



Vom Kult zur Kultur

SLAs in serviceorientierten Architekturen

Michel Alessandrini, Michael Mehlert, Willi Nüßer

Mit Hilfe einer serviceorientierten Architektur (SOA) lassen sich IT-Systeme von Unternehmen so strukturieren, dass ihre Leistungen in Form von Diensten („Services“) Kunden oder internen Unternehmensabteilungen zur Verfügung gestellt werden können. Essentiell für den Erfolg einer SOA ist eine tragfähige Vereinbarung der Dienste. Ein Service Level Agreement (SLA) ist genau solch eine Vereinbarung. Mit Hilfe spezieller Sprachen lassen sich SLAs formal beschreiben. Allerdings gibt es bisher nur Ansätze für vollständige Verwaltungssysteme für SLAs.

► Der SOA-Hype geht dem Ende entgegen. Diese zweifellos richtige Aussage kann auf unterschiedliche Weisen gelesen werden. Negativ betrachtet, stellt diese Entwicklung die weitere Beschäftigung mit SOA-Ansätzen grundsätzlich in Frage. Realistischer scheint aber eine andere Lesart zu sein: SOA kommt sukzessive mit allen Vor- und Nachteilen in der Wirklichkeit der Unternehmen an. Dies belegt auch die steigende Diskussion um SOA-Governance [KoSt07] und die – erneute – Beschäftigung mit Service Level Agreements (SLA) in einer SOA. Ohne eine Infrastruktur, die SLAs unterstützt, werden die organisatorischen und technischen Versprechungen von SOA-basierten Lösungen in der Praxis nur bedingt einzulösen sein.

Dieser Artikel widmet sich den Entwicklungen im Umfeld von SOA und SLAs. Er beschreibt dazu nach einer kurzen Diskussion der grundlegenden Begriffe aktuelle Entwicklungen und stellt einen eigenen pragmatischen Ansatz für die Arbeit mit SLAs in einer SOA vor.

Grundlagen von SLAs

Unter einem *Service Level Agreement* (SLA) versteht man in der Regel eine – formale – Vereinbarung zwischen einem Dienstleister (Provider) und Dienstnehmer (Consumer), die funktionale und nicht-funktionale Eigenschaften des Dienstes nachvollziehbar beschreibt und Richtwerte vorgibt [Kay03]. Ein SLA ist in der Regel eine 1:1-Absprache zwischen einem Provider und einem Consumer.

Durch die Vereinbarung eines SLAs soll die Dienstgüte (Quality of Service, QoS) für den oder die Anwender des Dienstes sichergestellt werden. Dabei darf ein SLA aber nicht als Garantie verstanden werden, sondern eher als Versicherung, die im Falle einer Nichteinhaltung der vereinbarten QoS eine Kompensation vorsieht.

Zu einem SLA gehören typischerweise mehrere Dokumente, so beispielsweise:

- ▼ Festlegung der relevanten QoS-Parameter und deren Richtwerte,
- ▼ Rahmenwerke bei Nichteinhaltung der Richtwerte (Vertragsstrafen, Kompensationen usw.),
- ▼ Supportverträge.

Die Festlegung der Richtwerte geschieht in *Service Level Objectives* (SLOs), die sich aus dem betreffenden QoS-Parameter und dem ausgehandelten Richtwert zusammen setzen. Ein guter SLA beinhaltet zudem auch die Angabe der Art der Messung des QoS-Parameters. Dies wird als *Service Level Indicator* (SLI) bezeichnet.

Eines der bekanntesten Beispiele für eine SLA-Vereinbarung ist wohl die Festlegung der Antwortzeit eines Dienstes. Das SLO hat z. B. die folgende Gestalt „Antwortzeit kleiner als 2 sec“. Diese einfache Forderung bringt aber schon eine Vielzahl der für SLAs typischen Fragen mit sich:

- ▼ Wo und wie wird die Antwortzeit gemessen (SLI): am Consumer, am Provider oder „irgendwo dazwischen“? Die Diskussion, die einer solchen Setzung in der Regel vorangeht, ist sicher den meisten bekannt. Aussagen über Eigenschaften eines Netzes, wie des Internet, sind statistischer Natur.
- ▼ Wer ist verantwortlich, wenn Router irgendwo im Netz ausfallen und die notwendige Rekonfiguration zu einigen Minuten schlechter Antwortzeit führt?
- ▼ Wie kann eine Kompensation aussehen? Reicht z. B. die simple Vorgabe: „Zahlung von NN EUR, wenn die Antwortzeit fünf Mal innerhalb einer Woche überschritten wird“?

Es ist klar, dass das Aushandeln und Management der SLAs ein aufwändiger und komplexer Prozess ist, der aber heute einen wesentlichen Bestandteil jedes Angebots einer IT-Dienstleistung bildet. In der Literatur wird dieser Prozess als *Service Level Management* (SLM) bezeichnet [SMJ00].

Weniger häufig wird explizit von der notwendigen Infrastruktur gesprochen, die für die Unterstützung des Lebenszyklus eines SLA notwendig ist. Der Lebenszyklus eines SLA umfasst die Phasen Erstellung, Aushandlung, Implementierung und Überwachung [LeBN02]. Diese notwendige Infrastruktur wird häufig als *SLA Management System* (SLA MS) bezeichnet. In vielen Fällen übernehmen bestehende Überwachungslösungen diese Aufgabe mit und konkurrieren mit speziellen Lösungen [Nim08,Com08]. Für viele Anwender gewinnt in letzter Zeit diese Überwachung und das Reporting von SLAs erhebliche Bedeutung [Gar06].

SLA innerhalb der SOA-Welt

Die Bedeutung von SLAs nimmt in einer serviceorientierten Welt natürlich noch einmal zu. Für Unternehmen ist es entscheidend, die Qualität eigener und fremder Dienste genau und verlässlich zu kennen.

Dieser natürlichen Anforderung zum Trotz gibt es im Bereich der SOA derzeit aber keine allgemein akzeptierte oder gar standardisierte Lösung. Dabei wurde schon 2003 vom W3C eine Sammlung von QoS-Parametern, wie Performance, Capacity, Availability usw., für Webservices vorgestellt, die eine natürliche Basis für SLAs im Bereich der Webservices bilden [LJL03].

Das Fehlen einer allgemeinen Lösung für SLM in einer SOA ist zum Teil in den grundsätzlichen Problemen eines SLM, zum anderen Teil aber in den spezifischen Herausforderungen einer SOA-Landschaft begründet. Diese Herausforderungen sind konzeptioneller, organisatorischer oder technischer Natur:

- 1 Eine wesentliche Voraussetzung für ein SLA-Management in einer SOA ist eine funktionsfähige SOA-Governance. Vereinfacht gesagt: Wenn ich nicht weiß, welche Services existieren oder wie sie beschrieben sind, dann kann ich auch nicht über deren Qualität reden.
- 2 Wie können die SLAs der einzelnen Services überhaupt präzise und standardisiert werden? Dies ist die Domäne der SLA-Spezifikationssprachen.
- 3 Welche QoS-Werte gehören in einen konkreten SLA hinein? Nicht alle QoS-Parameter sind im Einzelfall angezeigt. So muss der Parameter Antwortzeit bei einer Verbindung über ein öffentliches Netz kritisch betrachtet werden. Mitunter ist da die rein lokal gemessene Bearbeitungszeit besser geeignet.



- 4 Wie können die Automatisierung, die Vertragsverhandlungen und das Änderungsmanagement bei SLAs im SOA-Umfeld ablaufen?
- 5 Ein wesentlicher Pluspunkt einer SOA sollte die Integration vieler und auch dynamisch bestimmter Services sein. Wie wird mit dieser Variabilität umgegangen?
- 6 Neue Services ergeben sich durch Komposition bestehender Services. Wie werden die SLOs dieser neuen Services bestimmt und wie werden die Ausnahmen der einzelnen Services propagiert? Es ist sicher nicht im Sinne des Kunden, wenn er statt einer Ausnahme („Ausfall des Netz-Providers“) plötzlich deren mehrere im SLA vorfindet: Für jeden Bestandteil eines aggregierten Service einen („Ausfall des Netz-Providers A“ und „Ausfall des Netz-Providers B“ ...).
- 7 Wenn die SLAs festgelegt sind, stellt sich die Frage nach der Bereitstellung der notwendigen Ressourcen auf Seiten des Providers (Provisioning). Wie können – dynamisch? – Ressourcen zugeteilt werden, sodass die SLAs eingehalten werden?
- 8 Schließlich müssen Regelverletzungen z. B. durch eine Überwachung festgehalten werden. Diese Überwachung muss dazu neben dem eigentlichen SLO aber auch die übrigen relevanten Rahmenbedingungen im SLA kennen. Eine Überschreitung des Richtwerts für die Antwortzeit kann beispielsweise ohne Strafe bleiben, wenn die Ausfallursache beim Internet-Provider zu suchen ist. Welche Form der Überwachung wird in einer SOA verwendet und woher weiß die Überwachung um diese Nebenbedingung?

Die folgende Fallstudie zeigt einige dieser Probleme exemplarisch.

Fallstudie

Die dargestellte Fallstudie Kredit-Portal (s. Abb. 1) basiert auf Szenarien des EU-Forschungsprojekts SENSORIA [SEU05]. Ein Kunde gibt am Kreditportal Informationen ein, die für die Beantragung eines Kredits notwendig sind. Ein Teil der Eingabedaten kann durch Anfragen an interne Services, wie Kunden-datenbanken o. ä., ausgewertet werden. Für das Rating wird aber eine externe Rating-Agentur herangezogen. Dieser externe Service verwendet selbst möglicherweise weitere Helper-Services, die erst zur Laufzeit bestimmt werden, um die Anfrage beantworten zu können.

Aus Sicht des Kunden/Consumers seien in diesem Beispiel die Antwortzeit und die Verfügbarkeit des gesamten Service, d. h. des Kreditportals, relevant. Beide Parameter ergeben sich hier aus einer „Parallelschaltung“ des internen und des Rating-Service und einer Reihenschaltung des Rating-Service mit den Helper-Services.

Viele der oben genannten Fragen sind in diesem einfachen Beispiel leicht ersichtlich. Der Kunde will nur einen zentralen SLA mit dem Kreditportal abschließen. Wie werden die SLAs der einzelnen Dienste (SLA1 bis SLA3), die aus unterschiedlichen Unternehmen kommen, allgemein verständlich beschrieben? Wie ergeben sich die Werte des zentralen SLAs aus diesen Einzel-Werten? Wie wird mit einem Ausfall der Netzverbindung zu einem externen Service umgegangen? Wer trägt hier eventuelle Kompensationen? Wie kann die Aushandlung des zentralen SLA in diesem Fall unterstützt werden? Wie und wo müssen welche Ressourcen bereitgestellt werden, um den SLA einzuhalten?

Nur für wenige dieser Fragen existieren derzeit Lösungsansätze. Wir beschreiben im Folgenden einige davon.

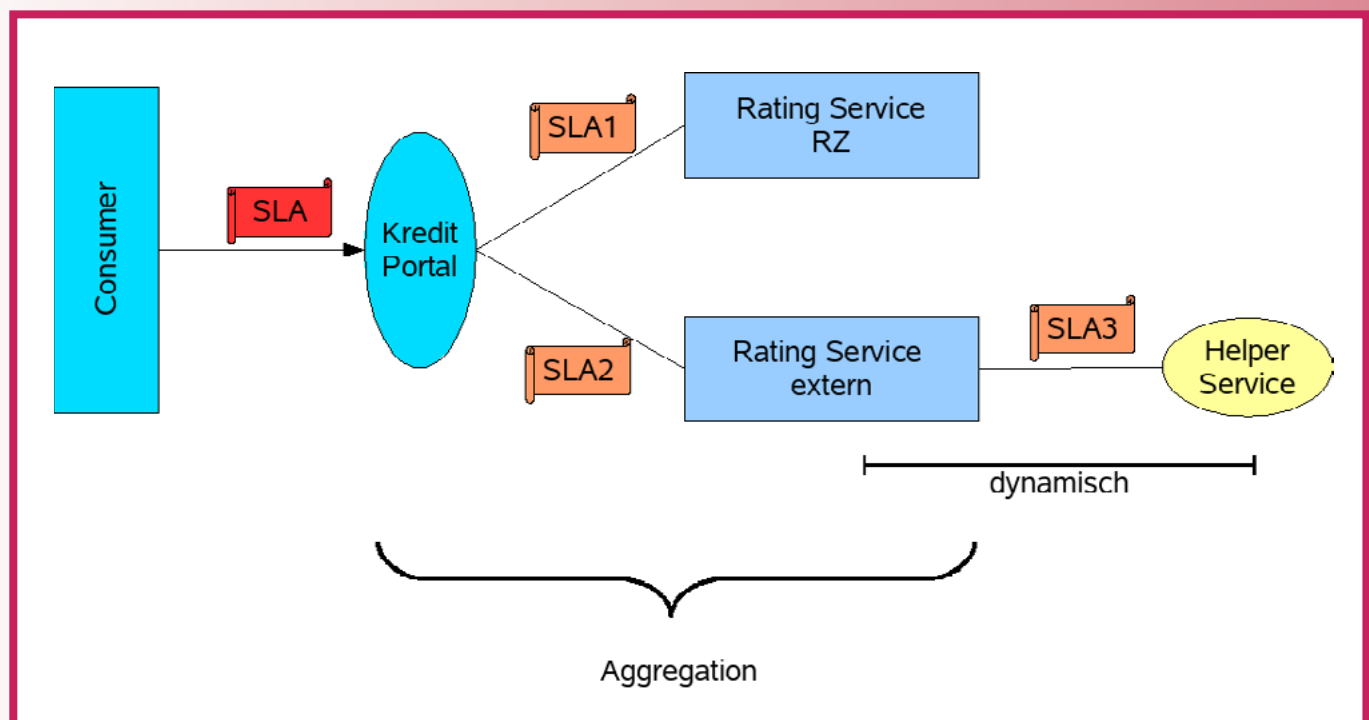


Abb. 1: SLAs am Beispiel eines Kreditportals

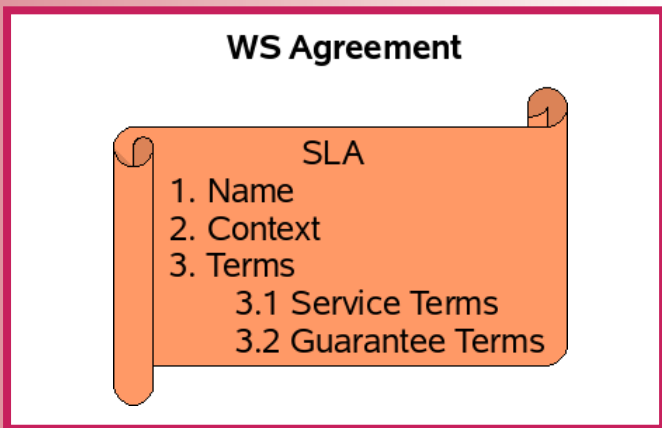


Abb. 2: SLAs in WS-Agreement

Aktuelle Lösungsansätze

Wie in vielen Bereichen der SOA- und Webservice-Welt entstand eine Vielzahl von Vorschlägen für den Umgang mit SLAs. Grundsätzlich muss hier zwischen SLA-Spezifikationssprachen und SLA-Managementsystemen unterschieden werden.

SLA-Spezifikationssprachen adressieren vor allem die Punkte 2, 3 und teilweise 4 der oben vorgestellten Fragenliste. Sie sollen

- ▼ standardisierte Strukturen und Formate bieten, um SLAs einheitlich beschreiben zu können,
- ▼ die wichtigsten inhaltlichen Aspekte von SLAs abdecken, wie Parteien, Status des SLA, Gültigkeitszeitraum, QoS-Parameter und SLOs, Verantwortlichkeiten und Ausnahmen,
- ▼ wiederverwendbare Elemente, Templates, beinhalten, die bei der Erstellung und Aushandlung neuer SLAs unterstützen können.

Beispiele für solche Sprachen sind die *Web Service Offering Language* [WSOL02], die *Web Service Management Language* (WSML)

als Teil des *Web Service Management Network* von HP [WSMN02] oder auch Sprachen wie *WS-Policy* [WSP006] und *SLAng* [LSE03]. Die beiden letztgenannten bieten zwar Möglichkeiten zur Beschreibung von SLAs, sind aber entweder nicht auf SLA speziell ausgerichtet (WS-Policy) oder wenig flexibel (SLAng).

Mehr Aufmerksamkeit als diese Beispiele hat die von IBM entwickelte *Web Service Level Agreement Language* (WSLA) erlangt. Sie wurde von IBM zugunsten des Nachfolgers *WS-Agreement* aufgegeben [WSA07]. *WS-Agreement* hat innerhalb der *Grid-Community* mittlerweile einen guten Stand und besitzt Implementierungen, die allerdings in den meisten Fällen noch im eher akademischen Bereich angesiedelt sind. Sie deckt, wie auch *WSOL* und *WSML*, die wichtigsten Anforderungen an eine SLA-Beschreibungssprache ab.

WS-Agreement beschreibt einen SLA durch einen Namen, einen Kontext, der allgemeine Informationen, wie Partner und Gültigkeitsdauer, angibt, und durch einen *Terms*-Abschnitt (s. Abb. 2). Diese *Terms* bestehen zum einen aus *Service Terms*, die den Webservice spezifizieren, für den der SLA erstellt wird, und seine QoS-Parameter nennen. Die eigentlichen SLOs werden in den *Guarantee Terms* festgelegt. Dort ist vermerkt, wie der SLO konkret lautet (z. B. welche Antwortzeit erwartet wird), wer für die Einhaltung zuständig ist und was für Strafen (*penalties*) existieren.

Speziell mit *WS-Agreement* liegt also eine Spezifikations-sprache vor, die für die meisten Bedürfnisse ausreicht. Anders sieht es bei den SLA MS aus. Hier werden in der Regel die folgenden Eigenschaften erwartet

- ▼ Unterstützung aller Lebensphasen eines SLAs: von Entwicklung und Verhandlung, über Implementierung, Ausführung und Provisioning bis hin zur Überwachung und zur Bewertung.
- ▼ Dabei Berücksichtigung der o. g. Besonderheiten von WS/SOA.
- ▼ Optional: gleichzeitige Durchführung des Accountings.

Die meisten der oben genannten Ansätze bringen Ideen für ein SLA MS gleich mit. So gehört eine *Web Service Offering Infrastructure* (WSOI) zur *WSOL* und die *WSML* zum umfangreicheren *WS Management Network*. Keiner dieser Ansätze hat aber bislang eine größere Verbreitung gefunden. Offen ist

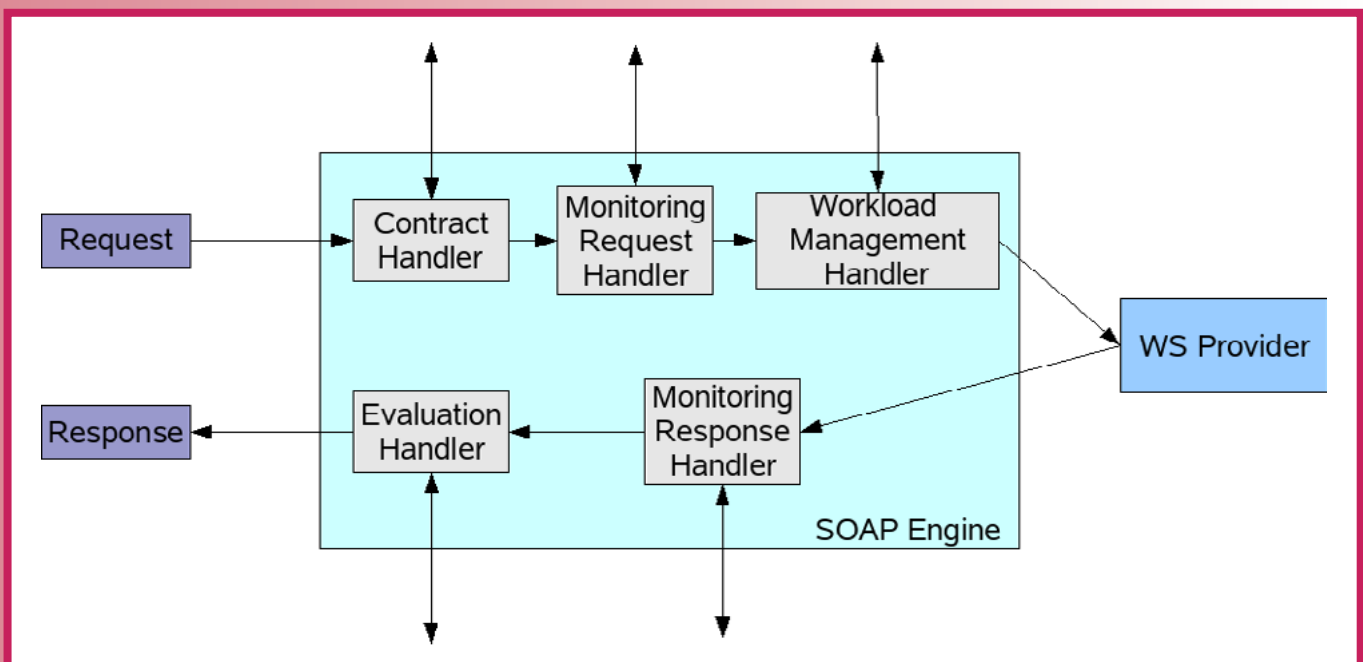


Abb. 3: Handler-Framework zur SLA-Unterstützung



derzeit, wie weit dies den oben schon genannten Implementierungen von WS-Agreement gelingen wird. Derzeit kann aber selbst im optimistischsten Fall davon ausgegangen werden, dass keine der genannten Implementierung so zentrale Fragen wie das Provisioning löst. SLA MS stellen also in unseren Augen eine wirkliche Mangelware dar.

In Praxisprojekten werden deshalb häufig pragmatische Lösungen herangezogen, die schon mit einfachen Mitteln einige zentrale Aufgaben recht weit lösen. Abbildung 3 zeigt einen solchen Ansatz. Hier werden Interceptoren, wie die Handler von Apache Axis, in den eingehenden und ausgehenden SOAP-Strom eingehängt. Der erste Handler (Contracting) stellt bei einem eingehenden Request den Bezug zum zugehörigen SLA her. Die Monitoring-Handler können z. B. Zeitstempel nehmen, um das SLO Antwort- oder Bearbeitungszeit zu messen. Ein Workload-Handler kann eine Priorisierung der eingehenden Anfragen vornehmen und damit auch einfache Provisioning-Aufgaben übernehmen. Der Evaluation-Handler schließlich startet auf der Basis der Monitoring-Werte die Auswertung der Request-Verarbeitung. Hier findet die Evaluation statt, ob der SLO eingehalten wurde oder nicht, und ggf. auch das Accounting.

Die Handler in diesem Szenario können natürlich beliebig ergänzt werden und selbst auch andere Dienste in Anspruch nehmen. Damit gewinnt dieser Ansatz dann schnell eine beachtliche Komplexität. Den Ansprüchen an ein volles SLA MS wird er aber dennoch sicherlich nicht gerecht.

Alternativen

Die bisher vorgestellten Ansätze beschreiben SLA deklarativ und bleiben damit weitgehend im bekannten SOA-Rahmen. Eine sehr interessante Alternative bietet da eine Regel-basierende Lösung, wie z. B. RBSLA [RBSLA08].

RBSLA kombiniert Elemente wie Regelsysteme und semantische Darstellungen mit den Fragestellungen des SLM. Die Regel-basierten Sprachen erlauben es, auch komplexe SLA-Sachverhalte effizient zu beschreiben. Durch Schlüsse können aus bekannten SLAs neue SLAs erstellt werden. Dies ist in den oben beschriebenen rein deklarativen Ansätzen nicht möglich. Damit wird der Vielfalt und Variabilität der Services in heutigen SOA-Landschaften Rechnung getragen.

Ebenso kann vergleichsweise einfach auf die Nichteinhaltung eines SLOs geschlossen werden, wenn dafür eine Regelbasis existiert. All dies macht RBSLA sicher zu einem Projekt, das sich weiter zu verfolgen lohnt. Auf der anderen Seite ist aber auch hier derzeit noch kein vollwertiges SLA MS erkennbar. Viele Projekte werden deshalb weiterhin ihre Flucht zu pragmatischen Teillösungen nehmen müssen.

Fazit

Service Level Agreements und SLA MS sind essentiell, wenn SOA-Lösungen den gesamten Bereich der IT überdecken sollen. Sprachen zur Beschreibung von SLAs existieren sowohl innerhalb der klassischen SOA-Community als auch im Bereich der Künstlichen Intelligenz, Regeln und Semantik. Hier zeigt sich die Entwicklung recht gut aufgestellt.

Vollständige Verwaltungssysteme für SLAs sind demgegenüber immer noch rar. Pragmatische Ansätze dominieren die Projektlandschaft. Es bleibt also abzuwarten, ob innovative Projekte, wie RBSLA, genügend Dynamik entwickeln, um eine echte Abhilfe zu sein.

Literatur und Links

- [Com08] Compuware Vantage, <http://www.compuware.com/products/vantage/clientvantage.htm>, 2008
- [Gar06] Curtis, Debra, Poll Reveals Buying Preferences for Business Service Management, Availability and Performance, Monitoring, Gartner, Inc., April 2006
- [Kay03] D. Kaye, Loosly Coupled: The Missing Pieces of Web Services, RDS Press, 2003
- [KoSt07] A. Koschel, G. Starke, SOA-Governance, in: JavaSPEKTRUM, 05/2007
- [LeBN02] J.J. Lee, R. Ben-Natan, Integrating Service Level Agreements: Optimizing Your OSS for SLA Delivery, Wiley, 2002
- [LJL03] K. Lee, J. Jeon, W. Lee, et al., QoS for Web Services: Requirements and Possible Approaches, W3C Working Group Note, 25.11.2003, <http://www.w3c.or.kr/kr-office/TR/2003/NOTE-ws-qos-20031125/>
- [LSE03] D. D. Lamanna, J. Skene, W. Emmerich, SLAng: A Language for Defining Service Level Agreements, in: Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems, 2003
- [Nim08] Nimsoft NimBus, <http://nimsoft.com/>, 2008
- [RBSLA08] A. Paschke, Rule Based Service Level Agreements, <http://ibis.in.tum.de/research/projects/rbsla/>
- [SEU05] Sensoria, Software Engineering for Service-Oriented Overlay Computers, <http://www.sensoria-ist.eu>
- [SMJ00] R. Sturm, W. Morris, M. Jander, Foundations of service level management, Sams, 2000
- [WSA07] Open Grid Forum, WS-Agreement Specification, März 2007, <http://www.ogf.org/documents/GFD.107.pdf>
- [WSMN02] HP Labs, Web Services Management Network: An Overlay Network for Federated Service Management, <http://www.hp1.hp.com/techreports/2002/HPL-2002-234.html>
- [WSOL02] V. Tosic, B. Pagurek, B. Esfandiari, K. Patel, W. Ma, Web Service Offerings language (WSOL) and Web Service Composition Management (WSCM), in: Proc. of the Object-Oriented Web Services Workshop (OOPSLA) 2002
- [WSP06] W3C: Web Services Policy 1.2 – Framework (WS-Policy), <http://www.w3.org/Submission/WS-Policy/>



Michel Alessandrini ist an der FHDW als wissenschaftlicher Mitarbeiter mit den Forschungsschwerpunkten SOA und künstlicher Intelligenz tätig. Des Weiteren ist er für die S&N AG als Projektleiter in einem EU-Forschungsprojekt im Einsatz.
E-Mail: michel.alessandrini@fhdw.de.



Michael Mehlert ist langjähriger Mitarbeiter der S&N AG und Absolvent der FHDW. Seit 2007 ist er als Softwareentwickler vor allem in Projekten für die Finanzindustrie tätig.
E-Mail: mmehlert@s-und-n.de.



Dr. Willi Nüßer ist seit 2002 Professor für angewandte Informatik an der Fachhochschule der Wirtschaft (FHDW) in Paderborn und Inhaber der Heinz Nixdorf Stiftungsprofessur. Er war von 1996 bis 2002 bei der SAP AG u. a. im LinuxLab in Walldorf tätig. Seit 2002 beschäftigt er sich in Theorie und Praxis mit der Webservice-Technologie.
E-Mail: wilhelm.nuesser@fhdw.de.