

End-to-end Monitoring (Vertiefung)

Seminar IT Service Management, ZHAW (Zürich)

Florian Lüthi*

13. Februar 2013

*luethifl@students.zhaw.ch

Inhaltsverzeichnis

1 Management Summary	3
2 Einführung	4
2.1 Kontext und Definitionen	4
2.2 Fragestellungen	4
2.2.1 Transparenz versus Transformabilität	4
2.2.2 Skalierbarkeit nach unten	5
3 Methodiken/Techniken	6
3.1 Service Level Agreements (SLAs)	6
3.2 Service Level Objectives (SLOs)	7
3.3 Metriken und Key Performance Indicators (KPIs)	7
3.4 Web Service Level Agreements (WSLAs)	9
3.5 Gestaltung von IT-Leistungskatalogen mit End-to-end Monitoring	16
3.6 Auswirkungen von End-to-end Monitoring auf Outsourcing	17
4 Fazit	19
Abbildungsverzeichnis	20
Literaturverzeichnis	21

1 Management Summary

2 Einführung

Diese Seminararbeit ist Teil des Seminars *IT Service Management*, gehalten an der ZHAW Standort Zürich im Winter 2012/13. Die Arbeit beschäftigt sich mit dem Teilthema *End-to-end Monitoring: Vertiefung*. Das Ziel ist, einen Überblick über End-to-end Monitoring von IT-Dienstleistungen vor allem aus Business-Sicht zu geben. Des Weiteren soll mit Web Service Level Agreements (WSLA) eine Methode vorgestellt werden, wie sich die daraus abgeleiteten Themata und Methodiken grosso modo in komplett automatisierbare Systeme übertragen lässt.

Die daraus resultierenden Fragestellungen und Aspekte sollen vor allem aus der Sichtweise der langfristig aufstrebenden klein- und mittelgrossen Firmen diskutiert werden.

Die Arbeit ist wie folgt aufgebaut: Im vorhergehenden Kapitel wurde der gesamte Inhalt der Arbeit für ein Publikum in Eile zusammengefasst. Der Rest dieses Kapitels wiederum widmet sich der Einbettung der für die Arbeit relevanten Begrifflichkeiten (inklusive ihrer Definitionen) in ihren grösseren Kontext sowie der Ansetzung der Fragestellungen, welche die Arbeit zu beantworten im Sinne hat. Im Kapitel 3 wird auf einige gängige und zukünftige Methodiken eingegangen, bevor schliesslich im Kapitel 4 versucht wird, eine Antwort auf die Fragestellungen zu geben oder aber ihre Nichtbeantwortbarkeit aufgezeigt wird. Ausserdem wird ebenda noch ein kurzes Fazit gezogen.

2.1 Kontext und Definitionen

Definition 1 (Business Service Management (BSM)). Das Modell des Business Service Management verknüpft die Geschäftsprozesse eines Unternehmens mit den darunterliegenden IT-Services. Dadurch ist es möglich, die Abhängigkeiten von Business zu IT darzustellen, sowie die Auswirkungen von IT-Störungen auf das Business aufzuzeigen. Das Ziel von Business Service Management ist, eine bessere Abstimmung zwischen Business und IT zu erzielen. [15]

BSM startete ungefähr im Jahr 2003 als Buzzword, bis sich Forrester Research 2006 dem Thema annahm und begann, darüber zu publizieren. [15, 7, 13]

Definition 2 (End-to-end Monitoring). Bla.,

2.2 Fragestellungen

2.2.1 Transparenz versus Transformabilität

2.2.2 Skalierbarkeit nach unten

3 Methodiken/Techniken

3.1 Service Level Agreements (SLAs)

Eine allgemein bekannte Definition sagt:

Ein Service Level Agreement (SLA) ist der Teil eines Dienstleistungsvertrags, in welchem die Dienstleistung formal definiert ist. In der Praxis wird der Begriff *SLA* manchmal für die vertraglich zugesicherte Lieferzeit verwendet. [22]

Eine typische Eigenschaft eines SLA ist, dass es – im Unterschied zu einem herkömmlichen Dienstleistungsvertrag – verschiedene Service-Levels zu unterschiedlichen Konditionen anbietet und so dem Kunden die Möglichkeit gibt, die Dienstleistung gemäss seinen Bedürfnissen und seiner finanziellen Potenz einzukaufen ([2]). Die SLA-Parameter, Objectives und KPIs sind dann abhängig vom gewählten Service-Level.

Es ist offensichtlich, dass diese Definition noch nicht enorm hilfreich ist. Forrester definiert darum zwei neue Begriffe, wobei Abbildung ?? die jeweilige Abgrenzung zueinander zeigt:

Definition 3 (Business service management (BSM)). Business service management is an intelligence layer that supports all the management processes involved in monitoring service delivery¹ [5].

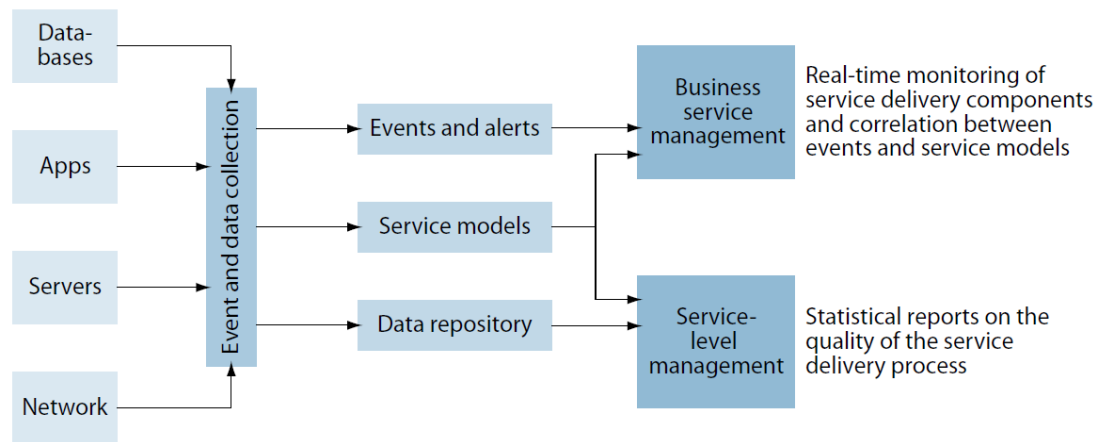
Definition 4 (Service-level Management (SLM)). Service-level management is a process layer that monitors and reports on the quality of the processes involved in service delivery² [5].

Die Idee ist nun, dass in diesem Kontext sowohl BSM als auch SLM exakt dieselben Qualitätsverbesserungs-Ziele haben müssen. Diese Ziele (*objectives*) werden in einem SLA ausgedrückt, wessen Parameter die Business-Zufriedenheit (*business satisfaction*) messen sollen [5, 2].

¹Da sich die beiden englischen Begriffe *intelligence* und *delivery* nicht so ins Deutsche übersetzen lassen, dass der Sinn des Satzes gewahrt bleibt, wurde vom Autor auf die Übersetzung verzichtet.

²ditto.

Abbildung 3.1: Abgrenzung von BSM und SLM gemäss Forrester



Source: Forrester Research, Inc.

3.2 Service Level Objectives (SLOs)

SLAs werden als das Instrument betrachtet, durch welches der Dialog zwischen dem Kunden und dem Dienstleister etabliert werden kann. Die *Service-level objectives* (SLOs) wiederum sind dann die Parameter, mit welchen die Zufriedenheit über den angebotenen Service gemessen wird [5]. Gemäss Forrester funktioniert das aber nur, wenn folgendes gilt:

- Die SLOs werden genau dort gemessen, wo die Dienstleistung verwendet wird.
- Die SLOs sind eine korrekte Repräsentation derjenigen Aspekte, die für den Kunden wichtig sind.

3.3 Metriken und Key Performance Indicators (KPIs)

Gemäss Forrester ([5]) soll eine in einem SLA verwendete Metrik die folgenden Anforderungen erfüllen:

1. Die Metrik ist objektiv messbar.
2. Die Metrik enthält eine klare Aussage über das erwartete Resultat.
3. Die Metrik unterstützt Business-seitige Anforderungen.
4. Die Metrik konzentriert sich entweder auf die Effektivität oder die Effizienz des zu messenden Prozesses.

5. Die Metrik erlaubt eine sinnvolle statistische oder Trend-Analyse.
6. Die Metrik wendet Industrie- und/oder andere Standards an.
7. Annahmen und Definitionen für zufriedenstellende Performance wird spezifiziert.
8. In die Festlegung der Metrik wurden diejenigen Stakeholder involviert, die nachher für die Performance des Prozesses verantwortlich sind.
9. Sowohl der Dienstleister wie auch der Abnehmer akzeptieren die Metrik.

Man ist sich in der Industrie einig ([5, 2]), dass die technische Antwortzeit von Services sowie deren grundsätzliche Verfügbarkeit die gängigsten Metriken sind. Auf der anderen Seite ist man sich aber ebenso einig, dass diese Metriken vor allem globaler Natur sind; eine wirkliche ständige Verbesserung der Services im Sinne von Business Process Management sowie [2] bringen sie noch nicht.

Aus diesem Grunde ist es erforderlich, für die erfolgreiche und ganzheitliche Überwachung eines Services auf Business-Level diesen Service als Summe aller involvierten technischen Services zu verstehen. Diese involvierten technischen Services werden dann *Komponenten* geheissen, und von diesen werden wiederum Metriken auf Komponenten-Ebene erhoben. Es ist von enormer Wichtigkeit, dass die Komponenten-Metriken ihren sinnvollen Platz in der Konstruktion des gesamten globalen SLOs finden.

Forrester ([5]) sieht darum zwei kritische Bereiche, denen während dem Aufbau einer solchartigen SLO-Landschaft Beachtung geschenkt werden muss:

Service/Applikations-Modelle. Die Idee ist, eine initiale Landkarte der momentanen Services, Applikationen und Komponenten zu entwickeln [5], damit Zusammenhänge schon *top-down* sichtbar werden und nicht erst in mühsamer Kleinarbeit *bottom-up* hergestellt werden müssen. Entschiedener Fokus wird auch auf die sogenannte *Configuration Management Database (CMDB)* gelegt. Als wesentliche Erfolgsfaktoren werden hierbei eine dezentrale Organisation (*Federation*) sowie Funktionalität genannt, welche die CMDB automatisch anhand einer geänderten Systemlandschaft aktualisieren soll (sogenanntes *Auto-discovery*) [13].

Historische Daten. Diese sind vor allem für die Festlegung von Performance-artigen Zielen beim Neuaufbau von SLOs relevant. Da in der ganzen prä-SLO-Welt die Performance eines Services nur End-to-end gemessen wurde (wenn überhaupt), liegen keine relevanten Daten über die Performance der einzelnen Komponenten vor. Die Festlegung dieser wäre also ein Schuss ins Blaue, welcher dann bei Nichteinhaltung (deren Wahrscheinlichkeit eben darum gegeben ist) wohl mehr organisatorische Probleme verursachen als lösen würde. Deshalb sollten historische Daten (möglicherweise anderer Indikatoren) herangezogen werden, um eine aussagekräftige Ist-Analyse vornehmen zu können.

Bezüglich der Thematik der ständigen Verbesserung einer Service-Landschaft kann zu guter Letzt auch die Wahl der richtigen Berechnungsmethodik der Metriken beitragen. Beispielsweise kann der ganze Aufbau eines SLOs komplett sinnfrei sein, wenn die

Performance-Ziele zu tief angesetzt sind. Wenn als Metrik einfach nur das arithmetische Mittel über eine gewisse Zeit herangezogen wird, könnte der Fall eintreten, dass auf dem Papier alle Ziele mit Bravour erfüllt werden – Komplettausfälle der Systeme mit negativen Business-Folgen aber (vor allem wenn sie in zeitlich kritischen Phasen auftreten) drücken zwar den Schnitt ein bisschen nach oben, aber nicht so sehr, dass das SLA eine Reaktion gebieten würde.

Forrester schlägt darum explizit die Implementierung der *Six-Sigma*-Methodologie vor [5, 23]:

1. Anstelle eines einzigen Zielwerts sollen ein Zielwert w_{Ziel} , ein maximal tolerierbarer Wert w_{max} sowie ein minimal tolerierbarer Wert w_{min} definiert werden.
2. Anstelle der simplen Berechnung des Durchschnitts der tatsächlich gemessenen Werte $w = \{w_1, w_2, \dots, w_n\}$ sollen sowohl Durchschnitt

$$\mu = \frac{1}{n} \sum_{i=1}^n w_i$$

als auch Standardabweichung ([25])

$$\sigma = \left(\frac{1}{n-1} \sum_{i=1}^n (w_i - \mu)^2 \right)^{\frac{1}{2}}$$

berechnet werden.

3. Es wird nun der Sigma-Faktor x eingeführt, sodass

$$x\sigma \geq \min \{w_{\text{Ziel}} - w_{\text{min}}, w_{\text{max}} - w_{\text{Ziel}}\}$$

gilt. Bei gegebener Normalverteilung der anfallenden Messwerte gibt x Aufschluss über die Performance-Abweichung des Services ([13]); beispielsweise gilt für $x = 1$, dass 31% aller Messwerte innerhalb der tolerierbaren Grenzen liegen, für $x = 6$ sind es bereits 99.99966% ([23]).³

3.4 Web Service Level Agreements (WSLAs)

SOA (Service Oriented Architecture) ist ein aktuelles Buzzword: Innerhalb von Business-Prozessen agierende Applikationen, die sich dynamisch an ihre benötigten Services binden. Als Quasi-Standard haben sich in diesem Umfeld Web-Services durchgesetzt.

Die dafür notwendigen Definitionen sind mittlerweile ebenfalls standardisiert oder quasi-standardisiert:

³Diese Berechnung ist stark vereinfacht – beispielsweise unterscheidet die Six-Sigma-Methodologie noch zwischen kurzfristigen und langfristigen Resultaten. Ausserdem divergiert die Definition des Sigma-Faktors je nach Quelle massiv, mit entsprechenden Auswirkungen auf die Prozentzahl der tolerierbaren Messwerte. Der Autor möchte dies an dieser Stelle unkommentiert lassen und verweist auf Kapitel 4.

SOAP ein XML-basiertes Austauschformat für Messages [24]

WSDL ein XML-basiertes Austauschformat für Daten- und Operationen-Definitionen [17]

Für sinnvolles End-to-End-Monitoring der Business-Prozesse müssen natürlich auch die daran beteiligten Web-Services überwacht werden können – und entsprechende SLAs zwischen dem Anbieter des Service (oder dem Service selbst) und dem Konsumenten vereinbart werden können.

Da die Benutzung von Web-Services aber dynamisch funktioniert (d.h. eine Fall-zu-Fall-Geschäftsbeziehung eingegangen wird), müssen schlussendlich auch Definition, Verhandlung, Festlegung, Überwachung und Durchsetzung der zugehörigen SLAs dynamisch (sprich automatisiert) funktionieren können [1].

Klassische SLAs sind als Freitext formuliert – was aus naheliegenden Gründen in direktem Widerspruch zur Anforderung der Automatisierbarkeit steht.

Die aktuelle Praxis (auch für alle anderen Arten von Agreements) ist darum, Templates zu entwickeln – Freitext, welcher die grundlegenden Züge der Abmachung beschreibt, kombiniert mit semantischen Feldern. Dieser Ansatz hat allerdings das Problem, dass er einigermassen unflexibel und nur für sehr grundlegende SLOs genügend ist [1].

Findige Köpfe von IBM haben darum 2003 dafür eine grundsätzliche Lösung vorgestellt: Das Web Service Level Agreement Framework [4, 1], mit den folgenden Anforderungen und Design-Zielen:

Flexible formale Sprache, die einen weiten Bereich von SLAs abdeckt. Es hat sich gezeigt, dass SLA-Parameter verschiedener Provider und unterschiedlicher Services, die identisch erscheinen, trotzdem stark in ihrer Semantik variieren. Darum ist es wichtig, dass ein SLA-Framework die Parteien nicht in der Formulierung der SLAs einschränkt. Handkehrum sind aber SLAs in ihrer Struktur stets identisch. Jedes SLA enthält:

- die involvierten Parteien,
- die SLA-Parameter,
- die Metriken, welche die Parameter vervollständigen,
- den Algorithmus, mit welchem die SLA-Parameter berechnet werden,
- die SLOs,
- sowie die Massnahmen, die im Falle einer Verletzung eines oder mehrerer SLOs zu treffen sind.

Möglichkeiten zur Integration mit eCommerce-Systemen und -Plattformen Bezüglich eCommerce und B2B ist die Automatisierung schon sehr weit fortgeschritten – ein SLA-Framework sollte also in seinen Grundzügen kompatibel zu dortigen existierenden Ansätzen sein. IBM sieht vor allem folgende Integrationsmöglichkeiten:

- Bewerbung und Verkauf von SLA-basierten Services,

- Integration in elektronische Marktplätze wie jene von Ariba (welche mittlerweile von SAP übernommen worden ist [18]) oder CommerceOne (welche mittlerweile konkurs gegangen ist [19]),
- weitere Marketplace- und Matchmaking-Anwendungen [6], auch für das Matching von Produkten mit komplexen Attributen (die eben in den SLAs spezifiziert werden können),
- automatisierte Verhandlung, wie beispielsweise vom Projekt SilkRoad vorgesehen [12, 11],
- Verknüpfung von Business-Prozessen über Organisationsgrenzen hinweg, beispielsweise über den ebXML-Stack [20, 1]

Übertragung von Überwachungs-Verantwortlichkeiten an Drittparteien Die enhanced Telecom Operations Map (eTOM) [1, 16] definiert eine Vielzahl von Rollen, die ein Service-Provider einnehmen kann; und momentaner Stand der Dinge ist auch, dass ein SLA vor allem als bilaterales Abkommen zwischen zwei Parteien betrachtet wird. Es kann nun aber Fälle geben, in denen die Daten für die Metriken weder von der einen noch von der anderen Partei sinnvoll geliefert werden können. Ausserdem können Fälle auftreten, in denen die eine Partei der anderen in Bezug auf die korrekte Bereitstellung der Daten misstraut.

Aus diesem Grund definiert [1] diejenigen Parteien, welche das SLA tatsächlich unterschreiben (typischerweise der Dienstleister und der Kunde) als sogenannte *Signatory parties*, und lässt daneben eine beliebige Anzahl sogenannter *Supporting parties* zu. Diese übernehmen beispielsweise Messungen oder die Auswertung der Daten.

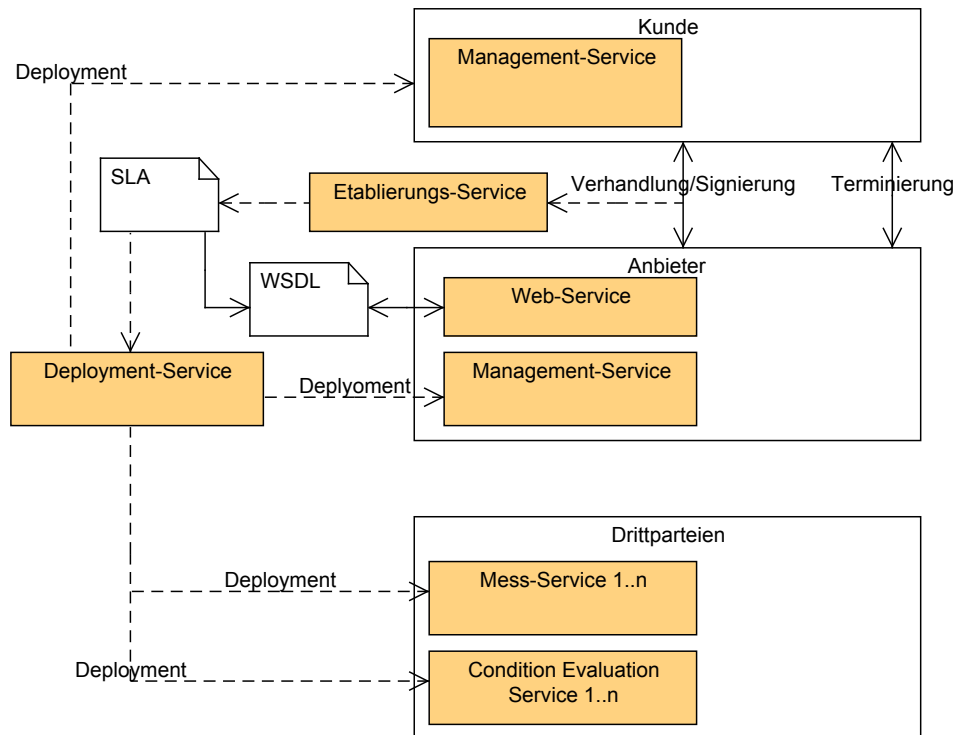
Need-to-know Principle für das Deployment von SLAs Es liegt auf der Hand, dass alle SLA-relevanten Definitionen sowie alle Parteien (ob Signatory oder Supporting im Sinne von oben) in einem einzigen SLA festgelegt sein sollten. Gleichzeitig kann aber die Landschaft der involvierten Parteien einigermassen komplex werden – und bestimmte Parteien werden das Bedürfnis haben, dass sie (oder Drittparteien) für andere nicht sichtbar sind.

Beispielsweise möchte ein Service Provider, der Teile (oder das Ganze) seines Services wiederum von einem dritten Service Provider einkauft, diesen nicht offenlegen. Des weiteren möchte wohl keine der Signatory parties den Supporting parties mehr Kennzahlen als nötig offenlegen (vor allem keine Preise) [1].

Diese Anforderung macht das Deployment eines SLAs bei den verschiedenen involvierten Parteien einigermassen kompliziert, denn es dürfen nur diejenigen Teile ausgerollt werden, die jeweils notwendig sind, damit eine Partei ihre Rolle wahrnehmen kann.

Aus diesem Grund schlagen die Autoren von IBM einerseits einen speziellen Deployment-Service vor (dessen Einbettung in den Gesamtkontext der Abbildung 3.2 entnommen werden kann), einerseits ein zweistufiges Deployment-Verfahren:

Abbildung 3.2: WSLA-Services und ihre Interaktionen

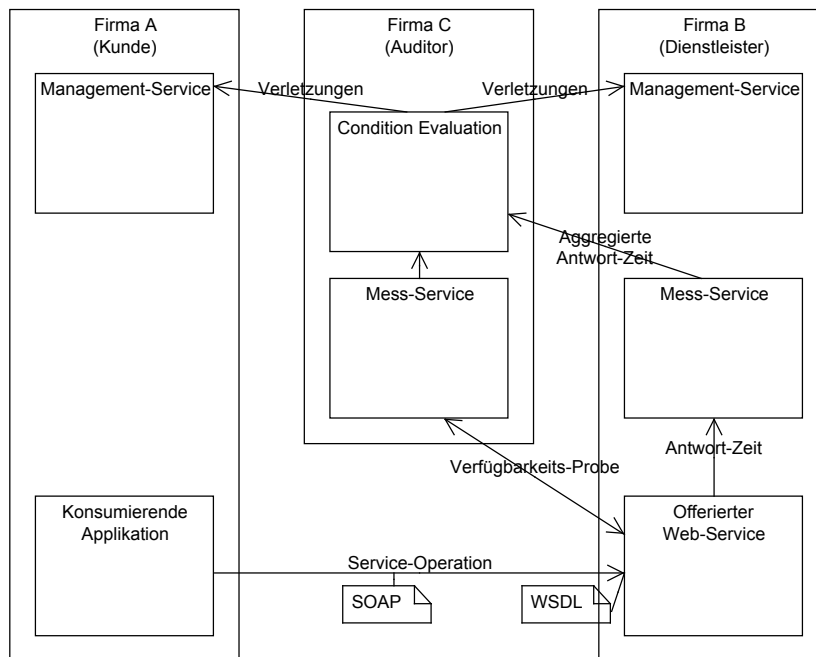


1. Der Deployment-Service generiert Konfigurationsinformation im sogenannten *Service Deployment Information (SDI)*-Format ([1, 4]) an die Supporting parties.
2. Die Supporting parties konfigurieren ihre eigenen Systeme anhand der Informationen im gesandten SDI.

SLA-getriebene Konfiguration von verwalteten Ressourcen Da die Etablierung eines SLAs weitreichende Konsequenzen auf die Konfiguration der verwalteten Ressourcen eines Service Providers haben kann, muss ein SLA-Management-Framework wie das besprochene die Möglichkeit bieten, einerseits sich auf bestehende Infrastruktur zu beziehen, andererseits die existierenden Metriken dieser existierenden Infrastruktur zu nutzen und zu den definierten Metriken innerhalb des SLAs in Beziehung zu setzen. Beispielsweise existiert ein RFC für ein MIB für SLA-Performance-Messung in SNMP-Umfeldern [1, 14].

Alle diese Anforderungen führen zu entsprechenden Services und Sprachdefinitionen. Das grundsätzliche Schema ist in Abbildung 3.3 dargestellt. Die wesentlichen Komponenten sind die folgenden:

Abbildung 3.3: Eine 3-Parteien-WSLA-Konstruktion



Offerierter Web-Service Der Web-Service, der zur Verfügung gestellt wird. Das WSLA dient primär der Überwachung dieses Services.

Konsumierende Applikation Die Applikation (oder Service oder Business-Prozess), welche den offerierten Web-Service konsumiert.

Mess-Service innerhalb Firma B Ein Service, welcher direkt durch Konsumation anfallende Metriken des offerierten Web-Services misst (beispielsweise die Antwort-Zeit).

Mess-Service innerhalb Firma C Ein Service, welcher indirekte und globale Metriken misst (beispielsweise, ob der Service überhaupt verfügbar ist).

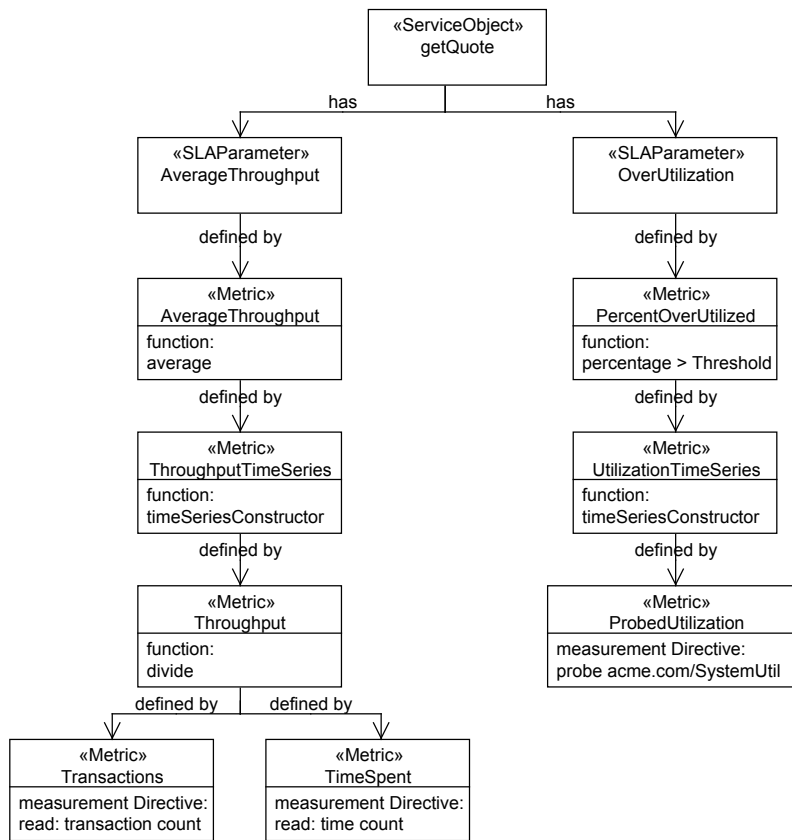
Condition Evaluation Service Ein Service, welcher die Metriken der verschiedenen Mess-Services erhält und gegen den Inhalt des SLAs überprüft.

Management-Service innerhalb Firma A Ein Service, der Notifikationen über Verletzungen des SLAs erhält und darauf entsprechend reagiert.

Management-Service innerhalb Firma B Ein Service, der Notifikationen über Verletzungen des SLAs erhält und darauf entsprechend reagiert.

Die Sprache besteht aus folgenden drei Teilbereichen:

Abbildung 3.4: Beispiel für eine WSLA-Service Description



Parties Dieser Bereich definiert alle in das SLA involvierten Parteien.

Service Description Dieser Bereich spezifiziert die Charakteristika des Services und seine beobachtbaren Parameter [1], welche für die Mess-Services von Bedeutung sind. Als Beispiel einer Service Description sei ein Service gemäss Abbildung 3.4 gegeben. Die wesentlichen Elemente sind **SLAParameter** und **Metric** – **SLAParameter** definiert vor allem die Kommunikationseigenschaften (welche Party wird darüber informiert), während **Metric** die dazugehörige Metrik mit einer Berechnungsfunktion sowie Referenzen auf darunterliegende Metriken enthält. Die XML-Listings in den Abbildungen 3.5 und 3.6 sollen dies illustrieren.

Obligations Dieser Bereich spezifiziert diverse Garantien und Einschränkungen, welche für SLA-Parameter gegeben werden können. Dieser Bereich ist vor allem für den Condition-Evaluation-Service von Bedeutung.

Abbildung 3.5: XML-Definition eines WSLA-SLAParameter

```

1 <SLAParameter name="OverUtilization" type="float" unit="Percentage">
2   <Metric>PercentOverUtilized</Metric>
3   <Communication>
4     <Source>Firma B</Source>
5     <Pull>Firma C</Pull>
6     <Push>Firma C</Push>
7   </Communication>
8 </SLAParameter>

```

Abbildung 3.6: XML-Definition einer WSLA-Metric

```

1 <Metric name="PercentOverUtilized" type="float" unit="Percentage">
2   <Source>Firma C</Source>
3   <Function xsi:type="percentage" Threshold="resultType="float">
4     <Schedule>BusinessDay</Schedule>
5     <Metric>UtilizationTimeSeries</Metric>
6     <Value>
7       <LongScalar>0.8</LongScalar>
8     </Value>
9   </Function>
10 </Metric>
11
12 <Metric name="UtilizationTimeSeries" type="TS" unit="">
13   <Source>Firma C</Source>
14   <Function xsi:type="timeSeriesConstructor" resultType="float">
15     <Schedule>Every5Minutes</Schedule>
16     <Metric>ProbedUtilization</Metric>
17     <Window>12</Window>
18   </Function>
19 </Metric>
20
21 <Metric name="ProbedUtilization" type="float" unit="">
22   <Source>Firma B</Source>
23   <MeasurementDirective xsi:type="Gauge" resultType="float">
24     <RequestURL>http://acme.com/SystemUtil</RequestURL>
25   </MeasurementDirective>
26 </Metric>

```

3.5 Gestaltung von IT-Leistungskatalogen mit End-to-end Monitoring

Ein IT-Leistungskatalog besteht grundsätzlich aus einer Liste von angebotenen Leistungen. Jede Leistung setzt sich zusammen aus:

- einer Beschreibung der Leistung,
- einem Zeitrahmen, der für die Erbringung der Leistung vorgesehen ist, oder einem SLA,
- einer Liste von Personen oder Gruppen, die dazu berechtigt sind, die Leistung zu beziehen,
- Kosten (wo anfallend),
- einer Beschreibung, wie die Leistung erbracht werden wird und was das Resultat sein wird. [21, 9]

Interaktive Leistungskataloge erlauben das direkte Bestellen von Dienstleistungen [3, 8]. Ein Benutzer kann beispielsweise einen neuen Laptop bestellen. Die Kosten dafür werden automatisch mit seiner Kostenstelle abgerechnet und ein entsprechender Beschaffungsprozess eingeleitet. Ausserdem wird der Benutzer darüber informiert, in welchem Stadium seine bezogene Dienstleistung steckt.

Eine Überwachung im Sinne von End-to-end Monitoring ist vor allem auch für die im Leistungskatalog angebotenen Services sehr sinnvoll (für Forrester ist ein funktionierendes BPM und SLM inklusive Attribuierung von Metriken sogar Voraussetzung dafür, um überhaupt ein Leistungskatalog-Management einführen zu können [10]). Damit ergäben sich beispielsweise folgende Möglichkeiten:

- Aus den gewonnenen historischen Metriken könnte die zu erwartende Zeit bis zum Abschluss des Prozesses vorausgesagt werden.
- Es könnten direkte Anreize geschaffen werden, um die Prozess-Durchlaufzeiten zu verringern (beispielsweise, indem im SLA vereinbart wird, dass die IT einen Teil der Kosten einer Beschaffung tragen muss, falls die IT-seitigen Teilprozesse zu langsam sind).

Dass dies in der Theorie zwar einfach ist und stringent tönt, sich in der Praxis aber doch komplexer präsentiert, soll das Beispiel eines grossen Unternehmens mit einer sehr dichten Applikationslandschaft veranschaulichen. Für diese Firma wurde ein Tool geschaffen, welches zentral verwendet wird, um die Berechtigungsprozesse für die verschiedenen Applikationen zu steuern und anzustossen. In diese Berechtigungsprozesse sind im Normalfall sehr viele unterschiedliche Personen involviert. Falls ein Mitarbeiter auf der untersten Organisationsebene eine Zulassung für eine Applikation beantragt, deren Business-Verantwortlicher nicht in demselben Teil der Aufbauorganisation zu Hause ist, geht der Prozess einige Stellen in der Linienorganisation nach oben, wobei jeder

Supervisor dem Antrag zustimmen muss. Danach geht das Begehren in der Linie des Applikationsverantwortlichen wieder auf dieselbe Art nach unten.

Es erscheint nun logisch, dass sich so eine Antragsstellung über Wochen hinziehen kann – denn die Wahrscheinlichkeit, dass mehrere der bewilligungsverantwortlichen Personen abwesend sind, ist durchaus gegeben, sei es weil sich die Abwesenheitsvertretungs-Regeln im Normalfall ebenfalls an der Linienorganisation orientieren, oder weil es in der Natur von Arbeitenden in grossen Unternehmen liegt, dass sie ihre Ferien möglichst disjunkt zu den Ferien ihrer Vorgesetzten planen.

Die Schulbuch-Lösung für diesen Fall wäre, ein entsprechendes SLO einzuführen. Dieses SLO müsste natürlich End-to-end funktionieren – sprich die entsprechende Metrik müsste die Komplet-Durchlaufzeit von der Antragsstellung bis zur erfolgten Berechtigung des Antragstellers in die entsprechende Applikation messen. Das Problem ist dann allerdings die Frage nach der Verantwortung bei einer Verletzung des SLO. Gemäss der Aufsetzung des Prozesses wären die beiden oben erwähnten Organisationslinien allesamt gleich verantwortlich. Da sie aber gemäss der Aufbauorganisation keine organisatorische Einheit bilden, sondern einen gerichteten Graphen, und erst noch in unterschiedlichen Kostenstellen angesiedelt sind, könnte die bei einer Verletzung des SLOs zu treffende Massnahme nur in der Reduktion der persönlichen Entlohnung entweder aller am Prozess beteiligter oder (im Sinne von Komponenten-SLOs) derjenigen Person, die die Verzögerung verursacht hat. Dies müsste für die Verantwortlichkeits-Kette jeder einzelnen Applikation gemacht werden, was das Entlohnungssystem des gesamten Unternehmens enorm kompliziert machen würde.

3.6 Auswirkungen von End-to-end Monitoring auf Outsourcing

Die Weiterentwicklung der Fähigkeiten der IT im Generellen hat viele neue Möglichkeiten gebracht, wie IT das Business direkt unterstützen und zur Wertschöpfungskette beitragen kann. Gemäss Forrester ([5]) zeitigen sich daraus allerdings zwei sich widersprechende Effekte:

Die IT-Abhängigkeit des Business nimmt zu. Praktisch sämtliche Unternehmen haben einen Punkt erreicht, an dem ihre Business-Prozesse so abhängig von einer funktionierenden IT sind wie von Elektrizität und fliessendem Wasser.

Der Kostendruck auf die IT nimmt zu. Externe Dienstleister haben für die Mehrheit der IT-Geschäftsfelder Lösungen entwickelt, die direkt mit den Dienstleistungen der internen IT-Abteilungen konkurrenzieren können. Gemäss der Logik des Marktes resultieren daraus sinkende Preise und steigende Qualität.

Die interne IT kann sich nun entweder dieses Konkurrenz- und Kostendrucks annehmen, sich auf den Einkauf von externen Services und deren Weitervermittlung an das Business konzentrieren, oder eine Mischform der ersten beiden anstreben.

Alle drei Strategien setzen aber voraus, dass die interne IT ihre operationelle Struktur derer eines IT-Dienstleisters angleicht. Dafür braucht sie entsprechendes Qualitätsmanagement, um dem Business ihren Wert beweisen zu können.

Auf der anderen Seite sind diejenigen IT-Dienstleister, welche bereits etabliert sind, aus folgenden Gründen nicht eben erpicht darauf, umfassendes End-to-end Monitoring einzuführen:

- Falls sie selber Teile ihrer Dienstleistungen wiederum an Drittanbieter ausgelagert haben, wären sie plötzlich voll verantwortlich für die gesamte Performance – und falls sie selber kein umfassendes End-to-end Monitoring ihres angebotenen Services angeboten haben, hätten sie keine Chance, diese Verantwortung sinnvoll zu managen.
- Konkurrenzlosigkeit macht bequem. Die Neueinführung eines funktionierenden End-to-end Monitorings auf Seiten des Kunden (vielleicht sogar mit automatisierter Bedingungs-Verhandlung im Sinne der WSLAs) hätte eine grössere Transparenz zur Folge – mit dem vielleicht unerwünschten Resultat, dass ein anderer Anbieter oder die interne IT-Abteilung des Kunden dieselbe Dienstleistung günstiger und mit höherer Qualität anbietet und es vorher einfach niemand gemerkt hat.

4 Fazit

Abbildungsverzeichnis

3.1	Abgrenzung von BSM und SLM gemäss Forrester	7
3.2	WSLA-Services und ihre Interaktionen	12
3.3	Eine 3-Parteien-WSLA-Konstruktion	13
3.4	Beispiel für eine WSLA-Service Description	14
3.5	XML-Definition eines WSLA-SLAParameter	15
3.6	XML-Definition einer WSLA-Metric	15

Literaturverzeichnis

- [1] Alexander Keller und Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management*, 11(1), November 2003.
- [2] Avy Ellis und Michael Kauferstein. *Dienstleistungsmanagement: Erfolgreicher Einsatz von prozessorientiertem Service Level Management (German Edition)*. Springer, 2004 edition, 11 2003.
- [3] Troy DuMoulin. Troy's Blog. The Hitch Hiker's Guide to the ITIL Galaxy and beyond. IT Service Catalog Examples. http://blogs.pinkelephant.com/index.php/troy/comments/it_service_catalog_examples/, 2010. [Online; Stand 12. Februar 2013].
- [4] Heiko Ludwig et al. Web Service Level Agreement (WSLA) Language Specification. <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>, 2003. [Online; Stand 11. Februar 2013].
- [5] Jean-Pierre Garbani. Best Practices For Service-Level Management. <http://i.zdnet.com/whitepapers/Forr051103489400.pdf>, 2004. [Online; Stand 11. Februar 2013].
- [6] Martin Bichler. *The Future of e-Markets: Multidimensional Market Mechanisms*. Cambridge University Press, 1 edition, 6 2001.
- [7] Peter O'Neill, Thomas Mendel et al. BSM Is Coming Of Age: Time To Define What It Is. http://www.ktsl.com/downloads/forrester_bsm_definition_whitepaper.pdf, 2006. [Online; Stand 11. Februar 2013].
- [8] Purdue University – Information Technology. ITaP Service Catalog. <http://www.itap.purdue.edu/service/catalog/>, 2013. [Online; Stand 12. Februar 2013].
- [9] Rodrigo Flores and Bill Fine and Troy DuMoulin. *Defining IT Success through the Service Catalog (Pink Elephant Guides)*. Van Haren Publishing, 1st edition, 3 2007.
- [10] Stephen Mann. Getting A Service Catalog: So Much More Than Buying A Tool! http://blogs.forrester.com/stephen_mann/12-07-11-getting_a_service_catalog_so_much_more_than_buying_a_tool, 2012. [Online; Stand 12. Februar 2013].

- [11] Michael Ströbel. A Framework for Electronic Negotiations Based on Adjusted-Winner Mediation, 2000.
- [12] Michael Ströbel. Design of Roles and Protocols for Electronic Negotiations. 1(3):335–353, July 2001.
- [13] Thomas Mendel, Peter O’Neill et al. Implementing BSM. http://www.kts1.com/downloads/forrester_bsm_definition_whitepaper.pdf, 2006. [Online; Stand 11. Februar 2013].
- [14] Kenneth D. White. Definitions of Managed Objects for Service Level Agreements Performance Monitoring. <http://tools.ietf.org/html/rfc2758>, 2000. [Online; Stand 12. Februar 2013].
- [15] Wikipedia. Business Service Management — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Business_Service_Management&oldid=96861954, 2011. [Online; Stand 11. Februar 2013].
- [16] Wikipedia. Enhanced Telecom Operations Map — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Enhanced_Telecom_Operations_Map&oldid=529955241, 2012. [Online; Stand 12. Februar 2013].
- [17] Wikipedia. Web Services Description Language — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Web_Services_Description_Language&oldid=530280073, 2012. [Online; Stand 11. Februar 2013].
- [18] Wikipedia. Ariba — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/w/index.php?title=Ariba&oldid=537869636>, 2013. [Online; Stand 12. Februar 2013].
- [19] Wikipedia. Commerce One — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Commerce_One&oldid=533102537, 2013. [Online; Stand 12. Februar 2013].
- [20] Wikipedia. ebXML — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/w/index.php?title=EbXML&oldid=531633145>, 2013. [Online; Stand 12. Februar 2013].
- [21] Wikipedia. Service catalog — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Service_catalog&oldid=535123491, 2013. [Online; Stand 12. Februar 2013].
- [22] Wikipedia. Service-level agreement — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Service-level_agreement&oldid=535847949, 2013. [Online; Stand 12. Februar 2013].
- [23] Wikipedia. Six sigma — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Six_Sigma&oldid=537707096, 2013. [Online; Stand 12. Februar 2013].

- [24] Wikipedia. SOAP — Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/w/index.php?title=SOAP&oldid=535478625>, 2013. [Online; Stand 11. Februar 2013].
- [25] Wikipedia. Standard deviation — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Standard_deviation&oldid=537713485, 2013. [Online; Stand 12. Februar 2013].