SLAs Empowering Services in the Future Internet[1]

Joe Butler[1], Juan Lambea[2], Michael Nolan[1], Wolfgang Theilmann[3], Francesco Torelli[4], Ramin Yahyapour[5], Annamaria Chiasera[6], and Marco Pistore[7]

[1] Intel, Ireland, {joe.m.butler, michael.nolan}@intel.com
[2] Telefónica Investigación y Desarrollo, Spain, juanlr@tid.com
[3] SAP AG, Germany, wolfgang.theilmann@sap.com
[4] ENG, Italy, francesco.torelli@eng.com
[5] Technische Universität Dortmund, Germany, ramin-yahyapour@udo.edu
[6] GPI, Italy, achiasera@gpi.it
[7] FBK, Italy, pistore@fbk.eu

Abstract. IT-supported service provisioning has become of major relevance in all industries and domains. However, the goal of reaching a truly service-oriented economy would require that IT-based services can be flexibly traded as economic good, i.e. under well defined and dependable conditions and with clearly associated costs. With this paper we claim for the need of creating a holistic view for the management of service level agreements (SLAs) which addresses the management of services and their related SLAs through the complete service lifecycle, from engineering to decommissioning. Furthermore, we propose an SLA management framework that can become a core element for managing SLAs in the Future Internet. Last, we present early results and experiences gained in four different industrial use cases, covering the areas of Enterprise IT, ERP Hosting, Telco Service Aggregation, and eGovernment.

Keywords: Service Level Agreement, Cloud, Service Lifecycle

1 Introduction

Europe has set high goals in becoming the most active and productive service economy in the world. Especially IT supported services have become of major relevance in all industries and domains. The service paradigm is a core principle for the Future Internet which supports integration, interrelation and inter-working of its architectural elements. Besides being the constituting building block of the so-called Internet of Services, the paradigm equally applies to the Internet of Things and the underlying technology cloud platform below. Cloud Computing gained significant attention and commercial uptake in many business scenarios. This rapidly growing service-oriented economy has highlighted key challenges and opportunities in IT-supported service provisioning. With more companies incorporating cloud based IT services as part of

[1] The research leading to these results is partially supported by the European Community's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement n° 216556.

J. Domingue et al. (Eds.): Future Internet Assembly, LNCS 6656, pp. 327–338, 2011.

their own value chain, reliability and dependability become a crucial factor in managing business. Service-level agreements are the common means to provide the necessary transparency between service consumers and providers.

With this paper, we discuss the issues surrounding the implementation of automated SLA management solutions on Service Oriented Infrastructures (SOI) and evaluate their effectiveness. In most cases today, SLAs are either not yet formally defined, or they are only defined by a single party, mostly the provider, without further interaction with the consumer. Moreover, the SLAs are negotiated in a lengthy process with bilateral human interaction. For a vivid IT service economy, better tools are necessary to support end-to-end SLA management for the complete service lifecycle, including service engineering, service description, service discovery, service composition, service negotiation, service provisioning, service operation, and service decommissioning.

We provide an approach that allows services to be described by service providers through formal template SLAs. Once these template SLAs are machine readable, service composition can be established using automatic negotiation of SLAs. Moreover, the management of the service landscape can focus on the existence and state of all necessary SLAs. A major aspect herein is the multi-layered service stack. Typically, a service is dependent on many other services, e.g. the offering of a software service requires infrastructure resources, software licenses or other software services.

We propose an SLA management framework that offers a core element for managing SLAs in the Future Internet. The framework supports the configuration of complex service hierarchies with arbitrary layers. This allows end to end management of resources and services for the business value chain. The scientific challenges include the understanding and modelling of the relationships between SLA properties. For instance assessing the performance of a service and its corresponding SLAs includes the whole stack and hierarchy. The deep understanding of this relationship needs to be modelled in a holistic way for reliability, performance etc.

The technical foundation to our approach is a highly configurable plug-in-based architecture, supporting flexible deployment options including into existing service landscapes. The framework's architecture mainly focuses on separation of concerns, related to SLAs and services on the one hand, and to the specific domain (e.g., business, software, and infrastructure) on the other.

With a set of four complementary use case studies, we are able to evaluate our approach in a variety of domains, namely ERP hosting, Enterprise IT, Service Aggregation and eGovernment. ERP Hosting is investigating the practicalities and benefits of holistic SLA planning and management when offering hosted ERP solutions for SMEs. Enterprise IT focuses on SLA-aware provisioning of compute platforms, managing decisions at provisioning time and runtime, as well as informing business planning. Service Aggregation demonstrates the aggregation of SLA-aware telecommunication and third party web services: how multi-party, multi-domain SLAs for aggregated services can best be offered to customers. eGovernment validates the integration of human-based services with those that are technology based, showcasing the automated, dynamic SLA-driven selection, monitoring and adjustment of third-party provisioned services.

The remainder of this paper is organized as follows. Chapter 2 introduces our reference architecture for an SLA management framework. Chapter 3 discusses the adoption of the framework, within the Future Internet but also in general System Management environments. Chapters 4-7 cover the respective use cases and evaluation results and Chapter 8 concludes the overall discussion.

2 Reference Architecture for SLA Management

The primary functional goal of our SLA management framework is to provide a generic solution for SLA management that (1) supports SLA management across multiple layers with SLA (de-)composition across functional and organizational Domains, (2) supports arbitrary service types (business, software, infrastructure) and SLA terms, (3) covers the complete SLA and service lifecycle with consistent interlinking of design-time, planning and run-time -management aspects; and (4) can be applied to a large variety of industrial domains.

In order to achieve these goals, the reference architecture follows three main design principles:

- a clear separation of concerns,
- a solid foundation in common meta models, and
- design for extensibility.

The primary building blocks of the architecture are the SLA Manager, responsible to manage a set of SLAs and corresponding SLA templates, and the Service Manager, responsible for service realizations. Both of these are realized as generic components which can be instantiated and customized for different layers and domains.

Figure 1 illustrates an example setup of the main components of the SLA@SOI framework and their relationships for three layers: business, software and infrastructure. The framework communicates to external parties, namely customers who (want to) consume services and 3rd party providers which the actual service provider might rely upon. Relationships are defined by stereotyped dependencies that translate to specific sets of provided and required interfaces.

On the highest level, we distinguish the Framework Core, Service Managers (infrastructure and software), deployed Service Instances with their Manageability Agents and Monitoring Event Channels. The Framework Core encapsulates all functionality related to SLA management, business management, and the evaluation of service setups. Infrastructure- and Software Service Managers contain all service-specific functionality. The deployed Service Instance is the actual service delivered to the customer and managed by the framework via Manageability Agents. Monitoring Event Channels serve as a flexible communication infrastructure that allows the framework to collect information about the service instance status.

Furthermore, the framework comes with a set of well linked meta-models, namely an SLA model, a service construction model (capturing provider-internal service aspects), the service prediction model, and an infrastructure model.

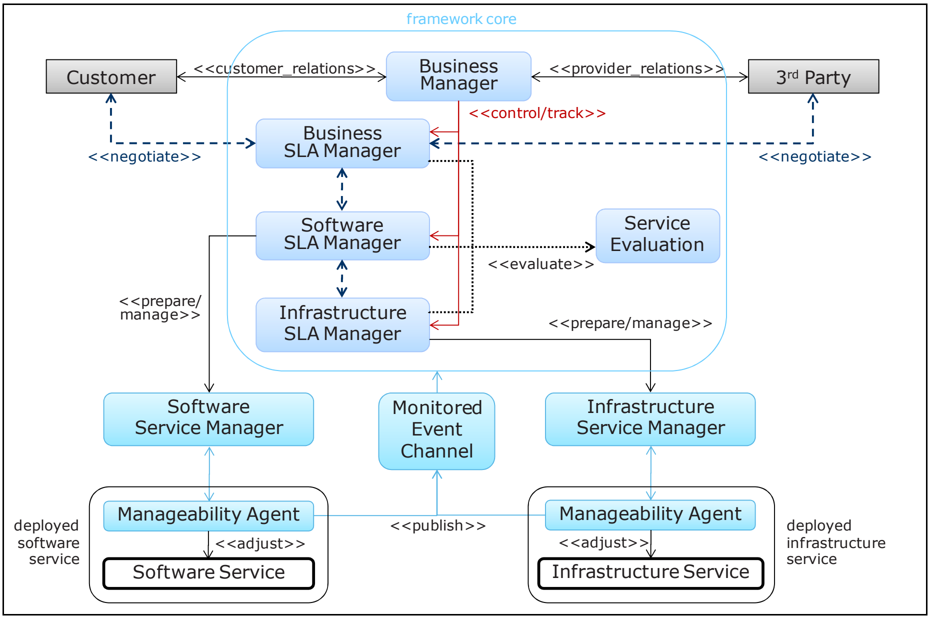

Fig. 1. Overview of the SLA Framework Reference Architecture

While all framework components come with default implementations they can also easily be extended or enhanced for more specific domain needs. Similarly, the provided meta-models come with clear extension mechanisms, e.g. to specify additional service level terms.

A typical negotiation/planning sequence realized via the framework starts with a customer querying for available products and corresponding SLA Templates. The framework then, initiates a hierarchical planning process that results in a specific offer. After agreement to an SLA offer, a provisioning sequence will be automatically triggered according to the SLA specification and will be executed in a bottom-up manner through the framework. Last, monitoring is conducted simultaneously at all framework layers; possible SLA warnings or violations either lead to local adjustment activities or are escalated to the next higher level. Further details on this architecture can be found at [2, 5].

3 Adoption Aspects

After introducing the reference architecture we now want to discuss various adoption issues. First we provide a sketch on how the architecture can be applied to the Future Internet. Second, we give an overview how SLA management relates to other management functions. Last, we provide a brief discussion on nun-functional aspects of the framework itself.

3.1 Adoption Considerations for the Future Internet

The SLA management framework architecture can easily be applied to different Future Internet scenarios. The SLA model is rich and extensible enough to be applied to e.g. infrastructure and networking resources, to sensor-like resources in the Internet of Things, to services in the Internet of Services, but also to describe people, knowledge, and other resources. Similarly, the service construction model can be adopted, which allows specification of arbitrary internal resource/service aspects.

Based on this model foundation, the framework components can be flexibly instantiated. Assuming to have Manageability Agents for the relevant artefacts in the Future Internet, a management environment consisting of SLA and Service Managers can be set up in different flavours. The setup can support autonomic scenarios, where specific SLA/Service managers are responsible for single artefacts, but also highly coordinated scenarios, where SLA/Service managers govern a larger set of entities collectively. Business aspects can be addressed at all layers by introducing a dedicated business manager. Service Evaluations can be also introduced at all layers: However there will most likely be different evaluation/optimization mechanisms for different actual domains.

Last, different framework instances can be flexibly created and connected as needed according to the requirements of the involved value chain stakeholders in the respective Future Internet scenario. In the following use-case chapters we also provide additional configuration examples of the framework.

3.2 Adoption Considerations for Cloud Computing

The SLA@SOI framework should become an intrinsic part of each cloud environment, whether it is about software-as-a-service, platform-as-a-service, or infrastructure-as-a-service. The Enterprise IT use-case (Section 3) is basically an infrastructure cloud use case that features SLA enabling. The ERP hosting use case (Section 4) contains many aspects of a software cloud.

3.3 Interlinkage with System Management

SLA-driven system management is the primary approach discussed in this paper. It actually tries to derive all kinds of management decisions from the requested or agreed service level agreements. However, there are also other management functions which are partially related to SLA management. As reference structure for these functions we use the 4 main categories of self-managing systems [3], namely self-configuring, self-healing, self-optimizing, and self-protecting.

Configuration management is closely related to SLA management. Possible configuration options are captured in the service construction model and are taken into consideration during the planning phase. Once, an SLA has been agreed and is to be provided, the configuration parameters derived during planning phase are used to set up the system. The same holds for replanning/adaptation cycles.

Self-healing is in the first place independent from SLA management as the detection and recovery from low-level unhealthy situations can be done completely independent from agreed SLAs. However, the detection of SLA violations and the automated re-planning could be also understood as self-healing process. Furthermore, low-level unhealthy situations might be used for predicting possible future SLA violations.

Self-optimizing as very closely related to SLA management and simply cannot be done without taking into account the respective constraints of the contracted SLAs.

Self-protecting is in the first place independant from SLA management. However, certain self-protecting mechanisms can be made part of an SLA.

3.4 Non-functional Properties of the SLA Framework Itself

The non-functional properties of the SLA@SOI framework play an important role for any eventual usage scenario. However, they heavily depend on the actual adoption style and cannot be simply described at the generic framework level.

The overhead introduced depends significantly on the granularity of the SLA management (how fine-grained the decomposition of an IT stack into services and SLAs is done) and the requested accuracy of the monitoring (significantly impacts on the number of events to be processed).

While the framework itself is fully scalable (in terms of its ability to run components in multiple instances), the actual scalability in a target environment depends on the number of interrelations between different artefacts. For example if many artefacts interrelate, their management cannot be easily parallelized but must be done from a central instance (e.g. a central SLA manager overseeing all SLAs in his domain).

Reliability aspects are mainly orthogonal and can be realized by state of the art redundancy mechanisms.

Last, flexibility has been a clear design goal of the framework, allowing for different setups of the framework instances which due to the underlying OSGI-based integration approach can be even changed during run-time.

4 Use Case – Enterprise IT

The Enterprise IT Use Case focuses on compute infrastructure provisioning in support of Enterprise services. We assume a virtualisation-enabled data centre style configuration of server capacity, and a broad range of services in terms of relative priority, resource requirement and longevity. As a support service in most enterprises, IT is expected to deliver application and data service support to other enterprise services and lines of business. This brings varied expectations of availability, mean-time-to-recover, Quality of Service, transaction throughput capacity, etc. A challenge for IT organisations in response to this, is how to deliver a range of service levels at the most optimal cost level. This challenge includes quick turnaround planning decisions such as provisioning and placement, run time adjustment decisions on workload migration for efficiency, and longer term strategic issues such as infrastructure refresh (in the case of internally managed clouds) and hosting service provider (external clouds).

This use case is therefore based around three distinct scenarios. The first scenario, titled "Provisioning", responds to the issue of efficient allocation of new services on IT infrastructure, SLA negotiation and provisioning of new services in the environment. The second scenario, "Run Time", deals with day-to-day, point in time operational efficiency decisions within the environment. These decisions maximise the value from the infrastructure investment. The final scenario, "Investment Governance" builds on the first two to demonstrate how they feed back into future business decisions. Taking a holistic cost view, it provides fine grained SLA based data to influence future investment decisions based on capital, security, compute power and energy efficiency.

In order to enable realistic and effective reasoning at provisioning and run time, a reference is included differentiates each of the supported Enterprise services in terms of their priority and criticality. This is the Enterprise Capability Framework or ECF.

From an implementation perspective, user interaction is via a web based UI, used by both IT customers and administrators. The Enterprise IT SLAT defines use case specific agreement terms which are loaded by the Business SLA manager to provide the inputs to provisioning requests in the form of PaaS services. Software services could potentially be selected by choosing a virtual machine template which contains pre-loaded applications, but software layer considerations are not considered core to this Use Case and are more comprehensively dealt with in the ERP Hosting Use Case. The Business SLA Manager passes service provisioning requests to the Infrastructure SLA Manager whose role is to carry out the creation of the new virtual machines which constitute the service along with monitoring and reporting for that service.

Evaluation of the framework is carried out with reference to parameters which align with IT and business priorities. The three scenarios on which the Use Case is based, are complementary and allow the framework to be assessed based on realistic objectives of an Enterprise IT function. Using Key Performance Indicators (KPIs) we evaluate the performance of the lab demonstrator in the areas of:

- IT enabling the Enterprise
- IT Efficiency
- IT Investment/Technology adoption

The Use Case identifies a hierarchy of KPIs which are measurable against established baseline and therefore result in a credible assessment of the impact of the SLA Management Framework in an Enterprise IT context.

Further details on this use case are available at [6].

5 Use Case – ERP Hosting

The ERP Hosting use case is about the dynamic provisioning of hosted Enterprise Resource Planning solutions. SLA management in this context promises great benefits to providers and customers: Providers are enabled to offer hosted solutions in a very cost-efficient and transparent way, which in particular offers new sales channels towards small and medium sized customers. Customers are enabled to steer their business in a more service-oriented and flexible manner that meets their business needs without spending too much consideration on IT matters. Furthermore, customers can flexibly negotiate the exact service details, in particular its service levels, so that they can eventually get the best fitting service for their needs.

The actual use case realizes a scenario with 4 layers of services. The top-level service considered is the so-called business solution. Such a solution typically consists of a software package (an application) but also some business-level activities, such as a support contract. At the next level, there are the actual software applications, such as for example a hosted ERP software package. At the next level, there are the required middleware components which are equally used for different applications. At the lowest layer, there are the infrastructure resources, delivered through an internal or external cloud. Each service layer is associated with a dedicated SLA, containing service level objectives which are specific to this layer. The business SLA is mainly about specifying support conditions (standard or enterprise support), quality characteristics (usage profile and system responsiveness), and the final price for the end customer. The Application SLA is mainly about the throughput capacity of the software solution, its response time, and the provider internal costs required for the offering. The Middleware SLA specifies the capacity of the middleware components, the response time guarantee of the middleware components and the costs required for the offering. The Infrastructure SLA specifies the characteristics of the virtual or physical resources (CPU speed, memory, and storage) and again the costs required for the offering.

The use case successfully applies the SLA framework by realizing distinct SLA Managers for the 4 layers and also 4 distinct Service Managers that bridge to the actual support department, the application, the middleware, and the infrastructure artefacts.

From a technical perspective, the most difficult piece in the realization of the whole use case was the knowledge discovery about the non-functional behaviour of the different components, e.g. the performance characteristics of the middleware. We collected a set of model-driven architecture artefacts, measurements, best practise rules and managed to consistently interlink them and to realize an overall hierarchical planning and optimization process. However, this process is still labour intensive and requires further automation tools in order to be applicable on a large scale.

From a business perspective the use case clearly proved tangible business benefits in different aspects. Time to market for quotation on service requests and provisioning of requested services can be significantly reduced. The dependability of provided services is increased proportional to the number of formally managed service level terms. The efficiency of service provisioning can be improved in the dimensions of environmental efficiency, resource efficiency, and process efficiency. The transparency for the end-to-end service delivery process involving different departments is largely improved due to the consistent interlinking of different views and artefacts. Last this transparency also supports an improved agility and allows for realizing changes much faster than in the past.

Further details on this use case can be found in [3]. Background information and a demo video are available at [7].

6 Use Case – Service Aggregation

The main aim of the Service Aggregation use case is the service-enabling of core Telco services and their addition with services from third parties (as Internet, infrastructure, media or content services). From the provider's point of view, they will be able to publish their services in the Service Aggregator and will be benefited in terms of reach new markets in which their services can be consumed and to be sold to the customers joined with reliable communication services offered by Telco providers. Customers can find the services and negotiate flexibly the terms of the consumption of the services included in the product. It is necessary to point out that negotiation takes place in three faces: Bank customer, Service Aggregator and Infrastructure provider. This implies the negotiation of the SLAs with quality of service aspects and the final price.

For the proof of concept we have chosen a combination of Telco capability (SMS) with a third party service (infrastructure) to build a product that is a service bundle to be offered to a Bank. There are two SLA@SOI framework instances implemented; the Service Aggregator and Infrastructure as a Service. Bank customer prototype uses several framework components mainly interfaces with Business Manager and Business SLA Manager. Service Aggregator and Infrastructure prototype have been implemented using business and infrastructure layers; additionally Service Aggregator integrates software layer (from SLA@SOI framework architecture). And finally Bank prototype is implemented using the top layer, business. Both providers utilize SLAT registries in their SLA Managers to publish the SLA templates of his services hierarchy. Business SLA template for SMS service includes some business terms like support, termination or price and other guarantee terms like availability, throughput, and response time. Communication of SLA templates between third party and Service Aggregator use advertising bus to share infrastructure templates. Then Service Aggregator can utilize local business SLA templates and third party business SLA templates to create the bundle of a product. Customer prototype is connected with Business Manager of the Service Aggregator to find and discover products; in this case it found the 'Communications and Infrastructure bundle' product. Customer retrieves the different SLA templates available for the product and the negotiation starts. Customer has to be previously registered and granted in the Service Aggregator. The negotiation is driven by the Business Manager of the Service Aggregator. It adopted provider requirements in shape of policy rules and promotions applicable to the final product. Infrastructure provider can also define the business negotiation of his services in the same way. When negotiation finished between the three actors, SLA has been signed between them in pairs and all the provision process is finished. The corresponding SLAs will be stored in SLA registries of each SLA manager used. In this way it is necessary to outline also is executed the provision of Telco web service wrappers by Software SLA Manager in an application server and also the provision of the infrastructure driven by Infrastructure SLA Manager (using the appropriate service manager). SMS wrappers deployed in the application server of the corresponding virtual machine has to connect and execute different tasks with core mobile network systems that are behind Telefónica Software Delivery Platform (SDP). The components that can be also connected in the use case are the monitors of the services (SMS and Infrastructure services). To take care about the violations, track interfaces are used to connect the adjustment components in each SLA Manager. Finally, Service Aggregator converts violations in penalties, and takes actions to adjust these violations and reports the situation to the customer.

Technical evaluation about SLA@SOI framework can be seen in a very positive way in terms of the functionality of the components and the outcome obtained by the use case. In the new ecosystems of Future internet of services the key will be the exporting and interconnection of services between different parties. It is necessary to care the service level agreements and the quality of the services guaranteed on those SLAs. SLA-aware aggregation of telecommunications services introduces a business opportunity for the agile and efficient co-creation of new service offerings and significant competitive advantages to all.

Further details on this use case including a demo video are available at [8].

7 Use Case – eGovernment

Public administrations often outsource human based services to 3rd party organizations. Such relationships are currently regulated with legal documents and human readable SLAs. The eGovernment use case aims at showing that the adoption of machine readable SLAs improves the agility and reduces costs, as it allows to automate several management activities also if the services are performed by humans.

In our proof of concept we considered a composed service allowing citizens to book medical treatments and to use and reserve at the same time the transport means for reaching the treatment place. Such a Health & Mobile Service is provided by a so called “Citizen Service Center” (CSC) and is composed of: a “Medical Treatment Service” provided by external Health Care Structures, a “Mobility Service” provided by specific Transport Providers and a “Contact Service” provided by phone call operators of the SCS and, when needed, also by a third party Call Center.. In this context, the SLA between the Government and the CSC regulates the provision of the health, mobile and contact services, as well as the expected overall satisfaction of the citizen. The SLA@SOI framework automates activities of the CSC that are usually performed manually or not performed at all, such as: the monitoring of SLAs, the allocation of the phone call operators, the dynamic selection of the mobility providers based on SLAs, the automatic re-negotiating of the SLA with the external Call Center.

The CSC fully adopts the SLA@SOI architecture. A SLA-aware BPEL Engine is used for the dynamic binding and execution of the composed service. A custom Human Service Manager allocates the human resources. A customised SLA Manager manages the negotiation with the Government and with the external Call Center. A Service Evaluation Component predicts, at negotiation time, the QoS obtainable with the available operators, in order to determine the SLA to negotiate with the external provider. At runtime the prediction feature warns about possible violations of the guarantee terms, triggering the automatic adjustment of internal human operators.

From the technical point of view, one of the main challenges of this use case has been the modelling of human-provided services, and the formalization of the strategies for handling human resources during negotiation and adjustment. This is still an ongoing task that has required several interviews with the operators working at the service providers. Also the SLAs defined in this use case have several peculiarities. For example, while typical software/hardware guarantee terms constraint the quality of each single execution of a service, in this use case the guarantee terms constraint the average value of KPIs computed for hundreds of executions measured on time periods of the order of months. Moreover, such measurements are relative to mobile time windows and periodic behaviours (consider, e.g., the difference between summer and winter in the delivery of the services considered in this use case). Extensions of the prediction model are under evaluation in order to cover new kinds of KPIs and guaranteed terms.

From the evaluation perspective, the application scenario is particularly critical due to sensitive data on the health status of the citizens and quite challenging for the key role of humans both in the provisioning and in the evaluation of the effectiveness of the platform. From the close interaction with the experts in the field, we derived an approach for the evaluation based on the feedback of the citizens and also of the operators (with focus groups and periodic interviews) in terms of: effort for the operators and citizens to interact with the system, usability and degree of acceptance of the system from the users, effectiveness of monitoring and adjustment functionalities. The results obtained from this evaluation is further integrated and compared with the trends in the real data extracted from the past behaviours of the systems at the service providers.

Overall, this scenario has been very valuable for SLA@SOI as a way to stress the project approach and core concepts to the case of human resources and human provided services.

Further details on this use case are available at [9].

8 Conclusions

Service level agreements are a crucial element to support the emerging Future Internet so that eventual services become a tradable, dependable good. The interdependencies of service level characteristics across layers and artefacts require a holistic view for their management along the complete service lifecycle. We explained a general-purpose SLA management framework that can become a core element for managing SLAs in the Future Internet. The framework allows the systematic grounding of SLA requirements and capabilities on arbitrary service artefacts, including infrastructure, network, software, and business artefacts. Four complementary industrial use cases demonstrated the applicability and relevance of the approach. Furthermore, the diverse and complementary nature of the Use Cases along with the consistent and structured evaluation approach ensures that the impact assessment is credible.

Future work concentrates on three aspects. Technology research will be deepened on the areas of SLA model extensibility, quality model discovery, business negotiation, and elastic infrastructure scaling. Use Case research will tackle additional scenarios, especially relevant for the Future Internet. Last, we plan to open up our development activities via an Open Source Project. The first framework version fully published as open source can be found at [5].

References

1. SLA@SOI Project Web Site. URL: http://www.sla-at-soi.eu
2. Theilmann, W., Happe, J., Kotsokalis, C., Edmonds, A., Kearney, K., Lambea, J.: A Reference Architecture for Multi-Level SLA Management. Journal of Internet Engineering 4(1) (2010), http://www.jie-online.org/ojs/index.php/jie/issue/view/8
3. Miller, B.: The autonomic computing edge: Can you CHOP up autonomic computing? Whitepaper IBM developerworks (March 2008), http://www.ibm.com/developerworks/autonomic/library/ac-edge4/
4. Theilmann, W., Winkler, U., Happe, J., Magrans de Abril, I.: Managing on-demand business applications with hierarchical service level agreements. In: Berre, A.J., Gómez-Pérez, A., Tutschku, K. (eds.) FIS 2010. LNCS, vol. 6369, Springer, Heidelberg (2010)
5. SLA@SOI Open Source Framework. First full release by December 2010, http://sourceforge.net/projects/sla-at-soi
6. SLA@SOI project: Enterprise IT Use Case. http://sla-at-soi.eu/research/focus-areas/use-case-enterprise-it/
7. SLA@SOI project: ERP Hosting Use Case. http://sla-at-soi.eu/research/focus-areas/use-case-erp-hosting/
8. SLA@SOI project: Service Aggregator Use Case, http://sla-at-soi.eu/research/focus-areas/use-case-service-aggregator/
9. SLA@SOI project: eGovernment Use Case, http://sla-at-soi.eu/research/focus-areas/use-case-e-government/