CoreGRID
European Research Network on Foundations, Software Infrastructures and Applications
for large scale distributed, GRID and Peer-to-Peer Technologies
A Network of Excellence funded by the European Commission

Extending WS-Agreement for dynamic negotiation of Service Level Agreements

Wolfgang Ziegler
`Wolfgang.Ziegler@scai.fraunhofer.de`
Fraunhofer Institute SCAI, Department of Bioinformatics
Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Philipp Wieder
`philipp.wieder@udo.edu`
Technical University Dortmund
44221 Dortmund, Germany

Dominic Battré
`dominic.battre@tu-berlin.de`
Berlin University of Technology
10587, Berlin, Germany

CoreGRID Technical Report
Number TR-0172
August 29, 2008

Institute on Resource Management and Scheduling

CoreGRID - Network of Excellence
URL: http://www.coregrid.eu

CoreGRID is a Network of Excellence funded by the European Commission under the Sixth Framework Programme

Project no. FP6-004265

Extending WS-Agreement for dynamic negotiation of Service Level Agreements

Wolfgang Ziegler
Wolfgang.Ziegler@scai.fraunhofer.de
Fraunhofer Institute SCAI, Department of Bioinformatics
Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Philipp Wieder
philipp.wieder@udo.edu
Technical University Dortmund
44221 Dortmund, Germany

Dominic Battré
dominic.battre@tu-berlin.de
Berlin University of Technology
10587, Berlin, Germany

CoreGRID TR-0172
August 29, 2008

Abstract

Services in commercial or scientific environments often need to be delivered at a guaranteed service level. This becomes even more important, if composition of services is required, as results will become available for the requester within a predictable timeframe only if the orchestration process may create Service Level Agreements (SLAs) for the individual services. WS-Agreement is a proposed recommendation of the Open Grid Forum (OGF) defining a language and a protocol to create SLAs. As the current version of the specification of WS-Agreement provides only a simple, one-step approach for the creation of SLAs the GRAAP working group of the OGF started working on a more sophisticated protocol allowing multi-step negotiations to create agreements. In this paper we present the approach for negotiations of SLAs using WS-Agreement under specification at the OGF. We describe the state of the work and give an outlook of the next steps until the specification will become a proposed recommendation.

1 Introduction

Service Level Agreements (SLA) may be used to establish agreements on the quality of a service between a service provider and a service consumer. Today these service level agreements are often created manually for individual services or are established through a framework contract between service provider and customer. Given the dynamic nature of the requirements towards service provisioning in Grids and the increased flexibility in selecting the most appropriate service provider for a required service new mechanisms are needed to create Service Level Agreements on the fly and to modify at a later state.

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

Inspired by previous developments in the web-service domain research and development in the Grid domain started addressing electronic Service Level Agreements on the quality or properties of services or resources to be used to solve a problem. The negotiation of the agreement between two parties is usually done automatically by appropriate software instances of the service provider and service consumer. In this paper we focus on WS-Agreement [3], which is the proposed recommendation of the Open Grid Forum (OGF) to create and monitor Service Level Agreements. It has been specified by the Grid Resource Allocation Agreement Working Group (GRAAP-WG) [8] where CoreGRID researchers have been driving the development together with industrial partners from NEC, IBM, HP and Platform since 2002.

Following the creation of an SLA with WS-Agreement both parties of the agreement need access to the state information of the agreement. In the case of WS-Agreement this information may be retrieved through functions provided by the Web Service Resource Framework (WSRF) on which the Web service expression of the WS-Agreement specification is based. Monitoring a SLA also empowers both parties to detect potential violations of the agreement and to take appropriate measures.

Negotiation and Monitoring of SLAs is of particular importance for commercial service providers. Standards are required to guarantee interoperability and the GRAAP-WG has become the focal point of these efforts.

The WS-Agreement specification version 1.0 already includes a protocol for negotiation of agreements. However, this protocol was designed to cover the most simple and general case: an offer for an SLA is made by either of the two parties and the respective other party may accept or reject the offer. No further negotiation, e.g. in form of a counter offer or request for modification is supported. Since there are situations where an agreement has to be modified at a later state or the process of creating an agreement needs more than the single step approach described above a more sophisticated protocol for negotiation and re-negotiation is needed in addition to the existing one.

The Grid Resource Allocation Agreement Group currently is working on a specification for an interoperable protocol for sophisticated negotiations and re-negotiation of SLAs: WS-Agreement-Negotiation.

In Section 2 we list other recent research and developments on negotiation of SLA, followed by a brief overview on WS-Agreement in Section 3. Section 4 we describe some use-cases for negotiation and re-negotiation of Service Level Agreements. Section 5 presents the requirements for a SLA negotiation while section 6 describes the protocol currently under consideration for specification. Section 7 gives an outlook on future work.

, and finally, the three approaches addressing different requirements, which have been selected for consolidation to form a single protocol during the last OGF meeting:

- A Contract Re-negotiation Protocol
- Dynamic SLA negotiation based on WS-Agreement
- Protocol considerations based on the current WS-Agreement specification

2 Related work

Different aspects of electronic Service Level Agreements have been investigated by a number projects over the last years. Some of them also presenting implementations of SLAs and their negotiation. NextGrid [11] proposed SLAs and a negotiation approach, which are modelled according to business objectives of both customers and service providers [9]. The development done within the Akogrimo [1] project focusses on Service Level Agreements in the context of mobile Grids [2]. The developments of the HPC4U [10] project include SLA negotiation, multi-site SLA-aware scheduling to support Grids as an environment for the execution of HPC applications. while the BREIN [6] project focusses on using SLA for supporting reliable Grids for businesses. The AssessGrid [4] project is addressing obstacles of a wide adoption of Grid technologies by bringing risk management and assessment into this field. BEinGRID [5] includes a large number of associated business experiments where some of them are using SLAs. To support these experiments BEinGRID has established a horizontal SLA cluster addressing also SLA Negotiation with a focus on the requirements of contract law. Finally, SmartLM [13] is targeting on the development of new business modes for using commercial software in Grids based and a corresponding license mechanisms using SLAs. An overview on 6 European projects with relevant research and development activities in the area of SLA is given in the CoreGRID Technical Report TR-0129 [12].

3 WS-Agreement overview

3.1 SLA life-cycle

Service Level Agreements have a certain life cycle, which they are running through from the initial preparation of the templates for an agreement up the evaluation whether the agreement has been fulfilled, or partly or completely violated. W. Sun et al [7] describe the SLA life cycle consisting of five phases; these phases are shown in Figure 1.

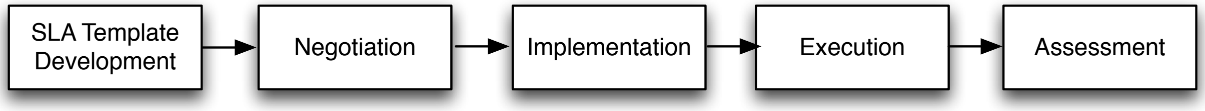

Figure 1: SLA lif-cycle.

The activities in the individual phases can be described as follows:

1. SLA Development: In this phase the SLA templates are developed.
2. Negotiation: In this phase the SLA is negotiated and the contracts are executed.
3. Implementation: where the SLA is generated.
4. Execution: The SLA is executed, monitored, and maintained.
5. Assessment: Evaluation of the SLA performance. In this phase, a re-evaluation of the initial SLA template might be done.

In contrast to the linear sequence of phases described above additional transitions between the Implementation or the Execution phase back to the Negotiation phase will be performed when re-negotiation of an agreement is required.

3.2 WS-Agreement

The WS-Agreement specification version 1.0 already includes a protocol for negotiation of agreements. However, this protocol was designed to cover the most simple and general case: an offer for an SLA is made by either of the two parties and the respective other party may accept or reject the offer. No further negotiation, e.g. in form of a counter offer or request for modification is supported. Since there are situations where an agreement has to be modified at a later state or the process of creating an agreement needs more than the single step approach described above a more sophisticated protocol for negotiation is needed in addition to the existing one.

The purpose of Web Services Agreement Specification is supporting establishing an agreement on the usage of Web Services between a service provider and a consumer. WS-Agreement defines a language and a protocol to represent the services of providers, create agreements based on offers and monitor agreement compliance at runtime. An agreement defines a relationship between two parties that is dynamically established and dynamically managed. The objective of this relationship is to deliver a service by one of the parties. In the agreement each party agrees on the respective roles, rights and obligations.

A provider in an agreement offers a service according to conditions described in the agreement. A consumer enters into an agreement with the intent of obtaining guarantees on the availability of one or more services from the provider. Agreements can also be negotiated by entities acting on behalf the provider and / or the consumer. An agreement creation process usually consists of three steps:

- The initiator retrieves a template from the responder, which advertises the types of offers the responder is willing to accept.
- The initiator then makes an offer,
- Which is either accepted or rejected by the responder.

The basic protocol of WS-Agreement is symmetric with respect to the initiating party, which allows both the resource consumer and the resource provider to act as initiator or responder respectively. WS-Agreement-Negotiation sitting on top of WS-Agreement keeps this symmetry allowing either of the parties on an Agreement to initiate the (re-)negotiation of agreements.

An agreement consists of the agreement name, its context and the agreement terms. The context contains information about the involved parties and metadata such as the duration of the agreement. Agreement terms define the content of an agreement: Service Description Terms (SDTs) define the functionality that is delivered under an agreement. A SDT includes a domain-specific description of the offered or required functionality (the service itself). Guarantee Terms define assurance on service quality of the service described by the SDTs. They define Service Level Objectives (SLOs), which describe the quality of service aspects of the service that have to be fulfilled by the provider. The Web Services Agreement Specification allows the usage of any domain specific or standard condition expression language to define SLOs. The specification of domain-specific term languages is explicitly left open. Figure 2 shows the structure of an Agreement.

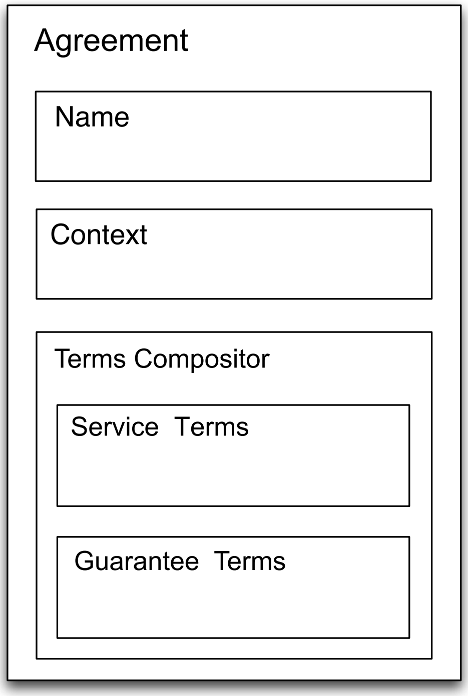

Figure 2: WS-Agreement Structure.

4 Use-cases for negotiaion

Basic use-cases are such like the extension or reduction of a previous reservation for resources guaranteed with a Service Level Agreement. This allows both a resource provider and the resource consumer more flexibility, e.g. to adapt the resource usage to actual requirements. The following paragraphs describe two more complex use-cases that are built on the basic ones.

4.1 Agreement on multiple QoS Parameters

In an environment consisting of several clusters operated in different administrative domains SLAs might be used for co-allocation or the resource allocation for workflows or distributed applications. A typical use-case is the co-allocation of multiple compute resources together with the network links between these resources with a dedicated QoS to run a distributed parallel application. The user specifies his request and the resource orchestrator starts the negotiation with the local scheduling systems of the compute resources and the with the network resource management system (NRMS) in order to find a suitable time-slot, where all required resources are available at the same time. Once a common time-slot is identified the orchestrator requires the reservation of the individual resources. Again, this reservation can be expressed as a Quality of Service and an SLA may be created where the reservation is fixed is a binding agreement for the two parties. The German VIOLA project [14] was addressing reservation and co-allocation of multiple resources across different administrative domains. In VIOLA a MetaScheduling Service was developed, which negotiates the time-slots with the different schedulers of the clusters and the NRMS and initiates the reservation of the nodes requested by the user. Figure 3 gives an overview over the architecture showing the UNICORE systems network job supervisor (NJS) and the target system interface (TSI), which are responsible for handling the job transfer to the local compute resources. At the time agreed upon the resources will then be available for the users to run their distributed applications. Another use-case is a workflow spanning across several resources. The only difference to the use-case described before is the kind of temporal dependencies: While for the distributed parallel application the resources must be reserved for the same time, for the workflow use-case the resources are needed in a given sequence. Thus the orchestration service needs to negotiate the reservations such that one workflow component can be executed on the required resource after the preceding component is completed.

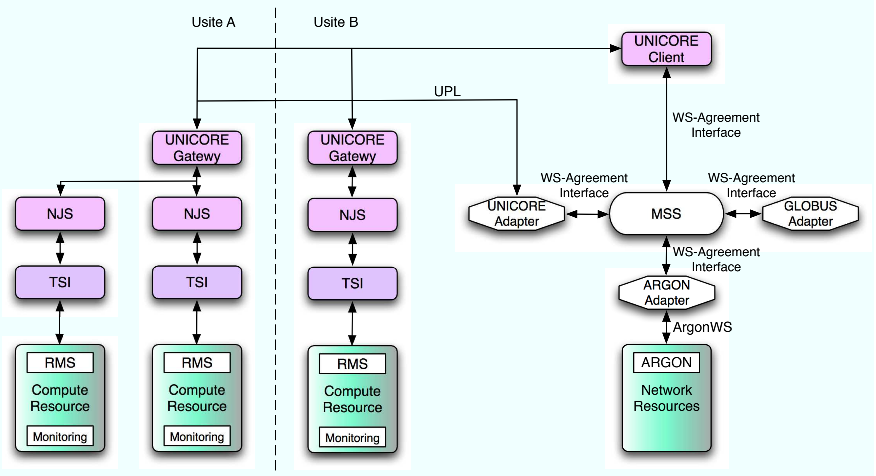

Figure 3: Architecture of the VIOLA MetaScheduling Service MSS.

4.2 Grid Scheduler interoperation

Since there is no single orchestrating service or Grid scheduler in a Grid spanning across countries and administrative domains we have to deal with multiple instances of independent Grid schedulers. Using resources from different domains requires co-ordination across multiple sites. There are two approaches either directly trying to negotiate with respective local scheduling systems or negotiation with the respective local orchestrator. The former solution requires local policies allowing a remote orchestrator to negotiate with local schedulers, which is in general not the case. In the second case there is one access point to the local resources, which then negotiates on behalf of the initiation orchestrator. The Grid Scheduling Architecture Research Group (GSA-RG) of the OGF has produced a preliminary definition of the Grid Scheduling Architecture. This architecture takes into account that the second approach described above also has a better scalability than the first one the. Figure 4 presents a high-level view on such a scheduling architecture.

For the communication between the different orchestration services or Grid schedulers, WS-Agreement as a language and a protocol to create SLAs was selected to achieve the necessary interoperability. Later at the end of the negotiation process the resulting SLAs may be combined by the initiating resource orchestrator into one single agreement wrapping the agreements created with all individual service providers. This composed agreement is then the SLA between the orchestrator and his client, the end-user.

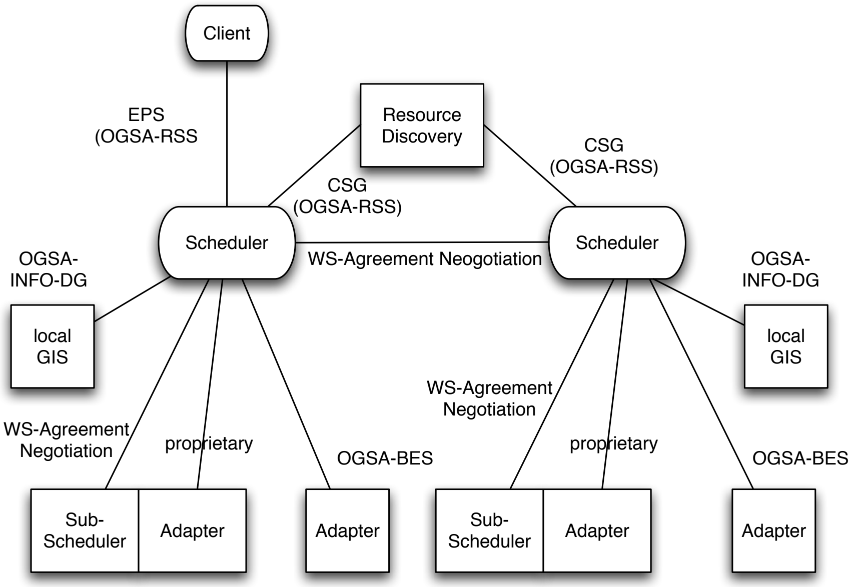

Figure 4: Grid Scheduling Architecture as discussed in the GSA-WG.

5 Approaches for the protocol

During the last three OGF meetings a number of proposals for a negotiation protocol have been discussed, two of them will be briefly presented in the following paragraphs. The discussion built the foundation for the work on the specification of the negotiation protocol outlined in Section 6.

5.1 A Contract Re-negotiation Protocol

The contract Re-negotiation protocol described an abstract, domain-independent protocol for the re-negotiation of contracts. Thus, the protocol is independent of the existing WS-Agreement specification. However, it is possible to use it together with WS-Agreement is some of the restrictions are modified from assumptions implicitly ruling the protocol to being implemented as domain specific policies that can be considered in the protocol depending on the domain. This protocol is based on the principles of contract law to make agreements with it legally compliant and allows for multi-round re-negotiation in an environment where messages may be lost, delayed and re-ordered. The protocol definition assumes an asymmetry of resource provision from resource consumption, which is reflected by putting a higher weight on the requirements of the service provider, e.g. with respect to avoiding potential loss due to the behaviour of a customer.

The protocol also allows the initiation of re-negotiation through non-binding enquiries to the other party so that an estimate as to how much it would cost to change the contract can be obtained before committing to a new contract. When re-negotiation is initiated the contract enters the re-negotiating state, which is a sub-state of contracted as the original contract is still in force, irrespective of the on-going re-negotiation. After successful re-negotiation current contract is in the superseded state as a new contract will have superseded this contract.

The protocol includes several basic assumptions derived from the contract law

- offer messages are binding if accepted,
- all offers are acknowledged and,
- because of the risk of cheating by the customer, contract laws mailbox rule is invoked where a contract is formed when the accept message is sent by the offeree (i.e. the resource provider) and not when the accept is received by the offeror (i.e. the customer).

The most important aspect of this approach for the future negotiation protocol was the fact, that the negotiation state is always a sub-state of contracted, e.g. the current agreement remains in force until it is superseded by a new one. As a major drawback the asymmetry was considered resulting in an early obligation of the customer vis-à-vis the provider.

5.2 Dynamic SLA negotiation based on WS-Agreementl

Negotiation requires an iterative process between the parties involved. To rely on WS-Agreement and minimise the extensions to the proposed standard, this proposal suggests not to negotiate SLAs but to negotiate and refine the templates that can be used to create an SLA. Here, the focus is on the bilateral negotiation of agreement templates using a simple offer/counter offer model.

In order to use this model in the WS-Agreement protocol, a new function is introduced: negotiateTemplate. This function takes one template as input (offer), and returns zero or more templates (counter offer). The negotiation itself is an iterative process. A basic negotiation process evolves in three major stages. During the negotiation process the agreement initiator becomes the negotiation initiator. Accordingly, the agreement providers are negotiation responders. Figure 5 shows a sequence diagram for the negotiation of the agreement template. The three steps are

1. Initialisation of the negotiation process
   First, the negotiation initiator initialises the process by querying a set of SLA templates from agreement providers by sending a standard WS-Agreement message, getResourceProperty request, to agreement providers (not shown in Figure 5). The initiator chooses the most suitable one as a starting point for the negotiation process. This template defines the context of the subsequent iterations. All subsequent offers must refer to this agreement template. This is required in order to enable an agreement provider to validate the creation constraints of the original template during the negotiation process, and therefore the validity of an offer.

2. Negotiation of the template
   After the negotiation initiator has chosen an agreement template, it will create a new agreement template based on the chosen one. The new created template must contain a reference to the originating template within its context. Furthermore, the agreement initiator may adjust the content of the new created template, i.e. service description terms, the service property terms, and the guarantee terms. These changes must be done according to the creation constraint defined in the original template. Additionally, the negotiation initiator may also include creation constraints within the new created template. These constraints provide hints for the negotiation responder, within which limits the negotiation initiator is willing to create an agreement. After the initiator created the new agreement template according to its requirements, the template is send to responders via a negotiateTemplate message. Now the agreement provider checks whether the service defined in the request could be provided or not. If the service can be provided, it just returns the agreement template to the client, indicating that an offer based on that template will potentially be accepted. Otherwise, the provider employs some strategy to create reasonable counter offers. The relationship between dynamically created templates and original ones must be reflected by updating the context of the new templates accordingly. After creating the counter offers the provider sends them back to the negotiation initiator (negotiateTemplate response).

3. Post-processing of the templates
   After the negotiation initiator received the counter offers from the negotiation responder, it checks whether one or more meets its requirements. If there is no such template, the initiator can either stop the negotiation process, or start again from step 1. If there is an applicable template, the initiator validates whether there is need for an additional negotiation step or not. If yes, the initiator uses the selected template and proceeds with step 2, otherwise the selected template is used to create a new SLA.

   For the final SLA creation the proposal suggests adding a new type of agreement that must be created in two phases: the first phase is a creation of the agreement triggered by a new prepareAgreement message resulting in a non-binding agreement (similar to the quote in the proposal 2.3.1. The second phase is initiated with a new non-standard Commit message.

Symmetry of the negotiation process and negotiation of templates instead of agreements are considered the most appealing aspects of this approach. Another important property is the usage of constraints to limit the negotiation space.

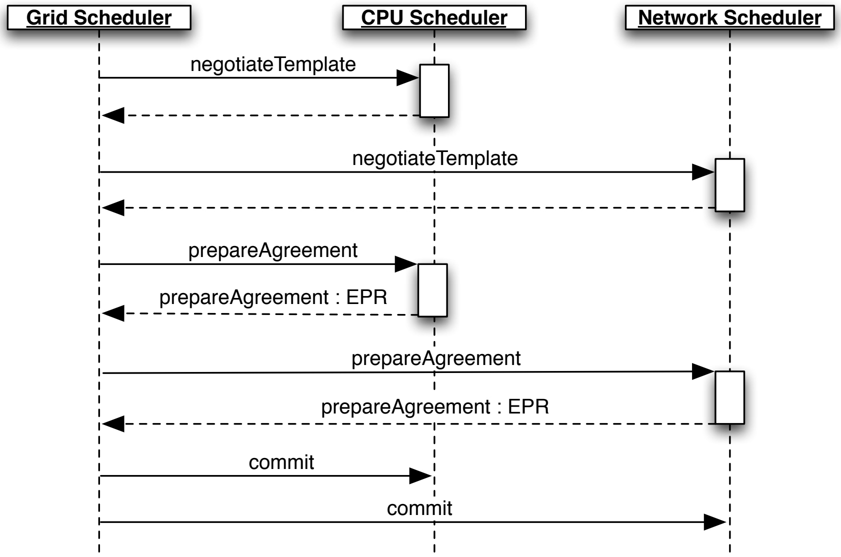

Figure 5: Extended WS-Agreement SLA negotiation.

6 Protocol proposal for WS-Agreement-Negotiation

Basically, both approaches for a negotiation protocol work together with WS-Agreement requiring only minor changes to WS-Agreement. It is also common for the approaches, to create a new Agreement as a result of a successful negotiation that is superseding the previous one instead of modifying the existing one. This also implies that the original agreement remains completely in force at least until the new superseding agreement has been created. The superseding agreement becomes effective once the negotiation has been completed successfully. In case the negotiation fails the original agreement remains in force.

Considering the differences as highlighted in the previous section, we suggest a separation of issues:

- Definition of the basic protocol, which will be common for all approaches
- Definition of policies, which may be used to guide the negotiation process

These policies that we expect to be domain dependent might be used, e.g. to

- restrict the possibility of initiation the negotiation to one of the parties (asymmetric approach),
- define the bindingness of the offers (for both cases AI=MR and AR=MR),
- define the time in the negotiation process when the superseding Agreement becomes effective

We suggest to implement WS-Agreement-Negotiation using the messages describing the Contract Re-negotiation Protocol. As basic sequence diagram for the negotiation process the one presented in Figure 6 will be used. Of course, there are additional issues that need further discussion with the GRAAP-WP, e.g.

- the agreement states of WS-Agreement might have to be extended to reflect an ongoing negotiation,
- the information that an agreement has been superseded by a new one after successful negotiation should available in the context together with the EPR of the superseding agreement
- the information that an agreement is superseding a previous one after successful negotiation should available in the context together with the EPR of the previous agreement

Figure 6 depicts the sequence diagram for the negotiation process. As said before, the negotiation may be initiated by either of the two parties.

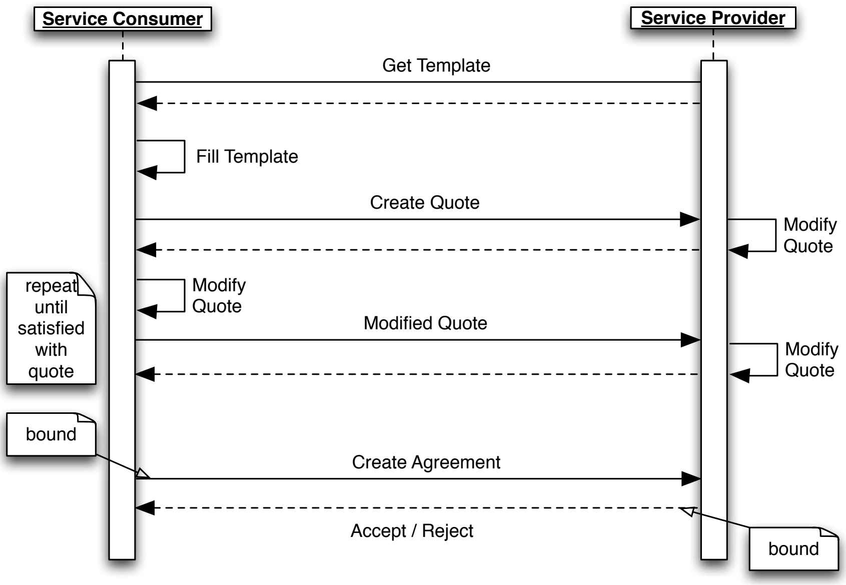

Figure 6: Sequence diagram of the negotiation process.

7 Future work

The next meeting of the GRAAP-WP will be dedicated to define the state machine for the protocol as well as the messages to exchange the modification offers. Based on this the WSDL and required XML elements for the definition of constraints will be created to have a first prototype implementation ready for experiments and tests early 2009. Since negotiation of license availability and subsequent reservation via SLA play a major role in the SmartLM project this project will along with the AssessGrid project provide the initial prototype.

8 Acknowledgements

Some of the work reported in this paper is funded by the German Federal Ministry of Education and Research through the D-Grid project under grant #01AK800A. This paper also includes work carried out jointly within the CoreGRID Network of Excellence funded by the European Commission's IST programme under grant #004265. Last not least the authors gracefully acknowledge the stimulating discussions in the GRAAP-WG.

References

[1] The Akogrimo Project. Web site, 18 July 2008 <www.akogrimo.org>.

[2] The Akogrimo Architecturet. Web site, 18 July 2008 <http://www.akogrimo.org/download/White_Papers_and_Publications/Ak
1.pdf>.

[3] A. Andrieux et al. Web Services Agreement Specification (WS-Agreement). Grid Forum Document GFD.107, Open Grid Forum, 2007.

[4] AssessGrid - Advanced Risk Assessment and Management for Trustable Grids. Web site, 18 July 2008 <http://www.assessgrid.eu/>.

[5] BEinGRID - Business Experiments in Grid. Web site, 18 July 2008 <http://www.beingrid.eu/>.

[6] The BREIN Project, Activity IST-2005-2.5.4. Web site, 18 July 2008 <http://www.gridsforbusiness.eu/>.

[7] W. Sun et al. The Role of XML in Service Level Agreements Management. Network Management Research Center, Beijing Jiaotong University, IEEE, 2005.

[8] Grid Resource Allocation Agreement Protocol Working Group, 2008. Web site, 18 July 2008 <https://forge.gridforum.org/projects/graap-wg/>.

[9] Peer Hasselmeyer, Henning Mersch, Bastian Koller, and Philipp Wieder. Implementing an SLA Negotiation Framework. Technical report. Web Site, 18 July 2008 <http://www.nextgrid.org/download/publications/Implementing_an%20SLA_Negotiation_Framework.pdf/>.

[10] The HPC4U Project, Reliable clusters for Grids. Web site, 18 July 2008 <http://www.hpc4u.eu/>.

[11] NextGRID: Architecture for Next Generation Grids. Web site, 18 July 2008 <http://www.nextgrid.org/>.

[12] Michael Parkin, Rosa Badia, and Josep Matrat. A Comparison of SLA Use in Six of the European Commissions FP6 Projects. Technical Report TR-0129, Institute on Resource Management and Scheduling, CoreGRID - Network of Excellence, April 2008.

[13] SmartLM - Grid-friendly software licensing for location independent application execution. Web site, 18 July 2008 <http://www.smartlm.eu/>.

[14] VIOLA – Vertically Integrated Optical Testbed for Large Application in DFN. Web site, 18 July 2008 <http://www.viola-testbed.de>.