Chapter 5
Management for Service Level Agreements

Igor Rosenberg, Antonio Conguista, and Roland Kuebert

Abstract  Electronic services, like other general-purpose services, often need to be delivered at a guaranteed service level. Service Level Agreements (SLAs) can be used to address this by defining Quality of Service (QoS); but they usually are paper contracts. The delivery of electronic services, automatically provisioned and managed, calls for a more agile system based on dynamic SLAs: electronic contracts generated on-the-fly. The approach taken within BEinGRID to identify the barrier for wide enterprise adoption is presented. This lead to the identification of requirements, capacities and design patterns. Components were also developed to bridge the gap. Finally, after analysing the uptake of the software provided, conclusions are drawn, and recommendations are presented.

5.1 The Overall Challenge

The dynamic aspects of aggregated electronic service provisioning compel the players to expect services to be offered with a guaranteed QoS. One way to provide these guarantees is to attach the service provision with the creation of an SLA, which describes the limits of the service, and the consequences of a failure to provide it. The inability to provide SLAs is an important barrier for the Grid uptake by industry in distributed e-business environments. SLA Management encompasses the SLA contract definition (basic schema with the QoS parameters), SLA negotiation, SLA monitoring and SLA enforcement according to defined policies. The main point is to build a new layer upon the grid middleware, which is able to create negotiation mechanisms between providers and consumer of services.

The SLAs considered in this chapter are electronic contracts describing service QoS, with a short duration (hours, or days), little human intervention, and rapid deployment. This contrasts with “paper” SLAs, which are signed by lawyers, involving many human actors, have long time span, and may include elements not related to the service QoS. The electronic contracts often reference a paper SLA, called in this case the framework SLA, which defines the basic boundaries in which electronic contracts can be signed.

I. Rosenberg (✉)
Atos Origin, calle Albaracín 25, 28037 Madrid, Spain
e-mail: igor.rosenberg@atosorigin.com
As the solution proposing dynamic SLAs was deemed quite specific and novel, the BEinGRID methodology was used for an exploration of the topic. Requirements were elicited from several Business Experiments. The initial analysis therefore started by fragmenting the whole SLA lifecycle into its separate steps. Once each challenge presented by the following Common Technical Requirements has been solved through the implementation of a component, a general framework can be built to add the SLA concepts to the whole service provisioning experience. Even more, the SLA functionality has to integrate with other capacities targeting other areas, like VO and Security. The public acceptance of SLAs for electronic services will be reached only through the integration of SLA features in a complete framework. This is what had previously been produced for the GRIA [15] and GRASP middlewares. In order to also provide a solution for a more general scenario, a solution has been implemented for the Globus Toolkit 4 middleware as BEinGRID interoperable components.

Two detailed examples are proposed in Chap. 9.

The whole SLA architecture, which should be perceived as a service management interface, must be centralised, powerful but intuitive, while being extremely verbose with monitoring information. SLAs are an elegant way of addressing variable service QoS in a competitive environment (service marketplace), but carry great possibilities of human errors inducing catastrophic economic consequences. They are still conceived, possibly rightly, as dangerous by the members of the deciding committees. Evidence is provided by the trivial and conservative SLAs offered actually (start 2009) to service customers (see for example the agreements offered by Amazon for the storage and execution services S3 [4] and EC2 [3]). In many cases these SLAs are not sufficient for the decision to rely on external providers for business-critical activities.

5.2 Technical Requirements

Providing SLAs with a service must follow the same steps as the service’s own lifecycle, from creation to decommission. The SLA must be seen as the shadow of the service, in the sense that each time an event concerns the service, the event also affects the SLA. As such, the Common Technical Requirements of the SLA technical area follow the steps of the service lifecycle. The ones which have gained the most important focus in BEinGRID are: SLA Negotiation, SLA Resource Selection Optimisation and SLA Evaluation.

The reader is invited to read Chap. 9 on integration of results from different technical areas of BEinGRID, to understand how the SLA aspects are only a piece of a much greater puzzle, and how SLAs should be implemented in parallel with at least Virtual Organisation and Security features. In the following we briefly explain the main concepts.