Overview of SLA@SOI and its framework architecture

SLAs empowering a dependable service economy

Dr. Wolfgang Theilmann (SAP)
SAP Research
The service marketplace is frustrating and cumbersome for both service providers and consumers.
## Motivation: Challenges, Goals, Vision

<table>
<thead>
<tr>
<th>Service Consumer</th>
<th>Flexible usage of dependable services</th>
</tr>
</thead>
<tbody>
<tr>
<td>• lack formal SLAs</td>
<td>• search standardized way to negotiate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Provider</th>
<th>Engineering of predictable services</th>
</tr>
</thead>
<tbody>
<tr>
<td>• struggle to understand the behaviour of their SOA stack</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Provider</th>
<th>Automated SLA negotiation and management</th>
</tr>
</thead>
<tbody>
<tr>
<td>• challenged to provide customized, dependable services at low cost</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Infrastructure Provider</th>
<th>SLA enforcement via adaptive infrastructures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• virtualization technologies allow for adaptive SOIs</td>
<td></td>
</tr>
</tbody>
</table>

### Vision of SLA@SOI

A n invigorated European economy thriving on a market of dependable services empowered by SLAs
Mission & Benefits

Mission

- 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

Expected Benefits

- **more dynamic:**
  - reduced preparation / setup time
- **more dependable**
  - through holistic SLA support
- **more automated and thus cost efficient**
  - automation of service management procedures
- **more flexible**
  - simplifying the adjustment or reprovisioning
- **more transparent**
  - understanding cost drivers and non-functional properties
Main innovations

- **SLA management framework**
  - harmonizing perspectives of relevant stakeholders (software/service/infrastructure provider and customer)
  - standards for SLA specification and negotiation & systematic multi-layer SLA management (planning, optimization, and provisioning), monitoring and accounting
  - guaranteed QoS in a dynamic and end-to-end fashion via consistent SLA handling across IT stack

- **adaptive SLA-aware infrastructures**
  - standardized interfaces for adaptive infrastructures with harmonized access to different virtualization technologies.
  - advanced technologies for SLA enforcement on infrastructure level
  - efficient resource usage w/ reliable SLA enforcement at infrastructure level

- **engineering methods for predictable service-oriented systems**
  - modelling techniques and prediction tools for SOA and SOI components

- **business management suite for e-contracting**
  - covers complete business lifecycle of a service provisioning/delivery
Main project results

**Open Source**

**SLA Core Architecture**

**Reference Implementation**

- **ERP Hosting**
  - ERP as a service
  - business value chains

- **Enterprise IT**
  - dynamic comprehension of service stack provisioning and business value

- **Serv. Aggreg.**
  - TaaS platform (telco)
  - aggregation across multiple providers

- **eGoverment**
  - agreements driven by social aspects (not market logics)
  - human services

**Standardization**

- DMTF
- IETF

**Reference demonstrator**

**Industrial Evaluation Report: “How to run an SLA-driven business”**
Key Features

Reference architecture for an SLA-management framework

1. multi-layered SLA management
2. arbitrary service types
3. complete SLA and service lifecycle
4. flexible deployment setups

What is a service in our scenario?

Which core functionality is needed?

How to support layers?

How to specialize and extend?
What is a Service?

Service

Dependancy
• e.g. DB impl. relies on block storage

Service (abstract)
• e.g. SQL DB

Service Implementation A
• e.g. MySQL DB

Service Implementation B
• e.g. Lucene DB

Service Instance 1
• e.g. MySQL instance 1 – tuned for read access

Service Instance 2
• e.g. MySQL instance 2 – tuned for write access

→ Sound understanding of service artefacts & relations
Which Functions are needed?

- **SLA Manager**
  - Manages SLAs & templates for services & instances

- **Service Manager**
  - Manages service implementations

- **Manageability Agent**
  - Manages service instances

- **Service Evaluation**
  - Prediction of quality aspects

Customer

- Negotiate

- Queries & orchestrates

- Uses

Few Building Blocks with clear responsibility
Example Setup: Web Shop as a Service

Flexible Framework Setups for different Domains
Altogether: Reference Architecture

- Customer
- Business Manager
- 3rd Party
- Business SLA Manager
- Service Evaluation
- Software SLA Manager
- Infrastructure SLA Manager
- Software Service Manager
- Monitored Event Channel
- Infrastructure Service Manager
- Manageability Agent
- Software Service
- Manageability Agent
- Infrastructure Service

Relations:
- <<customer_relations>>
- <<provider_relations>>
- <<negotiate>>
- <<control/track>>
- <<evaluate>>
- <<prepare/manage>>
- <<publish>>
- <<adjust>>
**Generic core elements**

- Generic SLA Manager
  - with abstract planning & provisioning module
- Generic Interfaces
  - Service Manager
  - Manageability Agent
  - Service Evaluation

**Domain-specific extensions**

- Business-layer
  - Business Manager (customer & provider relations)
  - Business SLA Manager (interweaving business logic into SLA management)
- Software-layer
  - Software SLA Manager, Service Manager, Service Evaluation
- Infrastructure / Cloud-layer
  - Infrastructure SLA Manager, Service Manager (OCCI compliant)

> A stable framework core with flexible extensions
Specialization Example: Managing heterogeneous Infrastructures

Provider perspective

Generic abstraction supporting multiple ways of managing heterogeneous infrastructures
Integrated Solutions for Requirement Prediction

Evaluation of Software Services

On-Demand Provisioning Support

- Machine learning techniques for run-time usage
- Continuous evaluation of prediction quality: training and testing
Conclusions

Reference Architecture

- Provides glue for consistent harmonization of all project innovations
- Covers design-time and run-time
- Relies on common modelling basis
  - SLA model, service construction model, infrastructure model (OCCI), prediction models (PCM)

Selected technical details/innovation => session 2

Adoption in 4 industrial use cases => session 3

Public availability as open source

- Initial component release on SourceForge – TODAY
- Full release – planned December 2010
Thank you!