

Web Services Directory based on Peer-to-Peer Technology

Joachim Götze, Markus Hillenbrand, Paul Müller
University of Kaiserslautern
PO Box 3049
D-67653 Kaiserslautern, Germany
{j_goetze, hillenbr, pmueller}@informatik.uni-kl.de

Abstract

Service-oriented architectures foster the interaction of a multitude of different services. In order to ensure flawless service interaction a directory service is needed that can be used to find suitable services for a desired task. A centralized solution for local services might be suitable for small systems, but on a larger scale – when systems start to interact via Federations of Trust – the need for an highly available and easily maintainable directory service arises.

1 Motivation

The directory service of a service-oriented architecture (SOA) is a core component due to the concept of service utilization in SOA. In this concept a directory service stores information about available services and allows the retrieval of this information for clients requesting access to these services. Figure 1 illustrates the concept and shows the importance of directory services that are involved in nearly every service request in a service-oriented architecture.

Although various solutions for directory services exist, there are some common drawbacks that a powerful solution for a highly distributed service-oriented architecture must overcome to be advantageous. Some of these drawbacks are:

- *Centralized service:* Typically a directory service is a centralized service providing information about the services in the local service domain. Because of the centralization this results in a single point of failure if this service is not available. Additionally, during operation a single service can be a bottleneck and account for a severe performance impact.
- *Uptodateness of information:* The quality of a directory service is closely related to the quality of the data it is providing. In common directory services a new

service is added manually and even if it is maintained regularly, this is only possible with a huge effort when the system becomes large. As a result a lot of entries in a directory service might be outdated.

- *Censorship of services:* A directory service for non-commercial services might be maintained without censorship even if there are services included belonging to different service domains. In a business environment most companies would suspect censorship if a directory service is maintained by a direct competitor, because missing core services could result in a weak position on the market and soil the service provider's reputation.
- *Incompatibility amongst different service domains:* Interoperability is one of the major goals of Web Services, but in order to foster interoperability amongst different service domains in a service-oriented architecture a common service directory is to be agreed upon that both parties can use to retrieve service information. If both parties use different directory services, these services might be incompatible and a new directory service might be introduced to support the cooperation resulting in more severe forms of the above explained drawbacks, e.g. maintenance of service information.

In order to handle the above drawbacks a new solution for a directory service has to incorporate various technologies. The directory service proposed here is using Peer-to-Peer (P2P) technology to overcome the problems of different service domains and introduces advantages of P2P networks to directory services like better availability and robustness. Additionally, P2P technology is used to create a decentralized directory service minimizing problems like performance bottlenecks and eliminating a single point of failure in the service-oriented architecture. To proof the applicability of this approach, a prototype has been developed in the project Venice – performed at the University of

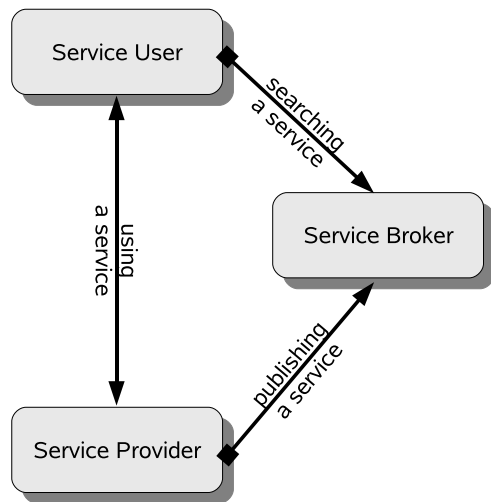


Figure 1. Concept of service utilization in SOA

Kaiserslautern in cooperation with Siemens AG – which focusses on the development of a service-oriented Voice over IP (VoIP) architecture.

In the following, section 2 gives a short introduction into the area of directory services and its related technologies. Section 3 outlines the Venice architecture on an abstract level. In section 4 the project JXTA is briefly explained, because of its major role in the implementation of the concept proposed here. Section 5 describes the concept thoroughly and gives an overview on implementation details. Section 6 concludes the paper with a summary and an outlook.

2 Related Work

Universal Description, Discovery, and Integration (UDDI) [1] is the common solution for a directory service in a service-oriented architecture. A lot of further research has been done to enhance and improve the basic concept of UDDI.

In [2] a Peer-to-Peer-based replacement for the central UDDI registry is introduced. Aiming at the weak points of a UDDI registry – namely scalability and being a potential single point of failure – a solution is provided implementing a decentralized UDDI registry using P2P technology.

Several other solutions like [3, 4, 5] also address replacing a central UDDI directory with P2P technology. [6] additionally uses semantic annotations for service discovery. While adhering to UDDI data types and terminology, they all do not allow for brokering arbitrary data structures like the solution proposed in this paper. Additionally, they use P2P technology directly while the solution proposed here wraps P2P technology using a standard Web service inter-

face.

3 Venice Architecture

In the Venice project different architectural approaches for VoIP are under investigation and tested on prototypes. The main focus lies on the service-oriented approach using Web Services and Peer-to-Peer technology. This section provides an overview of this approach and the resulting architectural design. Details and related work will not be discussed in detail, they have been published previously and can be gleaned in [7, 8, 9, 10, 11, 12]. Basically, the benefits of the architecture compared to existing H.323 or SIP-based solutions or products like Skype [13] are the open and service-oriented approach and the migration from client code into distributed services. This makes it easier to develop and integrate new supplementary services into the existing infrastructure and allows for thin clients running on diverse platforms.

The services offered by a VoIP provider can be classified into three distinct categories: *Management Services* are services necessary for the common usage and maintenance of VoIP services. The *Basic VoIP Services* contain all elementary functions that are responsible for making a phone call with other participants. The category of *Supplementary Services* finally contains all additional services that complement the Basic VoIP Services and contribute comfort before, during or after making a phone call. An overview of this architecture can be seen in figure 2 and will be discussed in the next subsections.

3.1 Management Services

Several Management Services help VoIP providers to actually create a distributed and open Voice over IP infrastructure. Currently, the following services can be identified:

- **Role-based Single Sign-on (SSO).** In order to access all services of a VoIP provider, a token-based single sign-on strategy is the most promising approach [14]. Because there may be several services involved in the process of making a phone call, it is very important to use an authentication strategy based on single sign-on. This strategy allows the user to be authenticated to any service inside the authentication domain without the need to enter his credentials more than once. In order to support dynamic interaction of users the VoIP providers themselves form a federation of trust. This federation of trust allows a user to initiate and receive calls or change the settings of the supplementary services he is using while being connected with a VoIP provider that is not the VoIP provider the user is articulated to by a contract. A more detailed description of

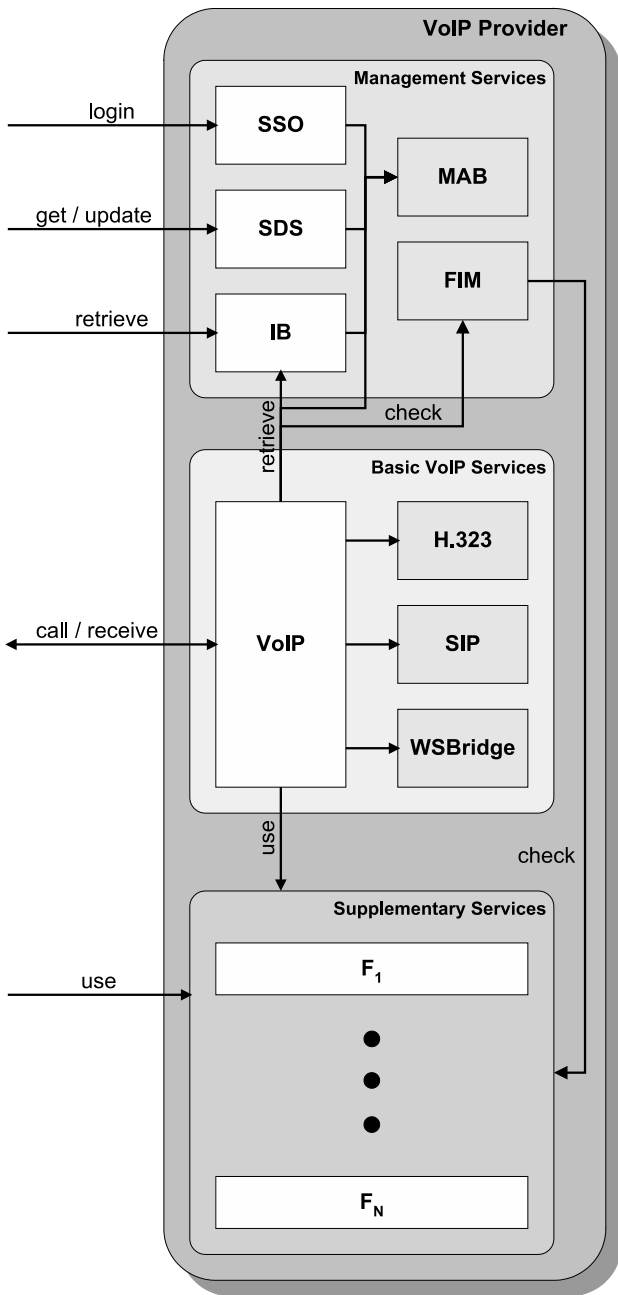


Figure 2. VoIP Architectural Overview

this token-based single sign-on solution can be found in [15].

- Metering, Accounting, and Billing (MAB).** Because of the service-oriented approach used in this architecture, it is possible to use services that provide the same service with different characteristics, e.g. quality of service. All these services do not need to be made available by a single provider, but it is possible to use services of different providers. As a result a multitude of different service providers can be involved in a single phone call. In order to have a reasonable accounting for services used, the VoIP providers need to have a metering system that keeps track of all requested and utilized services in the VoIP system.
- Software Deployment (SDS).** In order to provide an easy-to-use VoIP application to the end-user, it is important to dispense the user from tasks like installing and/or updating software. A Software Deployment Service allows the user to use an up to date application without any further effort. A comprehensive overview of the Software Deployment Service developed can be found in [11].
- Information Brokering (IB).** Because of the potentially huge number of information, it is important to have an Information Brokering Service that is able to support a large and variable amount of information, while also allowing to scale up to a growing quantity if for example new services are introduced. The Information Brokering Service is an extended directory service with the ability to broker additional data structures that go beyond typical service data. The concept proposed here also allows the brokering of additional data as explained later in section 5.
- Feature Interaction Manager (FIM).** This manager component is commissioned to check any request for a supplementary service of the caller or the callee with respect to their usability and applicability. The main focus is to ensure that there is no undesired side-effect because of the consecutive use of multiple supplementary services. Any recognized side-effect is being detected and resolved by using a knowledge base to provide the maximal functionality of the supplementary services requested.

As all of these components are implemented as Web Services, their interfaces [8, 11, 15] have to be well designed and agreed upon. Only this makes interoperability between service providers possible and allows the customer of one provider to talk to a customer of another domain.

3.2 Basic VoIP Services

The actual VoIP Service is the core component of the architecture. This component provides the basic functionality for Call Control, i.e. allows a client to initiate, perform and end a phone call. This implies that this component is directly connected to the client applications and provides abstraction from the underlying telephony technology. A special component using the Information Broker is responsible for finding VoIP users in other domains and routing the phone call through the system to the other provider's VoIP service (i.e. WSBridge in figure 2).

3.3 Supplementary Services

Supplementary services are placed around the core VoIP Service and make it worthwhile for customers to use a specific VoIP provider. They offer functionality that can be combined to satisfy a customer's needs. Typical supplementary services for private and business customers are Call Waiting, Call Forwarding, Call Hold, Three-Party, etc. And there are extensive supplementary services like a call center that can be found in larger companies. Currently, there are several hundred supplementary services available in ISDN and GSM networks – some more, some less known [16, 17].

4 JXTA

The JXTA project started as a research project incubated at Sun Microsystems, Inc. JXTA defines a common set of protocols for building Peer-to-Peer applications. These applications avoid the problem of existing Peer-to-Peer systems of creating incompatible protocols. The main goal of JXTA is the definition of a generic Peer-to-Peer overlay network, which may be used to design and implement a variety of Peer-to-Peer systems. Therefore JXTA offers the means to develop a hybrid overlay network and to orchestrate the deployed applications and services.

Figure 3 shows an example of a possible network topology. In order to distinguish "normal" peers from rendezvous peers, these are typically called edge peers, because of the position at the edge of the overlay network. The rendezvous peers (nodes in dark grey) are located at the center of the network acting as super-peers, while the normal edge peers (nodes in light grey) are connected to their corresponding rendezvous peers.

Peers in the JXTA environment organize themselves in peer groups, which represent a set of peers sharing a common interest and have agreed upon a common set of policies, e.g. a membership policy.

Based on the Resolver Service Protocol, which provides resolution operations such as resolving a peer name into an IP address, defined in the JXTA specifications, the JXTA

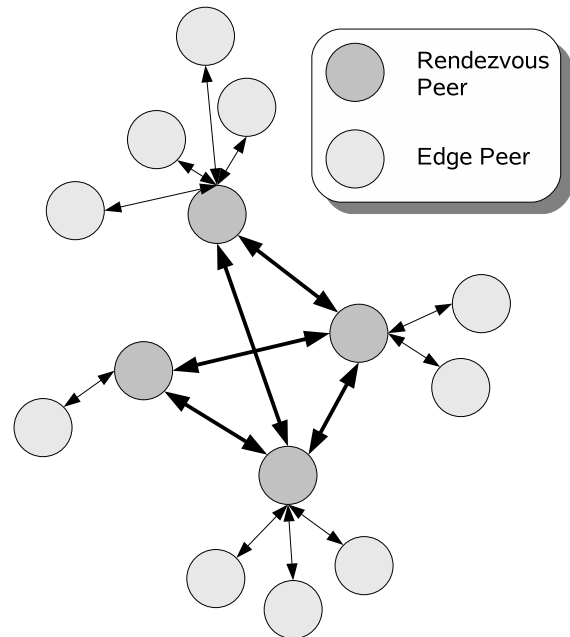


Figure 3. Hybrid P2P-Networktopology of JXTA

overlay network provides a default resolver service based on rendezvous peers. Rendezvous peers are peers that index advertisements to facilitate the discovery of resources in a peer group. Rendezvous peers are defined in the scope of peer groups to reduce the communication complexity. Any peer can potentially become a rendezvous peer, except prohibited due to security considerations.

Rendezvous peers maintain an index of published advertisements using the Shared Resource Distributed Index (SRDI) service. Peers use SRDI to push advertisement indices to their rendezvous peers. The rendezvous/edge peer hierarchy allows resolver queries to be propagated between rendezvous only, which reduces the amount of peers involved in a search operation significantly.

5 Information Brokering in Venice

The Information Broker is a core component of the Venice architecture. In this section the functionality of a directory service and the enhancement to allow for brokering of non-service data will be explained.

5.1 Directory Service

The service-oriented architecture of Venice consists of a large variety of distributed services belonging to a multitude of service providers. Therefore it is important to provide a directory service that enables the service user to

find and interact with a service suitable for the desired task. While a manually updated directory service might contain services currently not available or outdated services that are no longer provided, it is of huge importance to allow a service provider to easily manage the entries of the services provided. In order to support and encourage the interaction between a multitude of service providers a decentralized solution seems appropriate to prevent censorship of service entries by a competing service provider. Additionally, such a solution should be able to allow for an easy addition of entries into the directory service immediately when new services are deployed.

The directory service proposed here is based on a P2P network and therefore able to ensure these requirements. The integration of P2P into the service-oriented architecture is made transparently by providing a Web Service interface for information brokering that a service user accesses to retrieve the desired data. The services themselves do not have to participate in the P2P network, they only have to be registered at the information broker. In the following the advantages resulting from this approach will be explained more thoroughly:

- *Availability of data:* A typical directory service can be a single point of failure and prevent the interaction of services. One of the advantages of P2P networks is the robustness of the network during peer failures. While each service domain can maintain many instances of the information broker in order to get a load balancing effect for the local service domain, these instances also help to ensure the robustness of the P2P-network itself and by this the availability of the service data is provided.
- *Fast information retrieval:* A service-oriented architecture consists of a large number of services interacting with each other. Therefore the performance of information retrieval from the information broker can be a limitation of the architecture. The P2P technology can help to decrease the effort needed to find a suitable service. Typically a service is heavily interacting with local services, e.g. in the Venice architecture there is a heavy interaction between the Basic VoIP Services, the Management Services, and the Supplementary Services. This interaction is supported by the directory service, because the information about local services is stored locally and not in a centralized directory service residing in a remote location. Although the remote use of services is possible and service users are encouraged to use the most suitable service for their requests, this type of service interaction is much less likely and not critical to performance compared to heavily requested services.
- *Transparent usage of P2P:* Today the creation of soft-

ware utilizing P2P technology can be difficult. In order to decrease the difficulty of implementing a new service, the use of P2P is transparent for all services. Therefore the P2P technology is encapsulated within the service broker itself and can be accessed via a common Web Service interface. The transparency also prevents an intermixture of access concepts and protocols, because now every service is accessible via a Web Service interface and only the distributed directory service itself communicates internally via the P2P network.

Another advantage of this approach is that the security of clients behind firewalls is not diminished because of the need to allow access to a large number of communication ports. Clients still stay with their single connection that is needed for Web Service communication, while the P2P-communication is done by the service providers. Even service providers can gain some advantage from this approach, because only the hosts of the information broker need to be accessible via P2P, while all other services can reside on hosts protected by a strict firewall and can still utilize the P2P-functionality.

- *Local registration of services:* During the deployment of a new service, this service is registered at the local domain. This registration also includes the registration at the local information broker. The process is handled by the local domain administrator and the up-to-dateness of the service information is ensured locally for each domain itself. Any domain that participates in this directory service is now able to access this service (any authorization considerations are concern of the Single sign-on service which has been introduced in section 3.1). Any censorship of a single authority managing the directory service is prevented by the P2P concept used within the directory service, which conflates the locally available data of all participating service domains.
- *Set of available services is always up to date:* While the local directory service manages the locally available services, these data should be always up to date, because it is important for the flawless access and use of services in the local domain. The conflation of all local data results in an up to date directory service for all domains.

5.2 Extending a Directory Service to an Information Broker

Although the directory service is the basis for the concept of the Information Broker, it has additional functionality. The information provided by the directory service is therefore not the only information that can be brokered via

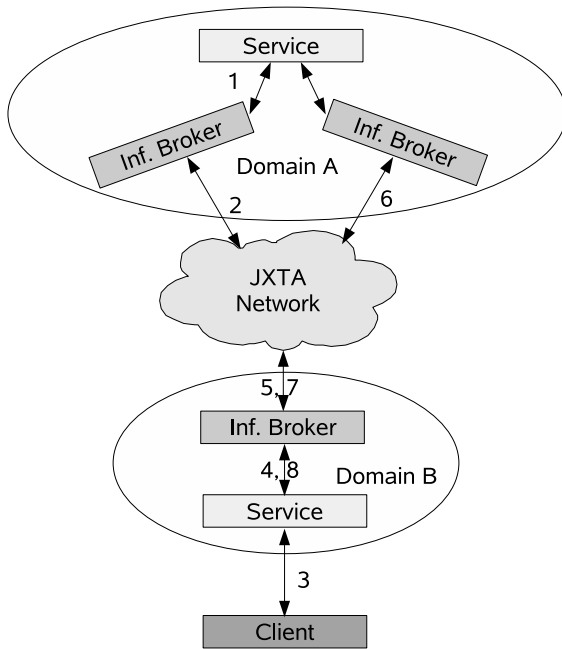


Figure 4. Transparent Usage of P2P-Technology through a Web Service Interface

this service. Based on this concept it is also possible to publish other data. Within the Venice project the Information Broker is not only publishing the typical data provided by a directory service, but also data of active clients. These data may range from basic status information, e.g. the online status of a user, to more sophisticated data like the personal details of the user. These personal details may include textual information, but also other data is possible, e.g. audio files, images, etc. The strategy how these information is provided is similar to the concept how the directory service brokers its information and will be explained in the following section by an example showing the brokering of service information.

5.3 Realization

In order to explain the functionality of the Information Broker in detail, the process of finding a newly deployed service will be illustrated. The order of events includes the initial service deployment, the registration and publication of the service within the network, and the search for this service and retrieval of service information by another service that needs to use the newly deployed service.

The service directory proposed here consists of two main components: the Web Service component, providing a generic interface for searching a service, and the P2P component, built on a JXTA basis. Additionally, a domain specific database is needed where the information about services

can be stored and retrieved. The process of finding a service is separated into multiple steps that will be explained in the following using figure 4:

1. A new service in *domain A* is deployed. In order to make this service available to other services the local *information broker* is requested to publish this service to the P2P network.
2. The local *information broker* is advertising the new service into the JXTA network. From now on the service is available not only locally but also for remote domains like *domain B*.
3. A client is requesting a service for a certain task. It is assumed in this context that the client has the necessary authorization to request this service. Therefore it is not of importance for this example if the client belongs to the local *domain B* (the domain of the requested service) or is currently in another domain.
4. The requested service in *domain B* needs to access another service to complete the requested task. In order to find the other service, the service in *domain B* is initiating the process of information retrieval by sending a request to a local *information broker*. If the requested information is about a local service, the result is immediately returned.
5. In order to handle a request about a remote service (in *domain A*), the *information broker* searches for the corresponding remote *information broker* that is responsible for *domain A*, which is the local domain of the remote service. This search process uses JXTA service advertisements to publish an *information broker* and provide a method to access the data.
6. When a corresponding *information broker* has been found, a bidirectional pipe is created to retrieve the desired information. A bidirectional pipe is a basic communication channel in JXTA that allows to transfer information using an XML document.
7. The returned data is cached locally at the *information broker* of *domain B* due to performance considerations. Typically a large number of service requests are directed to a small number of services. Therefore a small cache can allow for a huge increase in performance. The period of caching such information has been chosen to be short (typically 30 seconds to three minutes) as a tradeoff between uptodateness and performance.
8. The returned data is presented to the requesting service in *domain B* and can be used to access the remote service in *domain A* and provide the service needed by the client.

This example illustrated the process of finding and retrieving service information via the Information Broker. The process is very similar for other types of information that is published and brokered by the Information Broker.

6 Conclusion

Service-oriented architectures consist of various different services and their flawless interaction is of enormous importance for the quality of the provided services. A basis for service interaction is the directory service that stores access information for all available services in a service domain.

The information broker proposed here is a directory service based on P2P technology using JXTA and by this overcomes the typical drawbacks of common directory services like centralization (single point of failure), updateness of information, etc. Beyond the capabilities of a directory service, the information broker allows for brokering arbitrary data structures within the same P2P infrastructure, e.g. the online status or enhanced personal information of users. As a proof of the quality of this concept, a prototype has been developed and successfully brought into operation within the Venice architecture.

In the future some improvements to the prototype are intended. At first the caching operations will be improved to gain an additional increase in performance. Additionally, other P2P frameworks are evaluated and compared with the currently used JXTA, in order to investigate if changing the underlying P2P framework would result in an improvement in quality and performance.

References

- [1] Universal Description, Discovery, and Integration (UDDI), <http://www.uddi.org/>
- [2] Florian Forster, Hermann de Meer: Discovery of Web Services with a P2P Network, Proceedings of International Conference on Computational Science, June 2004, Krakow, Poland
- [3] Discovery von WebServices mit Gnutella, <http://vsis-www.informatik.uni-hamburg.de/~lgregor/>
- [4] Wolfgang Hoschek: A United Peer-to-Peer Database Framework and its Application for Scalable Service Discovery, Proceedings of the Int. IEEE/ACM Workshop on Grid Computing, November 2002, Baltimore, USA
- [5] Ludwig Mittermeier and Roy Oberhauser: Ad-hoc-Web-Services durch P2P-Technologien, in JavaSpektrum 04/2002, SIGS Verlag
- [6] http://lsdis.cs.uga.edu/library/download/VSSPOM_05-MWSDI-JITM.pdf
- [7] Markus Hillenbrand, Joachim Götze, Paul Müller: Voice over IP - Considerations for a Next Generation Architecture, 31st EUROMICRO Conference, September 2005, Porto, Portugal
- [8] Markus Hillenbrand, Ge Zhang: A Web Services Based Framework for Voice over IP, Proceedings of the 30th Euromicro Conference, September 2004, Rennes, France
- [9] Ge Zhang, Markus Hillenbrand: Implementing SIP and H.323 Signaling as Web Services, Proceedings of the 30th Euromicro Conference, September 2004, Rennes, France
- [10] Markus Hillenbrand, Paul Müller, Harald Müller: Voice over IP as Web Service, Proceedings of the 18. DFN Arbeitstagung, July 2004, Düsseldorf, Germany
- [11] Markus Hillenbrand, Paul Müller, Kristian Mihajloski: A Software Deployment Service for Autonomous Computing Environments, Proceedings of IAWTIC 2004, July 2004, Gold Coast, Australia
- [12] Ge Zhang, Markus Hillenbrand, Paul Müller: Facilitating the Interoperability amongst different VoIP Protocols with VoIP Web Services, Proceedings of the 1st International Conference on Distributed Frameworks for Multimedia Applications DFMA, February 2005, Besancon, France
- [13] Skype, Skype - Free Internet Technology, 2005, <http://www.skype.com>
- [14] Jan de Clercq: Single sign-on architectures, Proceedings of Infrastructure Security, International Conference, InfraSec, October 2002, Bristol, UK
- [15] Markus Hillenbrand, Joachim Götze, Jochen Müller, Paul Müller: A Single Sign-On Framework for Web-Services-based Distributed Applications, 8th International Conference on Telecommunications ConTEL, June 2005, Zagreb, Croatia
- [16] Wenyu Jiang, Jonathan Lennox, Sankaran Narayanan, Henning Schulzrinne et al.: Integrating Internet Telephony Services, IEEE Internet Computing, June 2002, page 64–72
- [17] ITU International Telecommunication Union: Recommendation H.450.x: Supplementary Services for H.323, <http://www.itu.int/rec/recommendation.asp?type=products&lang=e&parent=T-REC-H,1998-2001>