Dynamic Service Encapsulation

Alexander Kipp[1], Lutz Schubert[1], and Christian Geuer-Pollmann[2]

[1] HLRS–Höchstleistungsrechenzentrum Universität Stuttgart,
Nobelstraße 19, 70569 Stuttgart, Germany
{kipp,schubert}@hlrs.de
[2] European Microsoft Innovation Center (EMIC) GmbH,
Ritterstrasse 23, 52072 Aachen, Germany
Christian.Geuer-Pollmann@microsoft.com

Abstract. Service Provisioning over the internet using web service specifications becomes more and more difficult as real business requirements start to shape the community. One of the most important aspects relates to dynamic service provisioning: whilst the straight forward web service usage would aim at exposing individual resources according to a fixed description, real organizations would want to expose a flexible description of their complexly aggregated products. This paper presents an approach towards reducing the technological overhead in virtual service exposition over the internet, thus allowing for more flexibility. It therefore introduces a dynamic gateway structure that acts as virtual endpoint to message transactions and can encapsulate complex business process on behalf of the provider.

Keywords: Business communication, Communication standards, Communication system control, Communication system operations and management.

1 Introduction

Today's eBusiness scenarios require a consequent realization of the Service Oriented Architecture (SOA) paradigm. Such a consequent realization provides benefits for both sides, the service providers as well as for the service consumers. Service provider can easily provide their "products" in such a way that potential service consumers can integrate these services in their own products. This is done in an abstract manner which means in particular that no implementation details of the underlying service implementation need to be considered.

Service virtualisation goes even one step further. Here operational, integration and life cycle issues are faced which is critical regarding the success of SOA [1].

Service virtualization has already taken place in our everyday life. An example for such a virtual service is a banking service providing functionality allowing a client to execute financial transactions. Therefore in the background several underlying services are needed, like a transaction manager and a database system. The user of the banking service does not recognize these underlying subsystems since he only sees the interface of the banking service. Via this interface the complexity of the underlying infrastructure is hidden from the current user. Another example is a DNS

D.R. Avresky et al. (Eds.): Cloudcomp 2009, LNICST 34, pp. 217–230, 2010.

or virtual network capabilities. Without virtualization it would not be possible to handle such complex systems at all. Altogether virtualization can be seen as a more abstract view of the corresponding services and the underlying service infrastructure.

In modern eBusiness scenarios it is necessary to decouple service implementations and the corresponding service interfaces. The main reasons therefore are that such a decoupling increases fundamentally the maintainability of services as well as the flexibility of both, service providers and service consumers.

Actually Web services provide an infrastructure towards a SOA paradigm [17] but still have some gaps regarding the needed dynamicity in eBusiness and collaborative working scenarios [16]. An example of the latter one is the research project CoSpaces [2]. This projects aims to develop a framework allowing dynamic collaboration sessions for engineering teams being distributed all over the world. The issues being faced within this project are to bring together the involved people within such a collaborative working session as well as the corresponding applications. So a consequent realization of the SOA paradigm is here also very important. In this paper we provide an approach towards virtual services allowing a decoupling of service implementations from the corresponding service interfaces.

2 eBusiness and Web Services

In current eBusiness scenarios an abstract integration of collaboration partners is one of the main issues to be faced. In particular this means that partners within a collaboration want to consume the provided "product" of a partner without taking into account the corresponding service infrastructures. Web Services provide a first step towards such an approach. Web service technologies allow the consumption of services without the need to take into account the underlying service implementation. This is done by providing a standardized interface of these services (WSDL). These interfaces are integrated in the customers' code allowing him to consume the corresponding services. This interface just describes the functionality of the service in a syntactical manner. To announce a "product" consisting of the composition of several services enforces a more abstract view of the underlying services. One of the main disadvantages of the web service approach is that in the case of a change in a web service interface description the corresponding client code has also to be adapted to these changes.

Therefore abstract entities [3] have been introduced describing such a level of abstraction in a first instance. These abstract entities allow the integration of partners in an eBusiness process by assigning roles to partners and access the corresponding services or products via these abstract entities. This allows the design of collaborative eBusiness scenarios without the burden of taken into account the complexity of the underlying service infrastructures and the corresponding service implementations.

The main goals from an eBusiness perspective are

- The easy encapsulation and usage of services being distributed all over the world
- The easy composition of services in order to provide a "new product"

To realise these goals a new kind of infrastructure is needed with the goal to ease the maintenance of the underlying service infrastructures. In particular, changes of an interface or the service infrastructure should not affect the corresponding client applications. Additionally, service provider should also be able to easily adapt their infrastructures without affecting the corresponding interfaces and consequently the client applications consuming these services. The approach being presented in the following section is also going to ease the provision of new products regarding the currently available services.

3 A Dynamic WS Interface

Currently WSDLs describe a static interoperable interface to a service which is used in static manner. The interface is once proposed and linked in a static manner in the corresponding client code. This static approach does not provide the needed flexibility in a dynamic eBusiness scenario.

To provide such an adaptive and dynamic infrastructure just a contract should be proposed describing the name of this "virtual" service as well as the available operations and what they mean in particular. Additionally it should be mentioned how these operations can be invoked.

Service virtualization provides such an infrastructure by not directly proposing a static interface in the means of WSDL, instead a kind of contract is proposed describing the available functionality and how these services can be invoked as well as which information is needed to invoke these services. The introduced middleware maps in the next step after having intercepted an invocation of such a virtual service endpoint the calls to the corresponding service implementations.

The next sections are going to reflect this new approach in detail.

The New Gateway Architecture

In this section the Architecture of the new gateway is introduced and described in more detail. As mentioned before there is a concrete need in service virtualization and so consequently in an abstraction layer. This abstraction layer operates as an intermediary service between the service consumer and the service implementation by capturing the corresponding messages and mapping them to the corresponding services. This mapping also includes the necessary transformations since the virtualization gateway does not focus on a specific interface description.

Beside the mapping of messages to the corresponding service implementations within the service virtualization layer the following jobs can also be realised within this layer since the gateway describes a single point of entry to use the underlying services. This is preferable since most of the SOA infrastructures are some kind of "grown" nature with the restriction that some already existing implementations may not be compatible with current standards in interface definitions and messaging. So the gateway also provides functionality to encapsulate services.

In particular, this includes:

- *Policy enforcement*: The gateway acts as a policy enforcement point since it allows the definition of criteria that must be fulfilled before a potential service consumer is authorized to access a specific service. For example, it is possible to distinguish service consumers based on their reputation, e.g. in good and "not so" good customers.

Based on their reputation, the customers' requests are forwarded to services with different SLAs, such as "gold" services or "standard" services, where the "gold" rated services e.g. could provide a better quality of service as the "standard" services.

- *Message security, identity and access management*: In an ideal world, all deployed client applications and web services support the corresponding specifications like WS-Security, WS-Trust and WS-Federation. Ideally, each client application should be able to fetch security tokens that are necessary for service access, and each deployed service should be able to authorize an incoming request using a claims-based security model with fine-grained authorization. Unfortunately, many applications in production today do not yet adhere to these principles, and the gateway can serve as a migration path towards broader adoption of the claims-based access model. The customer-side gateway can authenticate internal requestors, request security tokens on their behalf and protect the outgoing messages. A service-side gateway can act as a policy-enforcement point to authenticate and authorize incoming callers. For example the gateway can establish a secure connection to the service consumer while the concrete client application does not support any secure data transmission.
- *Protocol translation*: Since standards in the area of web services are always a matter of change, the reflection of current needs of service consumers as well as of service provider are an essential criterion for such an infrastructure. In particular, the change of an addressing standard like WS-Addressing forces the adaption of the service implementations at the service provider side as well as the corresponding client applications consuming these services. In such a scenario the gateway allows the adaption of the corresponding service calls to the most current standards without affecting the concrete service implementation.
- *Transformation:* Since the gateway provides an universal interface for the underlying services a transformation has to be done before the message is forwarded to the corresponding service.
- *Filtering and information leakage protection:* The gateway can detect and remove private information from a request, offering a hook to install information leakage detection and prevention mechanisms.
- *Load balancing & fail over:* The gateway can act as a load balancer. If e.g. one service is currently heavy in use the gateway may decide to forward requests to this service to an equivalent one.
- *Routing:* If several equivalent services are available the routing of the messages to these services can be handled in this abstraction layer.
- *Login monitoring:* Often it is interesting for a service provider to see which version of a service is still used by the customers. Via the gateway this information is also available.

Figure 1 shows the structure of such a gateway. This structure enables service provider to *encapsulate* and *hide* their infrastructure in a way that also allows for *virtualization of products*. With the gateway being extensible, it provides the basis to non-invasively enact *security, privacy* and *business policies* related to message transactions. With the strong SOA approach pursued by the virtualization gateway, the structure furthermore meets the requirements of *minimal impact* and *maximum deployment flexibility*; through its filters, it furthermore supports the *standardized messaging support*. The gateway is

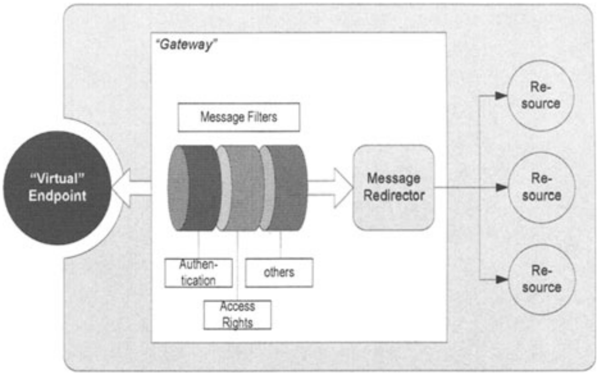

Fig. 1. Gateway Structure

furthermore constructed in a way that allows for participation in *multiple collaborations* at the same time without requiring reconfiguration of the underlying infrastructure.

The gateway of a service provider acts as the virtualization endpoint of the services exposed by the respective organization. Its main task consists in intercepting incoming and outgoing messages to enforce a series of policies related to access right restrictions, secure authentication etc. (cp. Figure 2) thus ensuring that both provider and collaboration specific policies are maintained in transactions.

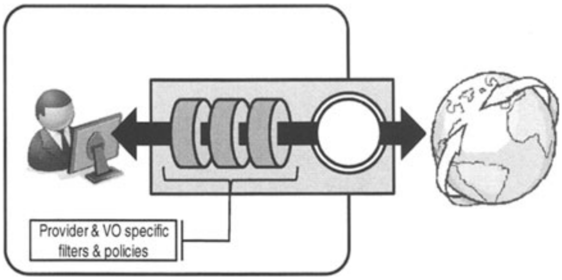

Fig. 2. The Gateway Principle

As a virtual endpoint, the gateway is capable of redirecting messages from virtual addresses to actual, physical locations (local or in the Internet), thus simplifying endpoint management from client side, i.e. applications / client services are not affected by changes in the collaboration, such as replacement of providers etc. An intrinsic handler in the gateway channel hence consists in an endpoint resolution mechanism that identifies the actual destination of a given message.

Figure 3 shows the conceptual overview of such an approach. In particular, the virtualization manager of a service provider announces a virtual service interface definition (WSDL). This virtual interface is also announced by the web server of the service provider to receive external service calls via the included virtual methods. These calls to the virtual interface are forwarded to the virtualization manager. In the following proceeding the virtualization manager transforms the incoming virtual message to a message that can be interpreted by the corresponding service implementation. Therefore the virtualization manager accesses a knowledge base containing all the necessary information like e.g. the mapping of the virtual name to a concrete service endpoint and the transformation of method names and parameters. The mapping of virtual service names to concrete service endpoints is also needed in the case when several service implementations on e.g. different machines hosting the same service are available as well as to avoid the client to take into account concrete service implementation aspects.

Via the knowledge base it is also possible to provide services dynamically. On the one hand new services can be announced via a new virtual interface. On the other hand it is also possible to develop new services for already announced virtual interfaces and map the calls from the old virtual interface to the new service implementations. So the mapping logic is encapsulated in the knowledge base providing the information needed to transform the corresponding message calls.

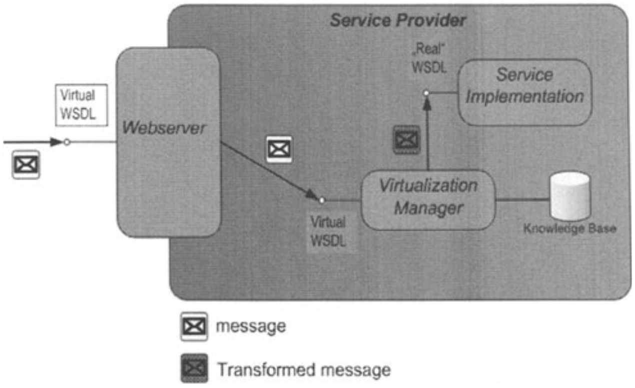

Fig. 3. General Architecture

Realisation of the New Gateway

Referring to the statistic of used web servers within the internet of April 2009 [8] there are most commonly used 2 web service infrastructures in current environments. In particular, those are the Apache Tomcat server with a contingent of 45.95% and Microsoft Internet Information Service (IIS) with a contingent of 29.97%. The remaining 24.06% are distributed over more than 30 other web server solutions, so they are not being taken into account for the following technical analysis considering in how far the service virtualization manager can be realized with existing and mainly used web services infrastructure solutions.

In the following it will be shown, how such a service virtualization manager can be realized with the mostly used web server solutions, namely the Apache Tomcat Server with AXIS and the IIS with WCF [11].

To provide a service virtualization manager, an ideally transparent intermediary service is needed acting as a message interceptor und as a message transformer. In particular, in the area of web services a HTTP router is needed doing this transformation without affecting the client calling the corresponding service as well as the underlying service implementation. Figure 4 illustrates an example of this processing:

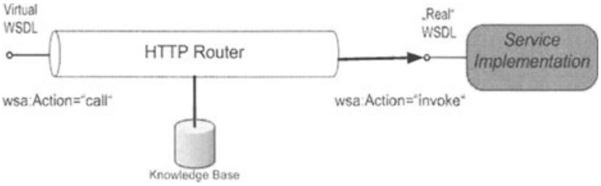

Fig. 4. Technical Realization

In particular, the HTTP router tunnels a request from a virtual WSDL to a concrete service call of a "real" service interface. Therefore in this example the virtual WSDL provides a method with the name "wsa:Action='call'. The HTTP router now maps this web service call to the corresponding "wsa:Action='invoke'" method call of the underlying service implementation. This is done completely transparent to the invoking client as well as to the service implementation.

Within the IIS / WCF realization the gateway infrastructure exposes virtual endpoints (URLs) similar to the (IIS) and may even be hosted inside the IIS like a simple service. The service administrator uses the capabilities of the virtualization gateway / IIS to decide which resources / services / workflows are exposed under which URL – all other services either remain hidden in the infrastructure or are exposed without a virtualization gateway intermediary. This way, the administrator can specify concretely which services are exposed in which manner (cp. Figure 5).

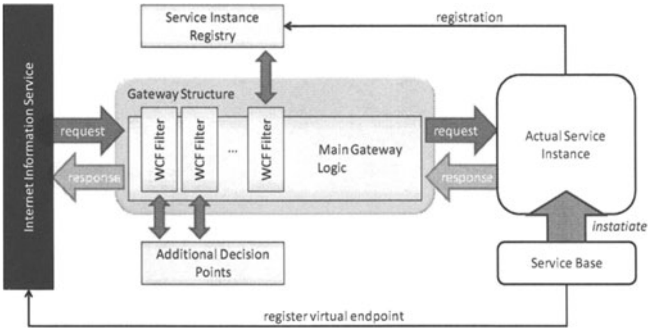

Fig. 5. Gateway Structure and its Relationship to IIS and Service Instances

Policy handlers can be registered at the virtualization gateway using the according management interface and the identifier of the specific gateway structure. Each service instance can thus principally be associated with its own gateway and policy handler chain, allowing for maximum flexibility.

The Service Instance Registry is a specific type of policy decision point that identifies the actual endpoints on basis of the transaction metadata (sender, related VO, addressed endpoint etc.). It will instruct the message chain about the next endpoint to forward the message to.

Axis [12] provides with the Handler concept an approach that allows to plug-in applications between the web server and the corresponding application services.

Therefore so called handler-chains can be realized describing a list of operations that can be executed on arriving messages for a specific service or for all web services being hosted on the corresponding web server. Figure 6 shows the general overview of the Axis architecture:

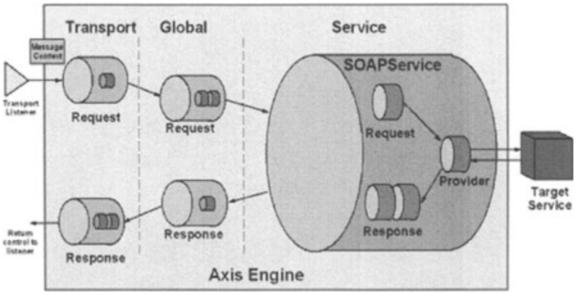

Fig. 6. Axis Engine Overview [12]

In particular, incoming messages are stored in a request queue. Before these messages are processed and forwarded to the corresponding service implementation the handlers being defined for this service are executed. These handlers are able to modify the incoming and outgoing messages, so at this point it is possible to plug in the knowledge support doing the mapping and the necessary transformations of the corresponding messages.

4 Trust Management

In distributed-system scenarios, the main security problem is cross-organizational authorization. Most identity and access systems available today provide flexible solutions for authorization-related problems within the boundaries of a single organization. Still, IT professionals who need security solutions for cross-organizational collaboration typically need to develop their own custom solutions.

The BREIN project extends the security work done in former projects, such as TrustCoM, MOSQUITO [6], NextGRID [5] or MYCAREVENT [7]. The security research in these projects addressed problems such as human-supported federation establishment and enactment, VO-centric identity and claims management, and authorization for cross-organizational service invocation. While that led to many insights into the VO security area, the BREIN project identified a couple of issues that needed further research: One open question is how to leverage the human user for context provisioning, such as why a particular service interaction happens, and subsequently utilizing that context for security decisions. The second broader issue for which a solution is needed is the access management for resources located outside of the data owner's organizational trust boundary. The third topic is related to the support for claims-based security in protocols that do not support WS-Security, such as MTOM-based streaming.

In the BREIN architecture, security-related implementation artifacts are located at various places and layers, so that BREIN can scale the flexibility of the solution depending on the concrete security requirements of the respective scenario. For example, it is clear that cross-organizational message exchanges always have to be integrity and confidentiality protected, and that the requestor needs to be authenticated and authorized. Depending on the capabilities and features of the web services stacks of both clients and application services, either the end-nodes take care of handling the cross-organizational security themselves, or big parts of that responsibility are factored into infrastructure components such as the gateway service. For example, if a web services-based client application cannot encrypt and sign SOAP messages using the appropriate cross-organizational security tokens, then that responsibility has to be handled by the gateway service which is sitting in the message path, on behalf of the client.

The Security Token Service (STS) issues claim-based tokens to authenticated users (or a gateway acting on behalf of the user) and is also involved in the process of establishing federations with other STSs'. Similarly to the gateway, the STS component needs to be installed within the security domains of the entities that want to communication and depending on the role they hold they perform different functions. Therefore the STS can play both the role of the client side STS as well as the server side STS performing different functions. The client-side role of the security token service issues tokens that are necessary to pass the access check on the service side. The tokens are generated based on the information that is extracted from the service call message. The Service-side role of the security token service performs an authorization decision on the ultimate service and issues a security token that will be understood by the service. It hence has the role of a policy decision point (PDP). The STS is a middleware component and is configured using its policy store. The policy store contains both the attribute information about clients in the own organization (i.e. the claims that can be issued), the capabilities of partner organizations (i.e. claims that the STS accepts from other issuers), and access policy for local resources, such as web services:

- User attributes and claims can be stored either within the STS' own configuration, or in external attribute stores such as Active Directory.
- The trust relationships with partner organizations describe e.g. which roles a partner company assumes in a given virtual organization, i.e. which statements and claims the partner is authorized to issue. Essentially this is similar to SecPAL's 'can-say' verb.
- The access policy for local services describes claim requirements for local services, i.e. which claims need to be present in the client's security token to access a particular service.

The STS will be queried for security token issuing by the *security handler*. This handler resides inside the Gateways and protects message that is about to be sent, and requests access control decisions for incoming messages.

The STS is implemented using .NET and WCF. The interaction will be through WS-* message. Most likely the component needs a network connection, although it could (theoretically) also communicate by local inter-process communication like named pipes. The WCF-based client-side security handler is implemented as a special SIR binding, which fetches the routing, security and binding information from the local SIR, creates the 'real' cross-organizational binding based on that endpoint information, and dispatches the message though this cross-organizational binding.

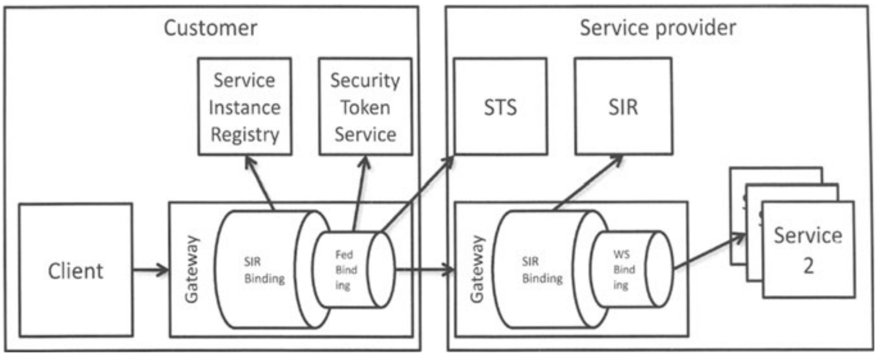

Fig. 7. SIR Binding Interactions

5 Brave New World

In order to evaluate the conceptual approach of the introduced virtualization infrastructure the WCF gateway prototype integrated within the Integrated Projects (IP) CoSpaces and BREIN [4], considering different of the mentioned benefits of such a virtualization infrastructure.

The IP CoSpaces is facing the challenging task in providing an infrastructure allowing for the support of collaboration of worldwide distributed engineering teams. Therefore CoSpaces aims to develop a framework that supports dynamic, ad-hoc collaborative working sessions [14]. This infrastructure stresses, beside the consideration of dynamic aspects within collaborations, security issues to be of the utmost priority and importance. Since security aspects usually affects every involved component within such a collaboration session, a new approach has been considered to allow application developers as well as collaboration participants to concentrate on their original tasks, e.g. the provision of a specific functionality within an application or the solving of a specific problem within a collaboration, without having to consider security aspects whilst being involved in a collaboration.

Since within collaborations between industrial partners often beside services also business critical data has to be shared, authentication, authorization and secure communication between participants has been determined as one of the most critical aspects that need to be considered by such a framework. Within CoSpaces Shibboleth has been chosen as the best suitable solution for providing an authentication infrastructure for authorization issues whilst considering dynamic aspects of such collaborations [15]. Therefore, the virtualization approach being presented within this paper is going to be used to transparently integrate an authentication and authorization infrastructure within the entire framework without affecting the underlying steering and coordination infrastructure components as well as the corresponding shared services and data. Consequently, the users as well as the application providers do not have to consider security aspects within their tasks whilst the framework ensures that only foreseen partners are allowed to access the corresponding services and data sets.

The IP BREIN faces the challenge that in today's world, enterprises, independent of their size, have to cooperate closely with other companies to keep their competitiveness, as no company is capable of fulfilling all requirements alone. But setting up these collaborations is still difficult and extremely costly. Especially for SMEs these collaborations are not really cost-efficient, as they have to put in high efforts to be able to compete on the market with other players. Therefore BREIN will enable service providers to reduce costs whilst maximizing profit by providing a framework that will automatically adapt to changes in individual business needs and/or environment in an intelligent manner. Cost and effort for service provisioning will be greatly reduced by simplifying business goal definition, intelligent optimization and decision making support. Therefore, BREIN is going to support the integration of "virtual" resources in workflows in order to achieve a higher degree of flexibility. This approach allows for both, an easier and more abstract usage of resources (e.g. a customer just invokes a "simulation" service without considering technical details) as well as an increased support of dynamism in such environments by easing the replacement of service providers (e.g. the customer still invokes a "simulation" service whilst his own company gateway redirects this request to a new service provider).

The "classical" WSDL approach would affect in such a dynamic environment that every client of a specific service provider has to adapt their applications to new service interfaces in case of any modification of the corresponding service provider infrastructure or in the case of a service provider change. Additionally a lot of added effort has to be spent for the corresponding service setup. But with the new gateway the client does not need to update his code, although the syntactical interface may have changed, since the messages of the calls via the old interface are mapped or, if needed, transformed, to the interface of the new service.

With this gateway the service provider is now able to implement any adoption needed, even regarding changes in inter-communication standards. Now it is possible to provide several interfaces for the same service, each adapting to another interface. E.g. one customer needs a secure connection to the service because sensible data has to be transferred while another one uses another version of WS-Addressing [9] or WS-Security [10].

eBusiness and the New Gateway

This approach introduces a new abstraction layer for SOAs facing the needs of eBusiness environments. In particular the main benefit is an *increment of flexibility*: Both, for the technical as well as for the business perspective, flexibility has been increased. From a technical point of view it is now possible to bind services statically in application codes while the corresponding service implementation can be migrated. Additionally the service provider can announce the available services independently from the protocol the potential service consumer are going to use. This way of announcing services allows the service provider to use and re-use already existing services in a very easy way. Beside this, the composition of services in a workflow has also been improved: Depending on the target outcome of a workflow services can now be combined regarding the announced contract. The service provider is consequently able to provide "new" products depending on the currently available resources, services and their current payload.

Resulting from this increase of flexibility, the main benefits of this approach are

- Increased customers satisfaction: service providers are now able to adapt very fast to different customers' needs.

- Easy and improved maintenance of provided services
- Efficient development since the customers' technical point of view does not need to be considered within a concrete service implementation.
- Easy adaptation of provided services to changing web standards. Since web standards in the area of security, addressing, reliable message transfer, etc. are continuously under development and improvement, the corresponding service provider has to support as most of these standards as possible.
- Decreased costs
- loose coupling can be better realised with such an approach
- Monitoring and logging in abstraction layer: enables the administrator to see which versions of a specific service are mostly used
- Governance guidelines force the realisation of specific functionality which is often not conforming with the current service realisation. The presented approach can realise this requirement without affecting the service implementation.
- Service consumer may use different end user systems to consume the corresponding services
- Many "grown" SOA infrastructures available are already existing and need to be integrated. This can be realised with an extremely reduced effort with the presented approach.

6 Conclusions

In this paper we presented an approach towards a "real" SOA paradigm and how this can only be realized with a corresponding support of a service virtualization infrastructure. We also presented a conceptual approach to realize this service virtualization taking into account the already existing, partly grown SOA realization with web service technologies. Finally we presented how this concept can be realized in principle taking into account the most common used web services infrastructures. The latter presentation showed that the current available concepts of these web service infrastructure implementations allows an adaptation of the "intelligence" of a service virtualization infrastructure in the sense that the corresponding knowledge support can be added in such a way that incoming messages of a virtual service definition can be mapped to a concrete service implementations.

Actually a first prototype of the WCF approach is available and in the testing phase within CoSpaces and BREIN. This first prototype actually allows the mapping of virtual EPRs to concrete EPRs including enhancements regarding security, policy enforcement, etc. The mentioned plug-in approach makes the introduced concept quite flexible regarding new requirements. A first prototype supporting the Shibboleth infrastructure is also be available. Additionally, the AXIS gateway is currently under development and will be available soon allowing a comparison of these two realizations.

We strongly believe in the success of SOA. The presented approach describes a necessary step towards an entire, SOA enabled infrastructure.

Acknowledgements

The results presented in this paper are partially funded by the European Commission under contract IST-5-034245 through the project CoSpaces as well as through the project BREIN under contract number IST-034556. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper. The authors want to thank all who contributed to this paper, especially all members of the corresponding consortiums.

References

[1] Nash, A.: Service Virtualization – Key to Managing Change in SOA (01.06.2006), `http://www.bitpipe.com/detail/RES/1130171201_512.html` (30.04.2009)

[2] CoSpaces – EU IST Project (IST-5-034245), `http://www.cospaces.org` (30.04. 2009)

[3] TrustCoM – EU IST Project (IST-2003-01945), `http://www.eu-trustcom.com` (30.04.2009)

[4] BREIN - EU IST Project (IST- 034556), `http://www.gridsforbusiness.eu` (30.04.2009)

[5] NextGRID - EU IST Project, `http://www.nextgrid.eu/` (30.04.2009)

[6] MOSQUITO - EU IST Project (IST-004636), `http://www.mosquito-online.org/` (30.04.2009)

[7] MYCAREVENT - EU IST Project (IST-04402), `http://www.mycarevent.com/` (30.04.2009)

[8] Netcraft –Web server statistic (April 2009), `http://news.netcraft.com/archives/2009/04/` (30.04.2009)

[9] Box, D., et.al.: WS-Addressing (10.08.2004), `http://www.w3.org/Submission/ws-addressing/` (30.04.2008)

[10] Nadalin, A., Kaler, C., Monzilo, R., Hallam, Baker, P.: WS-Security (01.02.2006), `http://www.oasisopen.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf` (30.04.2009)

[11] WCF – Windows Communication Foundation, `http://msdn.microsoft.com/wcf/` (30.04.2009)

[12] Axis Architecture Guide, `http://ws.apache.org/axis/java/architecture-guide.html` (30.04.2009)

[13] Schubert, L., Kipp, A., Wesner, S.: From Internet to Cross-Organisational Networking. In: Proceedings of the 15th ISPE International Conference on Concurrent Engineering: CE 2008, Belfast, Northern Ireland (August 2008)

[14] Kipp, A., Schubert, L., Assel, M.: Supporting Dynamism and Security in Ad-Hoc Collaborative Working Environments. In: Proceedings of the 12th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2008), Orlando, USA (July 2008)

[15] Assel, M., Kipp, M.A.: A Secure Infrastructure for Dynamic Collaborative Working Environments. In: Proceedings of the International Conference on Grid Computing and Applications 2007, Las Vegas, USA (June 2007)

[16] Schubert, L., Wesner, S., Dimitrakos, T.: Secure and Dynamic Virtual Organizations for Business. In: Cunningham, P., Cunningham, M. (eds.) Innovation and the Knowledge Economy: Issues, Applications, Case Studies, pp. 1201–1208. IOS Press, Amsterdam (2005)

[17] Golby, D., Wilson, M.D., Schubert, L., Geuer-Pollmann, C.: An assured environment for collaborative engineering using web services. In: CE 2006 (2006)

[18] Wesner, S., Schubert, L., Dimitrakos, T.: Dynamic Virtual Organisations in Engineering. In: 2nd Russian-German Advanced Research Workshop on Computational Science and High Performance Computing, March 14-16 (2005)