel*}*@hpi.uni-potsdam.de*

[2]SAP Research, Vincenz-Priessnitz-Str. 1,
76131 Karlsruhe, Germany
christian.wolter@sap.com

Abstract: Service Oriented Architectures (SOA) promise a flexible approach to utilize distributed capabilities that may be located in independent trust domains. These capabilities can be exposed using Web Service technologies, which provide functionality to describe, discover, and invoke exposed services across organisational boundaries. A broad range of SOA-platforms and toolkits are available focusing on Web Service enabling and orchestration within an organisation. This paper addresses an evaluation and classification of different SOA-platforms and security frameworks regarding secure cross-organisational service invocation. To overcome the revealed limitations of existing frameworks, a two layered security architecture is introduced that satisfies the identified security requirements and abstracts from local access control models to enable secure federated cross-organisational services compositions.

Keywords: SOA, Security, Access Control, Service Federation

1 Introduction

The standard set of Web Service technologies, such as WSDL, UDDI, and SOAP provides the means to describe, locate, and invoke a Web Service as part of a Service Oriented Architecture. These standards, based on XML, remove the dependencies of operating systems and system architectures, allowing a composition of different services. Service composition is one of the success factors of Service Oriented Architectures in order to enable the flexible integration of services provided by independent business partners. The term of service federation reflects that services are composed across organisational boundaries to create additional business value for all members of the federation.

However, the seamless and straightforward integration of cross-organisational federated services conflicts with the need to secure and control the access to provided services. Traditional access control is based on a dedicated model, such as role-based access control, access control lists, or task-based access control (1). In a service federation, however, there might exist multiple models with different semantics or expressiveness for each federation member. In addition, access control decisions in classical models are based on identities and permissions assigned to them. In a federated service the identities and their assigned permissions cannot be exposed to all participants of the federation in order to prevent information leakage of security policies or due to model inconsistencies (i.e. different role concepts, unknown attributes). Also, system evolution, such as new employees are hired or delegation of permissions takes place, must be mediated to all partners. Thus, classical access control architectures provide a non-feasible approach in terms of the general independence of services in a Service Oriented Architecture.

In this paper we propose a Two-Level Access Control (2LAC) architecture for cross-organisational federated service composition independent from local access control models - thus, preventing information leakage and allowing authorization-based cross-organisational service invocation (2). Therefore, we provide:

- A list of access control and authorization requirements for federated composite web service frameworks.
- An evaluation and categorization of existing SOA security frameworks and their capabilities to support cross-organisational federated composite services.
- A proposal of a two layered security architecture that addresses the identified security requirements and abstracts from local access control models in place.

The rest of the paper is organized as follows. The next section provides a list of security requirements for cross-organisational federated service compositions. In Section 3 we will briefly mention related SOA security architectures, such as Amberpoint, Crossvision, PERMIS, or the WS-Federation architecture. Each framework will be evaluated and categorized based on our identified security requirements. In Section 4, we introduce the concepts of our proposed 2LAC architecture for secure federated cross-organisational web service composition based on the authorization-based access control paradigm (2). The last section concludes this paper and gives an overview about the ongoing implementation of 2LAC and outlines future work.

2 Case Study for Federated Access Control

To illustrate the requirements for cross-organisational federated service invocation, consider the example presented in Figure 1, involving three business partners (a manufacturer and two suppliers). The manufacturer produce goods using components delivered by the suppliers. A service that composes the suppliers order services is used by the manufacture's staff to place orders.

Six basic requirements can be derived from the case study to facilitate secure, compliant, and flexible cross-organisational service invocation:

- Independent Access Control Models
  The dynamic nature of SOA demands a flexible and straightforward approach to establish and administrate service federations. Services provided by new business partners have to be integrated seamlessly without severe changes in the security infrastructure.

Received Septem 21, 2007

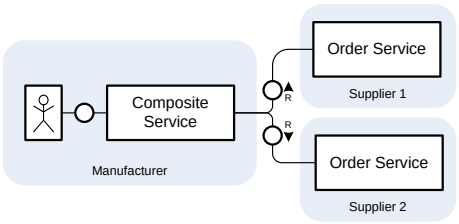

Figure 1. Use Case Service Composition

Especially, the security and access control models used by the respective service providers must be independent from each other. Dictating a common access control model is a non-suitable approach in a federation, hence domain specific access control models might be necessary.

- Service Level Agreements
  To establish a federation with a new business partner, the manufacturer and the new supplier have to agree on IT-level and business process-level service parameters. This agreement affects the access control decision that is also based on environment attributes, such as time and access statistics.
- Information Leakage
  Business roles, information about staff, and the service provider's security infrastructure are sensitive information which should be kept confidential to avoid information leakage. Especially, if the service provider deals with frequently changing business partners, continuously exchanging security policies is inapplicable.
- Dynamic Adaptation of Security Policies
  Once a federation is established, continuous changes relating to permissions assigned within a partner's domain might be necessary. For example changing personal, system evolution, and administrative, or ad-hoc delegation of access permissions (3) might be a reason, but this information must not be mediated to federation members in order to prevent flooding and revocation of authorization information.
- Trusted Service Invocation
  The trustworthy interaction with exposed services is essential for the federated business operations. To avoid business process disruption the access to services must be limited exclusively to authorized staff defined by security policies.
- Compliance
  As part of the Public Company Accounting Reform and Investor Protection Act that is known as SOX (4), auditing and reporting has to facilitate the verifiability that all actions in the federation comply with policy constraints, service level agreements, and law regulations. This is an important aspect regarding the federated delegation of authorizations in order to further guarantee compliance to corporate governance, internal control assessments, and public agencies.

3 Evaluation of Security Framework Architectures

In this section we analyze existing security frameworks with respect to access control for cross-organisational composite web services. Although, these frameworks and platforms focus on entirely different security and SOA aspects, they can be categorized in different groups based on their application to protect a service in a federated environment. We propose a classification depending on the distribution of authentication and authorization information. For each category we present existing example frameworks along with a short description.

3.1 Frameworks for Service Managed Policies

Frameworks based on *Service Managed Policies* facilitate the service to store and handle all information for access control. The identity of the service requester and its role is usually the most important aspect to grant access. Since all this information needs to be maintained for each user who is allowed to access the service, an initial registration of users is required (cf. Figure 2). Beside the exposure of identity related information, the inexistent possibility to delegate authorizations in the user domain is adverse. Furthermore, this approach requires the user to adopt the authentication method specified by the service provider. The interaction between user and service provider will fail if different security infrastructures are used, probably supporting incompatible ways for authentication.

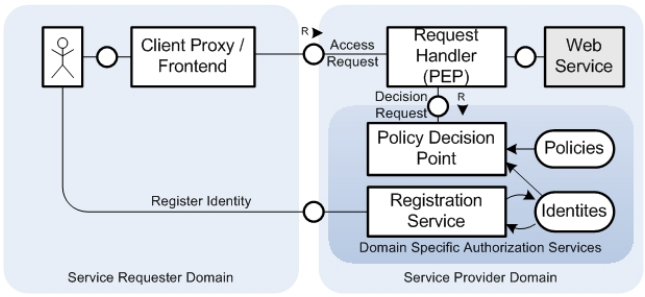

Figure 2. Service Managed Policies

A representative example for *Service Managed Policies* is the Software AG Crossvision Suite (5), which provides a set of tools and components for service enabling, orchestration, and integration in an enterprise environment. One of the central components of Crossvision is the Service Orchestrator that can be used to compose services to a process defined with BPEL [1]. The Service Orchestrator supports message-level security based on WS-Security, and a public key infrastructure (PKI). However, since these components are linked to a predefined java keystore containing the certificates of either authorized users or the issuing certificate authority, only an identity-based access control can be realized.

All aforementioned disadvantages related to service managed policies apply here. The secure invocation of Web Services across domain boundaries requires all participating organisations to deploy a PKI. Furthermore, each external user has to pass his certificate over to the service provider in an initial registration step to enable the verification and authorization of signed requests. Since the authorization decision is tied to the user's identity, access control in a federation is hard to manage in this way. Although a basic secure cross-domain invocation of Web Services is enabled by using a PKI, the general problem remains that such a trust domain cannot interact with a domain that is based on another security solution, such as Kerberos.

3.2 Frameworks for Equal Sharing of Policy Information

Equal sharing means the policy information is maintained by the client along with the service provider. This can be realized based on direct policy exchange, a central federation policy repository, or a dedicated authentication/authorization services (cf. Figure 3). This

[1] Business Process Execution Language

approach simplifies administrative aspects, although information has to be exposed to the service provider. The delegation of authorization is enabled since the user domain has the possibility to adapt security policies. Nevertheless, the establishment of a federation is complicated, due to the necessity to adopt the central security settings for each local infrastructure and domain-specific individual security requirements are hard to be supported by this approach.

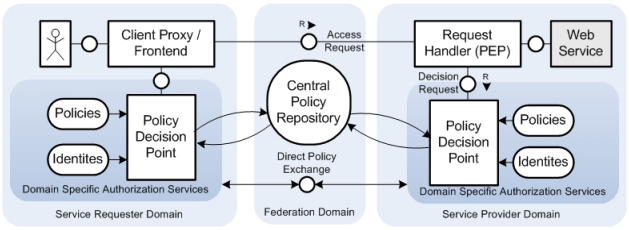

Figure 3. Equal Shared Policies

The Privilege and Role Management Infrastructure Standards Validation (PERMIS) framework is a policy-based authorization infrastructure implementing a hierarchical Role Based Access Control (RBAC) model (6). It is based on a privilege management infrastructure to maintain the user attributes in X.509 attribute certificates (AC), which are published in LDAP directories. These certificates describe the relationship between identities and their privilege attributes and are signed by an Attribute Authority. A central authority defines the access control rules that are translated into an authorization policy and stored as an attribute certificate by an LDAP directory service.

PERMIS has been designed to enable access control within a single organisation. Therefore, it provides an enhanced security functionality, such as an hierarchical RBAC model and delegation of authorities. However, an extension is described in (6) to enable PERMIS to issue SAML authorization tokens. The approach uses PERMIS as a central authorization service in a distributed environment. Although, a centralized solution avoids information leakage and supports security principles, such as Separation of Duties, some issues still remain. All organisations in a federation must use PERMIS as a centralized security solution and have to agree on a common role model.

Another approach is the proposed multipolicy authorization framework for Grid infrastructures by Lang *et al.* in (7). This framework is based on Web Service security specifications, such as XACML and SAML. The authorization mechanisms of the Grid computing platform supports multiple security policies and dynamic policy changes. The access decisions are based on a requester's attributes, such as the service, the resource, or the environment. Therefore, whenever a new identity is added to the service requester's side this identity information must be mediated to the service domain. The general concept of centralized resource and authorization management does not fit into the concept of Service Oriented Architectures. Nevertheless, the GT4 framework utilizes a promising set of security specifications that will be adapted by our proposed architecture.

3.3 Frameworks for User Managed Policies

In the context of *User Managed Policies* the service provider may store some local policies necessary on the provider's side, while the policy information stored at the service requester's domain is independent from the provider's side. This means that the identity of the service users - and therefore the authentication policies - are solely managed and known in the user domain. There are no cross domain policies used - the policies of all organisations in the federation are restricted to the respective security domain. This enables each domain to utilize an own security model independently from others. Asserted user attributes have to be conveyed along with each request to the service (cf. Figure 4). The service will grant access based on these asserted user information if the asserting authority in the user domain is trustworthy.

Although, this approach decouples the security infrastructure used in the different trust domains, a common understanding of the exchanged attributes is still required. For example, the involved organisations may have a different understanding of roles and identities. This would require complicated mapping mechanisms to translate these attributes. Information leakage is still an issue, since identity and role information must be provided to other organisations anyway.

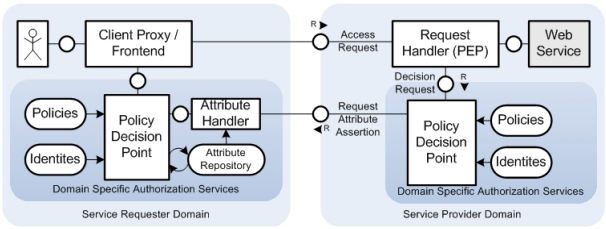

Figure 4. Client Managed policies

The *Amberpoint SOA Management System (ASMS)* (8) is a solution for SOA runtime governance. The ASMS is based on SOAP message interception and forwarding of service requests and responses. It is policy-driven to enforce the requirements concerning authentication, authorization, and integrity. Authentication can be performed by ASMS based on a local identity management system, WS-Security signed messages, or a SAML Authentication Assertion. Based on this information the authorization is performed using basically a simple RBAC model.

Since the supported identity management systems are not designed to be used in a federated environment, SAML is supported to enable secure service invocations across trust domains. An *User Managed Policy* approach can be realized in this way by requiring the authorization to rely on the user's role that is provided by the SAML-token. However, all organisations in a federation must have a common understanding of roles.

A standardized foundation for federation security is provided by the *Web Service Federation language (WS-Federation)* (9) and the *Liberty Alliance* (10) defining frameworks to federate independent trust domains. These specifications provide models for security token exchange to enable the brokering of identity, the discovery and retrieval of attributes, and the provision of security claims. A metadata model to describe and establish a federation is introduced as well (11). Altogether, WS-Federation as well as Liberty Alliance are designed to enable the use of identity attributes across trust domains to facilitate authorization decisions. Since common problems such as the need to map attributes or the provision of pseudonyms are addressed, these specifications provide a standardized model for federations based on user managed policies.

4 2-Level Access Control Architecture

Our classification reveals that each security solution has its own strength, but also specific weaknesses. Even WS-Federation as well as Liberty Alliance presume a common understanding of particular

attributes and claims. Since this standard is focused on the brokering of attributes, it does not fully meet the requirements defined in Section 2. Based on the WS-Trust standard, we propose an architecture that meets these requirements.

4.1 Cross-Organisational Access Control

In the domain of federated services, Kang *et al.* (12) describe a security architecture that proposed decoupled security infrastructures by introducing a shared role domain containing organisation independent roles and their relationships within the federation. Each federation partner maintains its own specific policies and translates it to the predefined shared roles of the federation. Although this approach does not fit into the concept of short living service federations, the basic idea to decouple security infrastructures by separating the local access control model from the cross-organisational one constitutes the foundation for our solution to enable secure cross-domain service invocation.

While the local access control model should depend solely on the local security infrastructure and requirements, the authorization-based access control model (2) is used for cross-organisational access control. The authorization decision is made and asserted in the service requester's domain and attached to the request that invokes the service in a service provider domain. Requesting domain and service domain form a federation which is characterized by a trust relationship. Therefore, requests are only accepted by the service provider if a valid security token from a trusted domain is provided along with each service request. Since the server-side authorization component handles the relation between the domains, service level agreements are enforced by this component according to parameters, such as access time or access statistics.

The user-to-service relation is handled by the user-side authorization component using the local security infrastructure to perform the authentication and the authorization. Since each organisation has its own sphere of local responsibility, we have a two-fold Policy Enforcement Point (PEP) hierarchy. In addition, the identity related policies reside in the domain of the service requester and no policy exchange takes place.

4.2 Service Invocation

A complete abstract of the 2LAC architecture is given in Figure 5. The *Token Handler* is a local component in the user domain that initiates the client-side authentication process and manages the authentication and authorization tokens for the user. These tokens are requested by contacting the Security Token Services (Identity Provider and Authorization Provider), which serve as an interface to the local authentication and authorization mechanisms. Once a valid authorization token is received, the token is conveyed with the request to the *Request Handler*. The *Request Handler* verifies that the request complies with the service level agreements, encrypts the message and invokes the service.

In the service domain the *Request Handler* acts as a second PEP and intercepts incoming requests. Similar to the *Request Handler* in the user domain, requests are decrypted and verified according to service level agreements. It is ensured that a trusted party has authorized this request by validating the security token provided by the requesting domain. Subsequently, the authorization decision is enforced by the *Request Handler*. If access is granted, information about the request such as the authorization token will be stored in the compliance store and the service will be invoked.

4.3 Trust Brokering

Authorization tokens are conveyed with a request and enable an user to access a particular service. These tokens reflect the user's permission to perform an operation on this service and are valid for a predefined period of time. Issued tokens are stored in the user's *Token Store* and are maintained by the *Token Handler*.

For each incoming request, the *Token Handler* tries to obtain a token from the *Token Store*. If a token for a request is not present in the store or has to be validated, the handler will commit an authorization token acquisition for a requested service by contacting the security token services in his domain.

The usage of security tokens enables the delegation of authorizations independent from the local security infrastructure. User can instruct the *Token Handler* to delegate authorizations to other users. This is realized by transferring the authorization token to the other user's *Token Store*. The handler utilize the *Delegation Manager* to enable revocations later on and the PDP to authorize the delegation, since it can be restricted by policy constraints, such as separation of duties.

Each authorization token is created with a unique number to manage and audit the delegation of these tokens. Since the service provider records all incoming authorization tokens in the compliance store, this information will be provided to the user domain if an abuse of authorizations is suspected.

5 Conclusion

The 2LAC Security Architecture enables the secure interaction of independent trust domains in a federation. In contrast to other security frameworks 2LAC decouples the organisational and the cross-organisational access control to allow domain-specific security solutions. All evaluated security solutions can be used in the 2LAC Architecture as local security infrastructure and are encapsulated using WS-Trust interfaces.

Since each organisation has its own sphere of responsibility - one domain handles the trust relationship user-to-organisation, while the other is responsible maintain the trust relationship organisation-to-service - there is no need to exchange user information or to perform an error-prone attribute mapping. Only the support of SAML as the used token type to express authorization claims is presumed. In addition, a flexible delegation of authorizations is enabled independently from the local security infrastructure. Exchanged service level agreements are also enforced. Therefore, 2LAC meets all the requirements specified in Section 2.

Nevertheless, these requirements depend on the trust relationship and the use case. In some cases, the provision of identity information may be required, e.g. identity information has to be provided to a credit card service. If the exposure and the exchange of identity attributes is an essential requirement in an use case than WS-Federation will provide a suitable framework for attribute based security. However, if any kind of information leakage has to be avoided and a clear separation of local and cross-organisational access control is needed, than 2LAC is an appropriate solution.

5.1 Future Work

There is an ongoing implementation effort for the proposed 2LAC architecture. Parts of the security infrastructure, such as the token exchange mechanisms, are already in place. We further plan to evaluate the performance implications of our security infrastructure to the overall service response time. Another aspect is the integration of existing SOA security frameworks, such as Amberpoint or Crossvision,

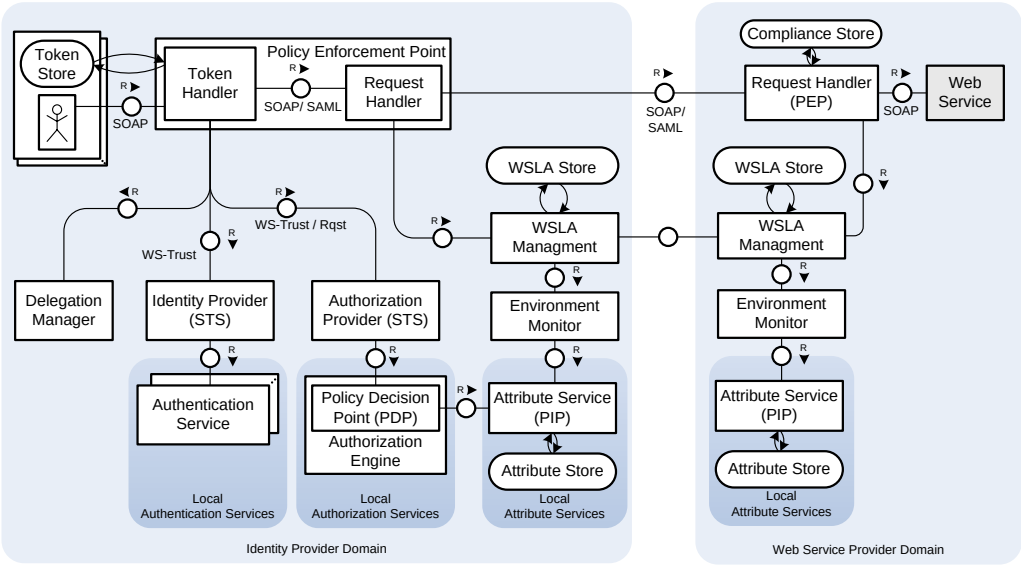

Figure 5. The 2LAC Security Architecture

by defining extension points to attach our proposed authorization-based components to the local authorization infrastructure. A third option would be to translate and apply our architecture into the context of virtual organisations, such as described by the European research projects TrustCom (13) and Serenity (14).

References

[1] C. Wolter, M. Kohler, A. Schaad, "Classification Model for Access Control Constraints", WIA - Proceedings of the 1st International Workshop on Information Assurance, pp. 410-417, 2007.

[2] A. H. Karp, "Authorization-Based Access Control for the Services Oriented Architecture", Proceedings of the Fourth International Conference on Creating, Connecting and Collaborating through Computing (C5'06), pp. 160-167, 2006, IEEE Computer Society.

[3] A. Schaad, J. D. Moffett, "A Framework for Organisational Control Principles", Proceedings of ACSAC, pp. 229-238, 2002.

[4] H. Taylor, *The Joy of SOX: Why Sarbanes-Oxley and Services Oriented Architecture May Be the Best Thing That Ever Happened to You*, John Wiley & Sons, Inc., New York, NY, USA, 2006.

[5] S. Rogers, "CentraSite: An Integrated SOA Registry and Repository", 2006, http://www.idc.com.

[6] D. Chadwick, S. Otenko, V. Welch, "Using SAML to Link the GLOBUS Toolkit to the PERMIS Authorisation Infrastructure", Proceedings of Eighth Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, pp. 251-261, 2005, Springer Boston.

[7] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, T. Freeman, "A Multipolicy Authorization Framework for Grid Security", Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications (NCA '06), pp. 269-272, 2006, IEEE Computer Society.

[8] AmberPoint Software, "Introduction to AmberPoint SOA Management System", June 2006, AmberPoint Software.

[9] A. Nadalin, C. Kaler, et al., "Web Services Federation Language Specification 1.1", 2007, Specification public draft.

[10] Liberty Alliance, "Liberty Alliance Specification", 2007, http://www.projectliberty.org/liberty/specifications__1.

[11] M. Goodner, M. Hondo, M. McIntosh, D. Schmidt, "White Paper: Understanding WS-Federation", 2007, www.ibm.com/developerworks/library/specification/ws-fed/.

[12] M. H. Kang, J. S. Park, J. N. Froscher, "Access Control Mechanisms for Inter-Organizational Workflow", Proceedings of the 6zh ACM Symposium on Access Control Models and Technologies (SACMAT '01), pp. 66-74, 2001.

[13] M. D. Wilson, A. Arenas, L. Schubert, "TrustCom Framework V2", 2006, http://www.eu-trustcom.com.

[14] P. Kostaki, S. Kokolakis, C. Pandolfo, "Serenity - System Engineering for Security & Dependability WP A2.D4.1", 2006, http://www.serenity-project.org.

Author Biographies

Michael Menzel was born in Duisburg, Germany in 1979. He studied computer science at the university of Trier, Germany and received his diploma degree in 2006. Since 2006 he is a Ph.D. in the Research School 'Service-Oriented Systems Engineering' at the Hasso-Plattner-Institute, University of Potsdam. His research work is focused on security in Service-oriented Architectures, in particular identity management, establishment of trust, policy negotiation and access control.

Christian Wolter was born in Berlin, Germany in 1979. He received his bachelor and master degree in computer science from the Hasso-Plattner-Institute at the University of Potsdam in 2005 and 2006 respectively. Since 2006 he is a Ph.D. candidate at SAP Research in Karlsruhe, Germany, in the Department of Security and Trust. His major

fields of study are business process management, business process modelling, security and access control.

Prof. Dr. Christoph Meinel, born in 1954, is CEO and Scientific Director of the Hasso-Plattner-Institute for IT-Systems Engineering at the University of Potsdam. He is full professor for computer science with a chair in Internet Technology and Systems. Meinel studied mathematics and computer science at Humboldt University in Berlin. He passed his doctorate in 1981 and was habilitated in 1988. After visiting positions at University of Paderborn and Max-Planck-Institute for computer science in Saarbrücken, Meinel became a full professor for computer science at the University of Trier. In that time his research focus was on complexity theory and on BDD-based data structures for VLSI design. Later he became interested in Internet research, particularly in Internet and information security as well as in innovative forms of teleteaching. From 1998 to 2002 he was founding director of the ”Institut für Telematik, e.V.” in Trier.