



SLAs Empowering a Dependable Service Economy



Project no. FP7- 216556

Instrument: Integrated Project (IP)

Objective ICT-2007.1.2: Service and Software Architectures, Infrastructures and Engineering

Deliverable D.A1a

Framework architecture

- evaluated architecture

Keywords:

Service Level Agreement, Service-Oriented Infrastructure, Architecture, Models

Due date of deliverable: 31st July 2011

Actual submission to EC date: 29th July 2011

Start date of project: 1st June 2008

Duration: 38 months

Lead contractor for this deliverable: SAP

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	Yes



Document Status	
Deliverable Lead	Wolfgang Theilmann
Reviewer 1	Sam Guinea
Reviewer 2	Franz Brosch
PMT Reviewer	Ramin Yahyapour
Complete version submitted to reviewers	June 17, 2011
Comments of reviewer 1 received	June 25, 2011
Comments of reviewer 2 received	June 29, 2011
Revised deliverable submitted to PMT	July 4, 2011
PMT Approval	July 14, 2011

Contributors	
Partner	Contributors
SAP	Wolfgang Theilmann, Jens Happe, Wasif Gilani
ENG	Francesco Torelli, Keven Kearney
TID	Juan Lambea, Beatriz Fuentes
XLAB	Miha Stopar
PMI	Sam Guinea
Intel	Andrew Edmonds, Michael Nolan
FZI	Franz Brosch
UDO	Peter Chronz, Costas Kotsokalis

Notices
<p>The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright 2010 by the SLA@SOI consortium.</p>
<p>* Other names and brands may be claimed as the property of others.</p>



This work is licensed under a [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/).

Document History			
Version	Date	Author	Changes
1.0	July 25	Wolfgang Theilmann	Final version

Executive Summary

This document describes the innovative architecture for the comprehensive SLA management framework created by the European research project SLA@SOI. It serves as formal deliverable D.A1a for reporting the work progress of work package A1 at project month 38. It is complemented by the deliverable D.A1b which provides the actual implementation of the SLA framework.

This architecture directly responds to the mission of SLA@SOI: “to deliver and showcase an innovative open SLA Management Framework that provides holistic support for service level objectives - enabling an open, dynamic, SLA-aware market for European service providers. SLAs will be managed autonomously throughout the complete service lifecycle, spanning the entire services stack from the business layer through to infrastructure. Arbitrary domains will be supported, as demonstrated by evaluations in wide-ranging, grounded, use cases”. Furthermore it integrates the contributions on the four top-level technical objectives of (1) Consistent SLA-management framework, (2) Adaptive SLA-aware infrastructure, (3) Engineering predictable service-oriented systems, and (4) Comprehensive business management suite for e-contracting into one consistent architecture.

The reference architecture definition was driven by the requirements of four industrial use cases but also by other internal and project-external stakeholders.

The reference architecture represents a key innovation of SLA@SOI as it realizes the first comprehensive architecture of a consistent SLA-management framework. Four main novelties can be highlighted: (1) the architecture supports multi-layered SLA management where SLAs can be composed and decomposed along functional & organizational domains; (2) it supports arbitrary service types (business, software, and infrastructure) and SLA terms; (3) the architecture covers the complete SLA and service lifecycle with consistent interlinking of design-time, planning and run-time management aspects; (4) the actual implementation supports flexible deployment setups, where actual components can be flexibly selected, extended and connected and where foundational data models can be extended.

This document together with its appendices provides a complete overview of the framework architecture including a high-level overview, a detailed discussion of relevant foundational concepts, a description of the modelling foundation i.e. the most important metamodels that are shared between different components of the architecture, and the actual architecture overview with its building blocks, components, interactions as well as supporting tools. Furthermore, it introduces an adoption guide that helps framework users to best realize the concepts for their target environment.

The technical architecture has been successfully implemented into an SLA management software framework. Details about the technical/scientific evaluation of the SLA framework can be found in deliverable D.B1a. Furthermore, the software framework has been successfully applied against four industrial use cases in the domains of ERP hosting, Enterprise IT management, Telco Service Aggregation and eGovernment.

The reference architecture has been prominently published in international conferences, journals, and books [4], [5], [6], [7].

Table of Contents

1	Introduction	9
1.1	Context and Scope	9
1.2	Document Overview	10
2	Contribution Overview	11
2.1	Key Innovations.....	11
2.2	Framework Contributions	11
2.3	Additional outputs.....	12
2.4	Task-level activities in 3rd project year.....	12
3	Reference Architecture and Adoption Guide	14
3.1	Objectives.....	14
3.2	Reference Architecture Overview	14
3.3	Adoption Guide Overview	15
4	Conclusions	16
4.1	Contributions and Achievements	16
4.2	Lessons learned	16
4.3	Outlook	17
5	References	19
	Appendix A: Glossary	20
	Appendix B: Abbreviations	23
	Appendix C: Reference Architecture Whitepaper.....	23
	Appendix D: Adoption Guide	23

List of Tables

Table 1: Document changes against previous version 10

1 Introduction

1.1 Context and Scope

This document describes the reference architecture for the SLA management framework created by the European research project SLA@SOI. The architecture definition includes foundational concepts on terms and lifecycles, a modelling foundation and the actual architecture in terms of components and interactions. In addition to the architecture specification, the document also provides an adoption guide – a methodology that supports a systematic decision process in order to define an appropriate instantiation of the architecture for a given situation.

This document serves as formal deliverable D.A1a for reporting the work progress of work package A1 during the third phase of the project. It is complemented by the deliverable D.A1b which provides the actual implementation of the SLA framework. Further insight into specific modules of the overall framework is given within the context of the other scientific work packages A2-A6 via deliverables D.A2a-D.A6a. The technical/scientific evaluation of the SLA framework is subject to deliverable D.B1a which also features the positioning of the framework against state-of-the-art including different IT management approaches. Use case-driven evaluations can be found in deliverables D.B2b and D.B3c-D.B6c. Last, a tutorial that applies the adoption guide to the open reference demonstrator is provided within D.B2b. Information on dissemination and exploitation activities regarding the architecture can be found in D.B8a, D.B8c, and D.B9b.

The reference architecture supports a clear 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. Service Managers are responsible for all management activities directly related to services. This includes the management of information about available services, supported types of services, as well as their offered functionality and their dependencies. SLA Managers are responsible for all actions that are related to the service-level agreements. They are involved in the negotiation with customers and they are responsible for the planning and optimization of new services that are to be provisioned. Furthermore, they monitor the terms a provider and customer have agreed upon and react in case of violations. SLA Managers can negotiate with each other in order to find the best offer for a customer. The provisioning of a service is a joint effort of all SLA Managers and Service Managers involved. In order to support multiple domains with our framework, multiple SLA managers and multiple Service Managers can collaborate inside the framework as well as across framework boundaries. Thereby, each SLA Manager and Service Manager is responsible for SLAs and services of a particular domain. Last, the framework also contains a set of tools that support the design, development and test of SLA-aware services. While those tools (e.g. Performance Cockpit, PCM, Manageability tools) are technically standalone, their results are directly needed to feed certain model artefacts and design decisions in the core framework.

The adoption guide provides a methodology for users of the SLA@SOI framework (such as use case developers) on how to best apply the overall framework and how to embed/enhance it into their target environment. The guide follows a systematic decision process covering six main phases, namely service analysis, system analysis, architecture specification, data model preparation, custom development, and framework configuration & setup.

Main achievement

The main achievement of this work package is the successful **design and implementation of a consistent SLA-management framework**, which integrates all SLA management related activities of the other scientific work packages into one consistent architecture.

The reference architecture has been prominently published in an international journal [6] and as a book chapter [5]. Furthermore, reference architecture and implementation have been published as open source project [1].

The reference architecture gained much interest by other stakeholders (SMEs, projects, initiatives) which plan to adopt it to their target environment/needs.

1.2 Document Overview

The remainder of this document is structured as follows. Chapter 2 provides an overview of the contributions made by this work package in terms of innovations, actual framework development, and also a task-level overview. Chapter 3 provides an overview of the two main whitepapers that have been produced in the context of the reference architecture, namely the architecture specification and the adoption guide. Chapter 4 concludes with a short summary, a discussion of lessons learned, and an outlook on future work.

The following table lists all the changes and new elements of this deliverable compared to the previous deliverable version.

Table 1: Document changes against previous version

Section	Change overview
Chapter 1	Significant revision to reflect overall project's lifetime
Section 2.3	New for current project phase
Chapter 3	new
Chapter 4	Revised; lessons learned completely new
Annex C	Largely based on year 2 deliverable; major revision regarding positioning of monitoring system and related interaction stereotypes
Annex D	new

2 Contribution Overview

This chapter provides an overview of the contributions that have been made by the work package A1 in terms of key innovations and contributions to the overall framework. Furthermore it summarizes the actual activities and progress at task-level for the reporting period.

2.1 Key Innovations

There is one single key innovation provided by work package A1: the **comprehensive reference architecture of a consistent SLA-management framework**.

This innovation directly responds to the mission of SLA@SOI: “to deliver and showcase an innovative open SLA Management Framework that provides holistic support for service level objectives - enabling an open, dynamic, SLA-aware market for European service providers. SLAs will be managed autonomously throughout the complete service lifecycle, spanning the entire services stack from the business layer through to infrastructure. Arbitrary domains will be supported, as demonstrated by evaluations in wide-ranging, grounded, use cases”. Furthermore, the A1 work package integrates the project’s contributions on the four top-level technical objectives of (1) Consistent SLA-management framework, (2) Adaptive SLA-aware infrastructure, (3) Engineering predictable service-oriented systems, and (4) Comprehensive business management suite for e-contracting into one consistent architecture.

The main novelty of this architecture consists of the following aspects:

1. Support for multi-layered SLA management where SLAs can be composed and decomposed along functional & organizational domains.
2. Support for arbitrary service types (business, software, and infrastructure) and SLA terms.
3. Support for the complete SLA and service lifecycle with consistent interlinking of design-time, planning and run-time management aspects.
4. Support for flexible deployment setups, where actual components can be flexibly selected, extended and connected and where founding data models (such as service construction model and the SLA model) can be extended.

More details on this innovation and it’s linkage to the state of the art can be found in [8], [9] respectively.

The reference architecture has been prominently published in international conferences, journals, and books [4], [5], [6], [7].

2.2 Framework Contributions

The actual contribution of this work package to the SLA@SOI framework is primarily of conceptual nature, namely the specification of foundational concepts and their relations, the specification and harmonization of basic metamodels, and the actual architecture specification in terms of building blocks and their interactions.

The analysis of the various data models and views provided by other technical work packages lead to the identification of a conceptual gap which hindered the proper interlinkage of different framework components, namely SLA managers, service managers and service evaluation components. To close this gap, work package A1 also specified and implemented the “service construction model”.

It is important to stress that the development of our framework components is done in a way that also supports the selected reuse of single components independently from the overall framework. Thus users can either adopt the complete framework or take it as a toolbox and select those elements relevant for them.

Furthermore, the work package provides an adoption guide, i.e. a methodology for systematic adoption of the framework in any kind of target scenario.

Last, this work package also included an analysis of externally available components that could be reused for our framework. The reuse of lower-level technology components (for logging, modelling, etc.) is further described in deliverable D.A1b [3]. The reuse of higher-level components related to SLA management was analysed in depth, in particular regarding the results provided on the platforms IT-Tude.com and gridsforbusiness.eu. Conceptually, both platforms provided very valid input to our framework architecture. However, the actual reuse of technical components was not reasonable, because of different scope (focus on grid services), conflicting licences (e.g. usage of GPL), and different technology decisions (e.g. usage of MS Visual Studio). Other components (such as Licence Management, Resource Selection Optimization, SLA Accounting for GT4, or Application Virtualization) could be well used as a complement to our framework for specific setups. Further details on this assessment are provided in [9].

2.3 Additional outputs

The main additional output of this work package, achieved together with all the other scientific work packages, was the set up of an **open source project** for the SLA@SOI management framework [1].

2.4 Task-level activities in 3rd project year

The main activities and their progress at task level are summarized below.

Task *A1.1 Terminology Alignment* cross-checked incoming feedback regarding the glossary. However, the glossary turned out to be pretty stable. Hence no major changes were required. In addition, this task contributed to the adoption guide (see Appendix D) by aligning the terminology used there to the project glossary.

Task *A1.2 Service Integration* performed a revision of the way how the monitoring system and its services are exposed within the reference architecture. A revision on the positioning was done including a refinement of the respective services and interaction stereotypes (see Appendix C).

Task *A1.3 Model Integration* continued the cross-checking of different meta-models. In particular an integration analysis on the newly suggested SLA model was done. Furthermore, contributions on modelling concepts were done for the adoption guide and the tutorial (part of D.B2b).

Task *A1.4 Architecture Definition* was the main driver behind the newly introduced adoption guide, which has been designed in close interlinkage with the concepts of the reference architecture. The result of this activity and the general architecture revisions are described in Chapter 3.

Task *A1.5 Technology Integration* continued the technical integration work for developing the full SLA management framework in the reporting period. Furthermore, it acted as key driver behind the open source initiative for the project (more details on this can be found in the exploitation deliverable D.B8a).

3 Reference Architecture and Adoption Guide

3.1 Objectives

The dissemination and exploitation goal of SLA@SOI is to provide its results to the widest possible audience and to allow interested stakeholders to adopt the results for further research activities or industrial usage.

One element in this dissemination strategy is the production of public whitepapers that can serve as entry point for interested stakeholders.

The first entry point is supported by the reference architecture whitepaper (see Appendix C). It addresses scientific and technical stakeholders that want to understand the concepts behind the SLA@SOI architecture, including its models, building blocks, algorithms, and detailed component specifications.

The second entry point is supported by the adoption guide (see Appendix D). It addresses potential framework users that want to apply the SLA@SOI framework in their target environment. It offers a systematic methodology from very high-level aspects around the service design down to very concrete aspects of framework configurations.

3.2 Reference Architecture Overview

The reference architecture whitepaper provides a complete overview of the concepts behind the SLA@SOI framework.

It consists of the following main chapters:

1. An overview of the context and the motivation under which the architecture has been developed.
2. A detailed introduction of the actual scope and foundational concepts, including core concepts of service and SLA, lifecycles, management domains, and building blocks.
3. The actual architecture specification including a top-level component diagram, the modelling foundation and the detailed specification of the core components.
4. A detailed specification of the interactions and protocols between different components.
5. A synthesizing overview on the end-to-end integration scenarios for negotiation, provisioning, adjustment, and renegotiation.
6. The specification of the overarching manageability approach including monitoring and adjustment concepts.
7. Adoption examples from three representative use cases, including ERP Hosting, Enterprise IT, and e-Government.

Furthermore, the whitepaper provides the relevant references to detailed concepts and algorithms (as described in deliverables or scientific publications) and also references to online component documentations within the Open Source project (component designs, Java documentation, deployment instructions).

The full reference architecture whitepaper can be found in Appendix C.

3.3 Adoption Guide Overview

The adoption guide provides information for users of the SLA@SOI framework (such as use case developers) on how to best apply the overall framework and how to embed/enhance it into their target environment. The guide attempts to clarify step by step the questions/issues use case adopters have to resolve when realizing an SLA management based on the SLA@SOI framework in their domain.

The adoption model covers the following phases

1. *Service Analysis*, where the general service offering is to be clarified.
2. *System Analysis*, where the capabilities and constraints of the underlying system are to be clarified.
3. *Architecture Specification*, where the general SLA management architecture is to be defined.
4. *Data Model Preparation*, where the required meta-models and actual model data is to be specified.
5. *Custom Development*, where the development of custom components is to be analysed.
6. *Framework Configuration & Setup*, where the overall configuration and setup of the framework shall be defined.

Each of these phases is further detailed into specific steps. For each step, the guide clearly specifies the required activities and the expected results.

The full adoption guide can be found in Appendix D.

4 Conclusions

Looking at the IT market evolution, cloud computing has become one of the most important trends. The paradigm of cloud computing started as delivery model for infrastructure resources but has become a very general delivery model for infrastructure, platform, software, and other kinds of services (sometimes denoted as XaaS). The delivery of services along the cloud paradigm exactly meets the anticipation of SLA@SOI of an “evolution towards a service-oriented economy, where IT-based services can be flexibly traded as economic good, i.e. under well-defined and dependable conditions and with clearly associated costs.”[10] Multiple reports highlighted the relevance of SLAs for the evolution of the cloud market (e.g. [11]).

4.1 Contributions and Achievements

The main contribution of this work package is the successful design and implementation of a consistent SLA-management framework, which supports the complete SLA management lifecycle including design, negotiation, provisioning, monitoring, and adjustment of SLA-aware services.

The framework comes as a pre-integrated environment to support negotiation, provisioning, monitoring, and adjustment of SLA-aware services across the domains of business, software, and infrastructure services. Furthermore, it delivers a set of tools to support the design, development and deployment of predictable SLA-aware services.

The realized framework fully achieves the integration of the four top-level technical objectives of (1) Consistent SLA-management framework, (2) Adaptive SLA-aware infrastructure, (3) Engineering predictable service-oriented systems, and (4) Comprehensive business management suite for e-contracting into one consistent architecture.

The framework has been widely published and disseminated as open source project. The adoption has been facilitated by dedicated guidelines and tutorials and community building with industrial and research partners have been actively driven.

4.2 Lessons learned

Research and development on this framework has been a great experience which provided a couple of lessons learned.

Lesson 1: The conception of an integrated, consistent framework is a highly valuable asset.

The topic of integrated SLA management is a highly complex one as it interlinks multiple views, layers, and stakeholders. Typically all our stakeholders initially feel to have a good understanding of SLA management. As soon as the discussion starts it turns out that they just consider SLAs from a certain viewpoint. In this context, the integrated conception developed by the project is a highly valuable asset. This feedback has been also supported by the advisory board.

Lesson 2: The framework architecture is comprehensive and future proof.

The adoption of the framework architecture was well possible in all the industrial use cases investigated by the project. Furthermore, also the feedback by external parties (projects, partners, advisory board) revealed no major missing piece. Instead, we received explicit feedback that the framework can “be considered a

'future proof' solution, able to 'embed' recent and future advances in the field of definition and control for SLA on services".

Lesson 3: Strong architecture governance is needed to achieve harmonized results.

The development of a coherent and integrated architecture for a project of this size and complexity is highly challenging. In addition to have good work package leads it proved to be very important to set up a small architecture governance board which had the authority to further evolve the architecture and to agree/disagree to individual proposals.

Lesson 4: Teaching on joint engineering capabilities is important.

The technical engineering background of the individuals in such a project is quite diverse. Explicit resources should be dedicated for teaching all project members on a common engineering approach including both process and tool / technology capabilities.

Lesson 5: A good balance between internal and external communication is needed.

Throughout the course of such a project it turns out that specific communication styles are developed which are highly efficient for communication within the project. On the other hand these styles are sometimes not compatible with externally known communication/documentation patterns. Therefore, specific emphasis must be put on establishing a good balance between these different needs.

Lesson 6: Integration at pre-industrial quality demands many resources.

The solid integration of the different contributions and the offering of a framework in pre-industrial quality demand much more resources than initially planned. These amount of resources needed here should be better reflected into the project plan from the beginning.

4.3 Outlook

The endeavour of SLA-driven service and system management is an ongoing journey which requires subsequent research into many areas.

While SLA@SOI provided the general conceptions and validated this in multiple use cases, there are still many open issues to be resolved.

In particular, SLA-driven service management must be specialized into other and new domains. As each domain comes with specific constraints on the predictability, monitorability, and adjustability, specific solutions need to be developed for those domains. A few prominent domains which we can already foresee right now are:

- Platform-as-a-Service (PaaS) environments, where higher level software services (for design and operation of applications) are provided. Here, specific mechanisms for deployment support and adjustment could and should be realized – based on the SLA@SOI framework.
- A specific aspect in the PaaS context is the SLA-driven engineering that could be achieved in a much more controlled way compared to traditional on-premise engineering approaches.
- Integration infrastructure management encompassing storage, network and compute resources is another area where further in-depth research is needed. A specific aspect herein is the integration of elasticity mechanisms with SLA adjustment capabilities.

- Hybrid cloud environments including different cloud providers and encompassing private and public clouds is another domain of relevant follow up activities.

Ultimately, SLA support should be embedded into Future Internet Scenarios. We already provided an initial analysis how a starting point for the adoption of the SLA@SOI framework could look like. However, the full vision and its related flexibility will require a couple of additional mechanisms in order to fully realize the SLA@SOI framework in this environment.

5 References

- [1] SLA@SOI project on SourceForge. URL: <https://sourceforge.net/projects/sla-at-soi/>
- [2] SLA@SOI project: *Project Web Site*. July 2011: URL: <http://www.sla-at-soi.eu/>
- [3] SLA@SOI project: *Deliverable D.A1b Integrated Framework – full lifecycle*. September 2010
- [4] W. Theilmann, R. Yahyapour, J. Butler, “Multi-level SLA Management for Service-Oriented Infrastructures”, In: Proc. of ServiceWave 2008 conference, 10.12.-13.12.2008, Madrid, URL: <http://www.servicewave.eu>, P. Mähönen, K. Pohl, and T. Priol (Eds.): ServiceWave 2008, LNCS 5377, pp. 324–335, 2008, Springer-Verlag Berlin Heidelberg 2008
- [5] Wolfgang Theilmann, Gabriele Zacco, Marco Comuzzi, Christoph Rathfelder, Constantinos Kotsokalis, and Ulrich Winkler: *A Framework for Multi-level SLA Management*. Book chapter in: Handbook of Research on Non-Functional Properties for Service-Oriented Systems: Future Directions, 2011
- [6] Wolfgang Theilmann, Jens Happe, Costas Kotsokalis, Andy Edmonds and Keven Kearney: *A Reference Architecture for Multi-Level SLA Management*. Journal of Internet Engineering (JIE), Vol 4, Nb 1, Dec 2010
- [7] Joe Butler, Juan Lambea, Michael Nolan, Wolfgang Theilmann, Francesco Torelli, Ramin Yahyapour, Annamaria Chiasera, Marco Pistore: *SLAs Empowering Services in the Future Internet*. Edition of the Future Internet Assembly book: Future Internet: Achievements and Promising Technology, May 2011
- [8] SLA@SOI project: *Scientific Goals Achievements and Plans – V2 – M26*. Annex of: *Project Management Report*. September 2010
- [9] SLA@SOI project: *State of the Art Analysis*. Annex of: *Deliverable D.B1a Scientific Evaluation Report*. September 2010
- [10] SLA@SOI project: *Description of Work*. November 14th, 2007
- [11] Burton Group: *Market Profile: Hardware Infrastructure as a Service (HIaaS)*. Version: 1.0, Jan 06, 2010
- [12] SLA@SOI Source Forge Project: Reference Architecture for an SLA Management Framework.
URL: https://sla-at-soi.svn.sourceforge.net/svnroot/sla-at-soi/platform/trunk/doc/SLA@SOI-Reference_Architecture.pdf
- [13] SLA@SOI Source Forge Project: SLA@SOI Adoption Guide.
URL: https://sla-at-soi.svn.sourceforge.net/svnroot/sla-at-soi/platform/trunk/doc/SLA@SOI-Adoption_Guide.pdf

Appendix A: Glossary

The following list shows the most important entries of the SLA@SOI glossary. Note that terms that are specific for the current document and not part of the overall project wide glossary are marked with an asterix *.

Agreement Initiator	An agreement initiator is a party to a <i>service level agreement</i> . The initiator creates and manages an agreement on the availability of a service on behalf of either the service customer or service provider, depending on the domain-specific signalling requirements.
Agreement Offer	An offer is the description of the agreement relationship that is sent from <i>agreement initiator</i> to <i>agreement responder</i> during agreement creation, indicating the relationship which the initiator would like to form.
Agreement Responder	The agreement responder is a party to a <i>service level agreement</i> . The responder implements and exposes an agreement on behalf of either the service provider or service customer, depending on the domain-specific signalling requirements.
Agreement Template	An agreement template is an XML document used by the <i>agreement responder</i> to advertise the types of offers it is willing to accept.
Agreement Term	Agreement terms define the content of a <i>service level agreement</i> .
Business Service	A business service is exposed/invoked via at least some non IT elements.
Business Manager	A specialization of <i>service provider</i> : person that defines the SLATs of products and joins available services in a product.
External Service	External services are exposed across the boundaries of an organization, i.e. across at least two administrative domains.
Framework Administrator	A specialization of <i>service provider</i> : person that configures/adapts the SLA@SOI framework for a specific application.
Guarantee Term	Guarantee terms define the assurance on service quality associated with the service described by the service definition terms. They refer to the service description that is the subject of the agreement and define service level objectives, qualifying conditions and business value expressing the importance of the service level objectives.
Hybrid Service	A hybrid service is a set or bundle of other services where all these services are exposed to the customer but have different service interface types (e.g. an IT service and a business service).
Infrastructure Manager	A specialization of <i>infrastructure provider</i> : person/system that is interested to measure and control infrastructure properties.
Infrastructure Provider	A specific kind of service provider that focuses on the provisioning of <i>infrastructure services</i> .

Infrastructure Service	An infrastructure service is a specific <i>IT service</i> which exposes resource/hardware-centric capabilities.
Internal Service	Internal services are exposed within the boundaries of an organization, i.e. within one administrative domain.
IT Service	An IT service is exposed/invoked by means of information technology. Specific classes of IT services may be software services, infrastructure services or media services.
Offered Service	An abstract service (more precisely: service type) which is offered by a specific <i>Service Provider</i> to its <i>Service Customers</i> .
Operation Level Agreements	A specification of the conditions under which an <i>internal service</i> or a component is to be used by its "customer".
Service	A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks. See also <i>service interface type, service concreteness, service exposure</i>
Service Concreteness	The stage a service reaches over time from a fully abstract type to actually instantiated. See also <i>service type, offered service, service implementation, service instance</i>
Service Consumer	Person(s) who actually consume/use the provided services. Typically they belong to the <i>service customer</i> .
Service Customer	Someone (person or group) who orders/buys services and defines and agrees the service level targets.
Service Description Term	Service Description Terms describe the functionality that will be delivered under the <i>service level agreement</i> . The agreement description may include also other non-functional items referring to the service description terms.
Service Exposure	Services can be exposed either internally (within the same administrative domain) or externally. See also <i>internal service, external service</i>
Service Implementation	A service implementation is a possible concrete realization of a given <i>service type</i> .
Service Instance	A concrete realization of an <i>offered service</i> which is ready for consumption by service users. It relies on the instantiations of all the resources required for a given <i>service implementation</i> .
Service Interface Type	Describes the nature of an actually exposed service, i.e. about the nature of his invocation interface. See also <i>business service, IT service, hybrid service</i>
Service Level Consequence	An action that takes place in the event that a service level objective is not met.
Service Level Agreement	An agreement defines a dynamically-established and dynamically managed relationship between parties. The object of this relationship is the delivery of a service by one of the parties within the context of the agreement. The management of this delivery is achieved by agreeing on the respective roles, rights and obligations of the parties. The agreement may specify not only functional properties for identification or creation of the service, but also non-functional properties of the service such as performance or

	availability. Entities can dynamically establish and manage agreements via Web service interfaces.
Service Level Objective	Service Level Objective represents the quality of service aspect of the <i>agreement</i> . Syntactically, it is an assertion over the agreement <i>terms</i> of the agreement as well as such qualities as date and time.
Service Provider	An organization supplying services to one or more internal customers or external customers.
SLA Manager	A specialization of <i>service provider</i> : person/system that is responsible for managing SLATs and SLA relationships.
Software Designer	A specialization of <i>software provider</i> : person that designs/develops the architecture and components of a specific SLA based application.
Software Manager	A specialization of <i>service provider</i> : person that defines software-based services, takes care of their management and supports the SLA manager in creating appropriate SLA templates.
Software Provider	An organization producing <i>software components</i> which might be used by a <i>service provider</i> to assemble actual <i>services</i> .
Software Service	A software service is a specific <i>IT service</i> which is exposed/invoked by means of software entities such as Web services, user interfaces, or software-based business processes.
Software Component	Software components are the entities produced at design-time by a <i>software provider</i> .
Service Type	A service type (or abstract service) specifies the external interface of a service possibly including non-functional aspects. It does not specify any means (components, resources) which are needed for the actual provisioning of that service.

Appendix B: Abbreviations

ERP	Enterprise Resource Planning
XaaS	Anything as a Service
SLA	Service Level Agreement
SLAT	Service Level Agreement Template
SME	Small and Medium-sized Enterprise

Appendix C: Reference Architecture Whitepaper

The reference architecture whitepaper is served as separated appendix to this document. The most current version can be retrieved from the SourceForge project [1] at [12].

Appendix D: Adoption Guide

The adoption guide is served as separated appendix to this document. The most current version can be retrieved from the SourceForge project [1] at [13].