Requirements for Policies in Cross-Domain Services Composition*

Ulrich Pinsdorf
Microsoft

Jan Schallabck
ULD

Stuart Short
SAP

1 Cross-Domain Service Composition

SOA is a technology-independent architecture concept adhering to the principle of service-orientation. It aims at enabling the development and usage of applications that are built by combining autonomous, interoperable, discoverable, and potentially reusable services. These services jointly fulfill a higher-level operation through communication. One core principle of SOA is the so-called loose coupling of partial services: Single services are not permanently bound to each others, but their binding happens only at run-time enabling a dynamic composition of services. Moreover, it is even feasible to dynamically bind services hosted in different security domains and by different legal entities ("cross-domain service composition"). One prominent example for this are services rendered via so-called "service chains" that comprise of several partial services offered by different organizations. To facilitate the use of such services, usually one legal entity might serve as single point of contact for (potential) customers. In times of the Internet, places of business of organizations providing partial services for one high-level service can be widely distributed around the globe.

2 Example: Electronic Job Search

The scenario of an electronic job search portal may serve here as an example for a cross-domain service composition (cf. Fig. 1). The user submits her CV in an electronic form to the eCV portal. The electronic CV is accompanied by a sticky policy defining the access control and/or data handling policy for the CV document. It may state that the electronic curriculum vita document a) may only be stored in Europe, b) that any entity passing on the document has to notify the user by email, and c) the document has to be deleted after 60 days. In our scenario a temping agency is trying to fill position for two employers, one located in Germany the other in UK. At some point in time the

*The research leading to the results presented in this paper has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) for PrimeLife project.

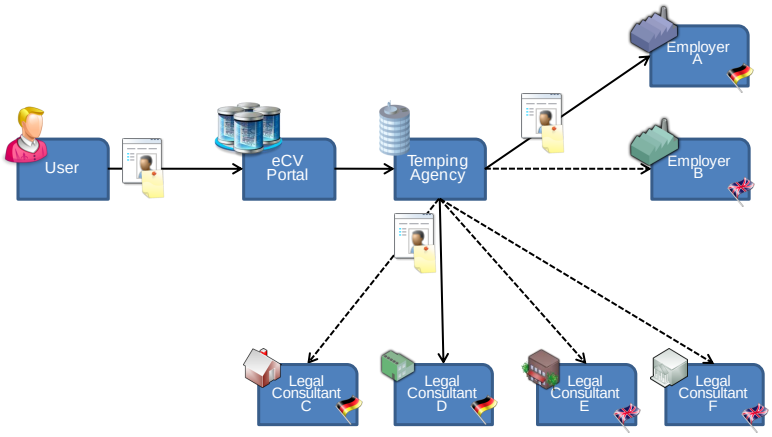

Figure 1: Example SOA for electronic job application

temping agency retrieves the electronic CV from the portal and passes it on to a legal consultant. The legal consultant is dynamically bound, based on two facts: the context and the policy of the legal consultant. In this example the context is simply the residence of the employer; if the user fits to the profile of the German employer the legal consultant should also be knowledgeable in German Law, while it would be vice versa for the UK employer. In the example depicted in Figure 1 the Temping agency may still choose between two legal consultants that both fit in the context. Next, the temping agency would pick the service which best adheres to the user's data handling policy. We assume that each service exposes it's data handling policy to the caller. The calling service (here the temping agency) can identify the service which is able to fulfill the user's requirements.

3 Policy Requirements

In this section we formulate and discuss a number of requirements for access control and privacy policies in service chains. These requirements are taken from Meissner and Schallaböck (2009), a deliverable of the PrimeLife project.[1] It features a list of 39 requirements covering also aspects such as logging and core policy requirements and builds on earlier research done in Bizer et al. (2007). We focus here only on the aspects of policies in cross-domain service composition.

> **Requirement 1:** It must be possible to maintain communicated policies even if the Service Oriented Architecture is dynamically adapted (refers to the constellation of a SOA being established by several entities).

It may happen that a member of a Service Architecture leaves the organization and is replaced by another entity. Dynamic changes of this kind should be

[1] `http://www.primelife.eu/`

possible without resulting in the need to negotiate policies once again with customers or even in the necessity to terminate contracts with customers. This requirement does not apply to the virtual organization: The formalization of policies e.g. may not restrict replacement of enterprises and their services during runtime.

Possible Solution: With the aid of semantic descriptions it is checked - as far as possible -, whether planned changes of the virtual organization are deemed to be possible if considering policies that have been communicated. Policies generated by means of an expert system facilitate such changes of the virtual organization because they do not unnecessarily restrict these facilities for alteration.

> **Requirement 2:** If it is not possible to maintain (all) communicated policies in case of an adaptation of the virtual organization, it must be possible to adapt the communicated policies (builds on requirement 1) through renegotiation, if this fails the service must be stopped.

This requirement complements the previous one: As already mentioned, it sometimes might not be feasible to retain policies when undertaking - possibly inevitable - alterations of the virtual organization. In such cases, mechanisms have to be in place allowing for adaptation of already communicated policies to the new conditions in mutual agreement. Alternative, it must be possible for customers to withdraw from a contract.

Possible Solution: Negotiation of new policies compliant with the law is technically enforced before data are processed in a manner that infringes old policies. At this, semantic descriptions of policies allow for identification of necessary changes and thus offer a basis for renegotiation.

> **Requirement 3:** A service provider whose service is a downstream part (those that process data later) of the overall workflow must adhere to policies given by service providers whose services are upstream parts (those that process data first) of the workflow.

As the service provider who is in contact with the customer makes binding policies for the whole workflow, service providers whose services are downstream parts of the overall workflow have to adhere to these policies.

Possible Solution: In order to achieve that common policies do not have to be negotiated in advance, a mechanism is applied that generates new preferences from existing preferences and policies: At the first service of a workflow customer preferences and policies of the service are matched. The result of the matching process then is matched as set of preferences with the policies of the second service. If preferences and policies are specified on the basis of the same semantic formalism, new preferences can be derived partly automated from them by means of a reasoner.

> **Requirement 4:** Multi-level-matching within a Service Oriented Architecture must be supported.

A multi-level-matching always takes place, when a Service A, which is approached by a user, launches another Service B. In this case Service A has to integrate the policies of Service B.

Possible Solution: Multi-level-matching of policies is enabled by means of formal methods as used for software verification.

> **Requirement 5:** The ability of the data subject to have access to information must be ensured for the future.

If subject access requests are answered on the basis of protocols this can invoke serious difficulties, if a Service Composition or a Virtual organisation is later decoupled. It could be difficult to identify all parties that participated in the specific service. Therefore mechanisms need to be implemented, that allow subject access requests for a longer period of time than the actual service composition is available.

Possible Solution: The logging is attached to the personal data themselves, as metadata. In a service flow, the final step is to deliver the whole data including this metadata back to the original service.

> **Requirement 6:** A ex post notice must be enabled by appropriate mechanisms.

If policies change, an ex-post information of the user becomes necessary (see requirements 1 and 2). Therefore mechanisms need to be included, that allow for notice in multi-level workflows, even if the user is not known to all services. Equally it must be possible for the user to accept the changes towards all included services.

Possible Solution: Standardized interfaces, allowing information against the stream of the workflow.

4 Conclusion

This position paper outlines six requirements for policies used in cross-domain service composition. All six requirements show that the design cross-domain SOA has a deep impact on the overall policy communication. The other way around, it might be impossible to combine services into an orchestration because their policies conflict with each other. It would be interesting to look into the question how both policy and orchestration influence each other and if a design methodology could be found that takes both aspects into account.

The presentation in this paper is very brief and we want to point the interested reader to Meissner and Schallaböck (2009) and Bizer et al. (2007).

References

Johann Bizer, Rüdiger Grimm, Steffen Staab, Sebastian Meissner, Daniel Pähler, Christoph Rigelstein, Martin Rost, Jan Schallaböck, and Felix Schwagereit. Chancen und risiken von service-orientierten architekturen in virtuellen architekturen. Project Report, 2007. URL `http://www.datenschutzzentrum.de/soa/SOAinVO-Analyse.pdf`.

Sebastian Meissner and Jan Schallaböck. Requirements for privacy-enhancing service-oriented architectures. Project report H6.3.1, PrimeLife Consortium, November 2009. URL `http://www.primelife.eu/results/documents`.