Distributed Trust Management for Validating SLA Choreographies

Haq, Irfan Ul; Alnemr, Rehab; Paschke, Adrian; Schikuta, Erich; Boley, Harold; Meinel, Christoph

Proceedings of the Service Level Agreements in Grids Workshop 2009,conjunction with The 10th IEEE International Conference on Grid Computing (Grid 2009), 2009-10-15

DISTRIBUTED TRUST MANAGEMENT FOR VALIDATING SLA CHOREOGRAPHIES

Irfan Ul Haq
Department of Knowledge and Business Engineering, University of Vienna, Austria

Rehab Alnemr
Hasso Plattner Institute, Potsdam University Germany

Adrian Paschke
Institute of Computer Science, Freie University Berlin, Germany

Erich Schikuta
Department of Knowledge and Business Engineering, University of Vienna, Austria

Harold Boley
Institute of Information Technology, National Research Council, Canada

Christoph Meinel
Hasso Plattner Institute, Potsdam University Germany

Abstract

For business workflow automation in a service-enriched environment such as a grid or a cloud, services scattered across heterogeneous Virtual Organisations (VOs) can be aggregated in a producer-consumer manner, building hierarchical structures of added value. In order to preserve the supply chain, the Service Level Agreements (SLAs) corresponding to the underlying choreography of services should also be incrementally aggregated. This cross-VO hierarchical SLA aggregation requires validation, for which a distributed trust system becomes a prerequisite. Elaborating our previous work on rule-based SLA validation, we propose a hybrid distributed trust model. This new model is based on Public Key Infrastructure (PKI) and reputation-based trust systems. It helps preventing SLA violations by identifying violation-prone services at service selection stage and actively contributes in breach management at the time of penalty enforcement.

1. Introduction

A Service Level Agreement (SLA) is a formally negotiated contract between a service provider and a service consumer to ensure the expected level of a service. In a service enriched environment such as Grid, cooperating workflows may result into a service choreography spun across several Virtual Organisations and involving many business partners. Service Level Agreements are made between services at various points of the service choreography. Not much research has been carried out towards dynamic SLA composition of workflows [2][3][7]. We have demonstrated [9]how a single-layer SLA composition model is insufficient to comply with such a multilayered aggregation of services across many Virtual Organisations and why only a hierarchical structure of SLAs among different supply chain partners can fully describe its behavior. We have introduced the concept of *Hierarchical SLA Choreography* [9] or simply SLA Choreography, in accordance with the underlying Service Choreography as well as the notion of *SLA Views* [9] to protect the privacy of business partners across the supply chain. We have also demonstrated how SLA Views contribute to the process of hierarchical SLA aggregation and how a rule-based top-down validation process can be invoked across SLA choreographies [11].

In this paper we elaborate a hybrid distributed trust system based on PKI and reputation-based trust models to enable our rule-based runtime validation framework [11] for hierarchical SLA aggregations.

This paper discusses:

- the justification and significance of a hybrid trust model for the validation of hierarchical SLA aggregations in section 2,
- the conceptual elements of our hybrid PKI and reputation based trust model in section 3, and
- a use case elaborating the breach management role of PKI and reputation based trust model in connection with the SLA validation framework in section 4.

Section 5 concludes the paper with a summary of the proposed model.

2. A Framework for Validation of Hierarchical SLA Aggregations

Service choreography is usually distributed across several Virtual Organizations and under various administrative domains. The complete aggregation information of the SLAs below a certain level in the chain is known only by the

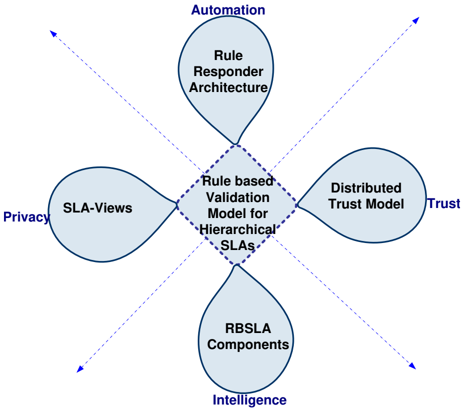

Figure 1. Validation as a Cross-section of Models

corresponding service provider and only a filtered part is exposed up towards the immediate consumer. This is the reason why during the validation process, the composed SLAs are required to be decomposed in an incremental manner down towards the supply chain of services and get validated in their corresponding service providers's domain. A validation framework for the composed SLAs, therefore, faces many design constraints and challenges: a trade-off between privacy and trust, distributed query processing, and automation to name the most essential ones. The aforementioned challenges bring in a cross-section of models depicted in figure 1. In our proposed model, the privacy concerns of the partners are ensured by the SLA View model [9], whereas the requirements of trust and security can be addressed through a reputation-based trust system built upon a distributed PKI (Public Key Infrastructure) based security system. Additionally, we use Rule Responder [14] to weave the outer shell of the validation system by providing the required infrastructure for the automation of role description of partners as well as steering and redirection of the distributed validation queries. The knowledge representation techniques of the RBSLA (Rule based Service Level Agreements) project [5] contribute at the core of validation system. Different parts of the WS-Agreement compliant SLAs can be transformed into corresponding sets of logical rules, which are composed together during the process of SLA composition and can be decomposed into separate queries during the process of validation.

A view in an SLA Choreography represents the visibility of a business partner, which in this case consists of a hierarchical collection of its SLAs both as a producer and consumer. Every service provider is limited only to its own view. In figure 2, two different views are highlighted in an example scenario where a client requires to render and host his videos by using online web services. The *rendering and computing service* S_1 is restricted to its view and

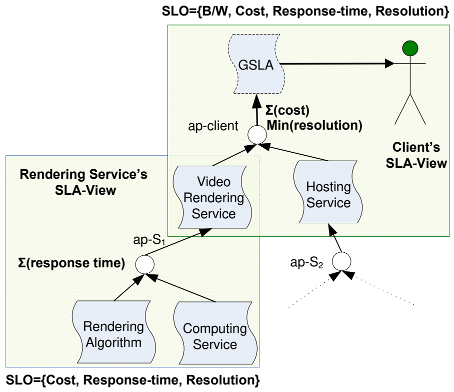

Figure 2. Example Scenario for SLA Views

the client is also shown here to have its own view. The central role during SLA aggregation is played by small circles shown in the figure, known as aggregation points. An aggregation point represents the control point of service provider. During the aggregation process, terms of the SLAs below aggregation points (called consumer-oriented SLAs) are aggregated within an aggregation point so that a feasible SLA offer can be presented to the client above the aggregation point. The whole SLA Choreography is seen as an integration of several SLA Views. In [9], details of the rigorous formal model elaborating SLA views and various aggregation patterns is elaborated. SLA-views can be implemented by using Rule Responder architecture.

Rule Responder adopts the approach of multi agent systems. There are three kinds of agents, namely: Organisational Agents (*OA*), Personal Agents (*PA*), and External Agents (*EA*). An *EA* is an entity that invokes the system from outside. A virtual organization is typically represented by an *OA*, which is the single (or main) point of entry for communication with the "outer" world i.e. an external agent. A *PA* corresponds to the SLA View of a service provider. Similar to an organizational agent, each individual agent (personal and external) is described by its syntactic resources of personal information about the agent, the semantic descriptions that annotate the information resources with metadata and describe the meaning with precise business vocabularies (ontologies) and a pragmatic behavioral decision layer to react autonomously. The flow of information is from External to Organisational to Personal Agent. In our scenario Rule Responder provides the rule-based enterprise service middleware for highly flexible and adaptive Web-based service supply chains.

Rule Responder utilizes RuleML [12] as *platform-independent rule Interchange* format and has the Mule open-source Enterprise Service Bus (ESB)

[13], as Communication Middleware and Agent/Service Broker to seamlessly handle message-based interactions between the responder agents/services and with other applications and services.

As depicted in figure 1, the fourth component in our framework is a distributed trust model. We need to choose a suitable trust model that integrates seamlessly with our aggregation and validation framework. Public Key Infrastructure (PKI) is a popular distributed trust model in Grids. Legitimate members of a Grid are certified by a Certification Authority (CA).

During service choreography, services may form temporary composition with other services, scattered across different VOs. The question of whose parent VO acts as the root CA in this case is solved by including *third party trust manager* like the case for dynamic ad hoc networks. The distributed trust system should work hand-in-hand with the breach management of the SLA validation framework. In case of SLA violation, in addition to enforcing penalty, the affected party is likely to keep a note of the violating service in order to avoid it in future. Moreover, a fair business environment demands even more and the future consumers of the failing service also have a right to know about its past performance. Reputation-based trust systems are widely used to maintain the reputation of different business players and to ensure this kind of knowledge. We propose a hybrid trust model based on PKI and reputation-based trust systems to harvest advantages from both techniques. The main points of the model are:

- the PKI based trust model has a third party trust manager that will act as a root CA and authenticate member VOs. These VOs are themselves CAs as they can further authenticate their containing services.

- Selection of services at the the pre-SLA stage is done by using reputation to prevent SLA violation. Services reputation are updated after each SLA validation process.

- SLA views integrate very closely with the trust model to maintain a balance between trust and security. While the trust model promises trust and security, the SLA views protect privacy.

3. A PKI and Reputation-based Distributed Trust Model

Trust management can be categorized into: policy-based and reputation-based management systems. The two approaches have been developed within the context of different environments and targeting different requirements. On one hand, policy-based trust relies on "strong security" mechanisms such as signed certificates and trusted certification authorities in order to regulate the access of users to services resulting in a binary decision i.e a party being trusted or not trusted whereas on the other hand, reputation-based trust relies on a rather "soft computational" approach where trust is typically computed from local experiences together with the feedback given by other entities in the network (e.g., users who have used services of that provider). The two trust management approaches address the same problem - establishing trust among interacting parties in distributed and decentralized systems. However, they assume different settings. While the policy based approach has been developed within the context of structured organizational environments the reputation systems have been proposed to address the unstructured user community [6].

The policy-based trust systems are very secure and hence are an essential requirement for the B2B and B2C relationships in virtual organisations and for this reason have been widely adopted in Grid Computing. On the other hand, the reputation-based trust is a lenient approach and are very suitable for self-emergent, automated, ad-hoc and dynamic business relationships across virtual enterprises. In the line of our work, we take the best features of both approaches and propose a PKI coupled Reputation-based Trust Management System. We use Rule Responders' agents to spawn trust across different stake-holders of a cross-enterprise business relationship.

In the following sub-sections, we elaborate how the best features of both PKI (policy-based approach) and reputation-based trust systems, along with Rule Responder architecture, are utilized to our advantage.

3.1 Single Sign-On and Delegation

In the proposed model, a third party acts as a root CA. This third party trust manager acts as a root Certification Authority (CA) and authenticates member VOs. These VOs are themselves CAs as they can further authenticate their containing services. Each member is given a certificate. Certificates contain the name of the certificate holder, the holder's public key, as well as the digital signature of a CA for authentication. The authentication layer in each VO middle-ware may be based on Grid Security Infrastructure (GSI) [8] where all resources need to install the trusted certificates of their CAs. GSI uses X.509 [4] proxy certificates to enable Single sign-on and Delegation. With Single Sign-On, the user does not have to bother to sign in again and again in order to traverse along the chain of trusted partners (VOs and services). This can be achieved by the Cross-CA Hierarchical [4] [8] Trust Model where the top most CA, called the root CA provides certificates to its subordinate CAs and these subordinates can further issue certificates to other CAs (subordinates), services or users.

3.2 Reputation Transfer using Trust Reputation Center

In previous work [1], we have presented a reputation-based model that facilitates reputation transfer. One of the main components of this model is

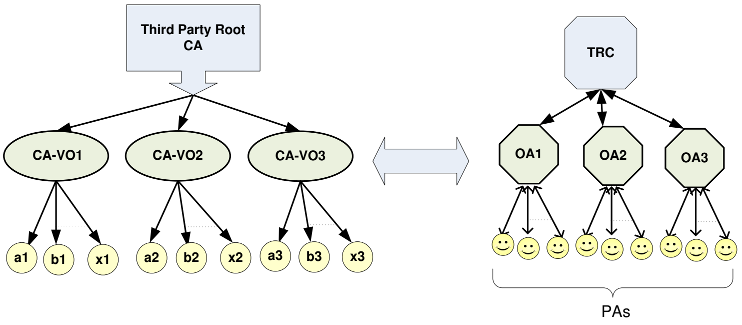

Figure 3. The correspondence between the PKI and reputation based systems and to the Rule Responder architecture

Trust Reputation Centers (TRC). It acts as a trusted third party. The TRC is a pool of users' reputation gathered from different platforms. Each user, agent, or service can have two values that define its reputation: an overall reputation (trusted or non-trusted for malicious users), and a context-based reputation object (RO). When two users from two different platforms (or organizations) establish an interaction, the TRC can be used as a transparent trusted third party. The hybrid system is currently implemented by extending the Rule Responder architecture as shown in figure 3.

As depicted in figure 3, this reputation-based trust model has direct correspondence with Rule Responder's agents and their mutual communication. The PAs consult OAs and OAs in return consult the TRC which is equivalent to the third party CA in PKI based system. In the rest of the paper, we refer to the channel direction flow between PA to OA to TRC, simply as communication among agent.

The word agent in this context refers to a software representation or a smart service. In [1] we illustrate how Agents can exchange lists of acquaintance agents. An Acquaintance Agent List (AAL) is a list of all previously dealt with trusted agents. Then the questioner agent cross-references the list with its own trusted agents, extracts the common ones and issues an inquiry about the agent in question. The answer is a Reputation Object (RO) that expresses the reputation value given by each agent and the context related to this value. The questioner analyzes the set of ROs and forms a decision whether to carry out the transaction or not. There can be more than one ways to represent trust (e.g. in form of numerical values) and hence there are multiple corresponding interpretation or reference models. So when we recommend someone, the name of the trust model can be used as a reference of what measures our trust, and its degree is based upon. We have also proposed the development of Reputation Reference Trust Models (RRTM) [1] that is used as a parameter

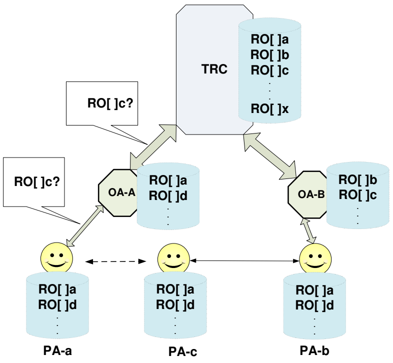

Figure 4. Query of PA-a about reputation of PA-c to OA-A and then redirected to TRC

when mentioning trust. Reputation is viewed as an object that contains the context related to each reputation value and reflects the dynamic nature of trust and its change through time. Reputation object contains a multidimensional array, a matrix, which represents the reputation linked with its context and the RRTM used to calculate this value.

```
Object Reputation {
TrustMatrix [context][reputation value][RRTM];
Time ValidTime;
Credentials PresentedCredentials;}
```

In figure 4, PA-a that corresponds to *service a* that makes an SLA with an unknown *service c* by checking first its credentials. For this purpose, it consults its corresponding organistational agent, which is OA-A in this case. OA-A too, does not have any information about *service c*'s reputation so it redirects *a*'s query to the trust reputation center *TRC* which then transfers the required reputation object tracing back the same channel.

4. Proposed Model via Use Case Scenario

Our final goal is to design a framework for the validation of hierarchical SLA aggregations. We achieve this goal by using the hybrid trust system introduced in Section 3. The processes involved in our model are:

- *Validation of complete SLA aggregation*: to do this the validation query is required to traverse through all the SLA views lying across heterogeneous administrative domains and get validated locally at each SLA view. The multi-agent architecture of Rule Responder provides communication middle-ware to the distributed stake-holders namely the client, the VOs, and various service providers. The validation process empowered by the *single sign-on and delegation* properties of the distributed trust model, helps the distributed query mechanism to operate seamlessly across different administrative domains.

- *Use of reputation in the selection phase*: reputation transfer is required at two stages: at service selection stage and at penalty enforcement stage. In the process of service selection, the reputation transfer helps to select the least violation-prone services, taking into account proactive measures to avoid SLA violations. Out of all the available services, the client (which is also a service in this case) first filters the best services complying its "happiness criteria" [10]. Then the client compares the credentials from reputation objects of the services. The reputation object is traced as discussed in section 3.2. Then the client can select the best service in accordance to its already devised criteria. We assume that out of redundant services which fulfil client's requirements, the service with the highest reputation is selected.
- *Use of PKI and reputation in breach management*: this hybrid Trust is used in the breach management after an occurrence of SLA violation. In figure 5, runtime validation of SLAs ensures that the service guarantees are in complete conformance with the expected levels. Our previous work discusses in detail [11]how the terms of aggregated SLA are represented as logical rules following the RBSLA specifications. These rules are composed together during the process of SLA aggregation [9].

In the scenario depicted in figure 2, the user is interested to render her videos and then host them on the web. Her requirements in terms of Service Level Objectives (SLOs) include a maximum cost of 45 €, maximum response time of 5 seconds, minimum resolution of 640x480 pixels and the minimum bandwidth (from hosting service) of 50 Mbps. In figure 5, we have depicted this scenario from validation point of view. The user-requirements are shown in the figure above the head of EA, as a derivation rule whose premises are SLOs of the aggregated SLA. The SLOs are a expresses as a conjuncted set of negated premises of the derivation rule. The predicates lt and gt denote lesser-than and greater-than respectively. The agents OA and PA representing the Rule Responder architecture, are shown to automate the distributed query processing. For the sake of simplicity, we outline the Rule Responder architecture just from agent-oriented perspective, and abstract various essential details such as the Rule-bases, the knowledge resources and the role of Enterprize Service Bus (ESB).

During the validation process, this rule is decomposed such that each premise will become a subgoal. This subgoal is sent as a message to the PA corresponding to the next SLA view in the hierarchy where it emerges as a conclusion

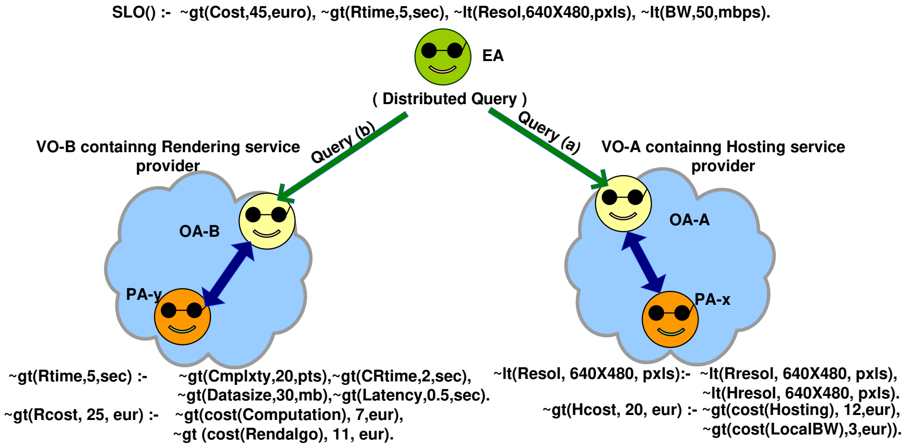
SLO() :- ~gt(Cost,45,euro), ~gt(Rtime,5,sec), ~lt(Resol,640X480,pxls), ~lt(BW,50,mbps).
EA
(Distributed Query)
Query (b)
Query (a)
VO-B containng Rendering service provider
VO-A containng Hosting service provider
OA-B
PA-y
OA-A
PA-x
~gt(Rtime,5,sec) :- ~gt(Cmplxty,20,pts),~gt(CRtime,2,sec), ~gt(Datasize,30,mb),~gt(Latency,0.5,sec).
~gt(Rcost, 25, eur) :- ~gt(cost(Computation), 7,eur), ~gt (cost(Rendalgo), 11, eur).
~lt(Resol, 640X480, pxls):- ~lt(Rresol, 640X480, pxls), ~lt(Hresol, 640X480, pxls).
~gt(Hcost, 20, eur) :- ~gt(cost(Hosting), 12,eur), ~gt(cost(LocalBW),3,eur)).

Figure 5. Validation through distributed query decomposition

of one of the rules in the local rule set, thus forming a distributed rule chain. The initial steps of decomposition procedure are depicted at the bottom of the figure. In the figure, OAs are shown to receive and track the distributed query whenever it enters a new VO e.g. OA-B receives a subgoal $\sim gt(Rtime, 5, sec)$ representing the requirement that the total response time of the system should not be more than 5 seconds. For each service provider, there is a personal agent. A PA, after finishing its job, reports to the corresponding OA that redirect the distributed query to the service provider's PA that comes next in the hierarchical chain. The single sign-on and delegation helps the backtracking to flow smoothly across trusted partners. The process continues until the query has found all the goals expressed in terms of logical rules or if there is a violation at any step in the chain. Active rules tracking these goals or SLOs, are then invoked locally within the administrative domains of the corresponding SLA views. The true or false results are conveyed back following the same routes. In case of a violation, an active rule is fired for the penalty enforcement. In addition to a fine, the reputation of the service is also decreased by the client service and the updated reputation objects is transferred to its corresponding VO from where it is passed to the TRC. If an alternate service is required by the client then the service can be recommended on the basis of its Reputation Object by the corresponding VO, which also keeps track of other services falling in the same category.

5. Conclusion and Future Work

In this paper, we presented the design of a hybrid trust management system as part of validation framework of hierarchical SLA aggregations corresponding to cross-VO workflow compositions. The trust system is based on PKI as well as reputation based trust models thus providing a single sign-on and maintaining service credentials based on their SLA compliance. Although the model presented here is strongly related to already existing trust models and frameworks, the application of this model, as part of validation framework of hierarchical SLA aggregations is innovative. We plan to implement this hybrid trust model through iterative development phases as part of a distributed rule-based validation system using RuleML/XML for interchange [9].

References

[1] Alnemr, Rehab, & Christoph Meinel 2008. Getting More from Reputation Systems: A Context—Aware Reputation Framework Based on Trust Centers and Agent Lists. Computing in the Global Information Technology, ICCGI'08., pages pp. 137–142.

[2] Blake, M. Brian, & David J. Cunnings 2007. Workflow Composition of Service Level Agreements. International Conference on Services Computing, 2007.

[3] Frankova, Ganna 2007. Service Level Agreements Web Services and Security. Springer Verlag, pages 556–562.

[4] Lioy, A., M.Marian, N.Moltchanova, & M. Pala 2006. PKI past, present and future. International Journal of Information Security, Springer Berlin, page 1829.

[5] Paschke, Adrian, & Martin Bichler March 2006. Knowledge Representation Concepts for Automated SLA Management. Int. Journal of Decision Support Systems (DSS).

[6] Piero Bonatti, Daniel Olmedilla, Claudiu Duma, & Nahid Shahmehri 2005. An Integration of Reputation-based and Policy-based Trust Management. Semantic Web and Policy Workshop, 2005.

[7] Unger, Tobias, Frank Leyman, Stephanie Mauchart, & Thorsten Scheibler 2008. Aggregation of Service Level Agreement in the context of business processes. Enterprise Distributed Object Computing Conference Munich, Germany, 2008.

[8] Zhao, S., A. Aggarwal, & R. D. Kent 2007. PKI-Based Authentication Mechanisms in Grid Systems. International Conference on Networking, Architecture, and Storage.

[9] Irfan Ul Haq, Altaf Huqqani, Erich Schikuta 2009. Aggregating hierarchical Service Level Agreements in Business Value Networks Business Process Management Conference (BPM2009), 2009.

[10] Kevin Kofler, Irfan Ul Haq, Erich Schikuta 2009. A Parallel Branch and Bound Algorithm for Workflow QoS Optimization The 38th International Conference on Parallel Processing (ICPP2009), Vienna 2009.

[11] Irfan Ul Haq, Adrian Paschke, Erich Schikuta, Harold Boley 2009. Rule-Based Workflow Validation of Hierarchical Service Level Agreements 4th International Workshop on Workflow Management (ICWM2009), Geneva 2009.

[12] H. Boley. The Rule-ML Family of Web Rule Languages. In *4th Int. Workshop on Principles and Practice of Semantic Web Reasoning*, Budva, Montenegro, 2006.

[13] Mule. Mule Enterprise Service Bus, http://mule.codehaus.org/display/MULE/Home, 2006.

[14] A. Paschke, H. Boley, A. Kozlenkov, and B. Craig. Rule responder: RuleML-based agents for distributed collaboration on the pragmatic web. Proceedings of the 2nd international conference on Pragmatic web Tilburg, The Netherlands, 2007.